A PHIGS Based Interactive Graphical Preprocessor for Spatial Mechanism Analysis and Synthesis

by

Brian R. Thatch

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
Master of Science
in
Mechanical Engineering

APPROVED:

Dr. Arvid Myklebust, Chairman

Dr. Charles F. Reinholtz

Dr. Hamilton H. Mabie

March 19, 1987

Blacksburg, Virginia

A PHIGS Based Interactive

Graphical Preprocessor

for Spatial Mechanism

Analysis and Synthesis

by

Brian R. Thatch

Dr. Arvid Myklebust, Chairman

Mechanical Engineering

(ABSTRACT)

This thesis presents the development and use of MECHIN, an interactive graphical preprocessor for data input to spatial mechanism analysis and synthesis codes. A goal in the development of this preprocessor is to produce a graphical data input program that is both graphics device-independent and not structured for the input of data to any particular mechanism processing program. To achieve device-independence, the proposed graphics standard PHIGS (Programmer's Hierarchical Interactive Graphics System) is used for the graphics support software. Program development strategies including screen layout and user interfaces for three-dimensional data input are discussed. The program structure is also described and presented along with a complete listing of the program code to aid in future modifications and additions. Finally, a description of the use of the program is presented along with several examples of mechanism data input for synthesis and analysis.

Acknowledgements

I would like to thank my advisor, Dr. Arvid Myklebust for his guidance throughout the research and preparation of this thesis. I would also like to thank Dr. Charles Reinholtz and Dr. Hamilton Mabie for serving on my advisory and examining committee.

Thanks also go to all the people in the CAD Lab for their help and advice with this thesis. In particular, I would like to thank Jeff Thompson, Dino Ciabattoni, Greg Sherman, Brad Coffey, Steve Wampler, and Mitch Keil.

I would also like to thank my parents for their support throughout my education.

Finally, I would like to thank my wife, Helen, whose love and hard work have made my graduate studies possible.

Table of Contents

List of Illustrations

Chapter 1

INTRODUCTION

A large number of general purpose programs for the analysis and synthesis of mechanisms have been developed over the past several years. Although these programs have concentrated on the solution of mechanism design problems, relatively little work has been done on the development of easy to use and effective mechanism data entry systems. This thesis will present the development and use of MECHIN, a general purpose graphical preprocessor for data entry of spatial mechanism analysis and synthesis parameters.

The basic problem for graphical kinematic data entry is providing the user with a simple system for specifying design parameters. This data may include joint positions, orientations, and connectivities for mechanism analysis and body positions and orientations for mechanism synthesis. Once entered, data must be presented so that it is easy to understand and modify. This problem is compounded in the case of spatial mechanisms

in trying to convey three-dimensional mechanism data on a two-dimensional computer graphics screen.

Another problem with mechanism data input is specifying the input parameters in a format specific to a processing code. These formats may include the use of special commands to describe the mechanism or a set order that mechanism parameters must be entered. By developing an input program which is tailored to a specific mechanism analysis or synthesis program, a unique input processor must be produced for each specific mechanism design program. A goal of this thesis is to develop a general mechanism data preprocessor which will handle a wide range of processing codes.

A further goal of this thesis is to develop a preprocessor which is not limited to a specific computer graphics system. This will be accomplished through the use of PHIGS (Programmer's Hierarchical Interactive Graphics System) which is a three-dimensional graphics standard proposed by the American National Standards Institute (ANSI). The use of PHIGS will not only help with the development of a device-independent graphics code, but will aid in the construction and viewing of three-dimensional mechanism input data.

This thesis will present the development of the interactive graphical preprocessor called MECHIN and describe both the program development strategy and the program structure. A description of the use of the program will also be presented along with several examples. A complete listing of the program source code is also included to aid in future development work and modifications.

Chapter 2

LITERATURE REVIEW

Since this thesis deals with the development of an interactive graphical preprocessor for spatial mechanism analysis and synthesis, two areas of literature review are necessary. The first area concentrates on the application of computer-aided design techniques as applied to mechanism design. The second area deals with the topic of interactive computer graphic software design.

Computer-Aided Mechanism Design

Before development of the digital computer, mechanical design and analysis consisted primarily of long hand calculations and time consuming graphical layouts. The advancement of computer hardware technology combined with decreases in the costs of

hardware and processing over the past few decades have given rise to a wide range of mechanical design and analysis tools.

In the area of mechanism design, several general purpose mechanism analysis and synthesis programs have been developed. These include IMP [1], DRAM [2], ADAMS [3], DYMAC [4], and DADS [5] for analysis and KINSYN [6], MECSYN [7,8], LINCAGES [9], and RECSYN [10] for synthesis. All of these programs are excellent for the design of mechanisms, but they require a substantial amount of work on the part of the user in the areas of problem formulation and data entry.

Coats and Cipra [11] present a more complete mechanism analysis system based on IMP. This system includes a graphical preprocessor for spatial mechanism input parameters and generates a data file in the IMP program language. Also included is a postprocessor for mechanism animation. The overall system is somewhat limited because it supports only mechanism analysis using IMP. The preprocessor is also limited because mechanism input data must be entered sequentially from joint to joint in the order of mechanism connectivity. This system is device-dependent and uses GRAFIC for graphics software support.

Work has also been done by Barris and Riley [12] on upgrading the graphical capabilities of LINCAGES. This work at the University of Minnesota includes the development of modularized graphics libraries and is supported by the Apollo DN660 workstation.

A new computer-aided mechanism design system has been proposed by Myklebust, Keil, and Reinholtz [13]. This system features device-independent graphical input, a common synthesis and analysis database, and realistic automatic modeling and animation for

rapid evaluation of results. This system, although partially complete, at present lacks a complete graphical input processor for synthesis and analysis specifications.

Interactive Graphics Software

The advancements in the area of computer graphics technology over the past several years have led to extensive use in many computer software applications. Although human factors research in the area of hardware interaction (i.e. chair height, screen tilt, lighting, etc.) has been extensive, human factors research in the area of interactive software design has been lacking.

Several papers have been written that attempt to address man-machine interaction and to quantify guidelines for human factors in interactive software design. Spencer [14] and Halter [15] present brief sets of guidelines on interactive software design. An attempt to apply these and other existing guidelines to a graphics system under development are presented by Swezey and Davis [16]. Although some guidelines do exist, current work in the area suggests that more research needs to be done.

Foley, Wallace, and Chan [17] present a very complete work which is an attempt to aid software designers in selecting interactive graphics techniques and devices. An attempt is made to quantify different interactive graphic techniques and to identify which are best suited for different applications.

Research is also being done in the area of visual design for computer graphics screens. Reilly and Roach [18] discuss screen design that borrow on the ideas of advertising

design for data presentation. Guidelines for screen design are also presented by Galitz [19]. This work addresses areas such as eyeball fixation studies which suggest that the human eye usually first moves to the upper-left center of a graphic display. These works stress the use of human factors design in screen layout as the key to increased user productivity.

The psychological effects of computer display menus have been addressed by Snowberry, Parkinson, and Sisson [20]. This research attempts to quantify different types of menu strategies and to rate selection performance. This research indicates that increased menu breadth (the number of choices per menu) improves search speed and accuracy over the use of menu depth (the number of menu levels).

A final topic in the area of visual computer graphic design is the use of color. Christ [21] presents a paper on the effect of color in visual search and identification performance. Although this work does not specify optimal colors for graphic design considerations, many sources suggest the use of bright colors, such as yellow and white, over a dark background, such as gray.

Several textbooks have also been written in the area of interactive computer graphics and are excellent references for interactive computer graphic software development. Two very useful texts were written by Newman and Sproull [22] and Foley and Van Dam [23].

Chapter 3

MECHIN PROGRAM DEVELOPMENT

In the development of any software system, the initial design considerations and decisions are important to the future system success. In order to describe the development strategy of the mechanism preprocessor MECHIN, this chapter presents the background considerations used in program development. These considerations include an introduction to the PHIGS graphic standard and the use of three-dimensional data display. Also discussed are program development considerations for screen design, mechanism analysis data input, and mechanism synthesis data input.

Introduction to PHIGS

Since one of the main objectives in the development of MECHIN was to design a three-dimensional mechanism input processor that is independent of the graphic input

device, the choice of graphics support software was crucial. To aid in the development, the graphics support software must support device-independent graphics routines as well as be able to handle the three-dimensional data input and viewing required for a spatial mechanism preprocessor.

In light of these design considerations, the logical choice for graphical software support was PHIGS (Programmer's Hierarchical Interactive Graphics System). PHIGS is an advanced graphics application programming language that is currently being proposed by the American National Standards Institute (ANSI) and is expected to be internationally approved by 1989. PHIGS provides a useful set of device-independent programming routines and uses true three-dimensional capabilities to improve the design and visualization of graphics data.

For development of MECHIN, the IBM implementation of PHIGS called graPHIGS was used. GraPHIGS is currently installed on an IBM 4341 processor and running in the CAD/CAM Laboratory at Virginia Tech. The development work for MECHIN was performed on an IBM 5080 color raster graphics workstation. GraPHIGS also supports a variety of programming languages including PL/I, Pascal, and FORTRAN. For development of MECHIN, FORTRAN77 was used as the programming language.

Input devices for data entry were chosen to be a pick device for menu selections, a locator for zoom area selection, and string and valuator devices for data entry. The IBM 5080 uses a tablet for pick and locator entry, a keyboard for string entry, and dials for valuator entry. Implementation of MECHIN can be supported on any graphics device that supports the PHIGS standard and handles a minimum of string and pick data entry.

Three-Dimensional Data Display

The construction of an interactive graphical preprocessor for spatial mechanism design requires the input and viewing of three-dimensional position and orientation data for both mechanism synthesis and analysis routines. Since a computer graphics screen is two-dimensional, an effective way of displaying three dimensions on a two-dimensional plane was needed. Textbooks generally use an oblique projection of one coordinate axis when presenting three-dimensional data on a two-dimensional page. Tsang [24] also discusses the use of an oblique view in the design of a user interface for three-dimensional conceptual design. For implementation in MECHIN, the use of this oblique projection was applied to the Z-axis of the Cartesian coordinate system. The X-axis and Y-axis coordinate directions remain perpendicular as in any oblique view. This implementation was chosen to give the appearance of the Z-axis being oriented out from the screen towards the user.

When positioning three-dimensional data, the user must be able to quickly comprehend the position of any data point. To achieve this goal, several graphical positioning cues were devised and are shown in Figure 1. These visualization cues include the projection of the three-dimensional data point on to each of the X-Y, X-Z, and Y-Z coordinate planes. Where these projections intersect the coordinate planes, a box is drawn to provide the user with a reference of the plane position and orientation. Another positioning cue is provided by highlighting each coordinate axis up to the value of the data point. Numerical coordinate data is also displayed to provide exact position specifications.

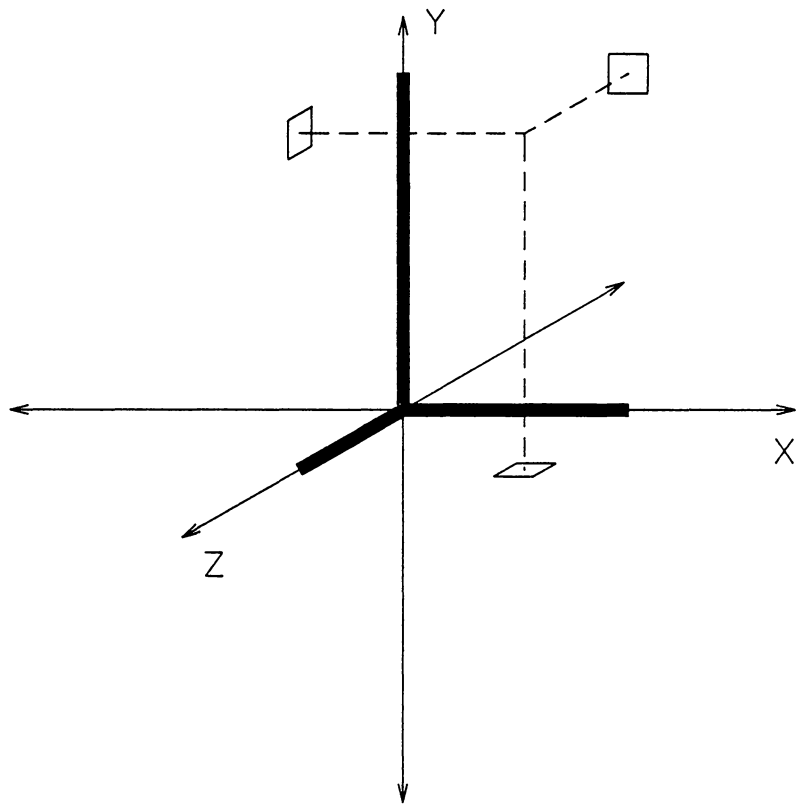


Figure 1. Three-Dimensional Positioning Cues

To display the orientation of vectors, an orientation scheme similar to the positioning scheme was chosen and is shown in Figure 2. This orientation method displays a unit vector in space drawn from the local coordinate system being positioned. The direction of the vector is indicated by highlighting the coordinate axes along the direction of the vector orientation. Again, numerical coordinate data for the vector orientation is also presented.

Screen Design

In order to make the mechanism preprocessor as useful as possible, the display screen was designed to enable the user to work efficiently. The major sections of the design screen determined to be needed were a command area, a menu area, a main viewing area, and a positioning area. The command area is used to interactively convey to the user what options are available in the present program state. This command area is used in conjunction with the menu area and provides the user with control of the program for mechanism data input and modification. The main viewing area is used to display all entered data and is the main work area of the program. The positioning area is used to position and orient the current data being entered. Once the data is entered, it is displayed in the main viewing area.

The final layout for the screen is shown in Figure 3. The command area was located near the upper left corner. This area was chosen since human factors research has shown the user usually first views this area, and that this is the area the user should first look at to determine the present program status. Immediately below the command box

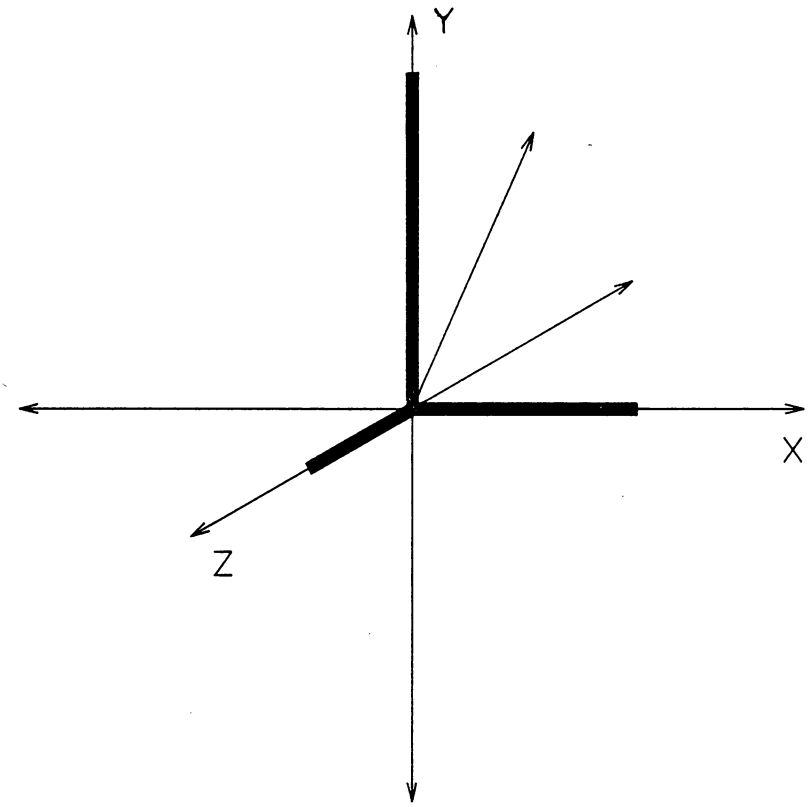


Figure 2. Three-Dimensional Orientation Cues

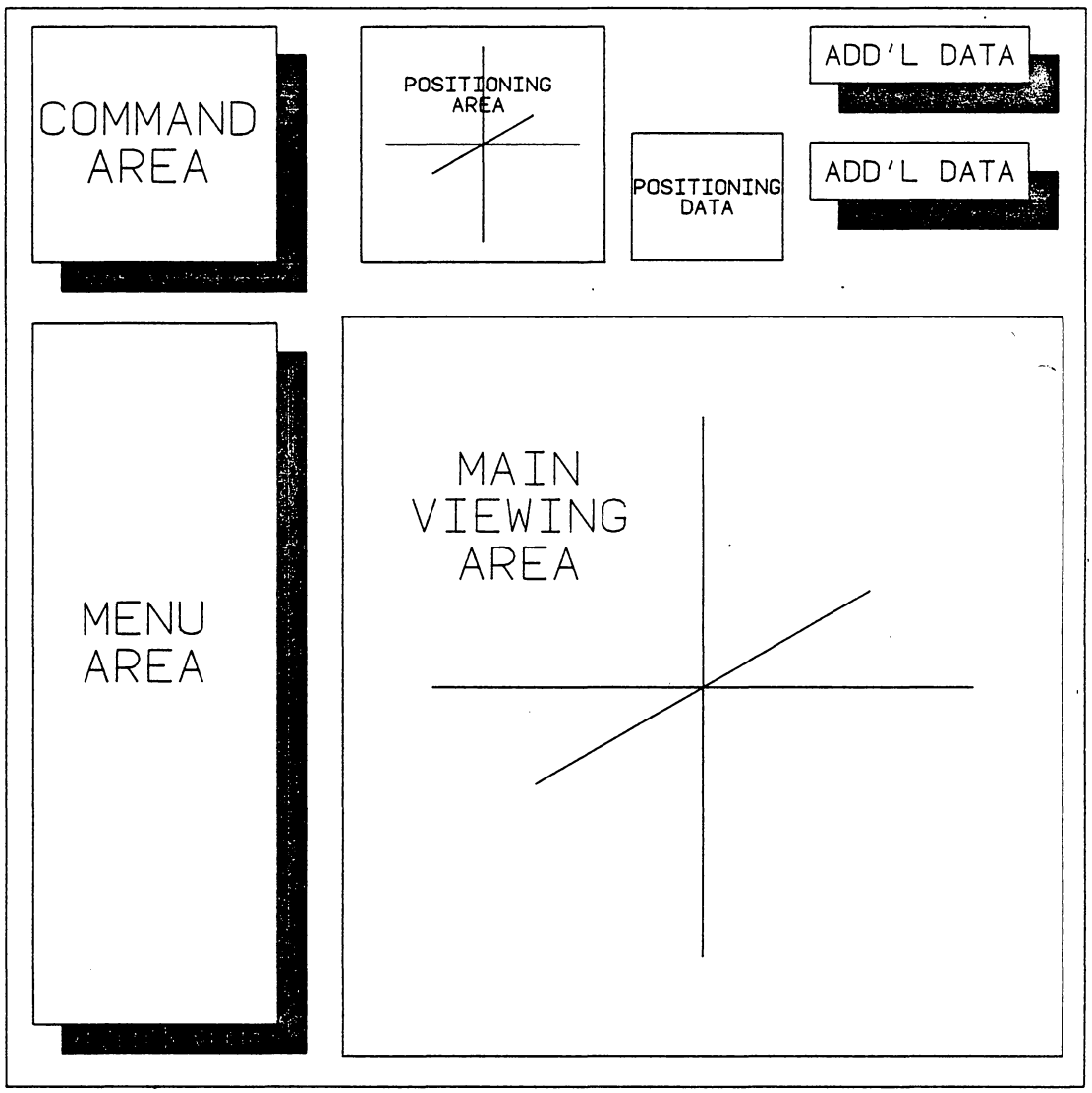


Figure 3. Data Entry Screen Design

and along the left side of the screen, the menu box was located. The menu box was located below the command box because all interaction with the program is prompted and controlled from these two areas.

The main viewing area was made as large as possible on the remaining screen area. Above the main viewing area, the positioning area was located. The upper right corner of the screen was reserved for additional display data such as the current file name and the range of the coordinate axes displayed in the main viewing area.

The use of color was also considered at this stage of development. To present the mechanism data in a useful and aesthetically appealing form, several different color combinations were tried. The final color choices consisted of a grey background for the command area, the menu area, the main viewing area, and the positioning area. The principal colors to be used on these backgrounds were chosen to be yellow and white with additional data to be displayed using red, black, turquoise, and magenta. These colors were chosen on the basis of which colors were most visible and aesthetically appealing on the grey background screen. All areas behind the command, menu, and viewing areas were colored blue, again for aesthetic reasons.

An additional detail added to the screen layout was the use of background shadowing to provide a three-dimensional effect. The command and menu areas were given a shadow area behind the grey background to give the appearance that these planes are lifted above the screen. This use of background shadowing was not applied to the main viewing and position areas because this reduced the illusion of three-dimensional depth.

Analysis Data Input

The input of data for mechanism analysis requires the specification of the position, orientation, and type of joints along with the connectivity of these joints within the mechanism. The goal of this research was to develop a general way of specifying the required mechanism analysis data that is easy to use and understand. In order to accomplish this goal, a novel method of interactively specifying joint input variables and connectivities was devised.

To be completely general, the program user has the option of creating and deleting joint and link data at any time. This method was chosen rather than the current implementation in many mechanism programs of specifying joint and link data sequentially around the mechanism loops. This method provides the user with greater flexibility in construction of a mechanism for analysis. However, the user must be careful that all mechanism loops are complete.

A large number of joint types were also implemented to give the user the option of creating many different mechanisms. The types of joints implemented include revolute, prismatic, cylindric, spheric, screw, flat (or planar), and a gear pair. The representations of these joints as they appear in MECHIN are shown in Figure 4.

It should be noted that within MECHIN, the ground points of the mechanism are also treated as joints. To attach a mechanism to ground, a link must be drawn between the joint which is to be grounded and a specified ground point.

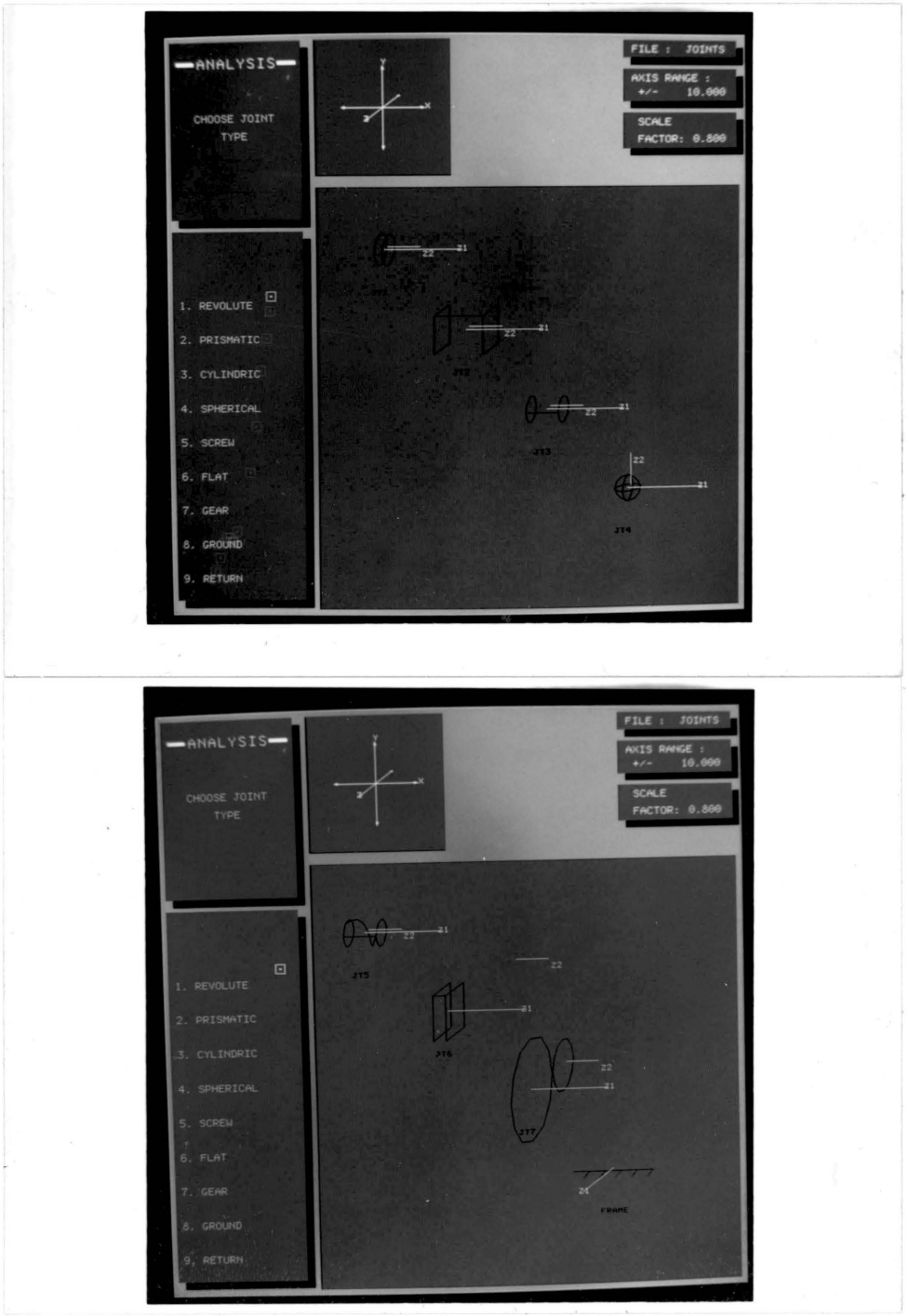


Figure 4. Joint Icons

Analysis data input for MECHIN is based on the idea of specifying two local coordinate systems at each joint and defining the connectives between the joints. This is accomplished by having the user orient the joint z-axis (typically the axis of rotation or sliding) when specifying the joint position. The user then inputs the direction of the joint x-axis for both local coordinate systems when specifying the mechanism connectivity. A complete description of how this is implemented within MECHIN along with several examples will be described in Chapter 5, Using MECHIN.

Synthesis Data Input

Depending on the implementation of the mechanism synthesis processor, the specification of mechanism synthesis parameters usually falls into one of three areas. These three areas are function generation, path generation, and rigid body guidance. Function generation typically requires the specification of the output angle of one or more links as a function of the link input angle within the mechanism. Path generation requires the input of the sequence of points that a point on the synthesized mechanism will pass through. Rigid body guidance requires the input of a series of body positions and orientations that a link of the synthesized mechanism will pass through.

For implementation within MECHIN, rigid body guidance synthesis was chosen because it lends itself to interactive computer graphics application. Also, function generation problems often can be treated as inversions of body guidance problems and path generation can be specified as incomplete body guidance problems. Synthesis input data for MECHIN is based upon entering the position of points the body will be guided through

by the synthesized mechanism. Orientation of these body positions is entered by defining a local coordinate system at each defined point. A complete description of how synthesis data input is implemented within MECHIN along with several examples will be described in Chapter 5, Using MECHIN.

Chapter 4

MECHIN PROGRAM STRUCTURE

To aid in the understanding of MECHIN and to assist in future program modifications, this chapter presents the program structure. Also presented is a complete description of the methods of data storage within MECHIN. MECHIN is written in FORTRAN77 using PHIGS for graphical support. The complete program listing is presented in Appendix A for user reference.

Program Organization

A flowchart of the top level organization of MECHIN is presented in Figure 5. As can be seen from this flowchart, the main program MECHIN is divided into separate processors for specification of analysis and synthesis parameters. Both of these processors

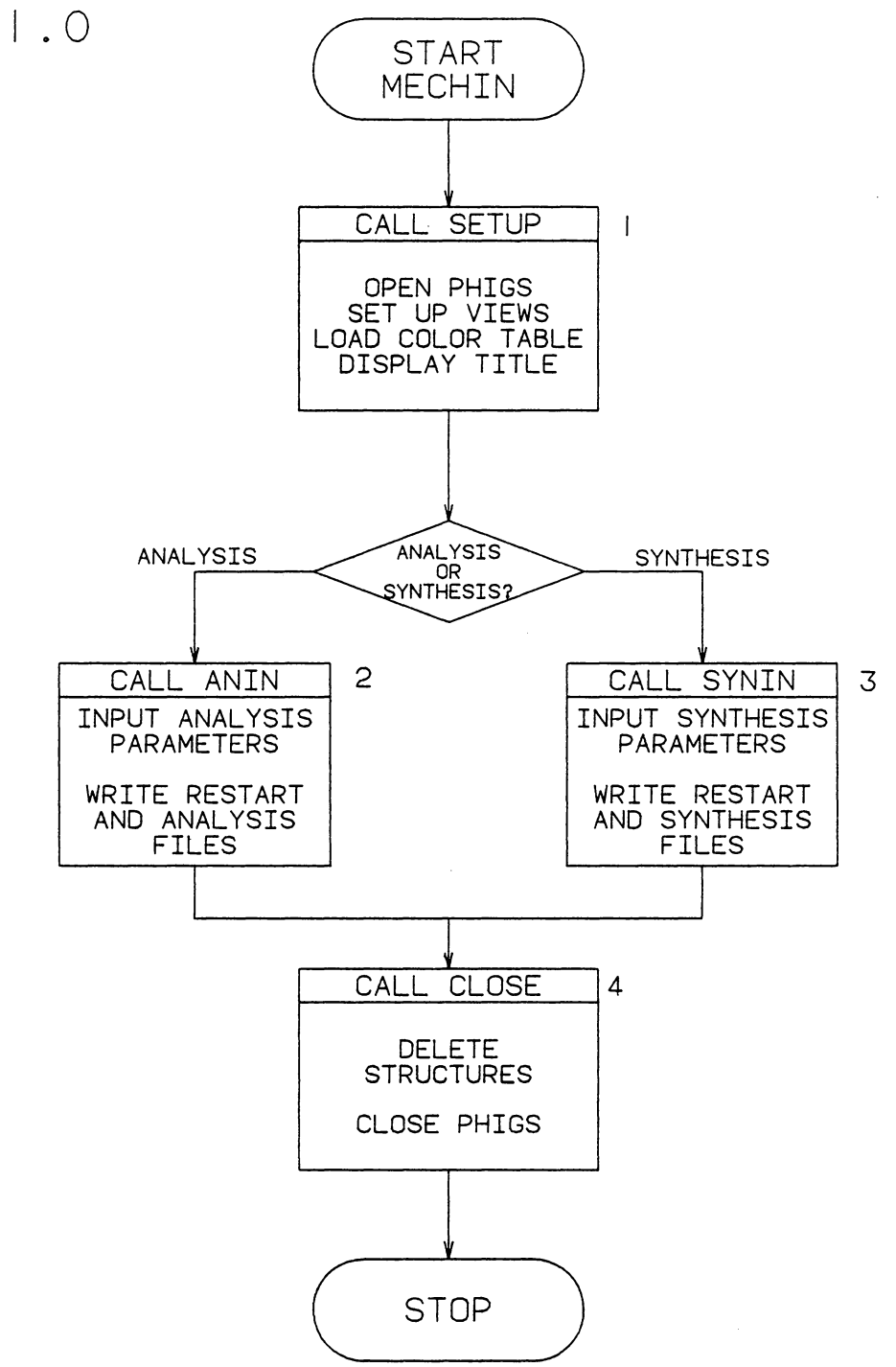


Figure 5. Flowchart of PROGRAM MECHIN

are supported by common routines for setting up initial program parameters and closing the program.

SUBROUTINE SETUP is used to open graPHIGS and is flowcharted in Figure 6. Once graPHIGS is opened, SUBROUTINE SETUP opens and initializes the graphics workstation, loads the color table, and sets up all initial views. After all setup steps are completed, the MECHIN title screen is displayed. A photograph of the title screen is shown in Figure 7. Following the title screen, SUBROUTINE SETUP then displays the analysis and synthesis choice screen as shown in Figure 8. When a choice is made, control is passed back to the main program MECHIN.

If analysis is selected, the user enters the analysis branch of the main program and control is passed to SUBROUTINE ANIN. A flowchart of SUBROUTINE ANIN is presented in Figure 9 and a photograph of the analysis screen with a mechanism to be analyzed is shown in Figure 10. The user is first queried for the name of a new file or the name of a previously defined model for restart. Control is then passed over to SUBROUTINE DATA in which the user interactively enters the mechanism analysis data by accessing lower level subroutines for the addition and deletion of points, joints, links and mechanism initial conditions. A flow chart of SUBROUTINE DATA is given in Figure 11.

Before leaving SUBROUTINE ANIN, a graphical restart file for MECHIN analysis data is created in SUBROUTINE WRREST. The creation of this restart file enables the user to recall the created mechanism for additional modifications. An analysis file is also created in SUBROUTINE WRANAL. At present MECHIN supports the creation of analysis data files for IMP and RSCR. RSCR is a mechanism processor for analysis and synthesis of revolute-spheric-cylindric-revolute mechanisms developed by Jeff Thompson

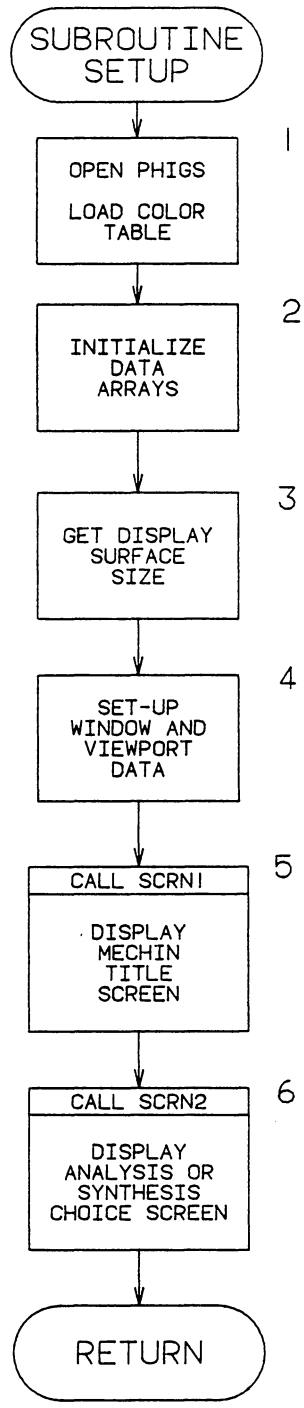


Figure 6. Flowchart of SUBROUTINE SETUP

Figure 7. Title Screen

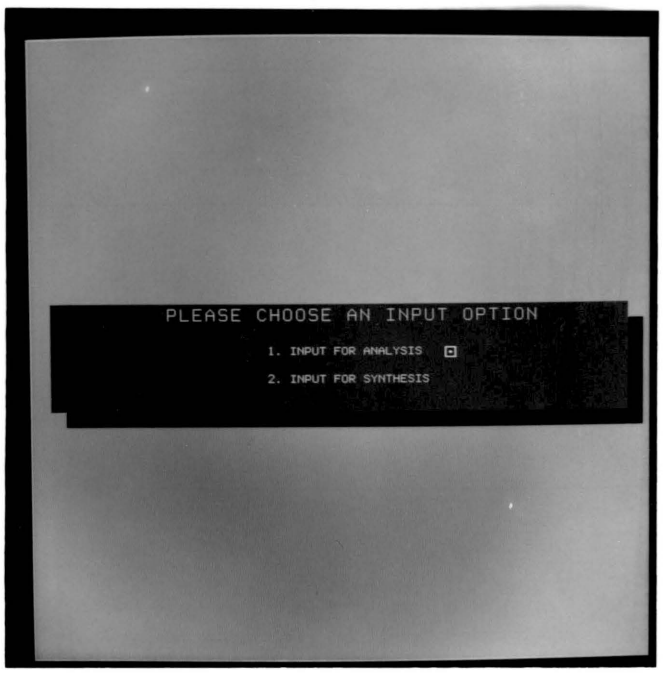


Figure 8. Analysis and Synthesis Choice Screen

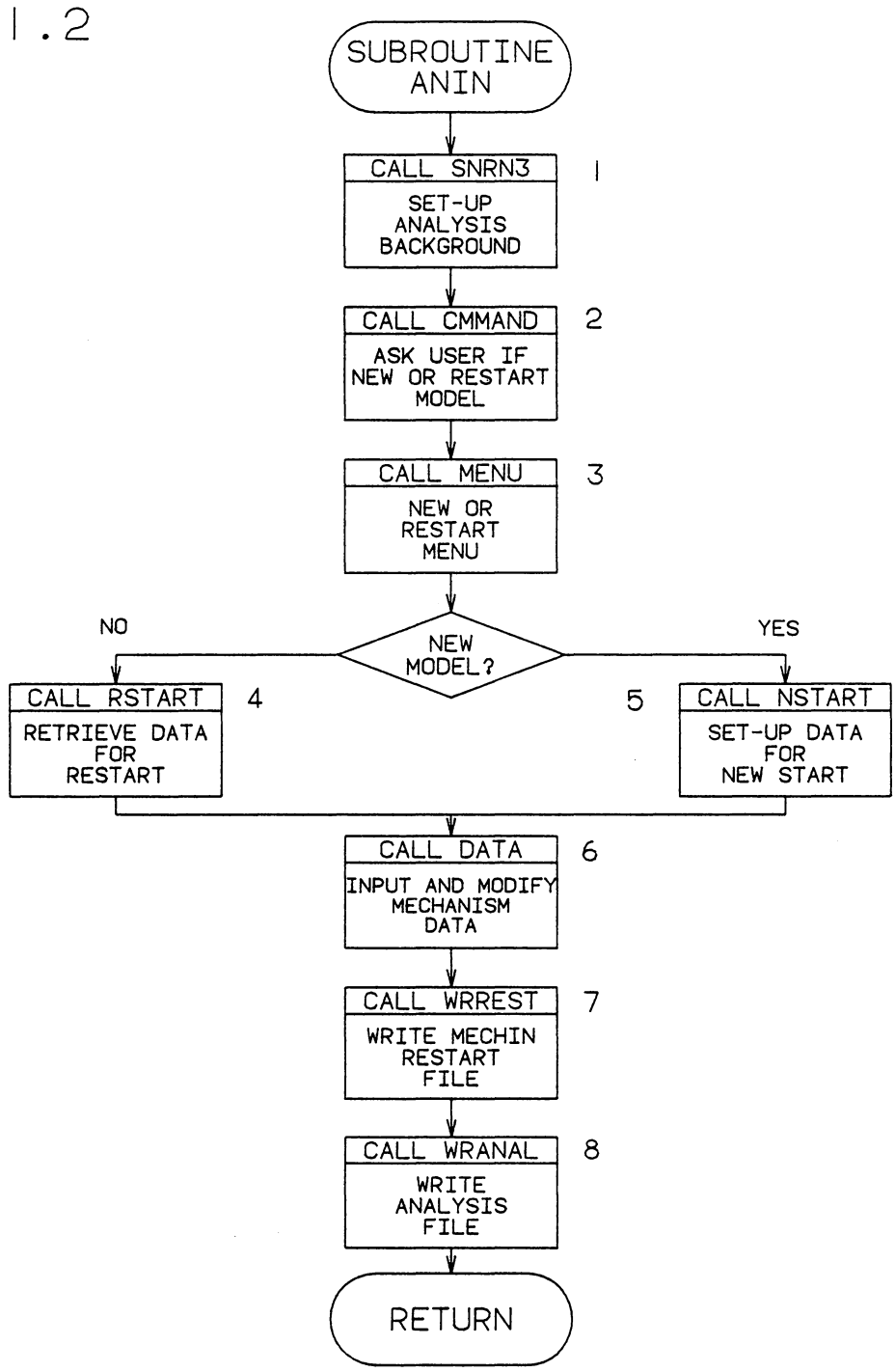


Figure 9. Flowchart of SUBROUTINE ANIN

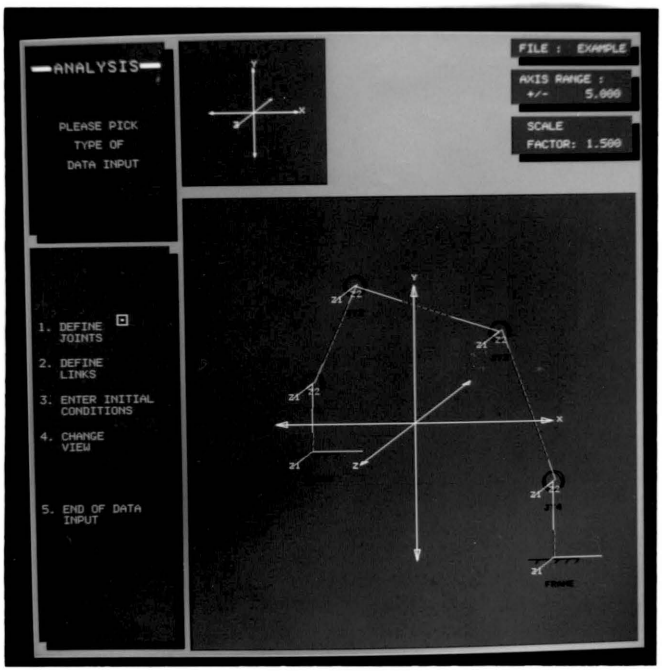


Figure 10. Analysis Background Screen

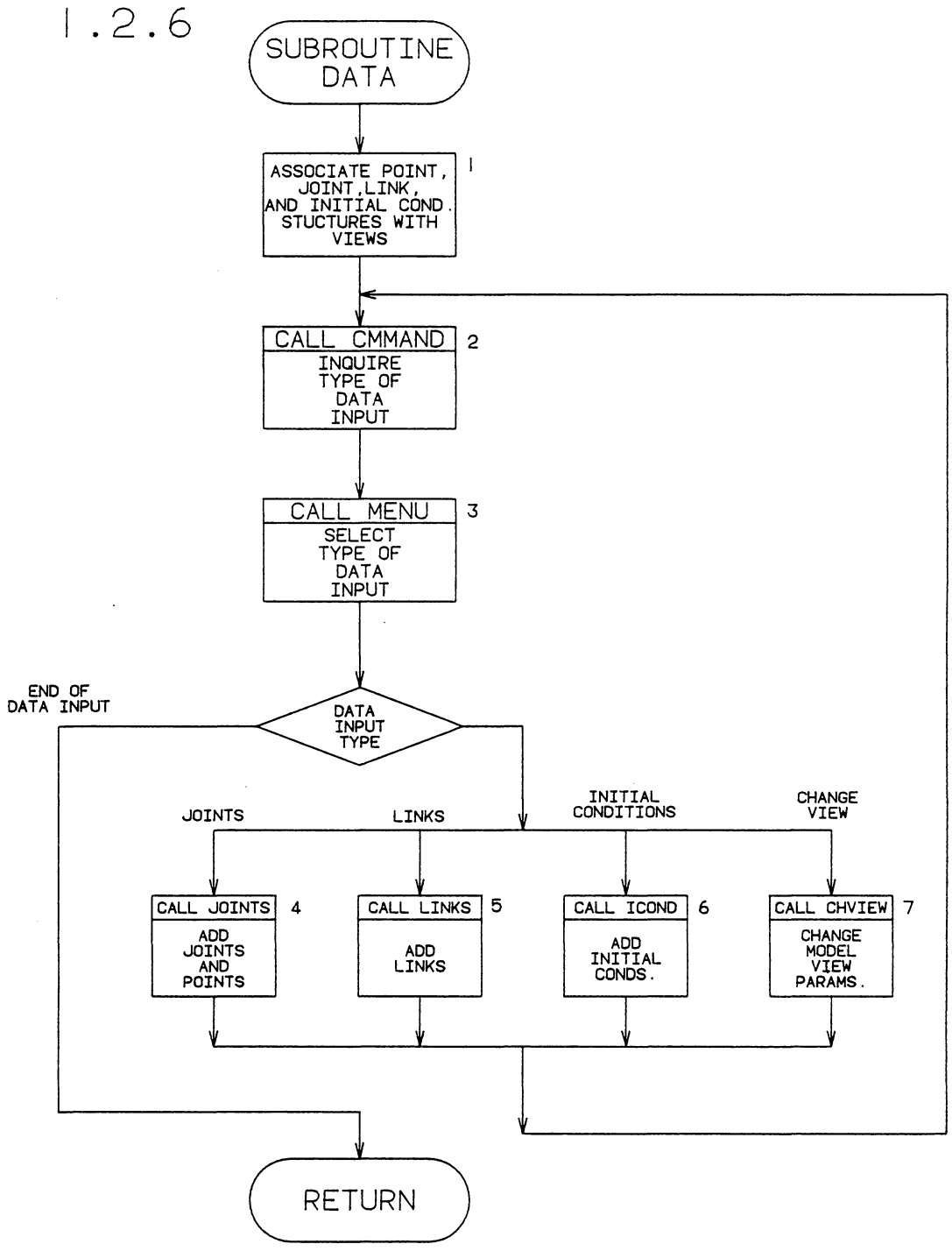


Figure 11. Flowchart for SUBROUTINE DATA

in the Department of Mechanical Engineering at Virginia Tech [25]. Two application examples for generating analysis input for IMP and RSCR are SUBROUTINE IMP and SUBROUTINE RSCRA presented in Appendix A. The use of these data formatting subroutines will aid in the addition of future analysis processing programs.

The structure of the synthesis input routine is very similar to the analysis routine structure. Once input for synthesis has been chosen, control of the program is passed to SUBROUTINE SYNIN. A flowchart of SUBROUTINE SYNIN is presented in Figure 12. The user is again prompted for either a new model name or the restart of previously defined synthesis specifications. The program then enters SUBROUTINE SDATA which prompts the user for the mechanism synthesis data and is flowcharted in Figure 13.

Before leaving SUBROUTINE SYNIN, a graphical synthesis restart file and the synthesis data file are then generated by SUBROUTINE SWRRES and SUBROUTINE WRSYN, respectively. MECHIN presently supports data input for the synthesis routine RSCR and an example for data formatting of synthesis data is presented in SUBROUTINE RSCRS in Appendix A.

Data Storage

To aid in future program modifications to MECHIN, and to allow the addition of more mechanism processing programs, a complete description of the storage techniques for

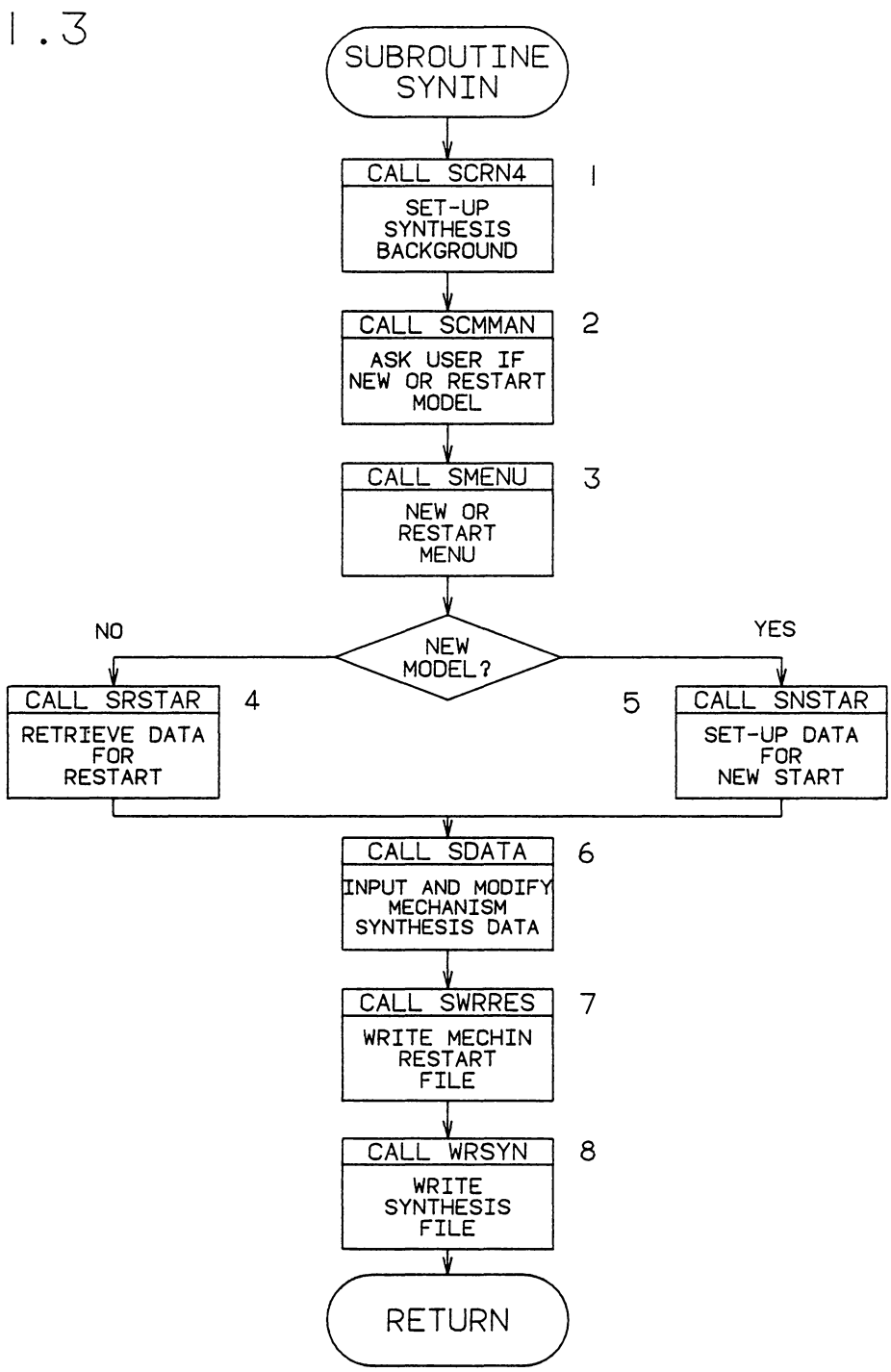


Figure 12. Flowchart of SUBROUTINE SYNIN

1.3.6

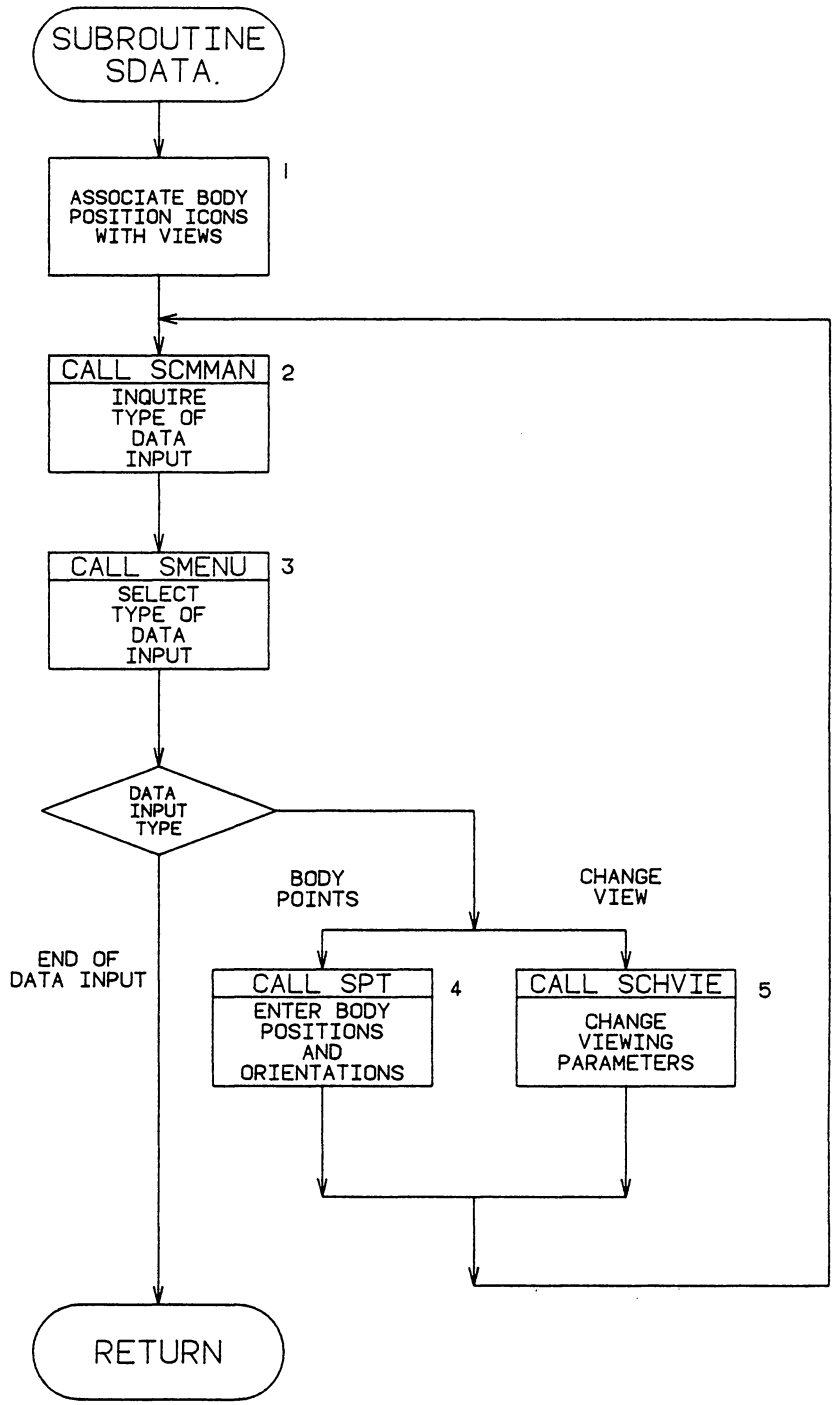
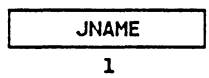


Figure 13. Flowchart of SUBROUTINE SDATA

mechanism analysis and synthesis data are presented. This includes the method of internal storage within MECHIN for analysis and synthesis parameters.

In the analysis input processor of MECHIN, data for joints, points, links, and initial conditions are stored. All four types of input data are stored using parallel arrays for character, real, and integer data. Parallel arrays are used because a combination of real, integer, and character data is needed to describe mechanism parameters, and storage of all three data formats is impossible in a single array. Therefore, three arrays are used with common data stored in identical array row positions.

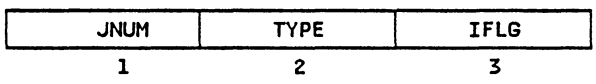
For the storage of joint data, the three arrays are called JNAM, JNUM, and JOINT for character data, integer data, and real data, respectively. JNAM is a character array used to store the name of the joint. JNAM is a one-dimensional array and a row in the array appears as shown below.



where

JNAME is the name of the joint (up to 7 characters)

JNUM is an integer array used to store the joint number, the joint type, and a flag for invisibility. This flag is used within MECHIN to temporarily make joints invisible for viewing considerations. A row of the JNAM array appears below.



where

JNUM	is the joint number
TYPE	is the joint type
	1 = revolute
	2 = prismatic
	3 = cylindric
	4 = spheric
	5 = screw
	6 = flat
	7 = gear pair
	8 = ground
IFLG	is the invisibility flag
	0 = visible
	1 = invisible

JOINT is a real array used to store the position of the joint and both local coordinate systems defined at the joint. Other data stored includes gear ratio for a gear pair, screw lead for a screw joint, and the initial offset for cylindric, prismatic, and screw joints. A row in the JOINT array is stored as shown below.

POS1	ZAX1	XAX1	XAX2	SCRL	IOS	POS2	ZAX2	GR
1-3	4-6	7-9	10-12	13	14	15-17	18-20	21

where

POS1	global x,y, and z coordinates 1st joint axis origin
ZAX1	unit vector along the z-axis of the 1st joint axis
XAX1	unit vector along the x-axis of the 1st joint axis
XAX2	unit vector along the x-axis of the 2nd joint axis
SCRL	lead of a screw joint
IOS	initial offset in cylindric, prismatic, and screw joints
POS2	global x,y, and z coordinates 2nd joint axis origin
ZAX2	unit vector along the z-axis of the 2nd joint axis

GR gear ratio for a gear pair

The storage of point data is similar to that of joint data storage. Arrays PNAM, PNUM, POINT are used for character, integer, and real storage, respectively. PNAM is a one-dimensional character array used to store the name of the point. A row of the array appears below.

PNAME
1

where

PNAME is the name of the point (up to 7 characters)

PNUM is used to store the number of the point and is also a one-dimensional array. A row in this integer array is shown below.

PNUM
1

where

PNUM is the number of the point

POINT is used to store the location of points and a typical row in the point array appears below.

XCOOR	YCOOR	ZCOOR
1	2	3

where

XCOOR	is the x coordinate of the point position
YCOOR	is the y coordinate of the point position
ZCOOR	is the z coordinate of the point position

Storage of link data is again in three parallel arrays LNAM, LNUM, LINK for character, integer and real data, respectively. LNAM is used to store the name of the link and a row in the LNAM array appears below.

LNAME
1

where

LNAME	is the name of the link (up to 7 characters)

LNUM is used to store the link number and the mechanism connectivity data. This connectivity data includes the joint and axis number at each end of the link. MECHIN also allows for the addition of as many as six joints per link for multiple loop mechanisms. An invisibility flag is also included as in the joint integer array. The data positions in the LNUM array appear below.

LNUM	JEN1	JAX1	JEN2	JAX2	....	JEN6	JAX6	IFLG
1	2	3	4	5	6-11	12	13	14

where

LNUM	is the link number
JEN1	is the number of the joint at the first end of the link
JAX1	is the axis number used at the first end of the link
JEN2	is the number of the joint at the second end of the link
JAX2	is the axis number used at the second end of the link

.

.

.

JEN6	is the joint that is the sixth coupler joint
JAX6	is the axis number used by this coupler point
IFLG	is the invisibility flag

LINK is a real array for storage of the positions of the ends of the link. The way MECHIN is presently implemented, this data is taken from the JOINT array and the LINK array is not used. The real array was left in the program for future program development work. A row in the present link array is shown below.

END1	END2
1-3	4-6

where

END1	are the coordinates of the first end of the link
END2	are the coordinates of the second end of the link

For initial condition data, the specified input positions, velocities, and accelerations are stored. The parallel array structure is again used with arrays INAM, INUM, and IC. INAM is a character array used to determine which joints have had initial conditions applied. If a row in the character array is filed with DDDDDDD, this signifies that initial condition data has been applied to the specified joint. If the array is filed with BBBBBBB or CCCCCCC, initial conditions have not been applied to the specified joint. Details of this data storage technique are discussed later in this section. A row in the INAM array is presented below.

INAME
1

where

INAME	is the initial conditional specification flag
	DDDDDDD = initial conditions are specified
	BBBBBBB = initial conditions are not specified
	CCCCCCC = initial conditions are not specified

INUM is an integer array used to specify the number of the joint at which the initial condition is specified and the number of increment steps of the initial condition that will be applied. A row in the INUM array appears below.

JNUM	INCR
1	2

where

JNUM	is the joint to which the initial conditions are applied
STEP	is the number of increments to be stepped through

IC is a real array used to store the initial conditions for position, velocity and acceleration at a given joint. A row in the IC array appears below.

POSI	INCR	VELO	ACCE
1	2	3	4

where

POSI	is the specified initial position
INCR	is the increment of the position

VELO	is the specified initial velocity
ACCE	is the specified acceleration

In MECHIN, all data storage is done by searching the appropriate array and finding the first unused row. This is accomplished by using the character array of the type of data being entered as the key for this search. In the current implementation of MECHIN, all data storage character arrays are initially filled with the string BBBBBBB. When a user deletes a mechanism analysis entity (joint, link, point, or initial condition), the character array for the deleted entity is filled with CCCCCCC in the appropriate row. When the user requests the addition of an analysis entity, the character array is searched for the first available slot, either a BBBBBBB or a CCCCCCC in the character array. When a slot is identified, the entered data is written to appropriate character, integer, and real array.

Data storage for synthesis is considerably simpler then data storage for analysis. The data stored are the body positions and orientations, again in parallel character, integer, and real arrays named PNAM, PNUM, and POINT. The reuse of the same array names as in the analysis routines causes no problems because analysis and synthesis input routines never run concurrently. Array PNAM is used to store the name of the body position. A row in the PNAM array appears below.

BNAME
1

where

BNAME	is the name of the body position (7 characters)

PNUM is used to store the number of the body position. A row in the PNUM array appears below.

BNUM	U	U	U
1	2	3	4

where

BNAME	is the name of the body position (7 characters)
U	is an unused array position

The POINT array is used to store the body position and the body orientation. A row in the POINT array appears below.

POSN	XAXE	YAXE	ZAXE	U	U
1-3	4-6	7-9	10-12	13	14

where

POSN	global x,y, and z coordinates of the body position
XAXE	unit vector along the local x-axis of the body position
YAXE	unit vector along the local y-axis of the body position
ZAXE	unit vector along the local z-axis of the body position
U	is an unused array position

Methods of data search and storage for synthesis data are the same as presented in the discussion of data search and storage for analysis data.

Chapter 5

USING MECHIN

This chapter presents an explanation of data input using MECHIN for the analysis and synthesis of spatial mechanisms. Because MECHIN is a completely interactive pre-processor, a step-by-step data entry procedure is not required. It is therefore the goal of this chapter to present an overview of the the data entry options available within MECHIN along with several examples.

MECHIN Menu Hierarchy

In order to clarify what options are available for data entry of mechanism parameters, menu hierarchies of the data entry portions of MECHIN are presented in Figure 14. It can be seen from this figure that the user has the option of entering four classes of

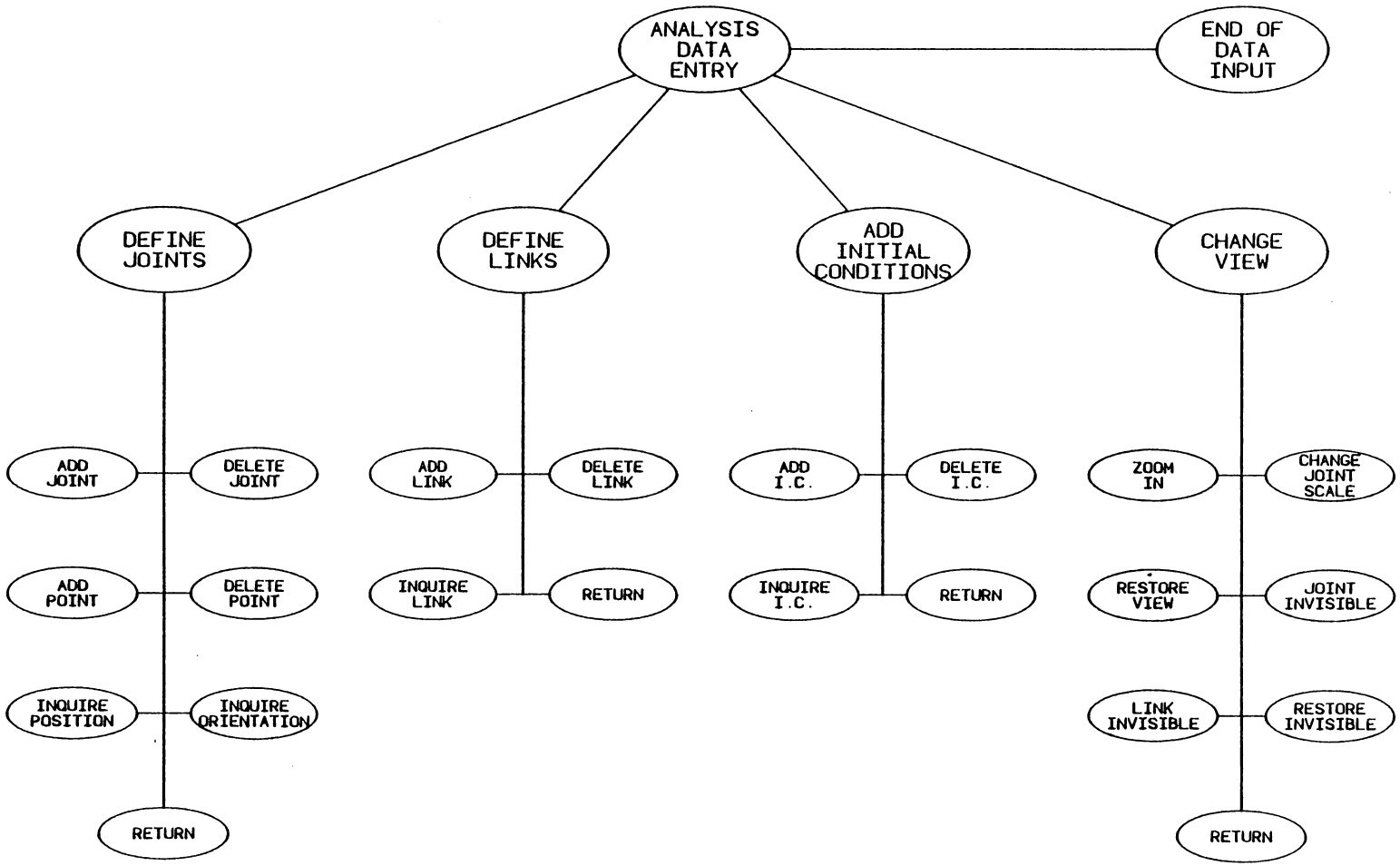


Figure 14. Data Entry Menu Hierarchy for Analysis

data along with the option of ending data input. These four data entry classes are the entry of joints, the entry of links, the entry of mechanism initial conditions, and the changing of data viewing parameters.

The first data entry class allows the user to enter or delete the location and orientation of revolute, prismatic, cylindric, spheric, screw, flat, and gear pair joints. In addition, the user is also asked to enter ground joints for the purpose of attaching joint local coordinate systems to ground.

The orientation entered is defined as the z-axis of the joint. This local joint z-axis is defined as the axis of rotation for revolute, cylindric, gear pair, and screw joints and the axis of sliding for the prismatic joint. The z-axis is also defined as the normal to the plane of sliding for the flat joint. For a spheric joint, the user enters two arbitrarily oriented z-axes. Figure 15 shows the icons and the z-axis definitions used for each joint type.

Also included in the first menu class is the option for the addition and deletion of points. These points are used as references for defining the orientations of joints. Once a point has been entered, the user may orient a local joint coordinate system toward the point by picking the point icon.

The final options in the add joint menu class are inquiry functions. These inquiry functions allow the user to inquire the position of a previously defined joint or point. The user also has the option of inquiring the orientation of a previously defined joint.

The second analysis menu class presented in Figure 14 allows the user to add and delete links between previously entered joints to define the mechanism connectivity. With the

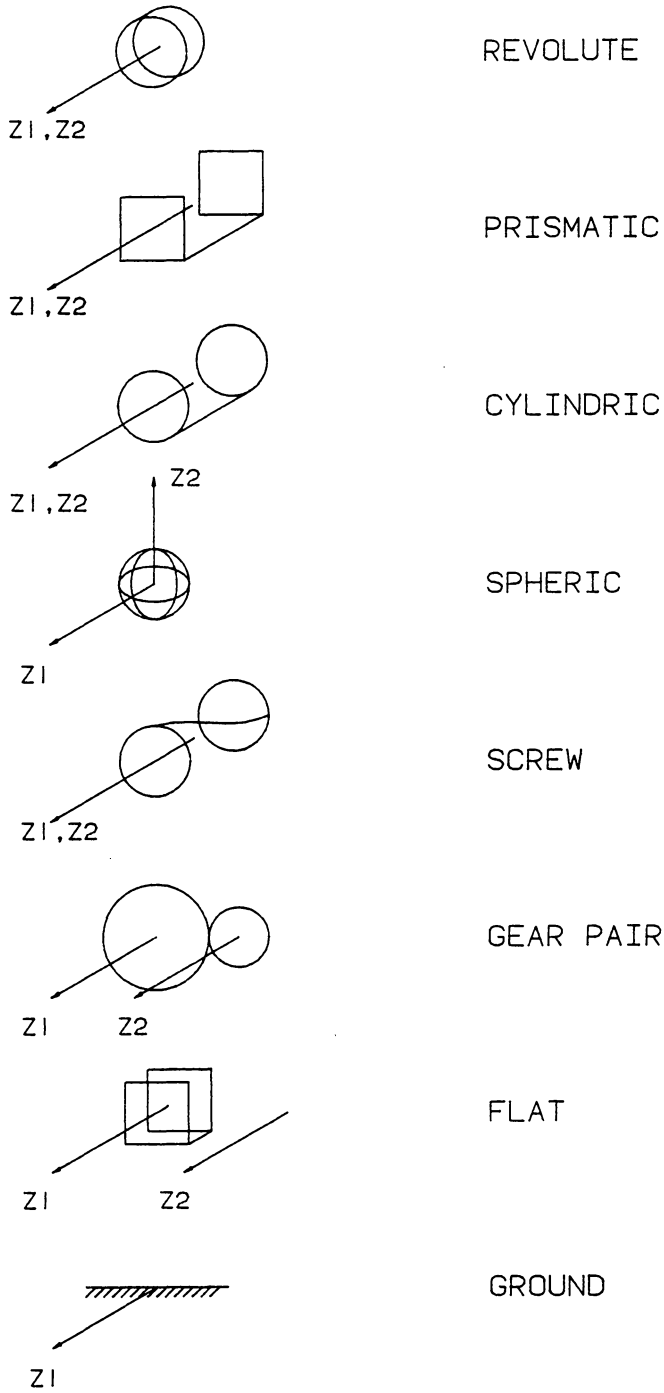


Figure 15. Joint Icons and Orientations

addition of links, the user completely specifies the local coordinate systems at each end of the link using a previously defined z-axis at a joint and by specifying the direction of the x axis. The user is also required to create a physical link between the the desired grounded joint axis and the previously discussed ground joint.

Also included in the add link menu class is an inquiry option. With this inquiry option, the user may request the name of previously defined links. This option is useful for the addition of multiple joints on a single link for multiple loop mechanisms. MECHIN can currently handle as many a six joints on a single link.

The third analysis menu class presented in Figure 14 allows the addition and deletion of kinematic initial conditions for revolute and prismatic joints. These initial conditions include the initial position, step increment, and the number of steps that an input joint will move through during a kinematic analysis. Input velocities and accelerations can also be entered. Again, an inquiry function is included to allow the inquiry of previously defined initial conditions.

The fourth and final menu class presented in Figure 14 allows the user to change the viewing parameters of the model under construction. This includes zooming in on sections of the model and scaling the size of joints. Also included is an invisibility function which allows the user to make previously defined joint and link icons invisible and visible to aid in model viewing and creation.

For data entry for synthesis, a description of the menu hierarchy for data entry is presented in Figure 16. This menu hierarchy is similar to that of the analysis data menu hierarchy and allows the user to enter two classes of data along with the option of

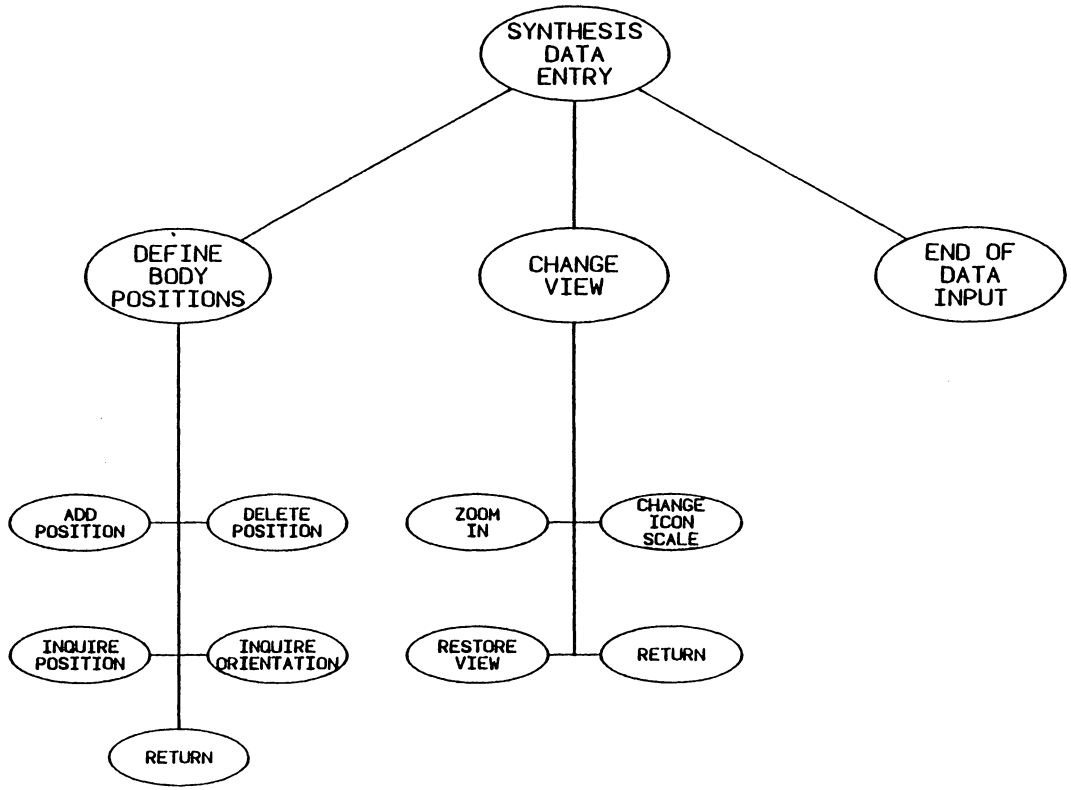


Figure 16. Data Entry Menu Hierarchy for Synthesis

ending data input. These two data entry options are the addition of body positions and the changing of viewing parameters.

In the addition of body positions, the user has the option of entering or deleting body positions used for body guidance synthesis. Each body position is described by entering the location and the orientation of a local coordinate system similar to the addition of joints in the analysis portion of MECHIN. The user also has the option of inquiring existing body locations and orientations.

The change of viewing parameters menu option allows the user to change the view of body positions previously entered. This includes zooming in on a section of the work area and scaling the size of the body position icons.

When all data has been entered for either analysis or synthesis, the end of data input menu selection is picked. MECHIN then requests the name of the graphical restart file to be written. The user then enters the name of a data file and the analysis or synthesis processing code for which the data will be formatted.

Examples of Analysis Data Input

In this section, several examples of mechanisms created in MECHIN are presented along with brief descriptions of the the data entry procedure. For the input of analysis data, an RSCR spatial mechanism, a slider crank mechanism, and a geared piston machine mechanism are shown.

In Figure 17 and Figure 18, a drawing and a MECHIN representation of a revolute-spheric-cylindric-revolute (RSCR) mechanism are presented [25]. To create this mechanism, the user first positions and orients the four mechanism joints and the two ground joints. It should be noted that when entering a spheric joint, MECHIN requests the orientation of both joint z-axes. For the rest of the joints in this model, the second z-axis is required to be collinear with the first z-axis.

After all joints and joint orientations are entered, the links between the joints are specified. To add a link, MECHIN first requests a name for the link being entered. The user is then requested to pick the z-axis of the local coordinate system at the first end of the link. To completely specify the local coordinate system at the first end of the link, MECHIN requests a vector in the X-Z plane of this local coordinate system from which the x-axis and the y-axis of the local coordinate system are calculated. This procedure is repeated for the other end of the link and the link is drawn.

The user then continues to add joints and links as desired to complete the mechanism. Once the mechanism is completed, the user selects the end of data input and MECHIN requests a restart file name and the type and name of an analysis data file. An IMP analysis file created for the mechanism shown in Figure 18 appears in Figure 19.

An example of a two loop mechanism is shown in Figure 20. This slider-crank mechanism demonstrates MECHIN's capability for handling mechanisms with multiple loops. To create a multiple loop mechanism, an additional joint is added to a previously defined link. This procedure can be implemented for as many as six joints per link.

A final example for analysis data input is shown in Figure 21 with a geared piston machine mechanism. Data entry for this model is handled the same as any other MECHIN

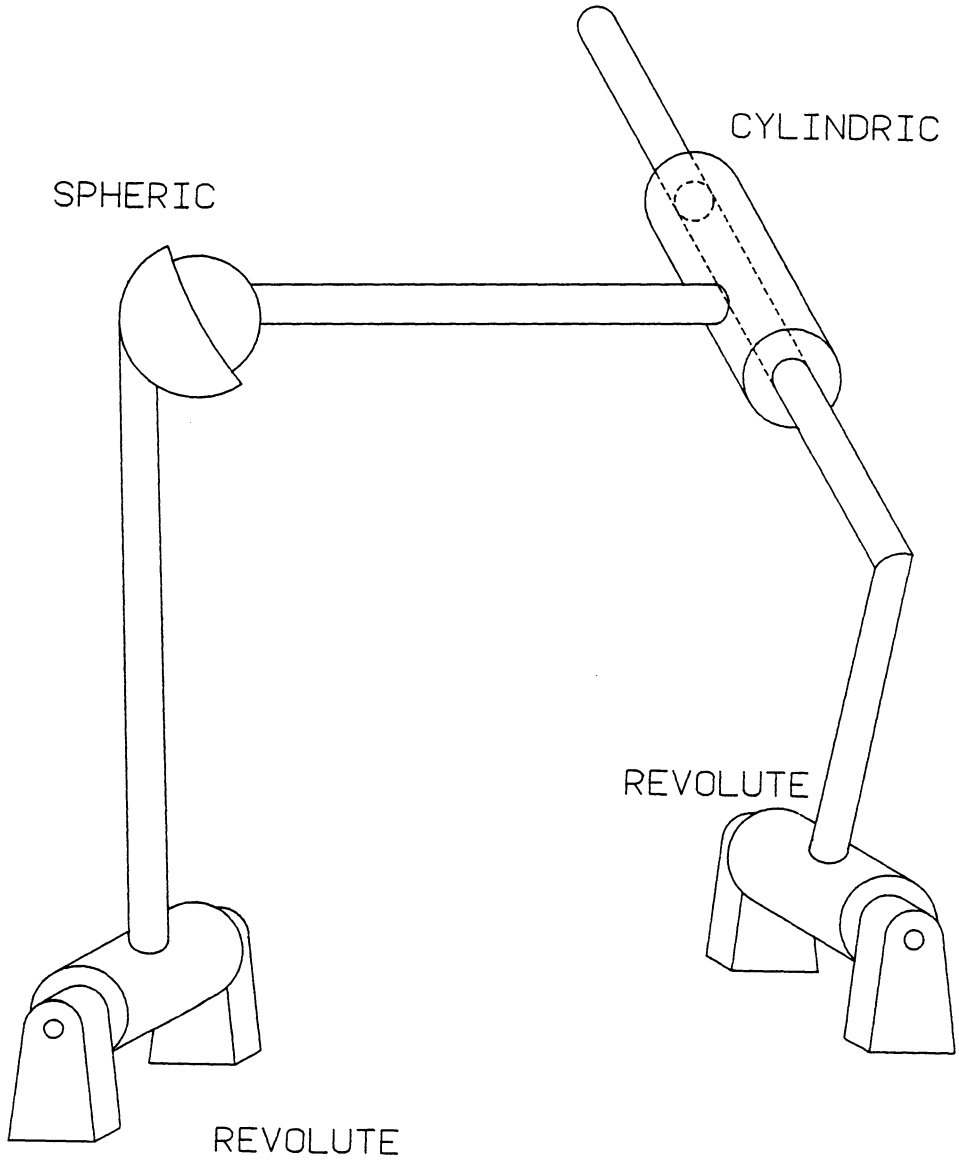


Figure 17. RSCR Mechanism

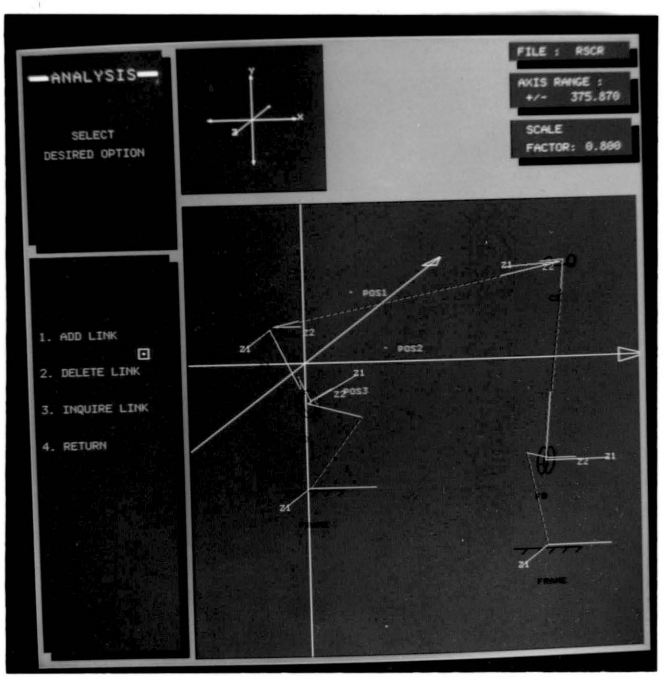


Figure 18. MECHIN Representation of an RSCR Mechanism

```
GROUND=FRAME
REVOLUTE(FRAME   ,LNK2    )=A0
SPHERE  (LNK2    ,LNK3    )=B1
CYLINDER(LNK3    ,LNK4    )=C1
REVOLUTE(LNK4    ,FRAME   )=F0
DATA: LINK   (FRAME   ,A0      ) =$
                7.21298027      , -41.0877991      ,  10.2034702      /$
                8.06142998      , -40.5964050      ,  10.4001293      /$
                7.71887016      , -41.5279999      ,  9.46165943
DATA: LINK   (LNK2    ,A0      ) =$
                7.21298027      , -41.0877991      ,  10.2034702      /$
                8.06142998      , -40.5964050      ,  10.4001293      /$
                6.73476982      , -40.2168579      ,  10.0903902
DATA: LINK   (LNK2    ,B1      ) =$
               -35.9375000      ,  37.5000000      , 0.000000000E+00 /$
               -35.9375000      ,  37.5000000      ,  1.00000000      /$
               -35.4592896      ,  36.6290588      , 0.113080025
DATA: LINK   (LNK3    ,B1      ) =$
               -35.9375000      ,  37.5000000      , 0.000000000E+00 /$
               -34.9375000      ,  37.5000000      , 0.000000000E+00 /$
               -35.3439789      ,  37.4496155      ,-0.803250015
DATA: LINK   (LNK3    ,C1      ) =$
                170.141846      ,  20.0068054      , -278.905273      /$
                169.640198      ,  20.2145844      , -278.065430      /$
                169.548325      ,  20.0571747      , -278.101807
DATA: LINK   (LNK4    ,C1      ) =$
                170.141846      ,  20.0068054      , -278.905273      /$
                169.640198      ,  20.2145844      , -278.065430      /$
                170.141846      ,  19.0360718      , -278.665039
DATA: LINK   (LNK4    ,F0      ) =$
                170.141846      , -182.863556      , -228.708023      /$
                171.141846      , -182.863556      , -228.708023      /$
                170.141846      , -181.892822      , -228.948212
DATA: LINK   (FRAME   ,F0      ) =$
                170.141846      , -182.863556      , -228.708023      /$
                171.141846      , -182.863556      , -228.708023      /$
                169.635956      , -182.423340      , -227.966202
DATA: POSITION(A0      ) =$
                0.000000000E+00 ,  10.0000000      ,          36
DATA: VELOCITY(A0      ) =$
                0.000000000E+00
DATA: ACCEL    (A0      ) =$
                0.000000000E+00
 STORE: POSITION (A0      )
 STORE: VELOCITY (A0      )
 STORE: ACCELER  (A0      )
POINT(LNK2    ) = PT22
POINT(LNK3    ) = PT23
DATA: POINT    (PT22,B1      ) = 0., 0., 0.
DATA: POINT    (PT23,B1      ) = 0., 0., 0.
STORE: POSITION (PT22,PT23)
STORE: VELOCITY (PT22,PT23)
STORE: ACCELER  (PT22,PT23)
POINT(LNK3    ) = PT33
POINT(LNK4    ) = PT34
DATA: POINT    (PT33,C1      ) = 0., 0., 0.
DATA: POINT    (PT34,C1      ) = 0., 0., 0.
STORE: POSITION (PT33,PT34)
STORE: VELOCITY (PT33,PT34)
STORE: ACCELER  (PT33,PT34)
 STORE: POSITION (F0      )
 STORE: VELOCITY (F0      )
 STORE: ACCELER  (F0      )
 RETURN
```

Figure 19. MECHIN Generated IMP Code for an RSCR Mechanism

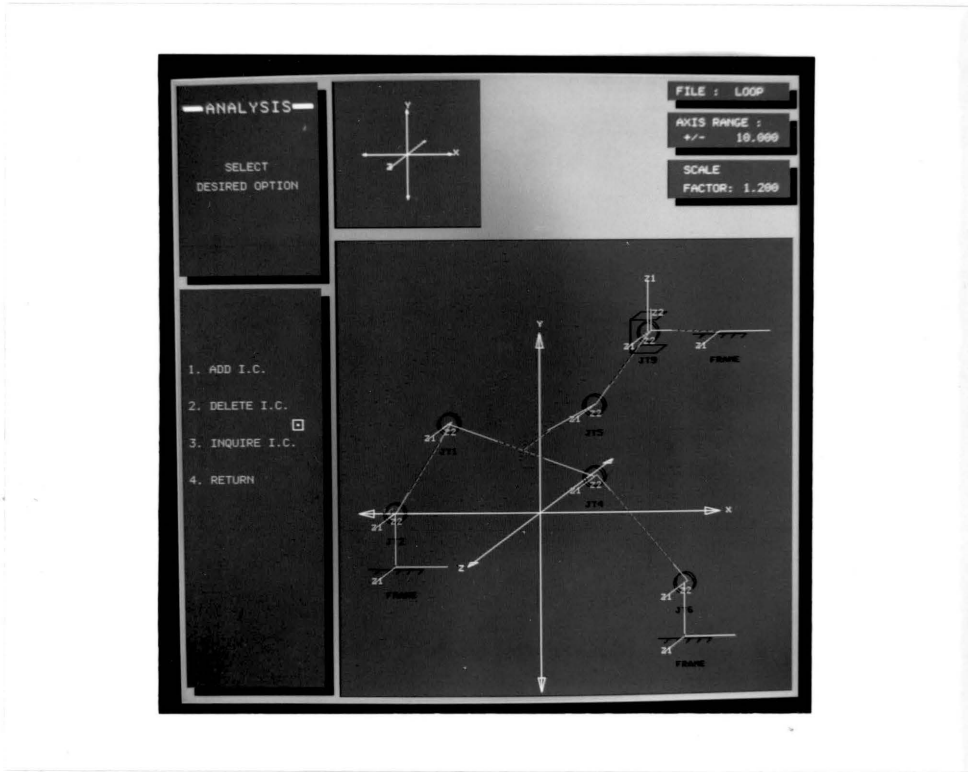


Figure 20. Slider-Crank Mechanism

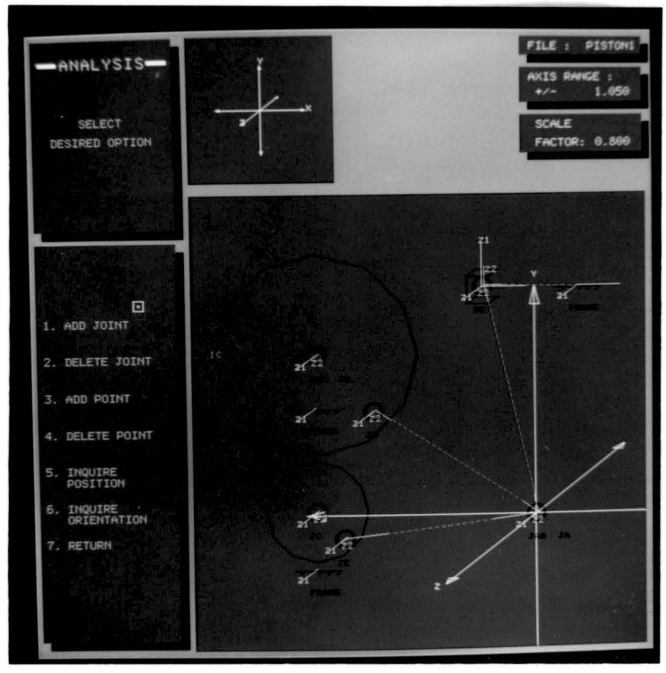


Figure 21. Geared Piston Machine Mechanism

data input where the user enters joint and link connectivity data. To aid in the creation of this model, the invisibility routines for joints and links were used to connect links to multiple joints with the same location and orientation. The invisibility functions were also used in this model to eliminate links to avoid clutter for presentation purposes.

Examples of Synthesis Data Input

For the input of synthesis data, the user is required to enter the body positions and orientations that the synthesized mechanism is to pass through. Examples of these body positions are shown in Figure 22.

To enter a body position, the user is first asked for the body position name. The user then enters the desired location of the body position. To complete the entry, the user defines the local coordinate system for the body position. This is accomplished by entering the local x-axis along with a vector in the local X-Y plane, from which the local y-axis and z-axis are calculated.

The above procedure is followed until all desired body positions are entered, at which time the end of data input menu item is selected. The user is then queried for a MECHIN graphical restart file name and the processing code to which the data will be written. MECHIN currently supports data entry to the synthesis processing code RSCR developed at Virginia Tech.

Once a desired synthesis processing code is chosen, MECHIN will query the user to pick either all or a subset of the body positions entered, depending on the number of body

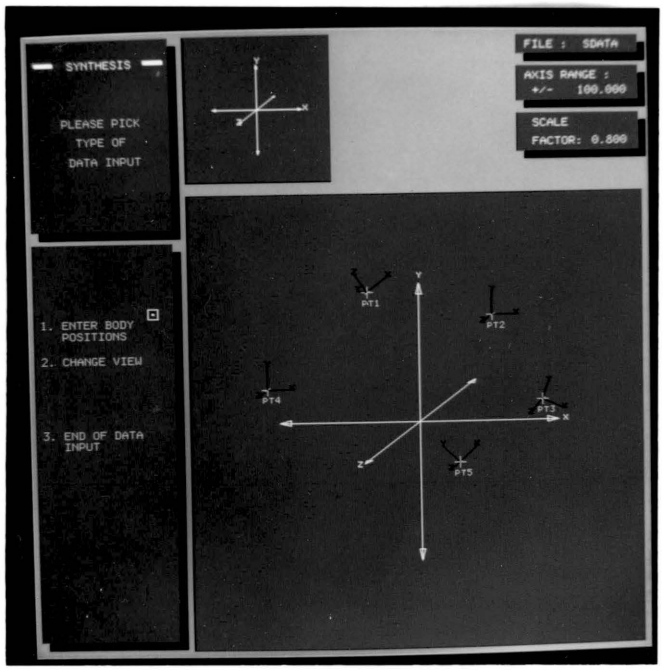


Figure 22. Body Positions for Mechanism Synthesis

positions the specific synthesis processor needs. Any additional data needed by the synthesis processor will also be queried at this time.

Chapter 6

CONCLUSIONS AND RECOMMENDATIONS

This thesis has presented the development and use of MECHIN, a new interactive graphical preprocessor for spatial mechanism analysis and synthesis. MECHIN was not only developed to be a useful data entry tool, but was designed to be graphics device-independent to make the program portable. It was also designed to be a general preprocessor and not structured for the data input of any particular processing code.

The use of MECHIN has proved to be an effective and beneficial method for data entry of spatial mechanism analysis and synthesis parameters. The mechanism design process is enhanced by allowing the user to graphically visualize mechanism data as it is entered into the computer. The user is also freed of the burden of writing input code for a variety of mechanism processing codes. Lastly, MECHIN offers graphical restart capabilities to allow the user to make minor changes in previously defined models to aid in the iterative design process.

The program structure of MECHIN has also been described and presented along with a complete program listing to allow future modifications and additions. Additional work needs to done to include more processing codes for both analysis and synthesis to make the mechanism design process more versatile. Work must also be done to tie MECHIN in with complete postprocessing software, including solid model generation and rendering along with mechanism animation, to complete the mechanism design cycle.

Future work may include the improvement of the data input strategies and may include the use of six-degree of freedom input devices with which the user enters both position and orientation data simultaneously. Work may also include development and improvement of three-dimensional viewing techniques including stereoscopic computer graphics displays and possibly the use of holographic display screens as computer graphic display technology develops.

BIBLIOGRAPHY

1. Sheth, P.N., and Uicker, J.J., "IMP (Integrated Mechanisms Program), A Computer-Aided Design and Analysis System for Mechanisms and Linkages," *Trans. of the ASME, Journal of Eng. for Ind.*, Vol 94, May 1972, pp. 454-464.

2. Chace, M.A., "DAMN-A Prototype Program for the Dynamic Analysis of Mechanical Networks," 7th Annual Share Design Automation Workshop, San Francisco, CA, June, 1970.

3. Orlandea, N., and Chace, M.A., "Simulation of a Vehicle Suspension with the ADAMS Computer Program," SAE Paper No. 770053, 1977.

4. Paul, B., "Dynamic Analysis of Machinery Via Program DYMAC," SAE Paper No. 770049, 1977.

5. Wehage, R.A., and Haug, E.J., "Generalized Coordinate Partitioning for Dimensional Reduction in Analysis of Constrained Dynamic Systems," *ASME Journal of Mechanical Design,*, Vol 104, No. 1, Jan. 1982.

6. Rubel, A.J., and Kaufman, R.E., "KINSYN III: A New Human-Engineered System for Interactive Computer-Aided Design of Planar Linkages," *ASME Journal of Eng. for Ind.*, Vol 99, No. 2, May 1977.

7. Myklebust, A., and Tesar, D., "The Analytical Synthesis of Complex Mechanisms for Combinations of Specified Geometric or Time Derivatives up to the Fourth Order," *Journal of Eng. for Ind., Trans. ASME,*, May 1970.

8. Sivertsen, O., and Myklebust, A., "MECSYN: An Interactive Computer Graphics System for Mechanism Synthesis by Algebraic Means," ASME Design Automation Conference, Los Angeles, CA, Paper No. 80-DET-68, September 1980.

9. Erdman, A.G., and Gustafson, J.E., "LINCAGES: Interactive Computer Analysis and Graphical Enhanced Synthesis Package," ASME Mechanisms Conference, Paper No. 77-DET-5, September 1977.

10. Chuang, J.C., Strong, R.T., and Waldron, K.J., "Implementation of Solution Rectification Techniques in an Interactive Linkage Synthesis Program," *Trans. of the ASME, Journal of Mechanical Design,* July 1981.

11. Coats, B.A., and Cipra, R.J., "A Computer Graphics Technique for Creating and Animating Spatial Linkages Using IMP (Integrated Mechanisms Program)," Proceeding of the 8th Applied Mechanisms Conference, September, 1983.

12. Barris, W.C., and Riley, D.R., "The Impact of the Workstation Environment on Mechanism Synthesis Strategy," ASME Mechanisms Conference, Paper No. 86-DET-150, October, 1986.

13. Myklebust, A., Keil, M.J., and Reinholtz, C.F., "MECSYN-IMP-ANIMEC: Foundation for a New Computer-Aided Spatial Mechanism Design System," *Mechanism and Machine Theory,* Vol. 20, No. 4, 1985, pp. 257-269.

14. Spencer, R.H. "Interactive Software Design: The Human Touch," *Computers in Mechanical Engineering,* September 1985, pp. 45-48.

15. Halter, R., "Man-Machine Interface Design Challenges," *Design News,* Vol. 41, No. 16, August 1985, pp. 63-70.

16. Swezey, R.W., and Davis, E.G., "A Case Study of Human Factors Guidelines in Computer Graphics," *IEEE Computer Graphics and Applications,* Vol. 3, No. 8, November 1983, pp. 21-30.

17. Foley, J.D., Wallace, V.L., and Chan, P., "The Human Factors of Computer Graphics Interaction Techniques," *IEEE Computer Graphics and Applications,* Vol. 4, No. 11, November 1984, pp. 13-48.

18. Reilly, S.S., and Roach, J.W., "Improved Visual Design for Graphics Display," *IEEE Computer Graphics and Applications,* Vol. 4, No. 2, February 1984, pp. 42-51.

19. Galitz, W., "Human Engineering in Screen Design," *Journal of System Management,* Vol. 34, No. 5, May 1983, pp. 6-11.

20. Snowberry, K., Parkinson, S.R., and Sisson, N., "Computer Display Menus," *Ergonomics,* Vol. 26, No. 7, 1983, pp. 699-712.

21. Christ, R.E., "Review and Analysis of Color Coding Research for Visual Displays," *Human Factors,* Vol. 17, No. 6, 1975, pp. 542-570.

22. Newman, W.M., and Sproull, R.F., *Principles of Interactive Computer Graphics,* McGraw Hill, New York, 1979.

23. Foley, J.D., and Van Dam, A., *Fundamentals of Interactive Computer Graphics,* Addison-Wesley, Reading, MA, 1984.

24. Tsang, Y., *Design and Implementation of An Advanced User Interface for Three-Dimensional Conceptual Design On CAD/CAM Systems,* Master's Thesis, Virginia Polytechnic Institute and State University, Blacksburg, Va., 1986.

25. Thompson, J.M., *Computer Aided Design and Synthesis of the RSCR Spatial Mechanism,* Master's Thesis, Virginia Polytechnic Institute and State University, Blacksburg, Va., 1987.

26. IBM Programmer's Reference for graPHIGS, SC33-8104-0, Program No. 5668-792.
27. IBM Messages and Codes for graPHIGS, SC33-8105-0, Program No. 5668-792.
28. IBM Understanding graPHIGS, SC33-8102-0, Program No. 5668-792.
29. IBM Writing Applications for graPHIGS, SC33-8103-0, Program No. 5668-792.
30. SDRC IMP (Integrated Mechanisms Program) User's Manual, 1979.

Appendix A

MECHIN PROGRAM LISTING

PROGRAM MECHIN

```
************************************************************************
*  ---- MECHIN -------                                                 *
*                                                                      *
*                                                                      *
*    A PHIGS BASED PREPROCESSOR FOR SPATIAL MECHANISM ANALYSIS         *
*                                                                      *
*    AND SYNTHESIS.                                                    *
*                                                                      *
*                                                                      *
*                          BRIAN THATCH                                *
*                                                                      *
*                     DEPT. OF MECHANICAL ENG.                         *
*                  VIRGINIA POLYTECHNIC INSTITUTE                      *
*                      AND STATE UNIVERSITY                            *
*                                                                      *
*                      BLACKSBURG, VIRGINIA                            *
*                                                                      *
************************************************************************

      REAL CSIZE(3)

************************************************************************
*    OPEN AND INITIALIZE PHIGS, DISPLAY TITLE SCREEN, GET TYPE OF      *
*    DATA ENTRY (ANALYSIS OR SYNTHESIS)                                *
************************************************************************

      CALL SETUP(IWSID,ICHO,CSIZE)


************************************************************************
*    ENTER ANALYSIS OR SYNTHESIS PORTION OF MECHIN.  WRITE RESTART     *
*    AND DATA FILES                                                    *
************************************************************************

      IF(ICHO.EQ.1)THEN
C                       INPUT FOR ANALYSIS
         CALL ANIN(IWSID,CSIZE)
      ELSE
C                       INPUT FOR SYNTHESIS
         CALL SYNIN(IWSID,CSIZE)
      ENDIF


************************************************************************
*    DELETE ALL STRUCTURES AND CLOSE PHIGS                             *
************************************************************************

      CALL CLOSE(IWSID,CSIZE)


*
*  Installation Note ---
*
*    MECHIN currently uses a system dependent routine SYSCAL in
*    SUBROUTINE FILEXS.  The purpose of SYSCAL is to check the
*    existence or nonexistence of files.
*
*

      STOP
      END
```

SUBROUTINE ADDIC

```
      SUBROUTINE ADDIC(IWSID,CSIZE,AXISR,JNAM,JNUM,JOINT,INAM,INUM,IC,
     &SCALE)
************************************************************************
* -----  SUBROUTINE ADDIC -----                                        *
*                                                                      *
*                                                                      *
*    THIS SUBROUTINE IS USED TO ADD INITIAL CONDITIONS TO A MECHIN     *
*                                                                      *
*    FILE                                                              *
*                                                                      *
************************************************************************

      INTEGER IWSID,NUMJ,RFLG,JNUM(20,3),CHOICE,CLASS(2),INUM(20,2)

      REAL AXISR,LOCAT(3),AREA(6),CSIZE(3),JOINT(20,21),
     &  ENDP1(3),ENDP2(3),LNK(6),XVEC(3),YVEC(3),X1(6),X2(6),
     &  JOINTT(20,21),BOX(8),SHD(8),LX(2),LY(2),LZ(2),LX1(2),LY1(2),
     &  LZ1(2),IC(20,4)
```

```
      CHARACTER*7 JNAME,JNAM(20),CLEAR,CLEAR1,INIT,VAL,INIT2,INAM(20)
      CHARACTER*7 VAL1,VAL2,VAL3,VAL4,VAL5
      CHARACTER*20 CLEAR2
      CHARACTER*20 X,Y,Z,V,A

      DATA X      /'POSITION =          '/
      DATA Y      /'INCREMENT=          '/
      DATA Z      /'STEPS    =          '/
      DATA V      /'VELOCITY =          '/
      DATA A      /'ACCELERATION=       '/

      DATA VAL        /'       '/
      DATA VAL1       /'0.     '/
      DATA VAL2       /'0.     '/
      DATA VAL3       /'0      '/
      DATA VAL4       /'0.     '/
      DATA VAL5       /'0.     '/
      DATA INIT       /'0.     '/
      DATA INIT2      /'0      '/
      DATA CLEAR1     /'       '/
      DATA CLEAR      /'       '/
      DATA CLEAR2     /'                  '/

C                             GREY VALUATOR BOX
      DATA BOX/0.00,0.545,0.48,0.545,0.48,0.73,0.00,0.73/

      LX(1)=0.05
      LX(2)=0.68

      LX1(1)=0.30
      LX1(2)=0.68

      LY(1)=0.05
      LY(2)=0.62

      LY1(1)=0.30
      LY1(2)=0.62

      LZ(1)=0.05
      LZ(2)=0.56

      LZ1(1)=0.30
      LZ1(2)=0.56

      VAL1='0.     '
      VAL2='0.     '
      VAL3='0      '
      VAL4='0.     '
      VAL5='0.     '

      DO 10 I=2,8,2
         SHD(I)=BOX(I)-0.025
   10 CONTINUE

      DO 20 I=1,7,2
         SHD(I)=BOX(I)+0.025
   20 CONTINUE


      RFLG=0

      LOCAT(1)=0.0
      LOCAT(2)=0.0
      LOCAT(3)=0.0

      AREA(1)=0.08*CSIZE(1)
      AREA(2)=0.90*CSIZE(1)
      AREA(3)=0.70*CSIZE(1)
      AREA(4)=0.90*CSIZE(1)
      AREA(5)=0.02*CSIZE(1)
      AREA(6)=0.90*CSIZE(1)

      JNAME=CLEAR

*************************************************************************
*    SET POINT NUMBER
*************************************************************************
C                     SEARCH PNAM ARRAY FOR NEXT AVAILABLE SLOT (BBBBBBBB)
      DO 21 NUMI=1,20,1
         IF(INAM(NUMI).EQ.'BBBBBBB'.OR.INAM(NUMI).EQ.'CCCCCCC')THEN
            GO TO 22
         ELSE
            CONTINUE
         ENDIF
   21 CONTINUE
   22 CONTINUE

*************************************************************************
*  OPEN STRUCTURE FOR THE VAL. BOX (VIEW 1)
*************************************************************************
C                             OPEN A STRUCTURE
      CALL GPOPST(8)
C                             SET SOLID INTERIOR STYLE FOR GREY BOX
```

```
      CALL GPIS(2)
C                               SET COLOR FOR INTERIOR OF SHADOW BOX
      CALL GPICI(4)
C                               DRAW GREY SHADOW BOX
      CALL GPPG2(1,4,2,SHD)
C                               SET COLOR FOR INTERIOR OF GREY BOX
      CALL GPICI(3)
C                               DRAW GREY TITLE BOX
      CALL GPPG2(1,4,2,BOX)
C                               CLOSE STRUCTURE
      CALL GPCLST

*********************************************************************
*  SCREEN DISPLAY
*********************************************************************
C                               UPDATE THE WORKSTATION
      CALL GPUPWS(IWSID,2)

*********************************************************************
*    SET STRING TO REQUEST MODE
*********************************************************************

C                     REQUEST STRING
      CALL GPSTMO(IWSID,1,1,2)
      CALL GPINST(IWSID,1,7,INIT,1,AREA,7,1,0,INIT)

*********************************************************************
*    SET STRING TO REQUEST MODE
*********************************************************************

C                     REQUEST STRING
      CALL CMMAND(IWSID,72)
      CALL MENU(IWSID,CSIZE,16,CHOICE)
      KKK=CHOICE

*********************************************************************
*    PICK REVOLUTE JOINT OR PRISMATIC JOINT OR RETURN
*********************************************************************
      IF(CHOICE.EQ.1)THEN

         CALL CMMAND(IWSID,61)
   50    CALL PICKJ2(IWSID,JOINT,AXISR,CSIZE,JNUM1,ENDP1,IAXIS1,RFLG)

         IF(RFLG.EQ.1)THEN
            GO TO 100
         ENDIF

         IF(JNUM(JNUM1,2).EQ.1)THEN
C                  INQUIRE THE INITIAL POSTITION
            CALL CMMAND(IWSID,63)
            VAL1=CLEAR
            CALL GPRQST(IWSID,1,7,ISTAT,NUM,VAL1)
C                  WRITE INPUT TO THE SCREEN
            CALL GPOPST(10)
            CALL GPTXCI(1)
            CALL GPAHSC(1.0)
            CALL GPAN2(LX,20,X)
            CALL GPAN2(LX1,7,VAL1)
            CALL GPCLST

C                  INQUIRE THE INCREMENT
            CALL CMMAND(IWSID,64)
            VAL2=CLEAR
            CALL GPRQST(IWSID,1,7,ISTAT,NUM,VAL2)
C                  WRITE INPUT TO THE SCREEN
            CALL GPOPST(10)
            CALL GPTXCI(1)
            CALL GPAHSC(1.0)
            CALL GPAN2(LY,20,Y)
            CALL GPAN2(LY1,7,VAL2)
            CALL GPCLST

C                  INQUIRE THE NO. OF STEPS
            CALL CMMAND(IWSID,65)
            CALL GPINST(IWSID,1,7,INIT2,1,AREA,7,1,0,INIT2)
            VAL3=CLEAR
            CALL GPRQST(IWSID,1,7,ISTAT,NUM,VAL3)
C                  WRITE INPUT TO THE SCREEN
            CALL GPOPST(10)
            CALL GPTXCI(1)
            CALL GPAHSC(1.0)
            CALL GPAN2(LZ,20,Z)
            CALL GPAN2(LZ1,7,VAL3)
            CALL GPCLST

            GO TO 80
         ENDIF

         IF(JNUM(JNUM1,2).EQ.2)THEN
C                  PRISMATIC JOINT
C                  INQUIRE THE INITIAL POSTITION
            CALL CMMAND(IWSID,67)
            VAL1=CLEAR
```

```
            CALL GPRQST(IWSID,1,7,ISTAT,NUM,VAL1)
C                 WRITE INPUT TO THE SCREEN
            CALL GPOPST(10)
            CALL GPTXCI(1)
            CALL GPAHSC(1.0)
            CALL GPAN2(LX,20,X)
            CALL GPAN2(LX1,7,VAL1)
            CALL GPCLST
C                 INQUIRE THE INCREMENT
            CALL CMMAND(IWSID,68)
            VAL2=CLEAR
            CALL GPRQST(IWSID,1,7,ISTAT,NUM,VAL2)
C                 WRITE INPUT TO THE SCREEN
            CALL GPOPST(10)
            CALL GPTXCI(1)
            CALL GPAHSC(1.0)
            CALL GPAN2(LY,20,Y)
            CALL GPAN2(LY1,7,VAL2)
            CALL GPCLST
C                 INQUIRE THE NO. OF STEPS
            CALL CMMAND(IWSID,65)
            CALL GPINST(IWSID,1,7,INIT2,1,AREA,7,1,0,INIT2)
            VAL3=CLEAR
            CALL GPRQST(IWSID,1,7,ISTAT,NUM,VAL3)
C                 WRITE INPUT TO THE SCREEN
            CALL GPOPST(10)
            CALL GPTXCI(1)
            CALL GPAHSC(1.0)
            CALL GPAN2(LZ,20,Z)
            CALL GPAN2(LZ1,7,VAL3)
            CALL GPCLST
            GO TO 80
         ENDIF

         IF(JNUM(JNUM1,2).NE.1.AND.JNUM(JNUM1,2).NE.2)THEN
C                 INVALID JOINT
            CALL CMMAND(IWSID,62)
            GO TO 50
         ENDIF

         ENDIF

      IF(CHOICE.EQ.2)THEN

         CALL CMMAND(IWSID,61)
  150    CALL PICKJ2(IWSID,JOINT,AXISR,CSIZE,JNUM1,ENDP1,IAXIS1,RFLG)

         IF(RFLG.EQ.1)THEN
            GO TO 100
         ENDIF

         IF(INAM(JNUM1).EQ.'BBBBBBB'.OR.INAM(JNUM1).EQ.'CCCCCCC')THEN
C                 POSITION MUST BE SPECIFIED BEFORE ACC AND VELOCITY
              CALL CMMAND(IWSID,79)
              GO TO 150
         ENDIF

         IF(JNUM(JNUM1,2).EQ.1)THEN
C                 INQUIRE THE VELOCITY
            CALL CMMAND(IWSID,73)
            VAL4=CLEAR
            CALL GPRQST(IWSID,1,7,ISTAT,NUM,VAL4)
C                 WRITE INPUT TO THE SCREEN
            CALL GPOPST(10)
            CALL GPTXCI(1)
            CALL GPAHSC(1.0)
            CALL GPAN2(LX,20,V)
            CALL GPAN2(LY1,7,VAL4)
            CALL GPCLST

            GO TO 80
         ENDIF

         IF(JNUM(JNUM1,2).EQ.2)THEN
C                 PRISMATIC JOINT
C                 INQUIRE THE VELOCITY
            CALL CMMAND(IWSID,74)
            VAL4=CLEAR
            CALL GPRQST(IWSID,1,7,ISTAT,NUM,VAL4)
C                 WRITE INPUT TO THE SCREEN
            CALL GPOPST(10)
            CALL GPTXCI(1)
            CALL GPAHSC(1.0)
            CALL GPAN2(LX,20,V)
            CALL GPAN2(LY1,7,VAL4)
            CALL GPCLST
         ENDIF

         IF(JNUM(JNUM1,2).NE.1.AND.JNUM(JNUM1,2).NE.2)THEN
C                 INVALID JOINT
            CALL CMMAND(IWSID,62)
            GO TO 150
         ENDIF
```

```
      ENDIF
      IF(CHOICE.EQ.3)THEN
         CALL CMMAND(IWSID,61)
  250    CALL PICKJ2(IWSID,JOINT,AXISR,CSIZE,JNUM1,ENDP1,IAXIS1,RFLG)
         IF(INAM(JNUM1).EQ.'BBBBBBB'.OR.INAM(JNUM1).EQ.'CCCCCCC')THEN
C                POSITION MUST BE SPECIFIED BEFORE ACC AND VELOCITY
           CALL CMMAND(IWSID,79)
           GO TO 150
         ENDIF

         IF(RFLG.EQ.1)THEN
           GO TO 100
         ENDIF
         IF(JNUM(JNUM1,2).EQ.1)THEN
C                INQUIRE THE ACCEL
           CALL CMMAND(IWSID,75)
           VAL5=CLEAR
           CALL GPRQST(IWSID,1,7,ISTAT,NUM,VAL5)
C                WRITE INPUT TO THE SCREEN
           CALL GPOPST(10)
           CALL GPTXCI(1)
           CALL GPAHSC(1.0)
           CALL GPAN2(LX,20,A)
           CALL GPAN2(LY1,7,VAL5)
           CALL GPCLST

           GO TO 80
         ENDIF

         IF(JNUM(JNUM1,2).EQ.2)THEN
C                PRISMATIC JOINT
C                INQUIRE THE ACCEL
           CALL CMMAND(IWSID,76)
           VAL5=CLEAR
           CALL GPRQST(IWSID,1,7,ISTAT,NUM,VAL5)
C                WRITE INPUT TO THE SCREEN
           CALL GPOPST(10)
            CALL GPTXCI(1)
           CALL GPAHSC(1.0)
           CALL GPAN2(LX,20,A)
           CALL GPAN2(LY1,7,VAL5)
           CALL GPCLST
         ENDIF

         IF(JNUM(JNUM1,2).NE.1.AND.JNUM(JNUM1,2).NE.2)THEN
C                INVALID JOINT
           CALL CMMAND(IWSID,62)
           GO TO 250
         ENDIF

      ENDIF
*************************************************************************
*    ACCEPT OR REJECT INPUT
*************************************************************************

   80 CALL CMMAND(IWSID,66)
      CALL MENU(IWSID,CSIZE,15,CHOICE)
      IF(CHOICE.EQ.1)THEN
C          FILL IC ARRAYS
         IF(KKK.EQ.1)THEN
           INUM(JNUM1,1)=JNUM1
           INAM(JNUM1)='DDDDDDD'
           REWIND 10
           WRITE(10,81)CLEAR2
           WRITE(10,81)CLEAR2
           WRITE(10,81)CLEAR2
   81      FORMAT(A20)
           REWIND 10
           WRITE(10,82)VAL1
           WRITE(10,82)VAL2
           WRITE(10,82)VAL3
   82      FORMAT(A7)
           REWIND 10
           READ(10,*)IC(JNUM1,1)
           READ(10,*)IC(JNUM1,2)
           READ(10,*)INUM(JNUM1,2)
C          DRAW AN ICON
           CALL ICICON(IWSID,JOINT,INAM,INUM,IC,SCALE)
         ENDIF
         IF(KKK.EQ.2)THEN
           REWIND 10
           WRITE(10,83)CLEAR2
   83      FORMAT(A20)
           REWIND 10
           WRITE(10,84)VAL4
   84      FORMAT(A7)
           REWIND 10
           READ(10,*)IC(JNUM1,3)
         ENDIF
         IF(KKK.EQ.3)THEN
```

```
            REWIND 10
            WRITE(10,85)CLEAR2
   85       FORMAT(A20)
            REWIND 10
            WRITE(10,86)VAL5
   86       FORMAT(A7)
            REWIND 10
            READ(10,*)IC(JNUM1,4)
         ENDIF
      ENDIF

  100 CALL GPEST(8)
      CALL GPEST(10)
      RETURN
      END
```

SUBROUTINE ADDJ

```
      SUBROUTINE ADDJ(IWSID,CSIZE,AXISR,JNAM,JNUM,JOINT,POINT,SCALE)
*****************************************************************************
* ----- SUBROUTINE ADDP -----                                                *
*                                                                            *
*                                                                            *
*   THIS SUBROUTINE IS USED TO ADD A JOINT TO A MECHIN FILE                  *
*                                                                            *
*****************************************************************************

      COMMON/JDEF/JNDEF(20)
      INTEGER IWSID,NUMJ,JNUM(20,3),RFLG,TYPE,CHOICE

      REAL AXISR,LOCAT(3),ORIENT(3),AREA(6),JOINT(20,21),CSIZE(3),
     &POINT(100,3),ORIEN2(3),LOCAT2(3),SCALE

      CHARACTER*7 JNDEF,JNAME,JNAM(20),CLEAR,CLEAR1
      CHARACTER*20 CLEAR2


      DATA CLEAR     /'       '/
      DATA CLEAR1    /'       '/
      DATA CLEAR2    /'                    '/

      RFLG=0

      LOCAT(1)=0.0
      LOCAT(2)=0.0
      LOCAT(3)=0.0

      AREA(1)=0.08*CSIZE(1)
      AREA(2)=0.90*CSIZE(1)
      AREA(3)=0.70*CSIZE(1)
      AREA(4)=0.90*CSIZE(1)
      AREA(5)=0.02*CSIZE(1)
      AREA(6)=0.90*CSIZE(1)

      JNAME=CLEAR

*****************************************************************************
*    GET JOINT TYPE
*****************************************************************************
C                  CALL PICK JOINT TYPE COMMAND AND MENU
      CALL CMMAND(IWSID,26)
      CALL MENU(IWSID,CSIZE,6,CHOICE)
C                  FIND OUT IF RETURN
      IF(CHOICE.EQ.9)THEN
         GO TO 1000
      ELSE
         TYPE=CHOICE
      ENDIF

*****************************************************************************
*    SET JOINT NUMBER
*****************************************************************************
C                  SEARCH PNAM ARRAY FOR NEXT AVAILABLE SLOT (BBBBBBBB)
      DO 10 NUMJ=1,20,1
         IF(JNAM(NUMJ).EQ.'BBBBBBB'.OR.JNAM(NUMJ).EQ.'CCCCCCC')THEN
            GO TO 20
         ELSE
            CONTINUE
         ENDIF
   10 CONTINUE
   20 IF(NUMJ.EQ.20)THEN
C                  NO MORE JOINTS AVAILABLE
         CALL CMMAND(IWSID,23)
C                  GET STROKE
         CALL GPINST(IWSID,1,7,CLEAR,1,AREA,7,1,0,CLEAR)
         CALL GPRQST(IWSID,1,7,ISTAT,NUM,CLEAR1)
         GO TO 1000
      ENDIF
```

```
*********************************************************************
*    GET THE JOINT NAME OR DEFAULT
*********************************************************************
      IF(TYPE.EQ.8)THEN
C                    GROUND LINK
         JNAME='FRAME'
         GO TO 500
      ENDIF
C                    REQUEST STRING
      CALL CMMAND(IWSID,24)
C                    INITIALIZE THE STRING DEVICE
      CALL GPSTMO(IWSID,1,1,2)
   25 CALL GPINST(IWSID,1,7,JNDEF(NUMJ),1,AREA,7,1,0,JNDEF(NUMJ))
C                    INQUIRE JOINT NAME
      CALL GPRQST(IWSID,1,7,ISTAT,NUM,JNAME)
      CALL GPSTMO(IWSID,1,1,1)

*********************************************************************
*    CHECK TO SEE IF NAME CAN BE USED
*********************************************************************
      DO 30 J=1,20
         IF(JNAME.EQ.JNAM(J))THEN
C                    NAME ALREADY EXISTS
              CALL CMMAND(IWSID,25)
              JNAME=CLEAR
              GO TO 25
         ENDIF
   30 CONTINUE

*********************************************************************
*    POSITION THE JOINT
*********************************************************************

      CALL POSIT(IWSID,AXISR,CSIZE,LOCAT,RFLG)
      IF(RFLG.EQ.1)THEN
         GO TO 1000
      ENDIF

*********************************************************************
*    DRAW THE JOINT
*********************************************************************
C
C     ORIENT(1)=1.00
C     ORIENT(2)=0.00
C     ORIENT(3)=0.00
C
C     CALL DRAWJT(IWSID,LOCAT,AXISR,JNAME,NUMJ,TYPE,ORIENT)

*********************************************************************
*    ORIENT THE JOINT
*********************************************************************

      CALL ORIEN(IWSID,LOCAT,POINT,JOINT,AXISR,CSIZE,ORIENT,RFLG)
      IF(RFLG.EQ.1)THEN
         GO TO 1000
      ENDIF

*********************************************************************
*    SPECIAL CASES (SCREW,SPHERICAL,GEAR,AND FLAT)
*********************************************************************
      IF(TYPE.EQ.1)THEN
C                    REVOLUTE
         JOINT(NUMJ,15)=LOCAT(1)
         JOINT(NUMJ,16)=LOCAT(2)
         JOINT(NUMJ,17)=LOCAT(3)
         JOINT(NUMJ,18)=ORIENT(1)
         JOINT(NUMJ,19)=ORIENT(2)
         JOINT(NUMJ,20)=ORIENT(3)
      ENDIF
      IF(TYPE.EQ.2.OR.TYPE.EQ.3.OR.TYPE.EQ.5)THEN
C                    PRIS.,CYLINDRIC,SCREW (INQUIRE INITIAL Z DISPLACEMENT)
         CALL CMMAND(IWSID,40)
         CALL CDATA(IWSID,CSIZE,Z,RFLG)
         JOINT(NUMJ,14)=Z
         JOINT(NUMJ,15)=LOCAT(1)
         JOINT(NUMJ,16)=LOCAT(2)
         JOINT(NUMJ,17)=LOCAT(3)
         JOINT(NUMJ,18)=ORIENT(1)
         JOINT(NUMJ,19)=ORIENT(2)
         JOINT(NUMJ,20)=ORIENT(3)
      ENDIF
      IF(TYPE.EQ.4)THEN
C                    SPHERICAL JOINT(INQUIRE SECOND LOCAL Z AXIS)
         CALL CMMAND(IWSID,42)
         CALL MENU(IWSID,CSIZE,9,CHOICE)
         IF(CHOICE.EQ.1)THEN
              JOINT(NUMJ,15)=LOCAT(1)
              JOINT(NUMJ,16)=LOCAT(2)
              JOINT(NUMJ,17)=LOCAT(3)
              JOINT(NUMJ,18)=ORIENT(1)
              JOINT(NUMJ,19)=ORIENT(2)
              JOINT(NUMJ,20)=ORIENT(3)
         ENDIF
         IF(CHOICE.EQ.2)THEN
             CALL ORIEN(IWSID,LOCAT,POINT,JOINT,AXISR,CSIZE,ORIEN2,RFLG)
```

```
                IF(RFLG.EQ.1)THEN
                      GO TO 1000
                ENDIF
                JOINT(NUMJ,15)=LOCAT(1)
                JOINT(NUMJ,16)=LOCAT(2)
                JOINT(NUMJ,17)=LOCAT(3)
                JOINT(NUMJ,18)=ORIEN2(1)
                JOINT(NUMJ,19)=ORIEN2(2)
                JOINT(NUMJ,20)=ORIEN2(3)
           ENDIF
      ENDIF
      IF(TYPE.EQ.5)THEN
C                     SCREW JOINT(INQUIRE SCREW LEAD)
           CALL CMMAND(IWSID,41)
           CALL CDATA(IWSID,CSIZE,Z,RFLG)
           JOINT(NUMJ,13)=Z
      ENDIF
      IF(TYPE.EQ.6)THEN
C                     FLAT JOINT(INQUIRE POSITION OF THE SECOND LOCAL SYS.
           CALL CMMAND(IWSID,43)
           CALL MENU(IWSID,CSIZE,10,CHOICE)
           IF(CHOICE.EQ.1)THEN
                JOINT(NUMJ,15)=LOCAT(1)
                JOINT(NUMJ,16)=LOCAT(2)
                JOINT(NUMJ,17)=LOCAT(3)
                JOINT(NUMJ,18)=ORIENT(1)
                JOINT(NUMJ,19)=ORIENT(2)
                JOINT(NUMJ,20)=ORIENT(3)
           ENDIF
           IF(CHOICE.EQ.2)THEN
                CALL POSIT(IWSID,AXISR,CSIZE,LOCAT2,RFLG)
                IF(RFLG.EQ.1)THEN
                      GO TO 1000
                ENDIF
                JOINT(NUMJ,15)=LOCAT2(1)
                JOINT(NUMJ,16)=LOCAT2(2)
                JOINT(NUMJ,17)=LOCAT2(3)
                JOINT(NUMJ,18)=ORIENT(1)
                JOINT(NUMJ,19)=ORIENT(2)
                JOINT(NUMJ,20)=ORIENT(3)
           ENDIF
      ENDIF
      IF(TYPE.EQ.7)THEN
C                     GEAR PAIR(INQUIRE POSITION OF THE SECOND LOCAL SYS.
C                               AND THE GEAR RATIO)
           CALL CMMAND(IWSID,44)
           CALL MENU(IWSID,CSIZE,10,CHOICE)
           IF(CHOICE.EQ.1)THEN
                JOINT(NUMJ,15)=LOCAT(1)
                JOINT(NUMJ,16)=LOCAT(2)
                JOINT(NUMJ,17)=LOCAT(3)
                JOINT(NUMJ,18)=ORIENT(1)
                JOINT(NUMJ,19)=ORIENT(2)
                JOINT(NUMJ,20)=ORIENT(3)
           ENDIF
           IF(CHOICE.EQ.2)THEN
                CALL POSIT(IWSID,AXISR,CSIZE,LOCAT2,RFLG)
                IF(RFLG.EQ.1)THEN
                      GO TO 1000
                ENDIF
                JOINT(NUMJ,15)=LOCAT2(1)
                JOINT(NUMJ,16)=LOCAT2(2)
                JOINT(NUMJ,17)=LOCAT2(3)
                JOINT(NUMJ,18)=ORIENT(1)
                JOINT(NUMJ,19)=ORIENT(2)
                JOINT(NUMJ,20)=ORIENT(3)
           ENDIF

           CALL CMMAND(IWSID,45)
           CALL CDATA(IWSID,CSIZE,Z,RFLG)
           JOINT(NUMJ,21)=Z
      ENDIF
  500 IF(TYPE.EQ.8)THEN
C                     GROUND LINK (INQUIRE POSITION)
           CALL POSIT(IWSID,AXISR,CSIZE,LOCAT,RFLG)
           ORIENT(1)=0.0
           ORIENT(2)=0.0
           ORIENT(3)=1.0
           IF(RFLG.EQ.1)THEN
                GO TO 1000
           ENDIF
      ENDIF

*************************************************************************
*    ADD POINT DATA TO JOINT ARRAYS
*************************************************************************

      JNAM(NUMJ)=JNAME
      JNUM(NUMJ,1)=NUMJ
      JNUM(NUMJ,2)=TYPE
      JNUM(NUMJ,3)=0
      JOINT(NUMJ,1)=LOCAT(1)
      JOINT(NUMJ,2)=LOCAT(2)
      JOINT(NUMJ,3)=LOCAT(3)
      JOINT(NUMJ,4)=ORIENT(1)
      JOINT(NUMJ,5)=ORIENT(2)
```

```
      JOINT(NUMJ,6)=ORIENT(3)

*********************************************************************
*    DRAW THE JOINT
*********************************************************************

      CALL DRAWJT(IWSID,AXISR,JNAME,NUMJ,TYPE,JOINT,SCALE)


 1000 RETURN
      END
```

SUBROUTINE ADDL

```
      SUBROUTINE ADDL(IWSID,CSIZE,AXISR,JNAM,JNUM,JOINT,PNAM,PNUM,POINT,
     &LNAM,LNUM,LINK,SCALE)
*********************************************************************
* ----- SUBROUTINE ADDL -----                                       *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED TO ADD A LINK TO A MECHIN FILE          *
*                                                                   *
*********************************************************************

      COMMON/LDEF/LNDEF(30)

      INTEGER IWSID,NUML,PNUM(100),RFLG,JNUM(20,3),LNUM(30,14),CLASS(1),
     &      CHOICE

      REAL AXISR,LOCAT(3),AREA(6),POINT(100,3),CSIZE(3),JOINT(20,21),
     &  LINK(30,6),ENDP1(3),ENDP2(3),LNK(6),XVEC(3),YVEC(3),X1(6),X2(6),
     &  JOINTT(20,21),SC(3),SCM(4,4),TR(3),TRM(4,4),TRM1(4,4),P(3),
     &  P1(3),P2(3),S(4,4),T(4,4)

      CHARACTER*7 LNDEF,LNAME,PNAM(100),JNAM(20),LNAM(30),CLEAR,CLEAR1
      CHARACTER*20 CLEAR2


      DATA CLEAR     /'       '/
      DATA CLEAR1    /'       '/
      DATA CLEAR2    /'                    '/
      DATA SCM/1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
      DATA TRM/1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
      DATA TRM1/1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
      DATA S  /1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
      DATA T  /1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./

      DATA CLASS  /5/

      RFLG=0

      LOCAT(1)=0.0
      LOCAT(2)=0.0
      LOCAT(3)=0.0

      AREA(1)=0.08*CSIZE(1)
      AREA(2)=0.90*CSIZE(1)
      AREA(3)=0.70*CSIZE(1)
      AREA(4)=0.90*CSIZE(1)
      AREA(5)=0.02*CSIZE(1)
      AREA(6)=0.90*CSIZE(1)

      LNAME=CLEAR

      SC(1)=SCALE
      SC(2)=SCALE
      SC(3)=SCALE


      CALL GPSC3(SC,SCM)

*********************************************************************
*    SET LINK NUMBER
*********************************************************************
C                    SEARCH LNAM ARRAY FOR NEXT AVAILABLE SLOT (BBBBBBBB)
      DO 10 NUML=1,30,1
         IF(LNAM(NUML).EQ.'BBBBBBB'.OR.LNAM(NUML).EQ.'CCCCCCC')THEN
            GO TO 20
         ELSE
            CONTINUE
         ENDIF
   10 CONTINUE
   20 IF(NUML.EQ.30)THEN
C                    NO MORE LINKS AVAILABLE
         CALL CMMAND(IWSID,46)
C                    GET STROKE
         CALL GPINST(IWSID,1,7,CLEAR,1,AREA,7,1,0,CLEAR)
```

```
         CALL GPRQST(IWSID,1,7,ISTAT,NUM,CLEAR1)
         RETURN
      ENDIF

*****************************************************************
*    GET THE LINK NAME OR DEFAULT
*****************************************************************
C                  REQUEST STRING
      CALL CMMAND(IWSID,47)
C                  INITIALIZE THE STRING DEVICE
      CALL GPSTMO(IWSID,1,1,2)
   25 CALL GPINST(IWSID,1,7,LNDEF(NUML),1,AREA,7,1,0,LNDEF(NUML))
C                  INQUIRE POINT NAME
      CALL GPRQST(IWSID,1,7,ISTAT,NUM,LNAME)
      CALL GPSTMO(IWSID,1,1,2)

*****************************************************************
*    IF AN EXISTING LINK NAME,  COUPLER JOINTS ON THIS LINK
*****************************************************************
      DO 26 I=1,30
         IF(LNAME.EQ.LNAM(I))THEN
              NUML=I
              CALL COUPLE(IWSID,LNAM,LNUM,LINK,JNAM,JNUM,JOINT,NUML,
     & CSIZE)
              CALL DRAWLK(IWSID,LNAM,LNUM,LINK,JOINT,JNUM,AXISR,NUML,
     &        SCALE)
              GO TO 100
         ENDIF
   26 CONTINUE

*****************************************************************
*    PICK FIRST END OF THE LINK
*****************************************************************

      CALL CMMAND(IWSID,49)
      CALL PICKJ(IWSID,JOINT,AXISR,CSIZE,JNUM1,ENDP1,IAXIS1,RFLG)
      IF(JNUM(JNUM1,2).EQ.8)THEN
C                  GROUND LINK
         TR(1)=JOINT(JNUM1,1)
         TR(2)=JOINT(JNUM1,2)
         TR(3)=JOINT(JNUM1,3)

         CALL GPTRL3(TR,TRM1)

         JOINTT(JNUM1,7)=1.0
         JOINTT(JNUM1,8)=0.0
         JOINTT(JNUM1,9)=0.0
         XVEC(1)=1.0
         XVEC(2)=0.0
         XVEC(3)=0.0

         X1(1)=0.
         X1(2)=0.
         X1(3)=0.
         X1(4)=(XVEC(1)*AXISR/10.)*2.4
         X1(5)=(XVEC(2)*AXISR/10.)*2.4
         X1(6)=(XVEC(3)*AXISR/10.)*2.4

         CALL GPOPST(19)
         CALL GPADCN(1,CLASS)
         CALL GPPKID(NUML)
         CALL GPPLCI(6)
         CALL GPMLX3(TRM1,3)
         CALL GPMLX3(SCM,1)
         CALL GPLT(1)
         CALL GPPL3(2,3,X1)
         CALL GPCLST

         CALL GPUPWS(IWSID,2)

         GO TO 50
      ENDIF

      IF(RFLG.EQ.1)THEN
         GO TO 100
      ENDIF

*****************************************************************
*    DEFINE THE X AXIS OF THE JOINT AT THE FIRST END OF THE LINK
*****************************************************************

      CALL CMMAND(IWSID,51)
      CALL MENU(IWSID,CSIZE,12,CHOICE)
      IF(CHOICE.EQ.1)THEN
         CALL CMMAND(IWSID,53)
         CALL TEJOX(IWSID,JNUM1,IAXIS1,JOINT,AXISR,CSIZE,XVEC,YVEC,RFLG)
         IF(IAXIS1.EQ.1)THEN
              JOINTT(JNUM1,7)=XVEC(1)
              JOINTT(JNUM1,8)=XVEC(2)
              JOINTT(JNUM1,9)=XVEC(3)
         ENDIF
         IF(IAXIS1.EQ.2)THEN
              JOINTT(JNUM1,10)=XVEC(1)
              JOINTT(JNUM1,11)=XVEC(2)
```

```
            JOINTT(JNUM1,12)=XVEC(3)
         ENDIF
         IF(RFLG.EQ.1)THEN
            GO TO 100
         ENDIF
      ENDIF
      IF(CHOICE.EQ.2)THEN
         CALL CMMAND(IWSID,52)
         CALL TEPOX(IWSID,JNUM1,IAXIS1,POINT,AXISR,CSIZE,XVEC,YVEC,
     &JOINT,RFLG)
         IF(IAXIS1.EQ.1)THEN
            JOINTT(JNUM1,7)=XVEC(1)
            JOINTT(JNUM1,8)=XVEC(2)
            JOINTT(JNUM1,9)=XVEC(3)
         ENDIF
         IF(IAXIS1.EQ.2)THEN
            JOINTT(JNUM1,10)=XVEC(1)
            JOINTT(JNUM1,11)=XVEC(2)
            JOINTT(JNUM1,12)=XVEC(3)
         ENDIF
         IF(RFLG.EQ.1)THEN
            GO TO 100
         ENDIF
      ENDIF

*********************************************************************
*     DRAW THE FIRST X AXIS
*********************************************************************

      IF(IAXIS1.EQ.1)THEN

         TR(1)=JOINT(JNUM1,1)
         TR(2)=JOINT(JNUM1,2)
         TR(3)=JOINT(JNUM1,3)

         CALL GPTRL3(TR,TRM1)

         X1(1)=0.
         X1(2)=0.
         X1(3)=0.
         X1(4)=(XVEC(1)*AXISR/10.)*2.4
         X1(5)=(XVEC(2)*AXISR/10.)*2.4
         X1(6)=(XVEC(3)*AXISR/10.)*2.4


         CALL GPOPST(19)
         CALL GPADCN(1,CLASS)
         CALL GPPKID(NUML)
         CALL GPPLCI(6)
         CALL GPMLX3(TRM1,3)
         CALL GPMLX3(SCM,1)
         CALL GPLT(1)
         CALL GPPL3(2,3,X1)
         CALL GPCLST
      ENDIF

      IF(IAXIS1.EQ.2)THEN

         TR(1)=JOINT(JNUM1,15)
         TR(2)=JOINT(JNUM1,16)
         TR(3)=JOINT(JNUM1,17)

         CALL GPTRL3(TR,TRM1)
         X1(1)=(.01*AXISR)+(JOINT(JNUM1,14)*JOINT(JNUM1,
     &        4))
         X1(2)=(.01*AXISR)+(JOINT(JNUM1,14)*JOINT(JNUM1,
     &        5))
         X1(3)=(JOINT(JNUM1,14)*JOINT(JNUM1,6))
         X1(4)=X1(1)+(XVEC(1)*AXISR/10.)
         X1(5)=X1(2)+(XVEC(2)*AXISR/10.)
         X1(6)=X1(3)+(XVEC(3)*AXISR/10.)

         CALL GPOPST(19)
         CALL GPADCN(1,CLASS)
         CALL GPPKID(NUML)
         CALL GPMLX3(TRM1,3)
         CALL GPMLX3(SCM,1)
         CALL GPPLCI(2)
         CALL GPLT(1)
         CALL GPPL3(2,3,X1)
         CALL GPCLST
      ENDIF

      CALL GPUPWS(IWSID,2)


*********************************************************************
*     PICK SECOND END OF THE LINK
*********************************************************************

   50 CALL CMMAND(IWSID,50)
      CALL PICKJ(IWSID,JOINT,AXISR,CSIZE,JNUM2,ENDP2,IAXIS2,RFLG)

      IF(RFLG.EQ.1)THEN
         LNAM(NUML)='CCCCCCC'
         CALL RDRWLK(IWSID,JNAM,JNUM,JOINT,AXISR,LNAM,LNUM,LINK,SCALE)
```

```
         GO TO 100
      ENDIF

      IF(JNUM(JNUM2,2).EQ.8)THEN
C                 GROUND LINK
         TR(1)=JOINT(JNUM2,1)
         TR(2)=JOINT(JNUM2,2)
         TR(3)=JOINT(JNUM2,3)

         CALL GPTRL3(TR,TRM)

         JOINTT(JNUM2,7)=1.0
         JOINTT(JNUM2,8)=0.0
         JOINTT(JNUM2,9)=0.0
         XVEC(1)=1.0
         XVEC(2)=0.0
         XVEC(3)=0.0

         X2(1)=0.
         X2(2)=0.
         X2(3)=0.
         X2(4)=(XVEC(1)*AXISR/10.)*2.4
         X2(5)=(XVEC(2)*AXISR/10.)*2.4
         X2(6)=(XVEC(3)*AXISR/10.)*2.4

         CALL GPOPST(19)
         CALL GPADCN(1,CLASS)
         CALL GPPKID(NUML)
         CALL GPPLCI(6)
         CALL GPMLX3(TRM,3)
         CALL GPMLX3(SCM,1)
         CALL GPLT(1)
         CALL GPPL3(2,3,X2)
         CALL GPCLST

         CALL GPUPWS(IWSID,2)

         GO TO 60
      ENDIF

      IF(RFLG.EQ.1)THEN
         LNAM(NUML)='CCCCCCC'
         CALL RDRWLK(IWSID,JNAM,JNUM,JOINT,AXISR,LNAM,LNUM,LINK,SCALE)
         GO TO 100
      ENDIF

*********************************************************************
*    DEFINE THE X AXIS OF THE JOINT AT THE SECOND END OF THE LINK
*********************************************************************

      CALL CMMAND(IWSID,51)
      CALL MENU(IWSID,CSIZE,12,CHOICE)
      IF(CHOICE.EQ.1)THEN
         CALL CMMAND(IWSID,53)
         CALL TEJOX(IWSID,JNUM2,IAXIS2,JOINT,AXISR,CSIZE,XVEC,YVEC,RFLG)
         IF(IAXIS2.EQ.1)THEN
              JOINTT(JNUM2,7)=XVEC(1)
              JOINTT(JNUM2,8)=XVEC(2)
              JOINTT(JNUM2,9)=XVEC(3)
         ENDIF
         IF(IAXIS2.EQ.2)THEN
              JOINTT(JNUM2,10)=XVEC(1)
              JOINTT(JNUM2,11)=XVEC(2)
              JOINTT(JNUM2,12)=XVEC(3)
         ENDIF
         IF(RFLG.EQ.1)THEN
            LNAM(NUML)='CCCCCCC'
           CALL RDRWLK(IWSID,JNAM,JNUM,JOINT,AXISR,LNAM,LNUM,LINK,SCALE)
            GO TO 100
         ENDIF
      ENDIF
      IF(CHOICE.EQ.2)THEN
         CALL CMMAND(IWSID,52)
         CALL TEPOX(IWSID,JNUM2,IAXIS2,POINT,AXISR,CSIZE,XVEC,YVEC,
     &JOINT,RFLG)
         IF(IAXIS2.EQ.1)THEN
              JOINTT(JNUM2,7)=XVEC(1)
              JOINTT(JNUM2,8)=XVEC(2)
              JOINTT(JNUM2,9)=XVEC(3)
         ENDIF
         IF(IAXIS2.EQ.2)THEN
              JOINTT(JNUM2,10)=XVEC(1)
              JOINTT(JNUM2,11)=XVEC(2)
              JOINTT(JNUM2,12)=XVEC(3)
         ENDIF
         IF(RFLG.EQ.1)THEN
            LNAM(NUML)='CCCCCCC'
           CALL RDRWLK(IWSID,JNAM,JNUM,JOINT,AXISR,LNAM,LNUM,LINK,SCALE)
            GO TO 100
         ENDIF
      ENDIF

*********************************************************************
*    DRAW THE SECOND X AXIS
*********************************************************************
```

```
      IF(IAXIS2.EQ.1)THEN

         TR(1)=JOINT(JNUM2,1)
         TR(2)=JOINT(JNUM2,2)
         TR(3)=JOINT(JNUM2,3)

         CALL GPTRL3(TR,TRM)

         X2(1)=0.
         X2(2)=0.
         X2(3)=0.
         X2(4)=(XVEC(1)*AXISR/10.)*2.4
         X2(5)=(XVEC(2)*AXISR/10.)*2.4
         X2(6)=(XVEC(3)*AXISR/10.)*2.4

         CALL GPOPST(19)
         CALL GPADCN(1,CLASS)
         CALL GPPKID(NUML)
         CALL GPMLX3(TRM,3)
         CALL GPMLX3(SCM,1)
         CALL GPPLCI(6)
         CALL GPLT(1)
         CALL GPPL3(2,3,X2)
         CALL GPCLST
      ENDIF

      IF(IAXIS2.EQ.2)THEN

         TR(1)=JOINT(JNUM2,15)
         TR(2)=JOINT(JNUM2,16)
         TR(3)=JOINT(JNUM2,17)

         CALL GPTRL3(TR,TRM)

         X2(1)=(.01*AXISR)+(JOINT(JNUM2,14)*JOINT(JNUM2,
     &          4))
         X2(2)=(.01*AXISR)+(JOINT(JNUM2,14)*JOINT(JNUM2,
     &          5))
         X2(3)=(JOINT(JNUM2,14)*JOINT(JNUM2,
     &          6))
         X2(4)=X2(1)+(XVEC(1)*AXISR/10.)
         X2(5)=X2(2)+(XVEC(2)*AXISR/10.)
         X2(6)=X2(3)+(XVEC(3)*AXISR/10.)

         CALL GPOPST(19)
         CALL GPADCN(1,CLASS)
         CALL GPMLX3(TRM,3)
         CALL GPMLX3(SCM,1)
         CALL GPPKID(NUML)
         CALL GPPLCI(2)
         CALL GPLT(1)
         CALL GPPL3(2,3,X2)
         CALL GPCLST
      ENDIF

      CALL GPUPWS(IWSID,2)

*************************************************************************
*     DRAW THE LINK
*************************************************************************

   60 P(1)=X1(4)*SCALE
      P(2)=X1(5)*SCALE
      P(3)=X1(6)*SCALE

      CALL GPXF3(P,TRM1,P1)

      P(1)=X2(4)*SCALE
      P(2)=X2(5)*SCALE
      P(3)=X2(6)*SCALE

      CALL GPXF3(P,TRM,P2)

      LNK(1)=P1(1)
      LNK(2)=P1(2)
      LNK(3)=P1(3)
      LNK(4)=P2(1)
      LNK(5)=P2(2)
      LNK(6)=P2(3)


      IF(JNUM(JNUM1,2).EQ.8)THEN

         P(1)=0.
         P(2)=0.
         P(3)=0.

         CALL GPXF3(P,TRM1,P1)

         LNK(1)=P1(1)
         LNK(2)=P1(2)
         LNK(3)=P1(3)
      ENDIF
```

```
      IF(JNUM(JNUM2,2).EQ.8)THEN

         P(1)=0.
         P(2)=0.
         P(3)=0.

         CALL GPXF3(P,TRM,P2)

         LNK(4)=P2(1)
         LNK(5)=P2(2)
         LNK(6)=P2(3)
      ENDIF

      CALL GPOPST(19)
      CALL GPADCN(1,CLASS)
      CALL GPPKID(NUML)
      CALL GPMLX3(T,3)
      CALL GPMLX3(S,1)
      CALL GPPLCI(7)
      CALL GPLT(2)
      CALL GPPL3(2,3,LNK)
      CALL GPCLST

      CALL GPUPWS(IWSID,2)

*********************************************************************
*    ADD POINT DATA TO POINT ARRAYS
*********************************************************************

      LNAM(NUML)=LNAME
      LNUM(NUML,1)=NUML
      LNUM(NUML,2)=JNUM1
      LNUM(NUML,3)=IAXIS1
      LNUM(NUML,4)=JNUM2
      LNUM(NUML,5)=IAXIS2
      LNUM(NUML,14)=0
      LINK(NUML,1)=LNK(1)
      LINK(NUML,2)=LNK(2)
      LINK(NUML,3)=LNK(3)
      LINK(NUML,4)=LNK(4)
      LINK(NUML,5)=LNK(5)
      LINK(NUML,6)=LNK(6)

      IF(IAXIS1.EQ.1)THEN
         JOINT(JNUM1,7)=JOINTT(JNUM1,7)
         JOINT(JNUM1,8)=JOINTT(JNUM1,8)
         JOINT(JNUM1,9)=JOINTT(JNUM1,9)
      ENDIF
      IF(IAXIS1.EQ.2)THEN
         JOINT(JNUM1,10)=JOINTT(JNUM1,10)
         JOINT(JNUM1,11)=JOINTT(JNUM1,11)
         JOINT(JNUM1,12)=JOINTT(JNUM1,12)
      ENDIF

      IF(IAXIS2.EQ.1)THEN
         JOINT(JNUM2,7)=JOINTT(JNUM2,7)
         JOINT(JNUM2,8)=JOINTT(JNUM2,8)
         JOINT(JNUM2,9)=JOINTT(JNUM2,9)
      ENDIF
      IF(IAXIS2.EQ.2)THEN
         JOINT(JNUM2,10)=JOINTT(JNUM2,10)
         JOINT(JNUM2,11)=JOINTT(JNUM2,11)
         JOINT(JNUM2,12)=JOINTT(JNUM2,12)
      ENDIF

  100 RETURN
      END
```

SUBROUTINE ADDP

```
      SUBROUTINE ADDP(IWSID,CSIZE,AXISR,PNAM,PNUM,POINT)
*********************************************************************
* ----- SUBROUTINE ADDP -----                                       *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED TO ADD A POINT TO A MECHIN FILE         *
*                                                                   *
*********************************************************************

      COMMON/PDEF/PNDEF(100)

      INTEGER IWSID,NUMP,PNUM(100),RFLG

      REAL AXISR,LOCAT(3),AREA(6),POINT(100,3),CSIZE(3)

      CHARACTER*7 PNDEF,PNAME,PNAM(100),CLEAR,CLEAR1
      CHARACTER*20 CLEAR2
```

```
      DATA CLEAR     /'        '/
      DATA CLEAR1    /'        '/
      DATA CLEAR2    /'                  '/

      RFLG=0

      LOCAT(1)=0.0
      LOCAT(2)=0.0
      LOCAT(3)=0.0

      AREA(1)=0.08*CSIZE(1)
      AREA(2)=0.90*CSIZE(1)
      AREA(3)=0.70*CSIZE(1)
      AREA(4)=0.90*CSIZE(1)
      AREA(5)=0.02*CSIZE(1)
      AREA(6)=0.90*CSIZE(1)

      PNAME=CLEAR

*****************************************************************
*    SET POINT NUMBER
*****************************************************************
C                  SEARCH PNAM ARRAY FOR NEXT AVAILABLE SLOT (BBBBBBBB)
      DO 10 NUMP=1,100,1
         IF(PNAM(NUMP).EQ.'BBBBBBB'.OR.PNAM(NUMP).EQ.'CCCCCCC')THEN
            GO TO 20
         ELSE
            CONTINUE
         ENDIF
   10 CONTINUE
   20 IF(NUMP.EQ.100)THEN
C                  NO MORE POINTS AVAILABLE
         CALL CMMAND(IWSID,16)
C                  GET STROKE
         CALL GPINST(IWSID,1,7,CLEAR,1,AREA,7,1,0,CLEAR)
         CALL GPRQST(IWSID,1,7,ISTAT,NUM,CLEAR1)
         RETURN
      ENDIF

*****************************************************************
*    GET THE POINT NAME OR DEFAULT
*****************************************************************
C                  REQUEST STRING
      CALL CMMAND(IWSID,13)
C                  INITIALIZE THE STRING DEVICE
      CALL GPSTMO(IWSID,1,1,2)
   25 CALL GPINST(IWSID,1,7,PNDEF(NUMP),1,AREA,7,1,0,PNDEF(NUMP))
C                  INQUIRE POINT NAME
      CALL GPRQST(IWSID,1,7,ISTAT,NUM,PNAME)
      CALL GPSTMO(IWSID,1,1,2)

*****************************************************************
*    CHECK TO SEE IF NAME CAN BE USED
*****************************************************************
      DO 30 J=1,100
         IF(PNAME.EQ.PNAM(J))THEN
C                  NAME ALREADY EXISTS
            CALL CMMAND(IWSID,14)
            PNAME=CLEAR
            GO TO 25
         ENDIF
   30 CONTINUE


*****************************************************************
*    POSITION THE POINT
*****************************************************************

      CALL POSIT(IWSID,AXISR,CSIZE,LOCAT,RFLG)
      IF(RFLG.EQ.1)THEN
         GO TO 100
      ENDIF

*****************************************************************
*    DRAW THE POINT
*****************************************************************

      CALL DRAWPT(IWSID,LOCAT,AXISR,PNAME,NUMP)

*****************************************************************
*    ADD POINT DATA TO POINT ARRAYS
*****************************************************************

      PNAM(NUMP)=PNAME
      PNUM(NUMP)=NUMP
      POINT(NUMP,1)=LOCAT(1)
      POINT(NUMP,2)=LOCAT(2)
      POINT(NUMP,3)=LOCAT(3)


  100 RETURN
      END
```

SUBROUTINE ANIN

```
      SUBROUTINE ANIN(IWSID,CSIZE)
*********************************************************************
* ----- SUBROUTINE ANIN -----                                       *
*                                                                   *
*                                                                   *
*   THIS IS THE ANALYSIS PORTION OF MECHIN.  THIS ROUTINE WILL      *
*                                                                   *
*   ALLOW THE INPUT OF SPATIAL MECHANISM PARAMETERS AND WILL        *
*                                                                   *
*   WRITE AN INPUT FILE FOR AN ANALYSIS PACKAGE BASED ON THESE      *
*                                                                   *
*   PARAMETERS.                                                     *
*                                                                   *
*********************************************************************

      COMMON/PNT/JOINT(20,21),PNAM(100),JNAM(20),LNAM(30),INAM(20)

      INTEGER IWSID,CHOICE,JNUM(20,3),LNUM(30,14),PNUM(100),INUM(20,2)
      REAL CSIZE(3),AXISR,POINT(100,3),JOINT,LINK(30,6),SCALE,IC(20,4)
      CHARACTER*7 PNAM,JNAM,LNAM,FILE,INAM

*********************************************************************
*  CALL BACKGROUND SCREEN
*********************************************************************
C                             CALL FILE WITH ANALYSIS BACKGROUND
      CALL SCRN3(IWSID)

*********************************************************************
*  FIND OUT IF A NEW OR OLD MODEL IS TO BE PROCESSED AND SET UP FOR DATA
*********************************************************************
C                             FIND IF NEW OR OLD MODEL
      CALL CMMAND(IWSID,1)
      CALL MENU(IWSID,CSIZE,1,CHOICE)
      IF(CHOICE.EQ.1)THEN
C                             NEW MODEL  --- REQUEST SIZE OF MODEL
         CALL NSTART(IWSID,FILE,CSIZE,AXISR,POINT,JOINT,LINK,IC,
     &                  PNUM,JNUM,LNUM,INUM)
         SCALE=.8
         CALL SCALNM(IWSID,SCALE)
      ELSE
C                             RESTART MODEL  --- READ RESTART FILE
         CALL RSTART(IWSID,FILE,CSIZE,AXISR,PNAM,PNUM,POINT,JNAM,JNUM,
     &                  JOINT,LNAM,LNUM,LINK,INAM,INUM,IC,SCALE)
      ENDIF

*********************************************************************
*  INPUT MODEL DATA
*********************************************************************

      CALL DATA(IWSID,AXISR,CSIZE,PNAM,PNUM,POINT,JNAM,JNUM,JOINT,
     &          LNAM,LNUM,LINK,INAM,INUM,IC,SCALE)

*********************************************************************
*  WRITE THE RESTART FILE
*********************************************************************

      CALL WRREST(PNAM,PNUM,POINT,JNAM,JNUM,JOINT,LNAM,LNUM,LINK,AXISR,
     &FILE,IWSID,CSIZE,INAM,INUM,IC,SCALE)

*********************************************************************
*  WRITE THE ANALYSIS FILE
*********************************************************************

      CALL WRANAL(PNAM,PNUM,POINT,JNAM,JNUM,JOINT,LNAM,LNUM,LINK,
     &CSIZE,INAM,INUM,IC,IWSID)

      RETURN
      END
```

SUBROUTINE ANOTH

```
      SUBROUTINE ANOTH(IWSID,CSIZE,CHOICE)
*********************************************************************
*  ---- SUBROUTINE ANOTH -----                                       *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE ASKS IF ANOTHER MECHANISM IS TO BE INPUT ELSE   *
*                                                                   *
*   QUIT                                                            *
*                                                                   *
*********************************************************************
```

```
      REAL CSIZE(3)
      INTEGER IWSID,CHOICE,CHO

      CALL CMMAND(IWSID,85)
      CALL MENU(IWSID,CSIZE,17,CHO)

      IF(CHO.EQ.1)THEN
         CALL GPDAST
         CALL GPEAV(IWSID)
         CALL GPDARW(IWSID)
         CALL GPVCH(IWSID,4,2,2,2,1,1,1,1,1)
         CALL GPVCH(IWSID,5,2,2,2,1,1,1,1,1)
         CALL GPVCH(IWSID,6,2,2,2,1,1,1,1,1)
         CALL GPUPWS(IWSID,2)
         CALL SASSO(IWSID,CSIZE)
         CALL SCRN2(IWSID,CSIZE,CHOICE)
      ENDIF

      IF(CHO.EQ.2)THEN
         CHOICE=0
      ENDIF

      RETURN
      END
```

SUBROUTINE AXES

```
      SUBROUTINE AXES(IWSID,AXISR)
*************************************************************************
* ----- SUBROUTINE AXES -----                                           *
*                                                                       *
*                                                                       *
*   THIS SUBROUTINE IS USED TO DISPLAY THE COORDINATE AXES              *
*                                                                       *
*************************************************************************
      INTEGER IWSID,PARELL,PRFORM

      REAL AXISR,XAXIS(24),YAXIS(24),ZAXIS(24),WINDOW(4),VP1(6),VP2(6),
     &BOX1(8),BOX4(8),NEAR,FAR,POINT(3),DIST,UP(3),NORMAL(3),
     &MATRIX(4,4),POINTP(3),PX(3),PY(3),PZ(3),WINDO1(4),NEAR1,FAR1

      CHARACTER*1 X,Y,Z

C                              FLAG FOR UPDATE
      DATA PRFORM/2/
C                              GREY MAIN SCREEN BOX
      DATA BOX1/-0.48,-0.98,0.98,-0.98,0.98,0.48,-0.48,0.48/
C                              GREY POSITION BOX
      DATA BOX4/-0.48,0.52,-0.02,0.52,-0.02,0.98,-0.48,0.98/
      DATA DIST/0./
      DATA PARELL/1/
      DATA POINT/0.,0.,0./
      DATA POINTP/2.0,1.5,5.0/
      DATA NORMAL/1.,0.,0./
      DATA UP/0.,0.,1./
      DATA X/'X'/
      DATA Y/'Y'/
      DATA Z/'Z'/

      XAXIS(1)=-AXISR
      XAXIS(2)=0.0
      XAXIS(3)=0.0
      XAXIS(4)=-AXISR+(.08*AXISR)
      XAXIS(5)=-.02*AXISR
      XAXIS(6)=0.0
      XAXIS(7)=-AXISR+(.08*AXISR)
      XAXIS(8)=.02*AXISR
      XAXIS(9)=0.0
      XAXIS(10)=-AXISR
      XAXIS(11)=0.0
      XAXIS(12)=0.0
      XAXIS(13)=AXISR
      XAXIS(14)=0.0
      XAXIS(15)=0.0
      XAXIS(16)=AXISR-(.08*AXISR)
      XAXIS(17)=-.02*AXISR
      XAXIS(18)=0.0
      XAXIS(19)=AXISR-(.08*AXISR)
      XAXIS(20)=.02*AXISR
      XAXIS(21)=0.0
      XAXIS(22)=AXISR
      XAXIS(23)=0.0
      XAXIS(24)=0.0
```

```
      YAXIS(1)=0.0
      YAXIS(2)=-AXISR
      YAXIS(3)=0.0
      YAXIS(4)=-.02*AXISR
      YAXIS(5)=-AXISR+(.08*AXISR)
      YAXIS(6)=0.0
      YAXIS(7)=.02*AXISR
      YAXIS(8)=-AXISR+(.08*AXISR)
      YAXIS(9)=0.0
      YAXIS(10)=0.0
      YAXIS(11)=-AXISR
      YAXIS(12)=0.0
      YAXIS(13)=0.0
      YAXIS(14)=AXISR
      YAXIS(15)=0.0
      YAXIS(16)=-.02*AXISR
      YAXIS(17)=AXISR-(.08*AXISR)
      YAXIS(18)=0.0
      YAXIS(19)=.02*AXISR
      YAXIS(20)=AXISR-(.08*AXISR)
      YAXIS(21)=0.0
      YAXIS(22)=0.0
      YAXIS(23)=AXISR
      YAXIS(24)=0.0

      ZAXIS(1)=0.0
      ZAXIS(2)=0.0
      ZAXIS(3)=-AXISR
      ZAXIS(4)=-.02*AXISR
      ZAXIS(5)=0.0
      ZAXIS(6)=-AXISR+(.08*AXISR)
      ZAXIS(7)=.02*AXISR
      ZAXIS(8)=0.0
      ZAXIS(9)=-AXISR+(.08*AXISR)
      ZAXIS(10)=0.0
      ZAXIS(11)=0.0
      ZAXIS(12)=-AXISR
      ZAXIS(13)=0.0
      ZAXIS(14)=0.0
      ZAXIS(15)=AXISR
      ZAXIS(16)=-.02*AXISR
      ZAXIS(17)=0.0
      ZAXIS(18)=AXISR-(.08*AXISR)
      ZAXIS(19)=.02*AXISR
      ZAXIS(20)=0.0
      ZAXIS(21)=AXISR-(.08*AXISR)
      ZAXIS(22)=0.0
      ZAXIS(23)=0.0
      ZAXIS(24)=AXISR

      PX(1)=AXISR+(.04*AXISR)
      PX(2)=(-0.01*AXISR)
      PX(3)=0.0

      PY(1)=-.01*AXISR
      PY(2)=AXISR+(.04*AXISR)
      PY(3)=0.0

      PZ(1)=-.02*AXISR
      PZ(2)=0.0
      PZ(3)=AXISR+(.07*AXISR)

      VP1(1)=(BOX1(1)+1.)/2.
      VP1(2)=(BOX1(3)+1.)/2.
      VP1(3)=(BOX1(4)+1.)/2.
      VP1(4)=(BOX1(6)+1.)/2.
      VP1(5)=(BOX1(1)+1.)/2.
      VP1(6)=(BOX1(3)+1.)/2.

      VP2(1)=(BOX4(1)+1.)/2.
      VP2(2)=(BOX4(3)+1.)/2.
      VP2(3)=(BOX4(4)+1.)/2.
      VP2(4)=(BOX4(6)+1.)/2.
      VP2(5)=(BOX4(1)+1.)/2.
      VP2(6)=(BOX4(3)+1.)/2.

      NEAR=AXISR+(.64*AXISR)
      FAR=-AXISR-(.64*AXISR)


      NEAR1=1.64
      FAR1=-1.64

      WINDOW(1)=-AXISR-(.64*AXISR)
      WINDOW(2)=AXISR+(.64*AXISR)
      WINDOW(3)=-AXISR-(.64*AXISR)
      WINDOW(4)=AXISR+(.64*AXISR)

      WINDO1(1)=-1.64
      WINDO1(2)=1.64
      WINDO1(3)=-1.64
      WINDO1(4)=1.64

*************************************************************************
*  SET THE VIEW CHARACTERISTICS FOR AXES VIEWS (4 ,5,AND 6)
```

```
*********************************************************************
C                              SET THE CHARACTERISTICS
      CALL GPVMP3(IWSID,4,WINDOW,VP1,PARELL,POINTP,DIST,NEAR,FAR)
      CALL GPVMP3(IWSID,5,WINDOW,VP2,PARELL,POINTP,DIST,NEAR,FAR)
      CALL GPVMP3(IWSID,6,WINDO1,VP2,PARELL,POINTP,DIST,NEAR1,FAR1)

C                              ACTIVATE THE VIEW
      CALL GPVCH(IWSID,4,2,2,2,1,0,2,5,2)
      CALL GPVCH(IWSID,5,2,2,2,1,0,2,5,2)
      CALL GPVCH(IWSID,6,2,2,2,1,0,2,5,2)


*********************************************************************
*  OPEN STRUCTURE FOR AXIS
*********************************************************************
C                              OPEN A STRUCTURE
      CALL GPOPST(7)
C                              SET AXIS COLOR WHITE
      CALL GPPLCI(1)
C                              DRAW X AXIS
      CALL GPPL3(8,3,XAXIS)
C                              DRAW Y AXIS
      CALL GPPL3(8,3,YAXIS)
C                              DRAW Z AXIS
      CALL GPPL3(8,3,ZAXIS)
C                              SET CHARACTER HEIGHT
      CALL GPAHSC(0.25)
C                              DRAW X AXIS IDENTIFIER
      CALL GPAN3(PX,1,X)
C                              DRAW Y AXIS IDENTIFIER
      CALL GPAN3(PY,1,Y)
C                              DRAW Z AXIS IDENTIFIER
      CALL GPAN3(PZ,1,Z)
C                              CLOSE STRUCTURE
      CALL GPCLST

      CALL GPARV(IWSID,4,7,0.)
      CALL GPARV(IWSID,5,7,0.)
      CALL GPVP(IWSID,4,1,1)
      CALL GPVP(IWSID,5,1,1)
      CALL GPVP(IWSID,6,1,1)
      CALL GPVP(IWSID,6,4,1)
      CALL GPVP(IWSID,6,5,1)

*********************************************************************
*  WRITE AXIS RANGE TO THE SCREEN
*********************************************************************

      CALL AXISNM(IWSID,AXISR)


      CALL GPUPWS(IWSID,PRFORM)

      RETURN
      END
```

SUBROUTINE AXISNM

```
      SUBROUTINE AXISNM(IWSID,AXISR)
*********************************************************************
* ----- SUBROUTINE AXISNM -----                                     *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED TO DISPLAY AXIS RANGE ON THE SCREEN     *
*                                                                   *
*********************************************************************

      INTEGER IWSID,LENGT,LENGF,PRFORM


      REAL POSF(2),BOX1(8),SHD1(8),POS1(2),POS2(2),AXISR
      CHARACTER*10 FILE
      CHARACTER*22 FILE1,FILE2,CLEAR

C                              FILE NAME LENGTH AND POSITION
      DATA LENGF/10/
      DATA POSF/.70,.79/
C                              FILE IDENT. LENGTH AND POSITION
      DATA LENGT/22/
      DATA POS1/.60,.84/
      DATA POS2/.60,.79/
C                              FLAG FOR UPDATE
      DATA PRFORM/2/
C                              FILE TITLE IDENTIFIER
      DATA FILE1/'AXIS RANGE :      '/
      DATA FILE2/'  +/-             '/
      DATA CLEAR/'                  '/
C                              GREY FILE BOX
      DATA BOX1/0.57,0.77,0.97,0.77,0.97,0.88,0.57,0.88/
```

```
C                              CALCULATION FOR THE TWO SHADOW BOXES
      DO 100 I=1,7,2
         SHD1(I)=BOX1(I)+0.025
  100 CONTINUE

      DO 200 I=2,8,2
         SHD1(I)=BOX1(I)-0.025
  200 CONTINUE

      REWIND 10
      WRITE(10,225)CLEAR
  225 FORMAT(A22)
      REWIND 10
      WRITE(10,250)AXISR
  250 FORMAT(F10.3)
      REWIND 10
      READ(10,275)FILE
  275 FORMAT(A10)


*********************************************************************
*  OPEN STRUCTURE FOR COMMAND BOX (ALSO PUT IN STRUCTURE 3)
*********************************************************************
C                              OPEN A STRUCTURE
      CALL GPOPST(3)
C                              SET SOLID INTERIOR STYLE FOR GREY BOX
      CALL GPIS(2)
C                              SET COLOR FOR INTERIOR OF SHADOW BOX
      CALL GPICI(4)
C                              DRAW GREY SHADOW BOX
      CALL GPPG2(1,4,2,SHD1)
C                              SET COLOR FOR INTERIOR OF GREY BOX
      CALL GPICI(3)
C                              DRAW GREY COMMAND BOX
      CALL GPPG2(1,4,2,BOX1)
C                              SET COLOR FOR WHITE OF FILE NAME
      CALL GPICI(1)
C                              SET TEXT COLOR TO WHITE
      CALL GPTXCI(1)
C                              SET ANNOTATION SIZE SCALE FACTOR
      CALL GPAHSC(1.00)
C                              DRAW TITLE TEXT
      CALL GPAN2(POS1,LENGT,FILE1)
      CALL GPAN2(POS2,LENGT,FILE2)
      CALL GPAN2(POSF,LENGF,FILE)
C                              CLOSE STRUCTURE
      CALL GPCLST

*********************************************************************
*  SCREEN DISPLAY
*********************************************************************
C                              UPDATE THE WORKSTATION
      CALL GPUPWS(IWSID,PRFORM)

      RETURN
      END
```

SUBROUTINE CDATA

```
      SUBROUTINE CDATA(IWSID,CSIZE,Z,RFLG)
*********************************************************************
* ----- SUBROUTINE CDATA -----                                      *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE WILL PROMPT FOR DATA ON THE COMMAND BOX         *
*                                                                   *
*********************************************************************

      INTEGER IWSID
      REAL AREA(6),CSIZE(3),Z
      CHARACTER*7 CLEAR,INIT
      CHARACTER*12 STRING,CLEAR2

      DATA CLEAR /'       '/
      DATA INIT  /'0.     '/
      DATA STRING/'           '/
      DATA CLEAR2/'           '/

      AREA(1)=0.08*CSIZE(1)
      AREA(2)=0.90*CSIZE(1)
      AREA(3)=0.70*CSIZE(1)
      AREA(4)=0.90*CSIZE(1)
      AREA(5)=0.02*CSIZE(1)
      AREA(6)=0.90*CSIZE(1)

      CALL GPEST(6)
```

```
*********************************************************************
*  GET REQUESTED DATA
*********************************************************************

    1 CALL GPSTMO(IWSID,1,1,2)
      CALL GPINST(IWSID,1,12,INIT,1,AREA,12,1,0,CLEAR)
      STRING=CLEAR2

      CALL GPRQST(IWSID,1,12,ISTAT,NUM,STRING)
      REWIND 10
      WRITE(10,8)CLEAR1
    8 FORMAT(A20)
      REWIND 10
      WRITE(10,12)STRING
   12 FORMAT(A12)
      REWIND 10
      READ(10,*,ERR=1000)Z

 1000 CALL GPSTMO(IWSID,1,1,2)
      RETURN
      END
```

SUBROUTINE CHSCAL

```
      SUBROUTINE CHSCAL(IWSID,AXISR,CSIZE,SCALE,JNAM,JNUM,JOINT,
     &LNAM,LNUM,LINK)
*********************************************************************
* ----- SUBROUTINE CHSCAL -----                                     *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE WILL PROMPT CHANGE OF JOINT ICON SCALE          *
*                                                                   *
*********************************************************************

      INTEGER IWSID,RFLG,CHOICE,JNUM(20,3),LNUM(30,14)
      REAL LOCAT(3),AXISR,AREA(6),CSIZE(3),LX(2),LY(2),LZ(2),LX1(2),
     & LY1(2),LZ1(2),JOINT(20,21),LINK(30,6)
      CHARACTER*7 CLEAR,AXIS,INIT,INIT1,JNAM(20),LNAM(30)
      CHARACTER*20 CLEAR1,X,Y,Z
      CHARACTER*12 STRING,CLEAR2

      DATA CLEAR /'       '/
      DATA INIT  /'0.     '/
      DATA INIT1 /'1.     '/
      DATA STRING/'           '/
      DATA AXIS  /'       '/
      DATA CLEAR1/'                   '/
      DATA CLEAR2/'           '/
      DATA Y     /'SCALE =            '/

      AREA(1)=0.08*CSIZE(1)
      AREA(2)=0.90*CSIZE(1)
      AREA(3)=0.70*CSIZE(1)
      AREA(4)=0.90*CSIZE(1)
      AREA(5)=0.02*CSIZE(1)
      AREA(6)=0.90*CSIZE(1)

      LOCAT(1)=0.0
      LOCAT(2)=0.0
      LOCAT(3)=0.0

      LY(1)=0.05
      LY(2)=0.62

      LY1(1)=0.15
      LY1(2)=0.62

      CALL GPEST(6)

*********************************************************************
*  GET SCALE FACTOR
*********************************************************************

      CALL CMMAND(IWSID,69)

    1 CALL GPSTMO(IWSID,1,1,2)
      CALL GPINST(IWSID,1,12,INIT,1,AREA,12,1,0,CLEAR)
      STRING=CLEAR2

      CALL GPRQST(IWSID,1,12,ISTAT,NUM,STRING)
      REWIND 10
      WRITE(10,8)CLEAR1
    8 FORMAT(A20)
      REWIND 10
      WRITE(10,12)STRING
   12 FORMAT(A12)
      REWIND 10
      READ(10,*,ERR=1000)SCALE
```

```
      IF(SCALE.LE.0.)THEN
C                              VALUE IS GREATER THAN 1
         CALL CMMAND(IWSID,70)
         GO TO 1
      ENDIF

      CALL SCALNM(IWSID,SCALE)

      CALL RDRWJT(IWSID,JNAM,JNUM,JOINT,AXISR,SCALE)
      CALL RDRWLK(IWSID,JNAM,JNUM,JOINT,AXISR,LNAM,LNUM,LINK,SCALE)

 1000 CALL GPSTMO(IWSID,1,1,2)
      RETURN
      END
```

SUBROUTINE CHVIEW

```
      SUBROUTINE CHVIEW(IWSID,AXISR,CSIZE,SCALE,JNAM,JNUM,JOINT,
     & LNAM,LNUM,LINK)
*************************************************************************
* ----- SUBROUTINE CHVIEW -----                                         *
*                                                                       *
*                                                                       *
*   THIS SUBROUTINE IS USED TO CHANGE THE MAIN VIEW                     *
*                                                                       *
*************************************************************************

      INTEGER IWSID,CHOICE,JNUM(20,3),LNUM(30,14)

      REAL CSIZE(3),AXISR,JOINT(20,21),LINK(30,6)

      CHARACTER*7 JNAM(20),LNAM(30)

    5 CALL CMMAND(IWSID,10)
      CALL MENU(IWSID,CSIZE,13,CHOICE)


      IF(CHOICE.EQ.1)THEN
         CALL ZOOM(IWSID,AXISR,CSIZE)
         GO TO 5
      ENDIF

      IF(CHOICE.EQ.2)THEN
         CALL RESTOR(IWSID,AXISR,CSIZE)
         GO TO 5
      ENDIF

      IF(CHOICE.EQ.3)THEN
         CALL CHSCAL(IWSID,AXISR,CSIZE,SCALE,JNAM,JNUM,JOINT,
     &               LNAM,LNUM,LINK)
         GO TO 5
      ENDIF

      IF(CHOICE.EQ.4)THEN
         CALL INVJ(IWSID,JNAM,JNUM,JOINT,AXISR,SCALE,CSIZE)
         GO TO 5
      ENDIF

      IF(CHOICE.EQ.5)THEN
         CALL INVL(IWSID,JNAM,JNUM,JOINT,AXISR,LNAM,LNUM,LINK,SCALE,
     &             CSIZE)
         GO TO 5
      ENDIF

      IF(CHOICE.EQ.6)THEN
         CALL REINV(IWSID,JNAM,JNUM,JOINT,AXISR,LNAM,LNUM,LINK,SCALE,
     &              CSIZE)
         GO TO 5
      ENDIF

      IF(CHOICE.EQ.7)THEN
         CONTINUE
      ENDIF

      RETURN
      END
```

SUBROUTINE CLOSE

```
      SUBROUTINE CLOSE(IWSID,CSIZE)
*************************************************************************
* ----- SUBROUTINE CLOSE -----                                          *
*                                                                       *
```

```
*                                                                      *
*   THIS SUBROUTINE IS USED TO CLOSE PHIGS AT THE END OF A             *
*                                                                      *
*   MECHIN SESSION AND TO HANDLE ERRORS INCURRED DURING MECHIN         *
*                                                                      *
*   OPERATION.                                                         *
*                                                                      *
************************************************************************

      INTEGER CHOICE,IWSID

      REAL CSIZE(3)

************************************************************************
*  SEE IF ANOTHER MECHANISM IS TO BE INPUT ELSE CLOSE
************************************************************************

  100 CALL ANOTH(IWSID,CSIZE,CHOICE)
      IF(CHOICE.EQ.1)THEN
         CALL ANIN(IWSID,CSIZE)
         GO TO 100
      ENDIF
      IF(CHOICE.EQ.2)THEN
         CALL SYNIN(IWSID,CSIZE)
         GO TO 100
      ENDIF

************************************************************************
*  CLOSE PHIGS AND WORKSTATIONS
************************************************************************
C                             DELETE ALL STRUCTURES
      CALL GPDAST
C                             CLOSE PHIGS
      CALL GPCLPH

      RETURN
      END
```

SUBROUTINE CMMAND

```
      SUBROUTINE CMMAND(IWSID,NUM)
************************************************************************
* ----- SUBROUTINE COMMAND -----                                       *
*                                                                      *
*                                                                      *
*   THIS SUBROUTINE IS USED TO DISPLAY COMMANDS IN THE COMMAND BOX     *
*                                                                      *
************************************************************************

      INTEGER IWSID,LENGT,PRFORM,NUM

      REAL POSL1(2),POSL2(2),POSL3(2),POSL4(2),POSL5(2),WINDOC(4),VPC(4)

      CHARACTER*19 C(91,5)

C                                   COMMANDS
      DATA C(1,1)/'                 '/
      DATA C(1,2)/'  PICK EITHER NEW '/
      DATA C(1,3)/' MODEL OR RESTART '/
      DATA C(1,4)/'   OF A PREVIOUS  '/
      DATA C(1,5)/'       MODEL      '/

      DATA C(2,1)/'  ENTER THE NAME  '/
      DATA C(2,2)/'   OF THE MECHIN  '/
      DATA C(2,3)/'   RESTART FILE   '/
      DATA C(2,4)/'   (UPPER CASE)   '/
      DATA C(2,5)/'                  '/

      DATA C(3,1)/'                  '/
      DATA C(3,2)/'                  '/
      DATA C(3,3)/'  READING MECHIN  '/
      DATA C(3,4)/'   RESTART FILE   '/
      DATA C(3,5)/'                  '/

      DATA C(4,1)/'                  '/
      DATA C(4,2)/'   ENTER A NAME   '/
      DATA C(4,3)/'  FOR THIS MECHIN '/
      DATA C(4,4)/'    INPUT FILE    '/
      DATA C(4,5)/'   (UPPER CASE)   '/

      DATA C(5,1)/'                  '/
      DATA C(5,2)/'  ENTER THE RANGE '/
      DATA C(5,3)/' OF THE COORDINATE'/
      DATA C(5,4)/'   AXES FOR THIS  '/
      DATA C(5,5)/'       MODEL      '/

      DATA C(6,1)/' FILE NOT A VALID '/
      DATA C(6,2)/'   RESTART FILE   '/
```

```
DATA C(6,3)/' ENTER VALID FILE '/
DATA C(6,4)/'   (UPPER CASE)   '/
DATA C(6,5)/'                  '/

DATA C(7,1)/'                  '/
DATA C(7,2)/'    PLEASE PICK   '/
DATA C(7,3)/'      TYPE OF     '/
DATA C(7,4)/'     DATA INPUT   '/
DATA C(7,5)/'                  '/

DATA C(8,1)/'   FILE ALREADY   '/
DATA C(8,2)/'      EXISTS      '/
DATA C(8,3)/'RE-ENTER FILE NAME'/
DATA C(8,4)/'TO USE IT OR ENTER'/
DATA C(8,5)/'  A NEW FILE NAME '/

DATA C(9,1)/'   FILE DOES NOT  '/
DATA C(9,2)/'      EXIST       '/
DATA C(9,3)/' ENTER VALID FILE '/
DATA C(9,4)/'   (UPPER CASE)   '/
DATA C(9,5)/'                  '/

DATA C(10,1)/'                 '/
DATA C(10,2)/'      SELECT     '/
DATA C(10,3)/'  DESIRED OPTION '/
DATA C(10,4)/'                 '/
DATA C(10,5)/'                 '/

DATA C(11,1)/'  INPUT LOCATION '/
DATA C(11,2)/'    DIRECTLY     '/
DATA C(11,3)/'(ENTER X LOCATION)'/
DATA C(11,4)/'       OR        '/
DATA C(11,5)/'  USE VALUATORS  '/

DATA C(12,1)/'  END OF PROGRAM '/
DATA C(12,2)/'     FOR NOW     '/
DATA C(12,3)/'                 '/
DATA C(12,4)/' PRESS ANY BUTTON '/
DATA C(12,5)/'     TO QUIT     '/

DATA C(13,1)/' ENTER THE NAME OF'/
DATA C(13,2)/'   THE POINT OR  '/
DATA C(13,3)/'  PRESS ENTER TO '/
DATA C(13,4)/'    USE DEFAULT  '/
DATA C(13,5)/'                 '/

DATA C(14,1)/'    POINT NAME   '/
DATA C(14,2)/'  ALREADY EXISTS '/
DATA C(14,3)/'                 '/
DATA C(14,4)/' ENTER NEW NAME OR'/
DATA C(14,5)/'  CHOOSE DEFAULT '/

DATA C(15,1)/'    POINT NAME   '/
DATA C(15,2)/'  CAN NOT BE USED '/
DATA C(15,3)/'                 '/
DATA C(15,4)/' ENTER NEW NAME OR'/
DATA C(15,5)/'  CHOOSE DEFAULT '/

DATA C(16,1)/'NUMBER OF POINTS '/
DATA C(16,2)/'   EXCEEDS THE   '/
DATA C(16,3)/'MAXIMUM ALLOWABLE'/
DATA C(16,4)/'                 '/
DATA C(16,5)/'   PRESS ENTER   '/

DATA C(17,1)/' POINT IS OUTSIDE '/
DATA C(17,2)/'  THE PREDEFINED  '/
DATA C(17,3)/' AXIS RANGE. PICK '/
DATA C(17,4)/' RE-ENTER AXIS OR '/
DATA C(17,5)/'  RE-ENTER POINT  '/

DATA C(18,1)/'                 '/
DATA C(18,2)/'     INPUT       '/
DATA C(18,3)/'       Y         '/
DATA C(18,4)/'    LOCATION     '/
DATA C(18,5)/'                 '/

DATA C(19,1)/'                 '/
DATA C(19,2)/'     INPUT       '/
DATA C(19,3)/'       Z         '/
DATA C(19,4)/'    LOCATION     '/
DATA C(19,5)/'                 '/

DATA C(20,1)/'  ENTER POINT OR '/
DATA C(20,2)/'     RETURN      '/
DATA C(20,3)/'   TO RE-ENTER   '/
DATA C(20,4)/'   POINT DATA    '/
DATA C(20,5)/'                 '/

DATA C(21,1)/'                 '/
DATA C(21,2)/'  PICK POINT TO  '/
DATA C(21,3)/'     DELETE      '/
DATA C(21,4)/'                 '/
DATA C(21,5)/'                 '/

DATA C(22,1)/'  PICK POINT OR  '/
DATA C(22,2)/'     JOINT       '/
```

```
DATA C(22,3)/'  FOR POSITION    '/
DATA C(22,4)/'    INQUIRY       '/
DATA C(22,5)/'                  '/

DATA C(23,1)/' NUMBER OF JOINTS '/
DATA C(23,2)/'   EXCEEDS THE    '/
DATA C(23,3)/' MAXIMUM ALLOWABLE'/
DATA C(23,4)/'                  '/
DATA C(23,5)/'   PRESS ENTER    '/

DATA C(24,1)/' ENTER THE NAME OF'/
DATA C(24,2)/'   THE JOINT OR   '/
DATA C(24,3)/'  PRESS ENTER TO  '/
DATA C(24,4)/'    USE DEFAULT   '/
DATA C(24,5)/'                  '/

DATA C(25,1)/'    JOINT NAME    '/
DATA C(25,2)/'  ALREADY EXISTS  '/
DATA C(25,3)/'                  '/
DATA C(25,4)/' ENTER NEW NAME OR'/
DATA C(25,5)/'  CHOOSE DEFAULT  '/

DATA C(26,1)/'                  '/
DATA C(26,2)/'   CHOOSE JOINT   '/
DATA C(26,3)/'      TYPE        '/
DATA C(26,4)/'                  '/
DATA C(26,5)/'                  '/

DATA C(27,1)/'                  '/
DATA C(27,2)/'     INPUT        '/
DATA C(27,3)/'       X          '/
DATA C(27,4)/' DIRECTION COSINE '/
DATA C(27,5)/'                  '/

DATA C(28,1)/'                  '/
DATA C(28,2)/'     INPUT        '/
DATA C(28,3)/'       Y          '/
DATA C(28,4)/' DIRECTION COSINE '/
DATA C(28,5)/'                  '/

DATA C(29,1)/'                  '/
DATA C(29,2)/'     INPUT        '/
DATA C(29,3)/'       Z          '/
DATA C(29,4)/' DIRECTION COSINE '/
DATA C(29,5)/'                  '/

DATA C(30,1)/'       PICK       '/
DATA C(30,2)/'     METHOD OF    '/
DATA C(30,3)/'    ORIENTATION   '/
DATA C(30,4)/'     OF JOINT     '/
DATA C(30,5)/'      Z AXIS      '/

DATA C(31,1)/'       VALUE      '/
DATA C(31,2)/'   GREATER THAN   '/
DATA C(31,3)/'       1.0        '/
DATA C(31,4)/'                  '/
DATA C(31,5)/'  PLEASE RE-ENTER '/

DATA C(32,1)/'ENTER ORIENTATION '/
DATA C(32,2)/'    OR RETURN     '/
DATA C(32,3)/'   TO RE-ENTER    '/
DATA C(32,4)/'   ORIENTATION    '/
DATA C(32,5)/'      DATA        '/

DATA C(33,1)/'ENTER ORIENTATION '/
DATA C(33,2)/'  WITH VALUATORS  '/
DATA C(33,3)/'   OR RETURN TO   '/
DATA C(33,4)/'     RE-ENTER     '/
DATA C(33,5)/'      DATA        '/

DATA C(34,1)/'   PICK A POINT   '/
DATA C(34,2)/'  OR JOINT TOWARD '/
DATA C(34,3)/'  WHICH THIS JOINT'/
DATA C(34,4)/'   IS ORIENTED    '/
DATA C(34,5)/'                  '/

DATA C(35,1)/'   PICK A JOINT   '/
DATA C(35,2)/'     TO WHICH     '/
DATA C(35,3)/'    THIS JOINT    '/
DATA C(35,4)/'    IS PARALLEL   '/
DATA C(35,5)/'                  '/

DATA C(36,1)/'                  '/
DATA C(36,2)/'  PICK JOINT TO   '/
DATA C(36,3)/'     DELETE       '/
DATA C(36,4)/'                  '/
DATA C(36,5)/'                  '/

DATA C(37,1)/'   PICK JOINT     '/
DATA C(37,2)/'       FOR        '/
DATA C(37,3)/'     INQUIRY      '/
DATA C(37,4)/' OF JOINT Z AXIS  '/
DATA C(37,5)/'  ORIENTATION     '/

DATA C(38,1)/'  ZERO MAGNITUDE  '/
DATA C(38,2)/'     VECTOR       '/
```

```
DATA C(38,3)/'      ENTERED       '/
DATA C(38,4)/'                    '/
DATA C(38,5)/'   RE-ENTER X       '/

DATA C(39,1)/'  ZERO MAGNITUDE    '/
DATA C(39,2)/'     VECTOR         '/
DATA C(39,3)/'     ENTERED        '/
DATA C(39,4)/'                    '/
DATA C(39,5)/' RE-ENTER VALUES    '/

DATA C(40,1)/' ENTER AN INITIAL   '/
DATA C(40,2)/'        Z           '/
DATA C(40,3)/'   DISPLACEMENT     '/
DATA C(40,4)/'       FOR          '/
DATA C(40,5)/'   THIS JOINT       '/

DATA C(41,1)/'    ENTER THE       '/
DATA C(41,2)/'      LEAD          '/
DATA C(41,3)/'     FOR THIS       '/
DATA C(41,4)/'      SCREW         '/
DATA C(41,5)/'      JOINT         '/

DATA C(42,1)/'  SECOND Z ORIENT.  '/
DATA C(42,2)/'VECTOR IS NEED FOR  '/
DATA C(42,3)/' A SPHERICAL JOINT  '/
DATA C(42,4)/'                    '/

DATA C(43,1)/' SECOND CENTER OF   '/
DATA C(43,2)/' LOCAL COORDINATE   '/
DATA C(43,3)/'  SYSTEM NEEDED     '/
DATA C(43,4)/'       FOR A        '/
DATA C(43,5)/'    FLAT JOINT      '/

DATA C(44,1)/' SECOND CENTER OF   '/
DATA C(44,2)/' LOCAL COORDINATE   '/
DATA C(44,3)/'  SYSTEM NEEDED     '/
DATA C(44,4)/'       FOR A        '/
DATA C(44,5)/'    GEAR PAIR       '/

DATA C(45,1)/'       ENTER        '/
DATA C(45,2)/' THE GEAR RATIO     '/
DATA C(45,3)/'  FOR THIS GEAR     '/
DATA C(45,4)/'       PAIR         '/
DATA C(45,5)/'                    '/

DATA C(46,1)/' NUMBER OF LINKS    '/
DATA C(46,2)/'   EXCEEDS THE      '/
DATA C(46,3)/'MAXIMUM ALLOWABLE   '/
DATA C(46,4)/'                    '/
DATA C(46,5)/'   PRESS ENTER      '/

DATA C(47,1)/' ENTER THE NAME OF  '/
DATA C(47,2)/'    THE LINK OR     '/
DATA C(47,3)/'  PRESS ENTER TO    '/
DATA C(47,4)/'    USE DEFAULT     '/
DATA C(47,5)/'                    '/

DATA C(48,1)/'      LINK NAME     '/
DATA C(48,2)/'  ALREADY EXISTS    '/
DATA C(48,3)/'                    '/
DATA C(48,4)/' ENTER NEW NAME OR  '/
DATA C(48,5)/'  CHOOSE DEFAULT    '/

DATA C(49,1)/'  PICK THE Z AXIS   '/
DATA C(49,2)/'  ON THE JOINT AT   '/
DATA C(49,3)/'   THE FIRST END    '/
DATA C(49,4)/'    OF THE LINK     '/
DATA C(49,5)/'     OR RETURN      '/

DATA C(50,1)/'  PICK THE Z AXIS   '/
DATA C(50,2)/'  ON THE JOINT AT   '/
DATA C(50,3)/'   THE OTHER END    '/
DATA C(50,4)/'    OF THE LINK     '/
DATA C(50,5)/'     OR RETURN      '/

DATA C(51,1)/'  CHOOSE METHOD     '/
DATA C(51,2)/'    OF LOCAL        '/
DATA C(51,3)/'     X AXIS         '/
DATA C(51,4)/'   ORIENTATION      '/
DATA C(51,5)/'                    '/

DATA C(52,1)/'    PICK A POINT    '/
DATA C(52,2)/'  IN THE POSITIVE   '/
DATA C(52,3)/'     X-Z PLANE      '/
DATA C(52,4)/'  OF THE DESIRED    '/
DATA C(52,5)/'   X ORIENTATION    '/

DATA C(53,1)/'    PICK A JOINT    '/
DATA C(53,2)/'  IN THE POSITIVE   '/
DATA C(53,3)/'     X-Z PLANE      '/
DATA C(53,4)/'  OF THE DESIRED    '/
DATA C(53,5)/'   X ORIENTATION    '/

DATA C(54,1)/'     CENTER OF      '/
DATA C(54,2)/'    JOINT CHOSEN    '/
DATA C(54,3)/' IS ON THE Z AXIS.  '/
```

```
DATA C(54,4)/'       PICK         '/
DATA C(54,5)/'   A NEW JOINT      '/

DATA C(55,1)/'    CENTER OF       '/
DATA C(55,2)/'   POINT CHOSEN     '/
DATA C(55,3)/' IS ON THE Z AXIS.  '/
DATA C(55,4)/'       PICK         '/
DATA C(55,5)/'   A NEW POINT      '/

DATA C(56,1)/'                    '/
DATA C(56,2)/'  PICK LINK TO      '/
DATA C(56,3)/'     DELETE         '/
DATA C(56,4)/'                    '/
DATA C(56,5)/'                    '/

DATA C(57,1)/'                    '/
DATA C(57,2)/'PICK FIRST CORNER   '/
DATA C(57,3)/'      OF THE        '/
DATA C(57,4)/'   ZOOM AREA        '/
DATA C(57,5)/'                    '/

DATA C(58,1)/'PICK THE OPPOSITE   '/
DATA C(58,2)/'      CORNER        '/
DATA C(58,3)/'      OF THE        '/
DATA C(58,4)/'   ZOOM AREA        '/
DATA C(58,5)/'                    '/

DATA C(59,1)/'                    '/
DATA C(59,2)/'   PICK LINK        '/
DATA C(59,3)/'       FOR          '/
DATA C(59,4)/'  NAME INQUIRY      '/
DATA C(59,5)/'                    '/

DATA C(60,1)/'   ENTER A NAME     '/
DATA C(60,2)/'    FOR MECHIN      '/
DATA C(60,3)/'   RESTART FILE     '/
DATA C(60,4)/'   (UPPER CASE)     '/
DATA C(60,5)/'                    '/

DATA C(61,1)/'                    '/
DATA C(61,2)/'  PICK JOINT FOR    '/
DATA C(61,3)/'INITIAL CONDITION   '/
DATA C(61,4)/'  SPECIFICATION     '/
DATA C(61,5)/'                    '/

DATA C(62,1)/'  INPUT JOINT       '/
DATA C(62,2)/'MUST BE A REVOLUTE  '/
DATA C(62,3)/' OR A PRISMATIC     '/
DATA C(62,4)/'      JOINT         '/
DATA C(62,5)/'PICK A NEW JOINT    '/

DATA C(63,1)/' ENTER THE INITIAL  '/
DATA C(63,2)/'   POSITION OF      '/
DATA C(63,3)/'THE REVOLUTE JOINT  '/
DATA C(63,4)/'   IN DEGREES       '/
DATA C(63,5)/'                    '/

DATA C(64,1)/'    ENTER THE       '/
DATA C(64,2)/'  INCREMENT FOR     '/
DATA C(64,3)/'  THIS REVOLUTE     '/
DATA C(64,4)/'      JOINT         '/
DATA C(64,5)/'   IN DEGREES       '/

DATA C(65,1)/'    ENTER THE       '/
DATA C(65,2)/'    NUMBER OF       '/
DATA C(65,3)/' INCREMENTS TO BE   '/
DATA C(65,4)/'MADE BY THIS JOINT  '/
DATA C(65,5)/'      VARIABLE      '/

DATA C(66,1)/'  ENTER INITIAL     '/
DATA C(66,2)/'    CONDITION       '/
DATA C(66,3)/'     OR PICK        '/
DATA C(66,4)/'     RETURN         '/
DATA C(66,5)/'                    '/

DATA C(67,1)/' ENTER THE INITIAL  '/
DATA C(67,2)/'   POSITION OF      '/
DATA C(67,3)/'  THE PRISMATIC     '/
DATA C(67,4)/'      JOINT         '/
DATA C(67,5)/'                    '/

DATA C(68,1)/'    ENTER THE       '/
DATA C(68,2)/'  INCREMENT FOR     '/
DATA C(68,3)/'  THIS PRISMATIC    '/
DATA C(68,4)/'      JOINT         '/
DATA C(68,5)/'                    '/

DATA C(69,1)/'    ENTER THE       '/
DATA C(69,2)/'   SCALE FACTOR     '/
DATA C(69,3)/'  FOR THE JOINT     '/
DATA C(69,4)/'      ICONS         '/
DATA C(69,5)/'                    '/

DATA C(70,1)/'  SCALE ENTERED     '/
DATA C(70,2)/'   IS LESS THAN     '/
DATA C(70,3)/' OR EQUAL TO ZERO   '/
```

```
DATA C(70,4)/'                    '/
DATA C(70,5)/'  PLEASE RE-ENTER   '/

DATA C(71,1)/'                    '/
DATA C(71,2)/'  PICK JOINT TO     '/
DATA C(71,3)/'     DELETE         '/
DATA C(71,4)/'INITIAL CONDITIONS  '/
DATA C(71,5)/'                    '/

DATA C(72,1)/'                    '/
DATA C(72,2)/'  PICK TYPE OF      '/
DATA C(72,3)/' DATA INPUT FOR     '/
DATA C(72,4)/'INITIAL CONDITIONS  '/
DATA C(72,5)/'                    '/

DATA C(73,1)/'    ENTER THE       '/
DATA C(73,2)/'   VELOCITY OF      '/
DATA C(73,3)/'THE REVOLUTE JOINT  '/
DATA C(73,4)/'                    '/
DATA C(73,5)/'                    '/

DATA C(74,1)/'    ENTER THE       '/
DATA C(74,2)/'   VELOCITY OF      '/
DATA C(74,3)/'  THE PRISMATIC     '/
DATA C(74,4)/'     JOINT          '/
DATA C(74,5)/'                    '/

DATA C(75,1)/'    ENTER THE       '/
DATA C(75,2)/' ACCELERATION OF    '/
DATA C(75,3)/'THE REVOLUTE JOINT  '/
DATA C(75,4)/'                    '/
DATA C(75,5)/'                    '/

DATA C(76,1)/'    ENTER THE       '/
DATA C(76,2)/' ACCELERATION OF    '/
DATA C(76,3)/'  THE PRISMATIC     '/
DATA C(76,4)/'     JOINT          '/
DATA C(76,5)/'                    '/

DATA C(77,1)/'   PICK JOINT       '/
DATA C(77,2)/'      FOR           '/
DATA C(77,3)/'    INQUIRY         '/
DATA C(77,4)/' OF JOINT INITIAL   '/
DATA C(77,5)/'   CONDITIONS       '/

DATA C(78,1)/'                    '/
DATA C(78,2)/'  PICK TYPE OF      '/
DATA C(78,3)/'    DATA FOR        '/
DATA C(78,4)/'INITIAL CONDITIONS  '/
DATA C(78,5)/'    INQUIRY         '/

DATA C(79,1)/' POSITION VALUES    '/
DATA C(79,2)/'MUST BE SPECIFIED   '/
DATA C(79,3)/'     BEFORE         '/
DATA C(79,4)/' VELOCITIES AND     '/
DATA C(79,5)/'  ACCELERATIONS     '/

DATA C(80,1)/'  PICK THE Z AXIS   '/
DATA C(80,2)/'   ON THE JOINT     '/
DATA C(80,3)/'   ALSO ON THE      '/
DATA C(80,4)/'    EXISTING        '/
DATA C(80,5)/'      LINK          '/

DATA C(81,1)/'   GROUND LINK      '/
DATA C(81,2)/'    INVALID         '/
DATA C(81,3)/'                    '/
DATA C(81,4)/' PICK A NEW JOINT   '/
DATA C(81,5)/'                    '/

DATA C(82,1)/'   NUMBER OF        '/
DATA C(82,2)/' JOINTS ON LINK     '/
DATA C(82,3)/'    EXCEEDS         '/
DATA C(82,4)/'       6            '/
DATA C(82,5)/'                    '/

DATA C(83,1)/'   ENTER A NAME     '/
DATA C(83,2)/'    FOR IMP         '/
DATA C(83,3)/'   DATA FILE        '/
DATA C(83,4)/'  (UPPER CASE)      '/
DATA C(83,5)/'                    '/

DATA C(84,1)/' ORIENTATION POINT  '/
DATA C(84,2)/' AND JOINT ORIGIN   '/
DATA C(84,3)/'   ARE THE SAME     '/
DATA C(84,4)/'                    '/
DATA C(84,5)/' PICK A NEW POINT   '/

DATA C(85,1)/'    ENTER A NEW     '/
DATA C(85,2)/'    MECHANISM       '/
DATA C(85,3)/'       OR           '/
DATA C(85,4)/'      QUIT          '/
DATA C(85,5)/'                    '/

DATA C(86,1)/'   PICK JOINT TO    '/
DATA C(86,2)/'   MAKE INVISIBLE   '/
DATA C(86,3)/'       OR           '/
```

```
      DATA C(86,4)/'      RETURN      '/
      DATA C(86,5)/'                  '/

      DATA C(87,1)/'    PICK LINK TO  '/
      DATA C(87,2)/'   MAKE INVISIBLE '/
      DATA C(87,3)/'        OR        '/
      DATA C(87,4)/'      RETURN      '/

      DATA C(88,1)/'   PICK ANALYSIS  '/
      DATA C(88,2)/'    ROUTINE FOR   '/
      DATA C(88,3)/'    INPUT FILE    '/
      DATA C(88,4)/'    GENERATION    '/
      DATA C(88,5)/'                  '/

      DATA C(89,1)/'  OPTION NOT YET  '/
      DATA C(89,2)/'    AVAILABLE     '/
      DATA C(89,3)/'                  '/
      DATA C(89,4)/'  PLEASE CHOOSE   '/
      DATA C(89,5)/' ANOTHER ROUTINE  '/

      DATA C(90,1)/' MECHANISM IS NOT '/
      DATA C(90,2)/'      AN RSCR     '/
      DATA C(90,3)/'                  '/
      DATA C(90,4)/'  PLEASE CHOOSE   '/
      DATA C(90,5)/' ANOTHER ROUTINE  '/

      DATA C(91,1)/'   ENTER A NAME   '/
      DATA C(91,2)/'     FOR RSCR     '/
      DATA C(91,3)/'    DATA FILE     '/
      DATA C(91,4)/'   (UPPER CASE)   '/
      DATA C(91,5)/'                  '/

C                               COMMAND LENGTH AND LINE POSITIONS
      DATA LENGT/19/
      DATA POSL1/.03,.7/
      DATA POSL2/.03,.6/
      DATA POSL3/.03,.5/
      DATA POSL4/.03,.4/
      DATA POSL5/.03,.3/
C                               FLAG FOR UPDATE
      DATA PRFORM/2/
C                               WINDOW AND VIEWPORT OF COMMAND BOX
      DATA WINDOC/0.00,1.00,0.00,1.00/
      DATA VPC/.01,.2375,0.67,0.975/

*****************************************************************
*  DELETE ALL PREVIOUS COMMENTS
*****************************************************************
C                               DELETE ALL PREVIOUS COMMENTS
      CALL GPEST(5)

*****************************************************************
*  OPEN STRUCTURE FOR COMMAND1
*****************************************************************
C                               OPEN A STRUCTURE
      CALL GPOPST(5)
C                               SET TEXT COLOR TO YELLOW
      CALL GPTXCI(2)
C                               SET ANNOTATION SIZE SCALE FACTOR
      CALL GPAHSC(1.0)
C                               DRAW TEXT
      CALL GPAN2(POSL1,LENGT,C(NUM,1))
      CALL GPAN2(POSL2,LENGT,C(NUM,2))
      CALL GPAN2(POSL3,LENGT,C(NUM,3))
      CALL GPAN2(POSL4,LENGT,C(NUM,4))
      CALL GPAN2(POSL5,LENGT,C(NUM,5))
C                               CLOSE STRUCTURE
      CALL GPCLST

*****************************************************************
*  SCREEN DISPLAY---------- VIEW INITIALIZATION GOES TO SETUP
*****************************************************************
C                               UPDATE THE WORKSTATION
      CALL GPUPWS(IWSID,PRFORM)

      RETURN
      END
```

SUBROUTINE COUPLE

```
      SUBROUTINE COUPLE(IWSID,LNAM,LNUM,LINK,JNAM,JNUM,JOINT,NUML,CSIZE)
*****************************************************************
* ----- SUBROUTINE COUPLE -----                                 *
*                                                               *
*                                                               *
*   THIS SUBROUTINE IS USED TO ADD A COUPLING LINK              *
```

```
*                                                                      *
************************************************************************

      COMMON/LDEF/LNDEF(30)

      INTEGER IWSID,NUML,PNUM(100),RFLG,JNUM(20,3),LNUM(30,14),CLASS(1),
     &     CHOICE

      REAL AXISR,LOCAT(3),AREA(6),POINT(100,3),CSIZE(3),JOINT(20,21),
     &  LINK(30,6),ENDP1(3),ENDP2(3),LNK(6),XVEC(3),YVEC(3),X1(6),X2(6),
     &  JOINTT(20,21),SC(3),SCM(4,4),TR(3),TRM(4,4),TRM1(4,4),P(3),
     &  P1(3),P2(3),S(4,4),T(4,4),VEC(3),ZVEC(3)

      CHARACTER*7 LNDEF,LNAME,PNAM(100),JNAM(20),LNAM(30),CLEAR,CLEAR1
      CHARACTER*20 CLEAR2

      DATA CLEAR     /'       '/
      DATA CLEAR1    /'       '/
      DATA CLEAR2    /'                    '/
      DATA SCM/1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
      DATA TRM/1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
      DATA TRM1/1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
      DATA S  /1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
      DATA T  /1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./

      DATA CLASS  /5/

      RFLG=0

      LOCAT(1)=0.0
      LOCAT(2)=0.0
      LOCAT(3)=0.0

      AREA(1)=0.08*CSIZE(1)
      AREA(2)=0.90*CSIZE(1)
      AREA(3)=0.70*CSIZE(1)
      AREA(4)=0.90*CSIZE(1)
      AREA(5)=0.02*CSIZE(1)
      AREA(6)=0.90*CSIZE(1)

      LNAME=CLEAR

      SC(1)=SCALE
      SC(2)=SCALE
      SC(3)=SCALE

      CALL GPSC3(SC,SCM)
************************************************************************
*    PICK FIRST END OF THE LINK
************************************************************************

   10 CALL CMMAND(IWSID,80)
      CALL PICKJ(IWSID,JOINT,AXISR,CSIZE,JNUM1,ENDP1,IAXIS1,RFLG)
      IF(JNUM(JNUM1,2).EQ.8)THEN
C                GROUND LINK INVALID
         CALL CMMAND(IWSID,81)
         GO TO 100
      ENDIF

      IF(RFLG.EQ.1)THEN
         GO TO 100
      ENDIF

************************************************************************
*    DEFINE THE X AXIS OF THE JOINT AT THE FIRST END OF THE LINK
************************************************************************

      CALL CMMAND(IWSID,51)
      CALL MENU(IWSID,CSIZE,12,CHOICE)
      IF(CHOICE.EQ.1)THEN
         CALL CMMAND(IWSID,53)
         CALL TEJOX(IWSID,JNUM1,IAXIS1,JOINT,AXISR,CSIZE,XVEC,YVEC,RFLG)
         IF(IAXIS1.EQ.1)THEN
              JOINT(JNUM1,7)=XVEC(1)
              JOINT(JNUM1,8)=XVEC(2)
              JOINT(JNUM1,9)=XVEC(3)
         ENDIF
         IF(IAXIS1.EQ.2)THEN
              JOINT(JNUM1,10)=XVEC(1)
              JOINT(JNUM1,11)=XVEC(2)
              JOINT(JNUM1,12)=XVEC(3)
         ENDIF
         IF(RFLG.EQ.1)THEN
            GO TO 100
         ENDIF
      ENDIF
      IF(CHOICE.EQ.2)THEN
         CALL CMMAND(IWSID,52)
         CALL TEPOX(IWSID,JNUM1,IAXIS1,POINT,AXISR,CSIZE,XVEC,YVEC,
     &JOINT,RFLG)
         IF(IAXIS1.EQ.1)THEN
              JOINT(JNUM1,7)=XVEC(1)
              JOINT(JNUM1,8)=XVEC(2)
              JOINT(JNUM1,9)=XVEC(3)
```

```
         ENDIF
         IF(IAXIS1.EQ.2)THEN
              JOINT(JNUM1,10)=XVEC(1)
              JOINT(JNUM1,11)=XVEC(2)
              JOINT(JNUM1,12)=XVEC(3)
         ENDIF
         IF(RFLG.EQ.1)THEN
            GO TO 100
         ENDIF
      ENDIF

*****************************************************************
*    ADD POINT DATA TO POINT ARRAYS
*****************************************************************

C                     FIND NEXT AVAILABLE SLOT IN NUML
      KKK=0

      DO 20 I=6,12,2
         IF(LNUM(NUML,I).EQ.0.AND.LNUM(NUML,I+1).EQ.0)THEN
              KKK=1
      WRITE(6,*)'FOUND A SLOT  '
              LNUM(NUML,I)=JNUM1
              LNUM(NUML,I+1)=IAXIS1
              GO TO 22
         ENDIF
   20 CONTINUE

   22 IF(KKK.EQ.0)THEN
C                     NUMBER OF JOINTS ON LINK EXCEEDS 6
         CALL CMMAND(IWSID,82)
         CALL GPAWEV(3.,IS,IT,IU)
         GO TO 100
      ENDIF

      RLC1=LINK(NUML,1)+(LINK(NUML,4)-LINK(NUML,1))/2.
      RLC2=LINK(NUML,2)+(LINK(NUML,5)-LINK(NUML,2))/2.
      RLC3=LINK(NUML,3)+(LINK(NUML,6)-LINK(NUML,3))/2.

  100 RETURN
      END
```

SUBROUTINE DATA

```
      SUBROUTINE DATA(IWSID,AXISR,CSIZE,PNAM,PNUM,POINT,JNAM,JNUM,
     &JOINT,LNAM,LNUM,LINK,INAM,INUM,IC,SCALE)
*****************************************************************
* ----- SUBROUTINE DATA -----                                   *
*                                                               *
*                                                               *
*   THIS SUBROUTINE IS USED FOR DATA ENTRY FOR ANALYSIS         *
*                                                               *
*****************************************************************

      INTEGER IWSID,CHOICE,PNUM(100),LNUM(30,14),JNUM(20,3),INUM(20,2)

      CHARACTER*7 PNAM(100),JNAM(20),LNAM(30),INAM(20)

      REAL AXISR,CSIZE(3),POINT(100,3),JOINT(20,21),LINK(30,6),SCALE,
     & IC(20,4)


*****************************************************************
*    ASSOCIATE ICON,POINT,JOINT,AND LINK STRUCTURES WITH VIEWS
*****************************************************************
C                     POSITION ICON
      CALL GPARV(IWSID,5,9,0.)
      CALL GPARV(IWSID,6,16,0.)
      CALL GPARV(IWSID,4,11,0.)
C                     POINTS
      CALL GPARV(IWSID,4,12,0.)
C                     JOINT NAMES
      CALL GPARV(IWSID,4,13,0.)
C                     JOINTS
      CALL GPARV(IWSID,4,14,0.)
C                     LINKS
      CALL GPARV(IWSID,4,15,0.)
C                     Z AXIS 1
      CALL GPARV(IWSID,4,17,0.)
C                     Z AXIS 2
      CALL GPARV(IWSID,4,18,0.)
C                     LINK
      CALL GPARV(IWSID,4,19,0.)
C                     I.C.
      CALL GPARV(IWSID,4,20,0.)
```

```
*********************************************************************
*    INQUIRE TYPE OF DATA INPUT
*********************************************************************

   10 CALL CMMAND(IWSID,7)
      CALL MENU(IWSID,CSIZE,2,CHOICE)
      IF(CHOICE.EQ.1)THEN
C                              JOINTS
         CALL JOINTS(IWSID,AXISR,CSIZE,JNAM,JNUM,JOINT,PNAM,PNUM,POINT,
     &SCALE)
         GO TO 10
      ENDIF

      IF(CHOICE.EQ.2)THEN
C                              LINKS
         CALL LINKS(IWSID,AXISR,CSIZE,JNAM,JNUM,JOINT,PNAM,PNUM,POINT,
     &LNAM,LNUM,LINK,SCALE)
         GO TO 10
      ENDIF

      IF(CHOICE.EQ.3)THEN
C                              INITIAL CONDITIONS
         CALL ICOND(IWSID,AXISR,CSIZE,JNAM,JNUM,JOINT,INAM,INUM,IC,
     &   SCALE)
         GO TO 10
      ENDIF

      IF(CHOICE.EQ.4)THEN
C                              CHANGE VIEWS
         CALL CHVIEW(IWSID,AXISR,CSIZE,SCALE,JNAM,JNUM,JOINT,
     &              LNAM,LNUM,LINK)
         GO TO 10
      ENDIF

      IF(CHOICE.EQ.6)THEN
C                              END OF INPUT DATA
         RETURN
      ENDIF


      RETURN
      END
```

SUBROUTINE DELIC

```
      SUBROUTINE DELIC(IWSID,CSIZE,AXISR,INAM,INUM,IC,JOINT,SCALE)
*********************************************************************
* ----- SUBROUTINE DELIC -----                                      *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED TO DELETE AN I.C.                       *
*                                                                   *
*********************************************************************

      INTEGER IWSID,PRFORM,PPATH(3),ACLASS(3),CHOICE,INUM(20,2),LENG

      REAL CSIZE(3),DATA(35),PAREA(6),L1(4),JOINT(20,21),AXISR,POSN(2),
     &IC(20,4)

      CHARACTER*7 INAM(20),C,D
      CHARACTER*16 RETURN

      DATA RETURN/'1.  RETURN      '/
      DATA LENG/16/
      DATA POSN/0.05,2.2/
      DATA C/'CCCCCCC'/
      DATA D/'DDDDDDD'/

      DATA DATA/32,0,0,          2,2,1,1,
     &                         1,1,1,1,1,1,
     &                         1,1,1,1,1,1,
     &                         1,1,1,1,1,1,
     &                         1,1,1,1,1,1,
     &                           1,1,1,1/
C                              FLAG FOR UPDATE
      DATA PRFORM/2/
C                              INCLUSION CLASS LIST
      DATA ACLASS/3,4,6/


*********************************************************************
*  REQUEST PICK INPUT
*********************************************************************
C                              SET PICK FILTER(ALL CLASS 2 DETECTABLE)
      CALL GPPKF(IWSID,1,3,ACLASS,0,ACLASS)
C                              MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                              DEFINE PICK AREA
      PAREA(1)=((-.98+1)/2)*CSIZE(1)
      PAREA(2)=((.98+1)/2)*CSIZE(1)
```

```
      PAREA(3)=((-.98+1)/2)*CSIZE(2)
      PAREA(4)=((.48+1)/2)*CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=CSIZE(3)
C                             INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                             PLACE PICK DEVICE IN THE EVENT MODE
      CALL GPPKMO(IWSID,1,3,2)
C                             TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)
C                             GET PICK OR RETURN COMMAND
      CALL CMMAND(IWSID,71)

*********************************************************************
*  INCLUDE A RETURN PICK
*********************************************************************

      CALL GPOPST(6)
      CALL GPADCN(1,ACLASS)
      CALL GPPKID(101)
      CALL GPTXCI(2)
      CALL GPAHSC(1.0)
      CALL GPAN2(POSN,LENG,RETURN)
      CALL GPCLST
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
      CALL GPUPWS(IWSID,PRFORM)

*********************************************************************
*  AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
*********************************************************************
C                             AWAIT AN EVENT
  200 CALL GPAWEV(1000.,IWS,ICLA,IDEV)
      IF(ICLA.EQ.5)THEN
C                             GET PICK
         CALL GPGTPK(1,1,PPATH)
C                             IF NO PICK, RETURN
         IF(PPATH(2).EQ.101)GO TO 1000
C                             CHOICE IS SECOND ITEM OF PICK PATH
         CHOICE=PPATH(2)
C                             DEACTIVATE POINT
         INAM(CHOICE)=C
         INUM(CHOICE,1)=0
         INUM(CHOICE,2)=0
         IC(CHOICE,1)=0.
         IC(CHOICE,2)=0.
         IC(CHOICE,3)=0.
         IC(CHOICE,4)=0.
C                             REDRAW ALL POINTS
         CALL ICICON(IWSID,JOINT,INAM,INUM,IC,SCALE)
C                             AWAIT ANOTHER EVENT
         GO TO 200
      ELSE
C                             IF NOT A PICK, GO TO AWAIT AN EVENT
         GO TO 200
      ENDIF


C                             MAKE SURE PICK IS DEACTIVATED
 1000 CALL GPPKMO(IWSID,1,1,2)
      CALL GPEST(6)

C                             SET PICK FILTER(ALL CLASS 2 UNDETECTABLE)
      CALL GPPKF(IWSID,1,0,ACLASS,3,ACLASS)


      RETURN
      END
```

SUBROUTINE DELJ

```
      SUBROUTINE DELJ(IWSID,CSIZE,AXISR,JNAM,JNUM,JOINT,SCALE)
*********************************************************************
* ----- SUBROUTINE DELJ -----                                       *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED TO DELETE A JOINT                       *
*                                                                   *
*********************************************************************

      INTEGER IWSID,PRFORM,PPATH(3),ACLASS(2),CHOICE,JNUM(20,3),LENG

      REAL CSIZE(3),DATA(35),PAREA(6),L1(4),JOINT(20,21),AXISR,POSN(2),
     & SCALE

      CHARACTER*7 JNAM(20),C
      CHARACTER*16 RETURN

      DATA RETURN/'1.  RETURN      '/
      DATA LENG/16/
```

```
      DATA POSN/0.05,2.2/
      DATA C/'CCCCCCC'/

      DATA DATA/32,0,0,         2,2,1,1,
     &                        1,1,1,1,1,1,
     &                        1,1,1,1,1,1,
     &                        1,1,1,1,1,1,
     &                        1,1,1,1,1,1,
     &                          1,1,1,1/
C                             FLAG FOR UPDATE
      DATA PRFORM/2/
C                             INCLUSION CLASS LIST
      DATA ACLASS/3,4/

*********************************************************************
*  REQUEST PICK INPUT
*********************************************************************
C                             SET PICK FILTER(ALL CLASS 3 DETECTABLE)
      CALL GPPKF(IWSID,1,2,ACLASS,0,ACLASS)
C                             MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                             DEFINE PICK AREA
      PAREA(1)=((-.98+1)/2)*CSIZE(1)
      PAREA(2)=((.98+1)/2)*CSIZE(1)
      PAREA(3)=((-.98+1)/2)*CSIZE(2)
      PAREA(4)=((.48+1)/2)*CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=CSIZE(3)
C                             INITIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                             PLACE PICK DEVICE IN THE EVENT MODE
      CALL GPPKMO(IWSID,1,3,2)
C                             TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)
C                             GET PICK OR RETURN COMMAND
      CALL CMMAND(IWSID,36)

*********************************************************************
*  INCLUDE A RETURN PICK
*********************************************************************

      CALL GPOPST(6)
      CALL GPADCN(1,ACLASS)
      CALL GPPKID(101)
      CALL GPTXCI(2)
      CALL GPAHSC(1.0)
      CALL GPAN2(POSN,LENG,RETURN)
      CALL GPCLST
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
      CALL GPUPWS(IWSID,PRFORM)

*********************************************************************
*  AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
*********************************************************************
C                             AWAIT AN EVENT
  200 CALL GPAWEV(1000.,IWS,ICLA,IDEV)
      IF(ICLA.EQ.5)THEN
C                             GET PICK
         CALL GPGTPK(1,1,PPATH)
C                             IF NO PICK, RETURN
         IF(PPATH(2).EQ.101)GO TO 1000
C                             CHOICE IS SECOND ITEM OF PICK PATH
         CHOICE=PPATH(2)
C                             DEACTIVATE POINT
         JNAM(CHOICE)=C
         DO 201 I=1,21
              JOINT(CHOICE,I)=0.0
  201    CONTINUE
C                             REDRAW ALL POINTS
         CALL RDRWJT(IWSID,JNAM,JNUM,JOINT,AXISR,SCALE)
C                             AWAIT ANOTHER EVENT
         GO TO 200
      ELSE
C                             IF NOT A PICK, GO TO AWAIT AN EVENT
         GO TO 200
      ENDIF


C                             MAKE SURE PICK IS DEACTIVATED
 1000 CALL GPPKMO(IWSID,1,1,2)
      CALL GPEST(6)

C                             SET PICK FILTER(ALL CLASS 2 UNDETECTABLE)
      CALL GPPKF(IWSID,1,0,ACLASS,2,ACLASS)


      RETURN
      END
```

SUBROUTINE DELL

```
      SUBROUTINE DELL(IWSID,CSIZE,AXISR,JNAM,JNUM,JOINT,LNAM,LNUM,LINK,
     &SCALE)
*********************************************************************
* ----- SUBROUTINE DELL -----                                       *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED TO DELETE A LINK                        *
*                                                                   *
*********************************************************************

      INTEGER IWSID,PRFORM,PPATH(3),ACLASS(1),CHOICE,JNUM(20,3),LENG,
     &        LNUM(30,14)

      REAL CSIZE(3),DATA(35),PAREA(6),L1(4),JOINT(20,21),AXISR,POSN(2),
     &       LINK(30,6)

      CHARACTER*7 JNAM(20),C,LNAM(30)
      CHARACTER*16 RETURN

      DATA RETURN/'1.  RETURN      '/
      DATA LENG/16/
      DATA POSN/0.05,2.2/
      DATA C/'CCCCCCC'/

      DATA DATA/32,0,0,          2,2,1,1,
     &                         1,1,1,1,1,1,
     &                         1,1,1,1,1,1,
     &                         1,1,1,1,1,1,
     &                           1,1,1,1/
C                             FLAG FOR UPDATE
      DATA PRFORM/2/
C                             INCLUSION CLASS LIST
      DATA ACLASS/5/


*********************************************************************
*  REQUEST PICK INPUT
*********************************************************************
C                             SET PICK FILTER(ALL CLASS 3 DETECTABLE)
      CALL GPPKF(IWSID,1,1,ACLASS,0,ACLASS)
C                             MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                             DEFINE PICK AREA
      PAREA(1)=((-.98+1)/2)*CSIZE(1)
      PAREA(2)=((.98+1)/2)*CSIZE(1)
      PAREA(3)=((-.98+1)/2)*CSIZE(2)
      PAREA(4)=((.48+1)/2)*CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=CSIZE(3)
C                             INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                             PLACE PICK DEVICE IN THE EVENT MODE
      CALL GPPKMO(IWSID,1,3,2)
C                             TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)
C                             GET PICK OR RETURN COMMAND
      CALL CMMAND(IWSID,56)

*********************************************************************
*  INCLUDE A RETURN PICK
*********************************************************************

      CALL GPOPST(6)
      CALL GPADCN(1,ACLASS)
      CALL GPPKID(101)
      CALL GPTXCI(2)
      CALL GPAHSC(1.0)
      CALL GPAN2(POSN,LENG,RETURN)
      CALL GPCLST
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
      CALL GPUPWS(IWSID,PRFORM)

*********************************************************************
*  AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
*********************************************************************
C                             AWAIT AN EVENT
  200 CALL GPAWEV(1000.,IWS,ICLA,IDEV)
      IF(ICLA.EQ.5)THEN
C                             GET PICK
         CALL GPGTPK(1,1,PPATH)
C                             IF NO PICK, RETURN
         IF(PPATH(2).EQ.101)GO TO 1000
C                             CHOICE IS SECOND ITEM OF PICK PATH
         CHOICE=PPATH(2)
C                             DEACTIVATE POINT
         LNAM(CHOICE)=C
         DO 230 I=1,13
            LNUM(CHOICE,I)=0
```

```
  230    CONTINUE
C                           REDRAW ALL POINTS
       CALL RDRWLK(IWSID,JNAM,JNUM,JOINT,AXISR,LNAM,LNUM,LINK,SCALE)
C                           AWAIT ANOTHER EVENT
       GO TO 200
      ELSE
C                           IF NOT A PICK, GO TO AWAIT AN EVENT
       GO TO 200
      ENDIF

C                           MAKE SURE PICK IS DEACTIVATED
 1000 CALL GPPKMO(IWSID,1,1,2)
      CALL GPEST(6)

C                           SET PICK FILTER(ALL CLASS 2 UNDETECTABLE)
      CALL GPPKF(IWSID,1,0,ACLASS,1,ACLASS)

      RETURN
      END
```

SUBROUTINE DELP

```
      SUBROUTINE DELP(IWSID,CSIZE,AXISR,PNAM,PNUM,POINT)
*************************************************************************
* ----- SUBROUTINE DELP -----                                           *
*                                                                       *
*                                                                       *
*   THIS SUBROUTINE IS USED TO DELETE A POINT                           *
*                                                                       *
*************************************************************************

      INTEGER IWSID,PRFORM,PPATH(3),ACLASS(1),CHOICE,PNUM(100),LENG

      REAL CSIZE(3),DATA(35),PAREA(6),L1(4),POINT(100,3),AXISR,POSN(2)

      CHARACTER*7 PNAM(100),C
      CHARACTER*16 RETURN

      DATA RETURN/'1.  RETURN      '/
      DATA LENG/16/
      DATA POSN/0.05,2.2/
      DATA C/'CCCCCCC'/

      DATA DATA/32,0,0,         2,2,1,1,
     &                        1,1,1,1,1,1,
     &                        1,1,1,1,1,1,
     &                        1,1,1,1,1,1,
     &                        1,1,1,1,1,1,
     &                          1,1,1,1/
C                           FLAG FOR UPDATE
      DATA PRFORM/2/
C                           INCLUSION CLASS LIST
      DATA ACLASS/2/

*************************************************************************
*  REQUEST PICK INPUT
*************************************************************************
C                           SET PICK FILTER(ALL CLASS 2 DETECTABLE)
      CALL GPPKF(IWSID,1,1,ACLASS,0,ACLASS)
C                           MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                           DEFINE PICK AREA
      PAREA(1)=((-.98+1)/2)*CSIZE(1)
      PAREA(2)=((.98+1)/2)*CSIZE(1)
      PAREA(3)=((-.98+1)/2)*CSIZE(2)
      PAREA(4)=((.48+1)/2)*CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=CSIZE(3)
C                           INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                           PLACE PICK DEVICE IN THE EVENT MODE
      CALL GPPKMO(IWSID,1,3,2)
C                           TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)
C                           GET PICK OR RETURN COMMAND
      CALL CMMAND(IWSID,21)

*************************************************************************
*  INCLUDE A RETURN PICK
*************************************************************************

      CALL GPOPST(6)
      CALL GPADCN(1,ACLASS)
      CALL GPPKID(101)
      CALL GPTXCI(2)
      CALL GPAHSC(1.0)
      CALL GPAN2(POSN,LENG,RETURN)
```

```
      CALL GPCLST
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
      CALL GPUPWS(IWSID,PRFORM)

*********************************************************************
*  AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
*********************************************************************
C                             AWAIT AN EVENT
  200 CALL GPAWEV(1000.,IWS,ICLA,IDEV)
      IF(ICLA.EQ.5)THEN
C                             GET PICK
         CALL GPGTPK(1,1,PPATH)
C                             IF NO PICK, RETURN
         IF(PPATH(2).EQ.101)GO TO 1000
C                             CHOICE IS SECOND ITEM OF PICK PATH
         CHOICE=PPATH(2)
C                             DEACTIVATE POINT
         PNAM(CHOICE)=C
C                             REDRAW ALL POINTS
         CALL RDRWPT(IWSID,PNAM,PNUM,POINT,AXISR)
C                             AWAIT ANOTHER EVENT
         GO TO 200
      ELSE
C                             IF NOT A PICK, GO TO AWAIT AN EVENT
         GO TO 200
      ENDIF


C                             MAKE SURE PICK IS DEACTIVATED
 1000 CALL GPPKMO(IWSID,1,1,2)
      CALL GPEST(6)

C                             SET PICK FILTER(ALL CLASS 2 UNDETECTABLE)
      CALL GPPKF(IWSID,1,0,ACLASS,1,ACLASS)

      RETURN
      END
```

SUBROUTINE DEO

```
      SUBROUTINE DEO(IWSID,AXISR,CSIZE,LOCAT,RFLG)
*********************************************************************
* ----- SUBROUTINE DEO  -----                                       *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE WILL PROMPT FOR THE DIRECT ENTRY OF JOINT       *
*                                                                   *
*   ORIENTATION COORDINATES                                         *
*                                                                   *
*********************************************************************

      INTEGER IWSID,RFLG,CHOICE
      REAL LOCAT(3),AXISR,AREA(6),CSIZE(3),LX(2),LY(2),LZ(2),LX1(2),
     & LY1(2),LZ1(2)
      CHARACTER*7 CLEAR,AXIS,INIT,INIT1
      CHARACTER*20 CLEAR1,X,Y,Z
      CHARACTER*12 STRING,CLEAR2

      DATA CLEAR /'       '/
      DATA INIT  /'0.     '/
      DATA INIT1 /'1.     '/
      DATA STRING/'           '/
      DATA AXIS  /'       '/
      DATA CLEAR1/'                '/
      DATA CLEAR2/'           '/
      DATA X     /'DX=              '/
      DATA Y     /'DY=              '/
      DATA Z     /'DZ=              '/

      AREA(1)=0.08*CSIZE(1)
      AREA(2)=0.90*CSIZE(1)
      AREA(3)=0.70*CSIZE(1)
      AREA(4)=0.90*CSIZE(1)
      AREA(5)=0.02*CSIZE(1)
      AREA(6)=0.90*CSIZE(1)

      LOCAT(1)=0.0
      LOCAT(2)=0.0
      LOCAT(3)=0.0

      LX(1)=0.05
      LX(2)=0.68

      LX1(1)=0.15
      LX1(2)=0.68

      LY(1)=0.05
      LY(2)=0.62
```

```
      LY1(1)=0.15
      LY1(2)=0.62

      LZ(1)=0.05
      LZ(2)=0.56

      LZ1(1)=0.15
      LZ1(2)=0.56

      CALL GPEST(6)

*************************************************************************
*  PROCESS X DIRECTION COSINE
*************************************************************************

      CALL CMMAND(IWSID,27)

    1 CALL GPSTMO(IWSID,1,1,2)
      CALL GPINST(IWSID,1,12,INIT,1,AREA,12,1,0,CLEAR)
      STRING=CLEAR2

      CALL GPRQST(IWSID,1,12,ISTAT,NUM,STRING)
      REWIND 10
      WRITE(10,8)CLEAR1
    8 FORMAT(A20)
      REWIND 10
      WRITE(10,12)STRING
   12 FORMAT(A12)
      REWIND 10
      READ(10,*,ERR=1000)LOCAT(1)

      IF(LOCAT(1).GT.1.0.OR.LOCAT(1).LT.-1.0)THEN
C                              VALUE IS GREATER THAN 1
         CALL CMMAND(IWSID,31)
         GO TO 1
      ENDIF
C                            WRITE X LOCATION TO THE SCREEN
      CALL GPOPST(10)
      CALL GPTXCI(1)
      CALL GPAHSC(1.0)
      CALL GPAN2(LX,20,X)
      CALL GPAN2(LX1,12,STRING)
      CALL GPCLST

      CALL GPEST(16)

      CALL GPOPST(16)
      CALL ICONO(LOCAT,AXISR)
      CALL GPCLST


*************************************************************************
*  PROCESS Y DIRECTION COSINE
*************************************************************************

      CALL CMMAND(IWSID,28)

  100 CALL GPSTMO(IWSID,1,1,2)
      CALL GPINST(IWSID,1,12,INIT,1,AREA,12,1,0,CLEAR)
      STRING=CLEAR2

      CALL GPRQST(IWSID,1,12,ISTAT,NUM,STRING)
      REWIND 10
      WRITE(10,108)CLEAR1
  108 FORMAT(A20)
      REWIND 10
      WRITE(10,112)STRING
  112 FORMAT(A12)
      REWIND 10
      READ(10,*,ERR=1000)LOCAT(2)

      IF(LOCAT(2).GT.1.0.OR.LOCAT(2).LT.-1.0)THEN
C                              VALUE IS GREATER THAN 1
         CALL CMMAND(IWSID,31)
         GO TO 100
      ENDIF

C                            WRITE Y LOCATION TO THE SCREEN
      CALL GPOPST(10)
      CALL GPTXCI(1)
      CALL GPAHSC(1.0)
      CALL GPAN2(LY,20,Y)
      CALL GPAN2(LY1,12,STRING)
      CALL GPCLST

      CALL GPEST(16)

      CALL GPOPST(16)
      CALL ICONO(LOCAT,AXISR)
      CALL GPCLST
```

```
*********************************************************************
*  PROCESS Z DIRECTION COSINE
*********************************************************************

      CALL CMMAND(IWSID,29)

  200 CALL GPSTMO(IWSID,1,1,2)
      CALL GPINST(IWSID,1,12,INIT,1,AREA,12,1,0,CLEAR)
      STRING=CLEAR2

      CALL GPRQST(IWSID,1,12,ISTAT,NUM,STRING)
      REWIND 10
      WRITE(10,208)CLEAR1
  208 FORMAT(A20)
      REWIND 10
      WRITE(10,212)STRING
  212 FORMAT(A12)
      REWIND 10
      READ(10,*,ERR=1000)LOCAT(3)

      IF(LOCAT(3).GT.1.0.OR.LOCAT(3).LT.-1.0)THEN
C                              VALUE IS GREATER THAN 1
         CALL CMMAND(IWSID,31)
         GO TO 200
      ENDIF

C                              WRITE Z LOCATION TO THE SCREEN
      CALL GPOPST(10)
      CALL GPTXCI(1)
      CALL GPAHSC(1.0)
      CALL GPAN2(LZ,20,Z)
      CALL GPAN2(LZ1,12,STRING)
      CALL GPCLST

      CALL GPEST(16)

      CALL GPOPST(16)
      CALL ICONO(LOCAT,AXISR)
      CALL GPCLST


      CALL GPSTMO(IWSID,1,1,1)
      CALL CMMAND(IWSID,32)
      CALL MENU(IWSID,CSIZE,8,CHOICE)

      IF(CHOICE.EQ.1)THEN
C                              ENTER ORIENTATION
         RMAG=((LOCAT(1)**2)+(LOCAT(2)**2)+(LOCAT(3)**2))**.5
         IF(RMAG.EQ.0.0)THEN
C                              VECTOR IS OF UNIT LENGTH
              CALL CMMAND(IWSID,38)
              CALL GPEST(10)
              GO TO 1
         ENDIF
         LOCAT(1)=LOCAT(1)/RMAG
         LOCAT(2)=LOCAT(2)/RMAG
         LOCAT(3)=LOCAT(3)/RMAG
         GO TO 1000
      ENDIF

      IF(CHOICE.EQ.2)THEN
C                              RETURN WITH NO ORIENTATION
         RFLG=1
         GO TO 1000
      ENDIF


 1000 CALL GPEST(10)
      CALL GPEST(16)
C     CALL GPSTMO(IWSID,1,1,2)
      RETURN
      END
```

SUBROUTINE DEP

```
      SUBROUTINE DEP(IWSID,AXISR,CSIZE,LOCAT,RFLG)
*********************************************************************
* ----- SUBROUTINE DEP  -----                                       *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE WILL PROMPT FOR THE DIRECT ENTRY OF LOCATION    *
*                                                                   *
*   COORDINATES                                                     *
*                                                                   *
*********************************************************************

      INTEGER IWSID,RFLG,CHOICE
      REAL LOCAT(3),AXISR,AREA(6),CSIZE(3),LX(2),LY(2),LZ(2),LX1(2),
     & LY1(2),LZ1(2)
```

```
      CHARACTER*7 CLEAR,AXIS,INIT,INIT1
      CHARACTER*20 CLEAR1,X,Y,Z
      CHARACTER*12 STRING,CLEAR2

      DATA CLEAR /'       '/
      DATA INIT  /'0.     '/
      DATA INIT1 /'1.     '/
      DATA STRING/'           '/
      DATA AXIS  /'       '/
      DATA CLEAR1/'                '/
      DATA CLEAR2/'           '/
      DATA X     /'X =             '/
      DATA Y     /'Y =             '/
      DATA Z     /'Z =             '/

      AREA(1)=0.08*CSIZE(1)
      AREA(2)=0.90*CSIZE(1)
      AREA(3)=0.70*CSIZE(1)
      AREA(4)=0.90*CSIZE(1)
      AREA(5)=0.02*CSIZE(1)
      AREA(6)=0.90*CSIZE(1)

      STRING='            '

      LX(1)=0.05
      LX(2)=0.68

      LX1(1)=0.15
      LX1(2)=0.68

      LY(1)=0.05
      LY(2)=0.62

      LY1(1)=0.15
      LY1(2)=0.62

      LZ(1)=0.05
      LZ(2)=0.56

      LZ1(1)=0.15
      LZ1(2)=0.56

      CALL GPEST(6)

*************************************************************************
*  PROCESS X COORDINATE
*************************************************************************

      CALL GPGTST(12,NUM,STRING)
      REWIND 10
      WRITE(10,8)CLEAR1
    8 FORMAT(A20)
      REWIND 10
      WRITE(10,12)STRING
   12 FORMAT(A12)
      REWIND 10
      READ(10,*,ERR=1000)LOCAT(1)
C                          WRITE X LOCATION TO THE SCREEN
      CALL GPOPST(10)
      CALL GPTXCI(1)
      CALL GPAHSC(1.0)
      CALL GPAN2(LX,20,X)
      CALL GPAN2(LX1,12,STRING)
      CALL GPCLST

      CALL GPEST(9)

      CALL GPOPST(9)
      CALL ICON(LOCAT,AXISR)
      CALL GPCLST


   15 IF(ABS(LOCAT(1)).GT.AXISR)THEN
C                          POINT IS OUTSIDE DEFINED AXES
         CALL CMMAND(IWSID,17)
         CALL MENU(IWSID,CSIZE,4,CHOICE)
         CALL GPSTMO(IWSID,1,1,1)
         IF(CHOICE.EQ.1)THEN
C                          REDEFINE THE AXIS RANGE
              AXIS=CLEAR
              CALL CMMAND(IWSID,5)
              CALL GPSTMO(IWSID,1,1,2)
              CALL GPINST(IWSID,1,7,INIT1,1,AREA,7,1,0,CLEAR)
              CALL GPRQST(IWSID,1,7,ISTAT,NUM,AXIS)
              REWIND 10
              WRITE(10,16)CLEAR1
   16         FORMAT(A20)
              REWIND 10
              WRITE(10,20)AXIS
   20         FORMAT(A7)
              REWIND 10
              READ(10,*,ERR=1000)AXISR
              CALL GPEST(7)
              CALL AXES(IWSID,AXISR)
              GO TO 15
```

```
         ENDIF
         IF(CHOICE.EQ.2)THEN
C                              RETURN WITH NO POINT
              RFLG=1
              GO TO 1000
         ENDIF
      ENDIF

*************************************************************************
*  PROCESS Y COORDINATE
*************************************************************************

      CALL GPSTMO(IWSID,1,1,2)
      CALL GPINST(IWSID,1,12,INIT,1,AREA,12,1,0,CLEAR)
      STRING=CLEAR2

      CALL CMMAND(IWSID,18)
      CALL GPRQST(IWSID,1,12,ISTAT,NUM,STRING)
      REWIND 10
      WRITE(10,108)CLEAR1
  108 FORMAT(A20)
      REWIND 10
      WRITE(10,112)STRING
  112 FORMAT(A12)
      REWIND 10
      READ(10,*,ERR=1000)LOCAT(2)
C                              WRITE Y LOCATION TO THE SCREEN
      CALL GPOPST(10)
      CALL GPTXCI(1)
      CALL GPAHSC(1.0)
      CALL GPAN2(LY,20,Y)
      CALL GPAN2(LY1,12,STRING)
      CALL GPCLST

      CALL GPEST(9)

      CALL GPOPST(9)
      CALL ICON(LOCAT,AXISR)
      CALL GPCLST

  115 IF(ABS(LOCAT(2)).GT.AXISR)THEN
C                              POINT IS OUTSIDE DEFINED AXES
         CALL CMMAND(IWSID,17)
         CALL GPSTMO(IWSID,1,1,1)
         CALL MENU(IWSID,CSIZE,4,CHOICE)
         IF(CHOICE.EQ.1)THEN
C                              REDEFINE THE AXIS RANGE
              AXIS=CLEAR
              CALL CMMAND(IWSID,5)
              CALL GPSTMO(IWSID,1,1,2)
              CALL GPINST(IWSID,1,7,INIT1,1,AREA,7,1,0,CLEAR)
              CALL GPRQST(IWSID,1,7,ISTAT,NUM,AXIS)
              REWIND 10
              WRITE(10,116)CLEAR1
  116         FORMAT(A20)
              REWIND 10
              WRITE(10,120)AXIS
  120         FORMAT(A7)
              REWIND 10
              READ(10,*,ERR=1000)AXISR
              CALL GPEST(7)
              CALL AXES(IWSID,AXISR)
              GO TO 115
         ENDIF
         IF(CHOICE.EQ.2)THEN
C                              RETURN WITH NO POINT
              RFLG=1
              GO TO 1000
         ENDIF
      ENDIF

*************************************************************************
*  PROCESS Z COORDINATE
*************************************************************************

      CALL GPSTMO(IWSID,1,1,2)
      CALL GPINST(IWSID,1,12,INIT,1,AREA,12,1,0,CLEAR)
      STRING=CLEAR2

      CALL CMMAND(IWSID,19)
      CALL GPRQST(IWSID,1,12,ISTAT,NUM,STRING)
      REWIND 10
      WRITE(10,208)CLEAR1
  208 FORMAT(A20)
      REWIND 10
      WRITE(10,212)STRING
  212 FORMAT(A12)
      REWIND 10
      READ(10,*,ERR=1000)LOCAT(3)
C                              WRITE Z LOCATION TO THE SCREEN
      CALL GPOPST(10)
      CALL GPTXCI(1)
      CALL GPAHSC(1.0)
      CALL GPAN2(LZ,20,Z)
      CALL GPAN2(LZ1,12,STRING)
```

```
      CALL GPCLST

      CALL GPEST(9)

      CALL GPOPST(9)
      CALL ICON(LOCAT,AXISR)
      CALL GPCLST

  215 IF(ABS(LOCAT(3)).GT.AXISR)THEN
C                              POINT IS OUTSIDE DEFINED AXES
         CALL CMMAND(IWSID,17)
         CALL GPSTMO(IWSID,1,1,1)
         CALL MENU(IWSID,CSIZE,4,CHOICE)
         IF(CHOICE.EQ.1)THEN
C                              REDEFINE THE AXIS RANGE
            AXIS=CLEAR
            CALL CMMAND(IWSID,5)
            CALL GPSTMO(IWSID,1,1,2)
            CALL GPINST(IWSID,1,7,INIT1,1,AREA,7,1,0,CLEAR)
            CALL GPRQST(IWSID,1,7,ISTAT,NUM,AXIS)
            REWIND 10
            WRITE(10,216)CLEAR1
  216       FORMAT(A20)
            REWIND 10
            WRITE(10,220)AXIS
  220       FORMAT(A7)
            REWIND 10
            READ(10,*,ERR=1000)AXISR
            CALL GPEST(7)
            CALL AXES(IWSID,AXISR)
            GO TO 215
         ENDIF
         IF(CHOICE.EQ.2)THEN
C                              RETURN WITH NO POINT
            RFLG=1
            GO TO 1000
         ENDIF
      ENDIF

      CALL GPSTMO(IWSID,1,1,1)
      CALL CMMAND(IWSID,20)
      CALL MENU(IWSID,CSIZE,5,CHOICE)

      IF(CHOICE.EQ.1)THEN
C                              ENTER POINT
         GO TO 1000
      ENDIF

      IF(CHOICE.EQ.2)THEN
C                              RETURN WITH NO POINT
         RFLG=1
         GO TO 1000
      ENDIF


 1000 CALL GPEST(10)
      CALL GPEST(9)
      RETURN
      END
```

SUBROUTINE DRAWJT

```
      SUBROUTINE DRAWJT(IWSID,AXISR,JNAME,NUMJ,JTYPE,JOINT,SC)
*************************************************************************
* ----- SUBROUTINE DRAWJT -----                                         *
*                                                                       *
*                                                                       *
*   THIS SUBROUTINE IS USED TO DRAW A JOINT IN THE MAIN WINDOW          *
*                                                                       *
*************************************************************************

      INTEGER IWSID,NUMJ,CLASS(1),CLASS1(1),JTYPE

      REAL LOCAT(3),LOCAT1(3),ORIENT(3),AXISR,SCALE(3),ORIGIN(3),
     &  MATSC(4,4),MATRIX(4,4),UP(3),Z(3),MAT1(4,4),MAT2(4,4),MAT3(4,4),
     &  MAT4(4,4),MAT5(4,4),Y(3),NEWX(3),NEWYU(3),LZ1(3),NEWY(3),
     &  LZ2(3),Z1(6),Z2(6),JOINT(20,21),ZO(3),LOCA1(3),LOCA2(3),
     &  ORIEN2(3),LOCAT2(3),SC,SC2(3),MSC2(4,4),NEWXU(3)

      CHARACTER*7 JNAME
      CHARACTER*2 Z1N,Z2N

      DATA Z1N/'Z1'/
      DATA Z2N/'Z2'/
      DATA CLASS/3/
      DATA CLASS1/4/
      DATA MAT1 /1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
      DATA MAT2 /1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
      DATA MAT3 /1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
      DATA MAT4 /1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
```

```
      DATA MATSC/1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
      DATA MSC2 /1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
      DATA MATRIX/1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./

      LOCAT(1)=JOINT(NUMJ,1)
      LOCAT(2)=JOINT(NUMJ,2)
      LOCAT(3)=JOINT(NUMJ,3)

      LOCAT2(1)=JOINT(NUMJ,15)
      LOCAT2(2)=JOINT(NUMJ,16)
      LOCAT2(3)=JOINT(NUMJ,17)

      ORIENT(1)=JOINT(NUMJ,4)
      ORIENT(2)=JOINT(NUMJ,5)
      ORIENT(3)=JOINT(NUMJ,6)

      ORIEN2(1)=JOINT(NUMJ,18)
      ORIEN2(2)=JOINT(NUMJ,19)
      ORIEN2(3)=JOINT(NUMJ,20)

      ZO(1)=JOINT(NUMJ,14)*ORIENT(1)
      ZO(2)=JOINT(NUMJ,14)*ORIENT(2)
      ZO(3)=JOINT(NUMJ,14)*ORIENT(3)

      LOCAT1(1)=-.04*AXISR
      LOCAT1(2)=-.14*AXISR
      LOCAT1(3)=0.0

      LOCA1(1)=LOCAT(1)
      LOCA1(2)=LOCAT(2)
      LOCA1(3)=LOCAT(3)

      LOCA2(1)=LOCAT2(1)+(ZO(1)*SC)
      LOCA2(2)=LOCAT2(2)+(ZO(2)*SC)
      LOCA2(3)=LOCAT2(3)+(ZO(3)*SC)

      Z1(1)=0.0
      Z1(2)=0.0
      Z1(3)=0.0
      Z1(4)=(ORIENT(1)*AXISR/10.)*2.4
      Z1(5)=(ORIENT(2)*AXISR/10.)*2.4
      Z1(6)=(ORIENT(3)*AXISR/10.)*2.4

      Z2(1)=(.01*AXISR)
      Z2(2)=(.01*AXISR)
      Z2(3)=0.0
      Z2(4)=(.01*AXISR)+(ORIEN2(1)*AXISR/10.)
      Z2(5)=(.01*AXISR)+(ORIEN2(2)*AXISR/10.)
      Z2(6)=(ORIEN2(3)*AXISR/10.)

      LZ1(1)=(ORIENT(1)*AXISR/10.)*2.4+(AXISR*.02)
      LZ1(2)=(ORIENT(2)*AXISR/10.)*2.4+(AXISR*.02)
      LZ1(3)=(ORIENT(3)*AXISR/10.)*2.4+(AXISR*.08)

      LZ2(1)=(ORIEN2(1)*AXISR/10.)+(AXISR*.02)
      LZ2(2)=(ORIEN2(2)*AXISR/10.)-(AXISR*.02)
      LZ2(3)=(ORIEN2(3)*AXISR/10.)
C                        CALCULATE AXIS TRANSLATION MATRIX
      CALL GPTRL3(LOCA1,MAT1)
      CALL GPTRL3(LOCA2,MAT2)

      SCALE(1)=AXISR*SC
      SCALE(2)=AXISR*SC
      SCALE(3)=AXISR*SC

      SC2(1)=SC
      SC2(2)=SC
      SC2(3)=SC

      ORIGIN(1)=0.0
      ORIGIN(2)=0.0
      ORIGIN(3)=0.0

      UP(1)=0.0
      UP(2)=0.0
      UP(3)=1.0

      Z(1)=0.0
      Z(2)=0.0
      Z(3)=1.0

      Y(1)=0.0
      Y(2)=0.0
      Y(3)=1.0

      IF(ORIENT(1).EQ.0.0.AND.ORIENT(2).EQ.0.0)THEN
         CALL GPTRL3(LOCAT,MAT3)
         GO TO 108
      ENDIF

      CALL VPROD(UP,ORIENT,NEWX)
      CALL VPROD(ORIENT,NEWX,NEWY)

  100 DO 101 I=1,3
         MAT3(I,1)=NEWX(I)
         MAT3(I,2)=NEWY(I)
```

```
         MAT3(I,3)=ORIENT(I)
         MAT3(I,4)=LOCAT(I)
  101 CONTINUE
  108 CALL GPSC3(SCALE,MATSC)
      CALL GPSC3(SC2,MSC2)

*********************************************************************
*    OPEN STRUCTURE 13 AND DRAW JOINT  (MAKE PICKABLE)
*********************************************************************
C                   OPEN STRUCTURE 13
      CALL GPOPST(13)
C                   INSERT CLASS NAME TO ALLOW PICKING JOINTS (CLASS 2)
      CALL GPADCN(1,CLASS)
C                   SET PICK I.D.
      CALL GPPKID(NUMJ)
      CALL GPMLX3(MAT1,3)
      CALL GPMLX3(MSC2,1)
C                   SET CHARACTER HEIGHT SCALE FACTOR
      CALL GPAHSC(.25)
C                   TEXT COLOR INDEX
      CALL GPTXCI(5)
C                   ASSOCIATE TEXT
      CALL GPAN3(LOCAT1,7,JNAME)
C                   CLOSE STRUCTURE
      CALL GPCLST

C                   OPEN STRUCTURE 14 (JOINT)
      CALL GPOPST(14)
C                   INSERT CLASS NAME TO ALLOW PICKING JOINTS (CLASS 2)
      CALL GPADCN(1,CLASS)
C                   SET PICK I.D.
      CALL GPPKID(NUMJ)
      IF(JTYPE.EQ.7)GO TO 110
      CALL GPMLX3(MAT3,3)
      CALL GPMLX3(MATSC,1)
C                   CALL JOINT ICON
  110 CALL JICONS(LOCAT,ORIENT,JTYPE,AXISR,JOINT,NUMJ,SC,MAT3)
C                   CLOSE STRUCTURE
      CALL GPCLST

C                   OPEN STRUCTURE 17 (Z AXIS 1)
      CALL GPOPST(17)
C                   INSERT CLASS NAME TO ALLOW PICKING AXES   (CLASS 4)
      CALL GPADCN(1,CLASS1)
C                   SET PICK I.D.
      CALL GPPKID(NUMJ)
C                   SET TRANSFORMATION
      CALL GPMLX3(MAT1,3)
      CALL GPMLX3(MSC2,1)
C                   SET LINE TYPE SOLID
      CALL GPLT(1)
C                   SET LINE COLOR
      CALL GPPLCI(6)
C                   ASSOCIATE FIRST Z AXIS
      CALL GPPL3(2,3,Z1)
C                   SET CHARACTER HEIGHT SCALE FACTOR
      CALL GPAHSC(.1)
C                   TEXT COLOR INDEX
      CALL GPTXCI(6)
C                   ASSOCIATE TEXT
      CALL GPAN3(LZ1,2,Z1N)
C                   CLOSE STRUCTURE
      CALL GPCLST

      IF(JTYPE.EQ.8)THEN
C                   GROUND LINK (DRAW JUST 1 AXIS)
        GO TO 1000
      ENDIF

C                   OPEN STRUCTURE 18 (Z AXIS 2)
      CALL GPOPST(18)
C                   INSERT CLASS NAME TO ALLOW PICKING AXES   (CLASS 4)
      CALL GPADCN(1,CLASS1)
C                   SET PICK I.D.
      CALL GPPKID(NUMJ)
C                   SET TRANSFORMATION
      CALL GPMLX3(MAT2,3)
      CALL GPMLX3(MSC2,1)
C                   SET LINE TYPE SOLID
      CALL GPLT(1)
C                   SET LINE COLOR
      CALL GPPLCI(2)
C                   ASSOCIATE FIRST Z AXIS
      CALL GPPL3(2,3,Z2)
C                   SET CHARACTER HEIGHT SCALE FACTOR
      CALL GPAHSC(.1)
C                   TEXT COLOR INDEX
      CALL GPTXCI(2)
C                   ASSOCIATE TEXT
      CALL GPAN3(LZ2,2,Z2N)
C                   CLOSE STRUCTURE
      CALL GPCLST

 1000 CALL GPUPWS(IWSID,2)

      RETURN
```

```
      END
```

SUBROUTINE DRAWLK

```
      SUBROUTINE DRAWLK(IWSID,LNAM,LNUM,LINK,JOINT,JNUM1,AXISR,NUML,
     &SCALE)
*************************************************************************
* ----- SUBROUTINE DRAWLK -----                                         *
*                                                                       *
*                                                                       *
*   THIS SUBROUTINE IS USED TO DRAW A LINK IN THE MAIN WINDOW           *
*                                                                       *
*************************************************************************

      INTEGER IWSID,NUML,CLASS(1),LNUM(30,14),JNUM(6),IAXIS(6),
     &JNUM1(20,3)

      REAL LINK(30,6),JOINT(20,21),XVEC(6),X1(6),X2(6),LNK(6),
     &SC(3),SCM(4,4),TR(3),TRM(4,4),TRM1(6,4,4),P(3),P1(3),P2(3),S(4,4),
     &T(4,4),XX1(6,6),TRMM(4,4)

      CHARACTER*7 LNAM(30)

      DATA CLASS/5/
      DATA SCM/1.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
      DATA TRM/1.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
      DATA TRMM/1.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
      DATA S  /1.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
      DATA T  /1.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./


      JNUM(1)=LNUM(NUML,2)
      JNUM(2)=LNUM(NUML,4)
      JNUM(3)=LNUM(NUML,6)
      JNUM(4)=LNUM(NUML,8)
      JNUM(5)=LNUM(NUML,10)
      JNUM(6)=LNUM(NUML,12)
      IAXIS(1)=LNUM(NUML,3)
      IAXIS(2)=LNUM(NUML,5)
      IAXIS(3)=LNUM(NUML,7)
      IAXIS(4)=LNUM(NUML,9)
      IAXIS(5)=LNUM(NUML,11)
      IAXIS(6)=LNUM(NUML,13)

      SC(1)=SCALE
      SC(2)=SCALE
      SC(3)=SCALE

      CALL GPSC3(SC,SCM)

*************************************************************************
*    DRAW THE X AXIS
*************************************************************************

      DO 50 I=1,6
         IF(JNUM(I).EQ.0)THEN
            GO TO 100
         ENDIF
         IF(IAXIS(I).EQ.1)THEN
            XVEC(1)=JOINT(JNUM(I),7)
            XVEC(2)=JOINT(JNUM(I),8)
            XVEC(3)=JOINT(JNUM(I),9)
         ENDIF
         IF(IAXIS(I).EQ.2)THEN
            XVEC(1)=JOINT(JNUM(I),10)
            XVEC(2)=JOINT(JNUM(I),11)
            XVEC(3)=JOINT(JNUM(I),12)
         ENDIF

         IF(IAXIS(I).EQ.1)THEN

            TR(1)=JOINT(JNUM(I),1)
            TR(2)=JOINT(JNUM(I),2)
            TR(3)=JOINT(JNUM(I),3)

            CALL GPTRL3(TR,TRM)

            DO 61 J=1,4
               DO 60 K=1,4
                  TRM1(I,J,K)=TRM(J,K)
60             CONTINUE
61          CONTINUE


            X1(1)=0.
            X1(2)=0.
            X1(3)=0.
            X1(4)=(XVEC(1)*AXISR/10.)*2.4
            X1(5)=(XVEC(2)*AXISR/10.)*2.4
```

```
            X1(6)=(XVEC(3)*AXISR/10.)*2.4

            XX1(I,1)=0.
            XX1(I,2)=0.
            XX1(I,3)=0.
            XX1(I,4)=(XVEC(1)*AXISR/10.)*2.4
            XX1(I,5)=(XVEC(2)*AXISR/10.)*2.4
            XX1(I,6)=(XVEC(3)*AXISR/10.)*2.4

            CALL GPOPST(19)
            CALL GPADCN(1,CLASS)
            CALL GPPKID(NUML)
            CALL GPMLX3(TRM,3)
            CALL GPMLX3(SCM,1)
            CALL GPPLCI(6)
            CALL GPLT(1)
            CALL GPPL3(2,3,X1)
            CALL GPCLST
         ENDIF

         IF(IAXIS(I).EQ.2)THEN
            TR(1)=JOINT(JNUM(I),15)
            TR(2)=JOINT(JNUM(I),16)
            TR(3)=JOINT(JNUM(I),17)

            CALL GPTRL3(TR,TRM)

            DO 63 J=1,4
               DO 62 K=1,4
                  TRM1(I,J,K)=TRM(J,K)
   62          CONTINUE
   63       CONTINUE


            XX1(I,1)=(.01*AXISR)+
     &     (JOINT(JNUM(I),14)*JOINT(JNUM(I),4))
            XX1(I,2)=(.01*AXISR)+
     &     (JOINT(JNUM(I),14)*JOINT(JNUM(I),5))
            XX1(I,3)=
     &     (JOINT(JNUM(I),14)*JOINT(JNUM(I),6))
            XX1(I,4)=(XVEC(1)*AXISR/10.)+(.01*AXISR)+
     &     (JOINT(JNUM(I),14)*JOINT(JNUM(I),4))
            XX1(I,5)=(XVEC(2)*AXISR/10.)+(.01*AXISR)+
     &     (JOINT(JNUM(I),14)*JOINT(JNUM(I),5))
            XX1(I,6)=(XVEC(3)*AXISR/10.)+
     &     (JOINT(JNUM(I),14)*JOINT(JNUM(I),6))


            X1(1)=(.01*AXISR)+
     &     (JOINT(JNUM(I),14)*JOINT(JNUM(I),4))
            X1(2)=(.01*AXISR)+
     &     (JOINT(JNUM(I),14)*JOINT(JNUM(I),5))
            X1(3)=
     &     (JOINT(JNUM(I),14)*JOINT(JNUM(I),6))
            X1(4)=(XVEC(1)*AXISR/10.)+(.01*AXISR)+
     &     (JOINT(JNUM(I),14)*JOINT(JNUM(I),4))
            X1(5)=(XVEC(2)*AXISR/10.)+(.01*AXISR)+
     &     (JOINT(JNUM(I),14)*JOINT(JNUM(I),5))
            X1(6)=(XVEC(3)*AXISR/10.)+
     &     (JOINT(JNUM(I),14)*JOINT(JNUM(I),6))

            CALL GPOPST(19)
            CALL GPADCN(1,CLASS)
            CALL GPPKID(NUML)
            CALL GPMLX3(TRM,3)
            CALL GPMLX3(SCM,1)
            CALL GPPLCI(2)
            CALL GPLT(1)
            CALL GPPL3(2,3,X1)
            CALL GPCLST
         ENDIF

   50 CONTINUE

*****************************************************************
*    DRAW THE LINK
*****************************************************************

  100 IF(LNUM(NUML,14).EQ.1)GO TO 1000

      P(1)=XX1(1,4)*SCALE
      P(2)=XX1(1,5)*SCALE
      P(3)=XX1(1,6)*SCALE

      DO 103 J=1,4
         DO 102 K=1,4
            TRMM(J,K)=TRM1(1,J,K)
  102    CONTINUE
  103 CONTINUE


      CALL GPXF3(P,TRMM,P1)

      P(1)=XX1(2,4)*SCALE
```

```
      P(2)=XX1(2,5)*SCALE
      P(3)=XX1(2,6)*SCALE

      DO 105 J=1,4
         DO 104 K=1,4
            TRM(J,K)=TRM1(2,J,K)
  104    CONTINUE
  105 CONTINUE

      CALL GPXF3(P,TRM,P2)

      LNK(1)=P1(1)
      LNK(2)=P1(2)
      LNK(3)=P1(3)
      LNK(4)=P2(1)
      LNK(5)=P2(2)
      LNK(6)=P2(3)

      IF(JNUM1(JNUM(1),2).EQ.8)THEN

         P(1)=0.
         P(2)=0.
         P(3)=0.

         CALL GPXF3(P,TRMM,P1)

         LNK(1)=P1(1)
         LNK(2)=P1(2)
         LNK(3)=P1(3)
      ENDIF

      IF(JNUM1(JNUM(2),2).EQ.8)THEN

         P(1)=0.
         P(2)=0.
         P(3)=0.

         CALL GPXF3(P,TRM,P2)

         LNK(4)=P2(1)
         LNK(5)=P2(2)
         LNK(6)=P2(3)
      ENDIF

      CALL GPOPST(19)
      CALL GPADCN(1,CLASS)
      CALL GPPKID(NUML)
      CALL GPMLX3(T,3)
      CALL GPMLX3(S,1)
      CALL GPPLCI(7)
      CALL GPLT(2)
      CALL GPPL3(2,3,LNK)
      CALL GPCLST

      CALL GPUPWS(IWSID,2)

*********************************************************************
*     DRAW THE COUPLING LINKS
*********************************************************************

      DO 150 I=3,6
         IF(JNUM(I).EQ.0)THEN
            GO TO 1000
         ENDIF

         P(1)=XX1(I,4)*SCALE
         P(2)=XX1(I,5)*SCALE
         P(3)=XX1(I,6)*SCALE

         DO 113 J=1,4
            DO 112 K=1,4
               TRMM(J,K)=TRM1(I,J,K)
  112       CONTINUE
  113    CONTINUE

         CALL GPXF3(P,TRMM,P1)

         P2(1)=JOINT(JNUM(1),1)+(JOINT(JNUM(2),1)-JOINT(JNUM(1),1))/2.
         P2(2)=JOINT(JNUM(1),2)+(JOINT(JNUM(2),2)-JOINT(JNUM(1),2))/2.
         P2(3)=JOINT(JNUM(1),3)+(JOINT(JNUM(2),3)-JOINT(JNUM(1),3))/2.


         LNK(1)=P1(1)
         LNK(2)=P1(2)
         LNK(3)=P1(3)
         LNK(4)=P2(1)
         LNK(5)=P2(2)
         LNK(6)=P2(3)


         CALL GPOPST(19)
         CALL GPADCN(1,CLASS)
         CALL GPPKID(NUML)
         CALL GPMLX3(T,3)
```

```
         CALL GPMLX3(S,1)
         CALL GPPLCI(7)
         CALL GPLT(2)
         CALL GPPL3(2,3,LNK)
         CALL GPCLST

         CALL GPUPWS(IWSID,2)

  150 CONTINUE

 1000 RETURN
      END
```

SUBROUTINE DRAWPT

```
      SUBROUTINE DRAWPT(IWSID,LOCAT,AXISR,PNAME,NUMP)
*********************************************************************
* ----- SUBROUTINE DRAWPT -----                                     *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED TO DRAW A POINT IN THE MAIN WINDOW      *
*                                                                   *
*********************************************************************

      INTEGER IWSID,NUMP,CLASS(1)

      REAL LOCAT(3),LOCAT1(3)

      CHARACTER*7 PNAME

      DATA CLASS/2/

      LOCAT1(1)=((LOCAT(1)/AXISR)+.04)*AXISR
      LOCAT1(2)=((LOCAT(2)/AXISR)-.01)*AXISR
      LOCAT1(3)=LOCAT(3)

*********************************************************************
*    OPEN STRUCTURE 12 AND DRAW POINT  (MAKE PICKABLE)              *
*********************************************************************
C                 OPEN STRUCTURE 12
      CALL GPOPST(12)
C                 INSERT CLASS NAME TO ALLOW PICKING POINTS (CLASS 2)
      CALL GPADCN(1,CLASS)
C                 SET PICK I.D.
      CALL GPPKID(NUMP)
C                 SET POLYMARKER TYPE
      CALL GPMT(1)
C                 SET POLYMARKER COLOR
      CALL GPPMCI(8)
C                 ASSOCIATE POLYMARKER
      CALL GPPM3(IWSID,3,LOCAT)
C                 SET CHARACTER HEIGHT SCALE FACTOR
      CALL GPAHSC(.25)
C                 TEXT COLOR INDEX
      CALL GPTXCI(8)
C                 ASSOCIATE TEXT
      CALL GPAN3(LOCAT1,7,PNAME)
C                 CLOSE STRUCTURE
      CALL GPCLST

      CALL GPUPWS(IWSID,2)

      RETURN
      END
```

SUBROUTINE ENTER

```
      SUBROUTINE ENTER(IWSID,CHOICE,CSIZE)
*********************************************************************
* ----- SUBROUTINE ENTER -----                                      *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED TO DETERMINE MENU CHOICE                *
*                                                                   *
*********************************************************************

      INTEGER IWSID,LENG,PRFORM,PPATH(3),ACLASS(1),CHOICE

      REAL POSN(2),CSIZE(3),PAREA(6),DATA(35)
```

```
      CHARACTER*35 TMENU(10)

C                               OPTIONS IN DATA SUBROUTINE
      DATA TMENU(1)/'1. ENTER POSITION  '/
      DATA TMENU(6)/'                   '/
      DATA TMENU(2)/'2. RETURN          '/
      DATA TMENU(7)/'                   '/
      DATA TMENU(3)/'                   '/
      DATA TMENU(8)/'                   '/
      DATA TMENU(4)/'                   '/
      DATA TMENU(9)/'                   '/
      DATA TMENU(5)/'                   '/
      DATA TMENU(10)/'                   '/
      DATA LENG/19/
      DATA DATA/32,0,0,          2,2,1,1,
     &                          1,1,1,1,1,1,1,
     &                          1,1,1,1,1,1,1,
     &                          1,1,1,1,1,1,1,
     &                          1,1,1,1/
C                               FLAG FOR UPDATE
      DATA PRFORM/2/
C                               INCLUSION CLASS LIST
      DATA ACLASS/1/

      POSN(1)=0.05
      POSN(2)=2.2

*********************************************************************
*  OPEN STRUCTURE FOR THE MENU OF ENTER POINT OR RETURN
*********************************************************************

C                               OPEN A STRUCTURE
      CALL GPOPST(6)
C                               INSERT A TEST POLYLINE


C                               INSERT CLASS NAME TO ALLOW PICKING MENU ITEM
      CALL GPADCN(1,ACLASS)
C                               SET TEXT COLOR TO YELLOW
      CALL GPTXCI(2)
C                               SET ANNOTATION SIZE SCALE FACTOR
      CALL GPAHSC(1.00)

      DO 100 I=1,5
C                               SET PICK IDENTIFIER
         CALL GPPKID(I)
C                               DRAW TEXT
         CALL GPAN2(POSN,LENG,TMENU(I))
C                               SET NEXT POSITION
         POSN(2)=POSN(2)-.10
C                               SET PICK IDENTIFIER
         CALL GPPKID(I)
C                               DRAW TEXT
         CALL GPAN2(POSN,LENG,TMENU(I+5))
C                               SET NEXT POSITION
         POSN(2)=POSN(2)-.20
  100 CONTINUE
C                               CLOSE STRUCTURE
      CALL GPCLST

*********************************************************************
*  SCREEN DISPLAY
*********************************************************************
C                               ASSOCIATE THE STRUCTURE 6 WITH A VIEW 3
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
C                               TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)

*********************************************************************
*  REQUEST PICK INPUT
*********************************************************************
C                               SET PICK FILTER(ALL CLASS 1 DETECTABLE)
      CALL GPPKF(IWSID,1,1,ACLASS,0,ACLASS)
C                               MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                               DEFINE PICK AREA
      PAREA(1)=0.0
      PAREA(2)=0.2
      PAREA(3)=0.0
      PAREA(4)=0.2
      PAREA(5)=0.0
      PAREA(6)=0.2
C                               INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                               PLACE PICK DEVICE IN THE EVENT MODE
      CALL GPPKMO(IWSID,1,3,2)
C                               TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)
```

```
      RETURN
      END
```

SUBROUTINE ENTERO

```
      SUBROUTINE ENTERO(IWSID,CHOICE,CSIZE)
*********************************************************************
* ----- SUBROUTINE ENTER -----                                      *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED TO DETERMINE MENU CHOICE                *
*                                                                   *
*********************************************************************
      INTEGER IWSID,LENG,PRFORM,PPATH(3),ACLASS(1),CHOICE

      REAL POSN(2),CSIZE(3),PAREA(6),DATA(35)
      CHARACTER*35 TMENU(10)

C                              OPTIONS IN DATA SUBROUTINE
      DATA TMENU(1)/'1. ENTER                '/
      DATA TMENU(6)/'   ORIENTATION          '/
      DATA TMENU(2)/'2. RETURN               '/
      DATA TMENU(7)/'                        '/
      DATA TMENU(3)/'                        '/
      DATA TMENU(8)/'                        '/
      DATA TMENU(4)/'                        '/
      DATA TMENU(9)/'                        '/
      DATA TMENU(5)/'                        '/
      DATA TMENU(10)/'                       '/
      DATA LENG/19/
      DATA DATA/32,0,0,          2,2,1,1,
     &                          1,1,1,1,1,1,1,
     &                          1,1,1,1,1,1,1,
     &                          1,1,1,1,1,1,1,
     &                            1,1,1,1/
C                              FLAG FOR UPDATE
      DATA PRFORM/2/
C                              INCLUSION CLASS LIST
      DATA ACLASS/1/

      POSN(1)=0.05
      POSN(2)=2.2

*********************************************************************
*  OPEN STRUCTURE FOR THE MENU OF ENTER POINT OR RETURN
*********************************************************************
C                              OPEN A STRUCTURE
      CALL GPOPST(6)
C                              INSERT CLASS NAME TO ALLOW PICKING MENU ITEM
      CALL GPADCN(1,ACLASS)
C                              SET TEXT COLOR TO YELLOW
      CALL GPTXCI(2)
C                              SET ANNOTATION SIZE SCALE FACTOR
      CALL GPAHSC(1.00)

      DO 100 I=1,5
C                              SET PICK IDENTIFIER
         CALL GPPKID(I)
C                              DRAW TEXT
         CALL GPAN2(POSN,LENG,TMENU(I))
C                              SET NEXT POSITION
         POSN(2)=POSN(2)-.10
C                              SET PICK IDENTIFIER
         CALL GPPKID(I)
C                              DRAW TEXT
         CALL GPAN2(POSN,LENG,TMENU(I+5))
C                              SET NEXT POSITION
         POSN(2)=POSN(2)-.20
  100 CONTINUE
C                              CLOSE STRUCTURE
      CALL GPCLST

*********************************************************************
*  SCREEN DISPLAY
*********************************************************************
C                              ASSOCIATE THE STRUCTURE 6 WITH A VIEW 3
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
C                              TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)
```

```
*********************************************************************
*  REQUEST PICK INPUT
*********************************************************************
C                               SET PICK FILTER(ALL CLASS 1 DETECTABLE)
      CALL GPPKF(IWSID,1,1,ACLASS,0,ACLASS)
C                               MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                               DEFINE PICK AREA
      PAREA(1)=0.0
      PAREA(2)=0.2
      PAREA(3)=0.0
      PAREA(4)=0.2
      PAREA(5)=0.0
      PAREA(6)=0.2
C                               INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                               PLACE PICK DEVICE IN THE EVENT MODE
      CALL GPPKMO(IWSID,1,3,2)
C                               TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)

      RETURN
      END
```

SUBROUTINE FILENM

```
      SUBROUTINE FILENM(IWSID,FILE)
*********************************************************************
* ----- SUBROUTINE FILENM -----                                     *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED TO DISPLAY FILENAME ON THE SCREEN       *
*                                                                   *
*********************************************************************
      INTEGER IWSID,LENGT,LENGF,PRFORM

      REAL POSF(2),BOX1(8),SHD1(8),POST(2)
      CHARACTER*7 FILE
      CHARACTER*16 FILET

C                               FILE NAME LENGTH AND POSITION
      DATA LENGF/7/
      DATA POSF/.79,.94/
C                               FILE IDENT. LENGTH AND POSITION
      DATA LENGT/16/
      DATA POST/.60,.94/
C                               FLAG FOR UPDATE
      DATA PRFORM/2/
C                               FILE TITLE IDENTIFIER
      DATA FILET/'FILE :     '/
C                               GREY FILE BOX
      DATA BOX1/0.57,0.92,0.97,0.92,0.97,0.98,0.57,0.98/

C                               CALCULATION FOR THE TWO SHADOW BOXES
      DO 100 I=1,7,2
         SHD1(I)=BOX1(I)+0.025
  100 CONTINUE

      DO 200 I=2,8,2
         SHD1(I)=BOX1(I)-0.025
  200 CONTINUE

*********************************************************************
*  OPEN STRUCTURE FOR COMMAND BOX (ALSO PUT IN STRUCTURE 3)
*********************************************************************
C                               OPEN A STRUCTURE
      CALL GPOPST(3)
C                               SET SOLID INTEROR STYLE FOR GREY BOX
      CALL GPIS(2)
C                               SET COLOR FOR INTERIOR OF SHADOW BOX
      CALL GPICI(4)
C                               DRAW GREY SHADOW BOX
      CALL GPPG2(1,4,2,SHD1)
C                               SET COLOR FOR INTERIOR OF GREY BOX
      CALL GPICI(3)
C                               DRAW GREY COMMAND BOX
      CALL GPPG2(1,4,2,BOX1)
C                               SET COLOR FOR WHITE OF FILE NAME
      CALL GPICI(1)
C                               SET TEXT COLOR TO WHITE
      CALL GPTXCI(1)
C                               SET ANNOTATION SIZE SCALE FACTOR
      CALL GPAHSC(1.00)
C                               DRAW TITLE TEXT
```

```
      CALL GPAN2(POST,LENGT,FILET)
      CALL GPAN2(POSF,LENGF,FILE)
C                              CLOSE STRUCTURE
      CALL GPCLST

*********************************************************************
*  SCREEN DISPLAY
*********************************************************************
C                              UPDATE THE WORKSTATION
      CALL GPUPWS(IWSID,PRFORM)

      RETURN
      END
```

SUBROUTINE FILEXS

```
                                                                      FIL00010
                                                                      FIL00020
C --- F I L E X S                                                     FIL00030
C                                                                     FIL00040
C     CHECK FOR FILE EXISTENCE  (AFILE = FILE TO BE CHECKED)          FIL00050
C                               (IRET = RETURN FLAG
C                                       0 -- FILE EXISTS
C                                       1 -- FILE DOES NOT EXIST
C                                                                     FIL00060
                                                                      FIL00070
      SUBROUTINE    FILEXS (AFILE,IRET)                             FIL00080
                                                                      FIL00090
      CHARACTER*7   AFILE                                           FIL00100
      CHARACTER*20  SYS                                             FIL00110
                                                                      FIL00120
      DATA          SYS / 'STATE FILE         A' /                  FIL00130
                                                                      FIL00140
        DO 20 I = 1,7                                               FIL00150
          SYS(I+11:I+11) = AFILE(I:I)                               FIL00160
   20   CONTINUE                                                     FIL00170
*********************************************************************
                                                                      FIL00180
        CALL SYSCAL (SYS,20,IRET)                                     FIL00190

C
C     SYSTEM DEPENDENT ROUTINE TO CHECK FILE EXISTENCE.
C
C         IRET = 0   --------  FILE EXISTS
C         IRET = 1   --------  FILE DOES NOT EXIST
C
*********************************************************************
        IF (IRET.EQ.0) THEN                                           FIL00200
          IRET = 0                                                    FIL00210
        ELSE                                                          FIL00220
          IRET = 1                                                    FIL00230
        ENDIF                                                         FIL00240
                                                                      FIL00250
                                                                      FIL00260
  200   RETURN                                                        FIL00270
        END                                                           FIL00280
```

SUBROUTINE ICICON

```
      SUBROUTINE ICICON(IWSID,JOINT,INAM,INUM,IC,SCALE)
*********************************************************************
* ----- SUBROUTINE ICICON  -----                                    *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED DISPLAY THE ICONS FOR THE IC            *
*                                                                   *
*********************************************************************

      INTEGER INUM(20,2),IWSID,CLASS(1)

      REAL IC(20,4),JOINT(20,21),P(3)

      CHARACTER*7 INAM(20)
      CHARACTER*2 TEXT

      DATA TEXT/'IC'/
      DATA CLASS/6/

      CALL GPEST(20)
```

```
      CALL GPUPWS(IWSID,2)

      DO 10 III=1,20
         IF(INAM(III).EQ.'BBBBBBB')THEN
C                          CONTINUE
         ENDIF

         IF(INAM(III).EQ.'CCCCCCC')THEN
C                          CONTINUE
         ENDIF

         IF(INAM(III).EQ.'DDDDDDD')THEN
C                          DRAW ICON
            P(1)=JOINT(INUM(III,1),1)-(.60*SCALE)
            P(2)=JOINT(INUM(III,1),2)
            P(3)=JOINT(INUM(III,1),3)
            CALL GPOPST(20)
            CALL GPADCN(1,CLASS)
            CALL GPPKID(III)
            CALL GPAHSC(.25)
            CALL GPTXCI(7)
            CALL GPAN3(P,2,TEXT)
            CALL GPCLST

            CALL GPUPWS(IWSID,2)

         ENDIF

   10 CONTINUE

 1000 RETURN
      END
```

SUBROUTINE ICON

```
      SUBROUTINE ICON(LOCAT,AXISR)
*************************************************************************
* ----- SUBROUTINE ICON  -----                                          *
*                                                                       *
*                                                                       *
*   THIS SUBROUTINE IS USED DISPLAY THE POSITION ICON                   *
*                                                                       *
*************************************************************************

      INTEGER TYPE

      REAL LOCAT(3),AXISR,LINEX(6),LINEY(6),LINEZ(6),XBOX(15),
     & YBOX(15),ZBOX(15),LX(6),LY(6),LZ(6)

      DATA LX/0.,0.,0.,0.,0.,0./
      DATA LY/0.,0.,0.,0.,0.,0./
      DATA LZ/0.,0.,0.,0.,0.,0./


      LX(4)=LOCAT(1)
      LY(5)=LOCAT(2)
      LZ(6)=LOCAT(3)

      LINEX(1)=LOCAT(1)
      LINEX(2)=LOCAT(2)
      LINEX(3)=LOCAT(3)
      LINEX(4)=0.
      LINEX(5)=LOCAT(2)
      LINEX(6)=LOCAT(3)

      LINEY(1)=LOCAT(1)
      LINEY(2)=LOCAT(2)
      LINEY(3)=LOCAT(3)
      LINEY(4)=LOCAT(1)
      LINEY(5)=0.
      LINEY(6)=LOCAT(3)

      LINEZ(1)=LOCAT(1)
      LINEZ(2)=LOCAT(2)
      LINEZ(3)=LOCAT(3)
      LINEZ(4)=LOCAT(1)
      LINEZ(5)=LOCAT(2)
      LINEZ(6)=0.

      XBOX(1)=0.00
      XBOX(2)=LOCAT(2)+(.06*AXISR)
      XBOX(3)=LOCAT(3)+(.06*AXISR)
      XBOX(4)=0.00
      XBOX(5)=LOCAT(2)-(.06*AXISR)
      XBOX(6)=LOCAT(3)+(.06*AXISR)
      XBOX(7)=0.00
      XBOX(8)=LOCAT(2)-(.06*AXISR)
      XBOX(9)=LOCAT(3)-(.06*AXISR)
```

```
      XBOX(10)=0.00
      XBOX(11)=LOCAT(2)+(.06*AXISR)
      XBOX(12)=LOCAT(3)-(.06*AXISR)
      XBOX(13)=0.00
      XBOX(14)=LOCAT(2)+(.06*AXISR)
      XBOX(15)=LOCAT(3)+(.06*AXISR)

      YBOX(1)=LOCAT(1)+(.06*AXISR)
      YBOX(2)=0.00
      YBOX(3)=LOCAT(3)+(.06*AXISR)
      YBOX(4)=LOCAT(1)-(.06*AXISR)
      YBOX(5)=0.00
      YBOX(6)=LOCAT(3)+(.06*AXISR)
      YBOX(7)=LOCAT(1)-(.06*AXISR)
      YBOX(8)=0.00
      YBOX(9)=LOCAT(3)-(.06*AXISR)
      YBOX(10)=LOCAT(1)+(.06*AXISR)
      YBOX(11)=0.00
      YBOX(12)=LOCAT(3)-(.06*AXISR)
      YBOX(13)=LOCAT(1)+(.06*AXISR)
      YBOX(14)=0.00
      YBOX(15)=LOCAT(3)+(.06*AXISR)

      ZBOX(1)=LOCAT(1)+(.06*AXISR)
      ZBOX(2)=LOCAT(2)+(.06*AXISR)
      ZBOX(3)=0.00
      ZBOX(4)=LOCAT(1)-(.06*AXISR)
      ZBOX(5)=LOCAT(2)+(.06*AXISR)
      ZBOX(6)=0.00
      ZBOX(7)=LOCAT(1)-(.06*AXISR)
      ZBOX(8)=LOCAT(2)-(.06*AXISR)
      ZBOX(9)=0.00
      ZBOX(10)=LOCAT(1)+(.06*AXISR)
      ZBOX(11)=LOCAT(2)-(.06*AXISR)
      ZBOX(12)=0.00
      ZBOX(13)=LOCAT(1)+(.06*AXISR)
      ZBOX(14)=LOCAT(2)+(.06*AXISR)
      ZBOX(15)=0.00

C                              SET POLYLINE COLOR
      CALL GPPLCI(1)
C                              SET LINE TYPE DASHED
      CALL GPLT(2)
C                              DRAW POLYLINE
      CALL GPPL3(2,3,LINEX)
      CALL GPPL3(2,3,LINEY)
      CALL GPPL3(2,3,LINEZ)
C                              SET LINE TYPE SOLID
      CALL GPLT(1)
      CALL GPPL3(5,3,XBOX)
      CALL GPPL3(5,3,YBOX)
      CALL GPPL3(5,3,ZBOX)

C                              SET POLYLINE COLOR
      CALL GPPLCI(5)
      CALL GPPL3(2,3,LX)
      CALL GPPL3(2,3,LY)
      CALL GPPL3(2,3,LZ)

      RETURN
      END
```

SUBROUTINE ICOND

```
      SUBROUTINE ICOND(IWSID,AXISR,CSIZE,JNAM,JNUM,JOINT,INAM,INUM,IC,
     &SCALE)
*************************************************************************
* ----- SUBROUTINE ICOND -----                                          *
*                                                                       *
*                                                                       *
*   THIS SUBROUTINE IS USED TO SET, DELETE OR INQUIRE I.C. ON           *
*                                                                       *
*   REVOLUTE OR PRISMATIC JOINTS                                        *
*                                                                       *
*************************************************************************
      INTEGER IWSID,CHOICE,JNUM(20,3),INUM(20,2)

      REAL CSIZE(3),AXISR,JOINT(20,21),IC(20,4)

      CHARACTER*7 JNAM(20),INAM(20)

    5 CALL CMMAND(IWSID,10)
      CALL MENU(IWSID,CSIZE,14,CHOICE)

      IF(CHOICE.EQ.1)THEN
         CALL ADDIC(IWSID,CSIZE,AXISR,JNAM,JNUM,JOINT,INAM,INUM,IC,
```

```
     &SCALE)
        GO TO 5
      ENDIF

      IF(CHOICE.EQ.2)THEN
         CALL DELIC(IWSID,CSIZE,AXISR,INAM,INUM,IC,JOINT,SCALE)
         GO TO 5
      ENDIF

      IF(CHOICE.EQ.3)THEN
         CALL INQIC(IWSID,CSIZE,AXISR,INAM,INUM,IC)
         GO TO 5

      ENDIF
      IF(CHOICE.EQ.4)THEN
         CONTINUE
      ENDIF

      RETURN
      END
```

SUBROUTINE ICONO

```
      SUBROUTINE ICONO(LOCAT,AXISR)
*************************************************************************
* ----- SUBROUTINE ICONO -----                                          *
*                                                                       *
*                                                                       *
*   THIS SUBROUTINE IS USED TO DISPLAY THE ORIENTATION ICON             *
*                                                                       *
*************************************************************************

      INTEGER TYPE

      REAL LOCAT(3),AXISR,LINEX(6),LINEY(6),LINEZ(6),XBOX(15),
     & YBOX(15),ZBOX(15),LX(6),LY(6),LZ(6),OR(6)

      DATA LX/0.,0.,0.,0.,0.,0./
      DATA LY/0.,0.,0.,0.,0.,0./
      DATA LZ/0.,0.,0.,0.,0.,0./

      DATA OR/0.,0.,0.,0.,0.,0./

      LX(4)=LOCAT(1)
      LY(5)=LOCAT(2)
      LZ(6)=LOCAT(3)

      OR(4)=LOCAT(1)
      OR(5)=LOCAT(2)
      OR(6)=LOCAT(3)

C                          SET POLYLINE COLOR
      CALL GPPLCI(1)
C                          SET LINE TYPE SOLID
      CALL GPLT(1)
C                          SET POLYLINE COLOR
      CALL GPPLCI(2)
      CALL GPPL3(2,3,OR)

      CALL GPPLCI(5)
      CALL GPPL3(2,3,LX)
      CALL GPPL3(2,3,LY)
      CALL GPPL3(2,3,LZ)


      RETURN
      END
```

SUBROUTINE IMP

```
      SUBROUTINE IMP(PNAM,PNUM,POINT,JNAM,JNUM,JOINT,LNAM,LNUM,LINK,
     &CSIZE,INAM,INUM,IC,IWSID)
*************************************************************************
* ----- SUBROUTINE IMP -----                                            *
*                                                                       *
*                                                                       *
*   THIS SUBROUTINE IS USED WRITE AN IMP ANALYSIS FILE                  *
*                                                                       *
*************************************************************************

      INTEGER PNUM(100),LNUM(30,14),JNUM(20,3),INUM(20,2),NP,NJ,NL,NI,
     &CONN(20,3)
```

```
      REAL LINK(30,6),JOINT(20,21),POINT(100,3),AXISR,CSIZE(3),AREA(6),
     &IC(20,4)
      CHARACTER*7 LNAM(30),JNAM(20),PNAM(100),FILE,FILEO,FILE1,INAM(20),
     &           JNAME,LNAME1,LNAME2,FILEN
      CHARACTER*19 FLINE
      DATA FILE/'IMP    '/
      NP=0
      NJ=0
      NL=0
      NI=0

      DO 2 I=1,20
         DO 1 J=1,3
            CONN(I,J)=0
    1    CONTINUE
    2 CONTINUE

      AREA(1)=.08*CSIZE(1)
      AREA(2)=.90*CSIZE(1)
      AREA(3)=.70*CSIZE(2)
      AREA(4)=.90*CSIZE(2)
      AREA(5)=.02*CSIZE(3)
      AREA(6)=.90*CSIZE(3)

      FILE1='       '

      REWIND 15
*****************************************************************
*  INITIALIZE THE STRING DEVICE. REQUEST NEW RESTART FILE NAME
*****************************************************************
C                              INITIALIZE STRING MODE
      CALL GPSTMO(IWSID,1,1,2)
C                              INITIALIZE THE STRING DEVICE
    3 CALL GPINST(IWSID,1,7,FILE,1,AREA,7,1,0,FILE)
C                              REQUEST THE FILE NAME
      CALL CMMAND(IWSID,83)
      FILEN='       '
C                              REQUEST STRING
      CALL GPRQST(IWSID,1,7,ISTAT,NUM,FILEN)
      IF(FILEN.EQ.'       ')GO TO 5000

      FILE=FILEN
C                              SEE IF FILE ALREADY EXISTS
    4 CALL FILEXS(FILE,IRET)
C                              IF FILE EXISTS, HAVE USER RETYPE TO USE
      IF(IRET.EQ.0)THEN
         CALL GPSTMO(IWSID,1,1,2)
         FILEO=FILE
         FILEN='       '
         CALL CMMAND(IWSID,8)
         CALL GPINST(IWSID,1,7,FILE,1,AREA,7,1,0,FILE)
         CALL GPRQST(IWSID,1,7,ISTAT,NUM,FILEN)
         IF(FILEN.EQ.'       ')GO TO 5000
         IF(FILE.NE.FILEO)THEN
            FILEN='       '
            GO TO 4
         ENDIF
      ENDIF

C                              OPEN A NEW FILE
      OPEN(UNIT=16,FILE=FILE,STATUS='NEW',ERR=2)

*****************************************************************
*    SORT OUT LINK CONNECTIVITIES
*****************************************************************
   30 DO 100 J=1,30
         IF(LNAM(J).EQ.'CCCCCCC')THEN
            GO TO 90
         ENDIF
         IF(LNAM(J).EQ.'BBBBBBB')THEN
            GO TO 105
         ENDIF

         LN=J
         DO 50 I=2,13,2
            IF(LNUM(LN,I).EQ.0)GO TO 90
            CONN(LNUM(LN,I),(LNUM(LN,I+1)+1))=LNUM(LN,1)
   50    CONTINUE

   90    CONTINUE
  100 CONTINUE

  105 DO 110 I=1,20
         CONN(I,1)=JNUM(I,2)
  110 CONTINUE

*****************************************************************
*    FILE JOINT TO GROUND WITH ZERO'S
*****************************************************************

      DO 200 I=1,20
```

```
         IF(CONN(I,1).EQ.8)THEN
            IGRD=CONN(I,2)
            CONN(I,2)=0
            DO 180 J=1,20
               IF(CONN(J,2).EQ.IGRD)THEN
                  CONN(J,2)=0
               ENDIF
               IF(CONN(J,3).EQ.IGRD)THEN
                  CONN(J,3)=0
               ENDIF
  180       CONTINUE
         ENDIF
  200 CONTINUE

*************************************************************************
*    WRITE FIRST LINE OF THE RESTART FILE AND AXIS RANGE
*************************************************************************

C      WRITE(16,*)'THIS WILL BE THE IMP FILE'
C      WRITE(16,*)'                        '
C
C      WRITE(16,*)'CONNECTION ARRAY '
C      DO 911 I=1,20
C         WRITE(16,*)(CONN(I,J),J=1,3)
C 911  CONTINUE
C      WRITE(16,*)'                        '
       WRITE(16,*)'GROUND=FRAME'
       DO 1000 I=1,20
          IF(CONN(I,2).EQ.0.AND.CONN(I,3).EQ.0)THEN
             GO TO 1000
          ENDIF
          IF(CONN(I,1).EQ.0.)THEN
             GO TO 2000
          ENDIF

          JNAME=JNAM(I)
          JNUMB=JNUM(I,1)

          LNAME1=LNAM(CONN(I,2))
          IF(CONN(I,2).EQ.0)THEN
             LNAME1='FRAME'
          ENDIF

          LNAME2=LNAM(CONN(I,3))
          IF(CONN(I,3).EQ.0)THEN
             LNAME2='FRAME'
          ENDIF

          JTYPE=CONN(I,1)

          IF(JTYPE.EQ.1)THEN
             WRITE(16,*)'REVOLUTE(',LNAME1,',',LNAME2,')=',JNAME
          ENDIF

          IF(JTYPE.EQ.2)THEN
             WRITE(16,*)'PRISM   (',LNAME1,',',LNAME2,')=',JNAME
          ENDIF

          IF(JTYPE.EQ.3)THEN
             WRITE(16,*)'CYLINDER(',LNAME1,',',LNAME2,')=',JNAME
          ENDIF

          IF(JTYPE.EQ.4)THEN
             WRITE(16,*)'SPHERE  (',LNAME1,',',LNAME2,')=',JNAME
          ENDIF

          IF(JTYPE.EQ.5)THEN
             WRITE(16,*)'SCREW   (',LNAME1,',',LNAME2,')=',JNAME
             WRITE(16,*)'DATA: SCREW (',JNAME,') =',JOINT(JNUMB,13)
          ENDIF

          IF(JTYPE.EQ.6)THEN
             WRITE(16,*)'FLAT    (',LNAME1,',',LNAME2,')=',JNAME
          ENDIF

          IF(JTYPE.EQ.7)THEN
             WRITE(16,*)'GEAR    (',LNAME1,',',LNAME2,')=',JNAME
             WRITE(16,*)'DATA: GEAR  (',JNAME,') =',JOINT(JNUMB,21)
          ENDIF

 1000 CONTINUE

 2000 DO 3000 I=1,20
          IF(CONN(I,2).EQ.0.AND.CONN(I,3).EQ.0)THEN
             GO TO 3000
          ENDIF
          IF(CONN(I,1).EQ.0.)THEN
             GO TO 3100
          ENDIF

          JNAME=JNAM(I)
          JNUMB=JNUM(I,1)

          LNAME1=LNAM(CONN(I,2))
          IF(CONN(I,2).EQ.0)THEN
             LNAME1='FRAME'
```

```
      ENDIF

      LNAME2=LNAM(CONN(I,3))
      IF(CONN(I,3).EQ.0)THEN
         LNAME2='FRAME'
      ENDIF

      JTYPE=CONN(I,1)

      WRITE(16,*)'DATA: LINK  (',LNAME1,',',JNAME,') =','$'
      WRITE(16,*)'                ',
     &  JOINT(JNUMB,1),',',
     &  JOINT(JNUMB,2),',',
     &  JOINT(JNUMB,3),'/','$'
      WRITE(16,*)'                ',
     &  JOINT(JNUMB,1)+JOINT(JNUMB,4),',',
     &  JOINT(JNUMB,2)+JOINT(JNUMB,5),',',
     &  JOINT(JNUMB,3)+JOINT(JNUMB,6),'/','$'
      WRITE(16,*)'                ',
     &  JOINT(JNUMB,1)+JOINT(JNUMB,7),',',
     &  JOINT(JNUMB,2)+JOINT(JNUMB,8),',',
     &  JOINT(JNUMB,3)+JOINT(JNUMB,9)

      WRITE(16,*)'DATA: LINK  (',LNAME2,',',JNAME,') =','$'
      WRITE(16,*)'                ',
     &  JOINT(JNUMB,15),',',
     &  JOINT(JNUMB,16),',',
     &  JOINT(JNUMB,17)+JOINT(JNUMB,14),'/','$'
      WRITE(16,*)'                ',
     &  JOINT(JNUMB,15)+JOINT(JNUMB,18),',',
     &  JOINT(JNUMB,16)+JOINT(JNUMB,19),',',
     &  JOINT(JNUMB,17)+JOINT(JNUMB,20)+JOINT(JNUMB,14),
     &  '/','$'
      WRITE(16,*)'                ',
     &  JOINT(JNUMB,15)+JOINT(JNUMB,10),',',
     &  JOINT(JNUMB,16)+JOINT(JNUMB,11),',',
     &  JOINT(JNUMB,17)+JOINT(JNUMB,12)+JOINT(JNUMB,14)

3000 CONTINUE

3100 DO 3500 I=1,20
      IF(INAM(I).EQ.'DDDDDDD')THEN
         WRITE(16,*)'DATA: POSITION(',JNAM(I),') =','$'
         WRITE(16,*)'                ',
     &     IC(I,1),',',
     &     IC(I,2),',',
     &     INUM(I,2)
         WRITE(16,*)'DATA: VELOCITY(',JNAM(I),') =','$'
         WRITE(16,*)'                ',
     &     IC(I,3)
         WRITE(16,*)'DATA: ACCEL   (',JNAM(I),') =','$'
         WRITE(16,*)'                ',
     &     IC(I,4)
      ENDIF
3500 CONTINUE

     IIII=IIII+1

     DO 3600 I=1,20
      IF(CONN(I,2).EQ.0.AND.CONN(I,3).EQ.0)THEN
         GO TO 3600
      ENDIF

      IF(CONN(I,1).EQ.0.)THEN
         GO TO 4000
      ENDIF

         LNAME1=LNAM(CONN(I,2))
         IF(CONN(I,2).EQ.0)THEN
            WRITE(16,*)'STORE: POSITION (',JNAM(I),')'
            WRITE(16,*)'STORE: VELOCITY (',JNAM(I),')'
            WRITE(16,*)'STORE: ACCELER  (',JNAM(I),')'
            GO TO 3600
         ENDIF

         LNAME2=LNAM(CONN(I,3))
         IF(CONN(I,3).EQ.0)THEN
            WRITE(16,*)'STORE: POSITION (',JNAM(I),')'
            WRITE(16,*)'STORE: VELOCITY (',JNAM(I),')'
            WRITE(16,*)'STORE: ACCELER  (',JNAM(I),')'
            GO TO 3600
         ENDIF

         WRITE(16,3550)LNAME1,I,CONN(I,2)
         WRITE(16,3550)LNAME2,I,CONN(I,3)
3550     FORMAT('POINT(',A7,') = ','PT',I1,I1)
         WRITE(16,3551)I,CONN(I,2),JNAM(I)
         WRITE(16,3551)I,CONN(I,3),JNAM(I)
3551     FORMAT('DATA: POINT   (PT',I1,I1,',',A7,') = 0., 0., 0.')
         WRITE(16,3552)I,CONN(I,2),I,CONN(I,3)
3552     FORMAT('STORE: POSITION (PT',I1,I1,',','PT',I1,I1,')')
         WRITE(16,3553)I,CONN(I,2),I,CONN(I,3)
3553     FORMAT('STORE: VELOCITY (PT',I1,I1,',','PT',I1,I1,')')
         WRITE(16,3554)I,CONN(I,2),I,CONN(I,3)
3554     FORMAT('STORE: ACCELER  (PT',I1,I1,',','PT',I1,I1,')')
```

```
 3600 CONTINUE

 4000 WRITE(16,*)'RETURN'
 5000 RETURN
      END
```

SUBROUTINE INQ

```
      SUBROUTINE INQ(IWSID,CSIZE,AXISR,PNAM,PNUM,POINT,JOINT)
*********************************************************************
* ----- SUBROUTINE INQ -----                                        *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED TO INQUIRE POSITIONS                    *
*                                                                   *
*********************************************************************

      INTEGER IWSID,PRFORM,PPATH(3),ACLASS(3),CHOICE,PNUM(100),LENG

      REAL CSIZE(3),DATA(35),PAREA(6),L1(4),POINT(100,3),AXISR,POSN(2),
     & LOCAT(3),BOX(8),SHD(8),LX(2),LY(2),LZ(2),LX1(2),LY1(2),LZ1(2),
     & JOINT(20,21)

      CHARACTER*7 PNAM(100)
      CHARACTER*16 RETURN
      CHARACTER*20 X,Y,Z,CLEAR
      CHARACTER*14 XVAL,YVAL,ZVAL

      DATA RETURN/'1.  RETURN      '/
      DATA LENG/16/
      DATA POSN/0.05,2.2/

      DATA X    /'X =                '/
      DATA Y    /'Y =                '/
      DATA Z    /'Z =                '/
      DATA CLEAR/'                   '/

      LX(1)=0.05
      LX(2)=0.68

      LX1(1)=0.14
      LX1(2)=0.68

      LY(1)=0.05
      LY(2)=0.62

      LY1(1)=0.14
      LY1(2)=0.62

      LZ(1)=0.05
      LZ(2)=0.56

      LZ1(1)=0.14
      LZ1(2)=0.56

      DATA DATA/32,0,0,          2,2,1,1,
     &                          1,1,1,1,1,1,
     &                          1,1,1,1,1,1,
     &                          1,1,1,1,1,1,
     &                          1,1,1,1,1,1,
     &                            1,1,1,1/
C                              FLAG FOR UPDATE
C     DATA PRFORM/2/           INCLUSION CLASS LIST
C     DATA ACLASS/2,3,4/
C                              GREY VALUATOR BOX
      DATA BOX/0.00,0.545,0.48,0.545,0.48,0.73,0.00,0.73/

      DO 10 I=2,8,2
         SHD(I)=BOX(I)-0.025
   10 CONTINUE

      DO 20 I=1,7,2
         SHD(I)=BOX(I)+0.025
   20 CONTINUE


*********************************************************************
*  REQUEST PICK INPUT
*********************************************************************
C                              SET PICK FILTER(ALL CLASS 2,3 DETECTABLE)
      CALL GPPKF(IWSID,1,3,ACLASS,0,ACLASS)
C                              MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                              DEFINE PICK AREA
      PAREA(1)=((-.98+1)/2)*CSIZE(1)
      PAREA(2)=((.98+1)/2)*CSIZE(1)
      PAREA(3)=((-.98+1)/2)*CSIZE(2)
```

```
      PAREA(4)=((.48+1)/2)*CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=CSIZE(3)
C                              INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                              PLACE PICK DEVICE IN THE EVENT MODE
      CALL GPPKMO(IWSID,1,3,2)
C                              TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)
C                              GET INQUIRY OR RETURN COMMAND
      CALL CMMAND(IWSID,22)

*********************************************************************
*  INCLUDE A RETURN PICK
*********************************************************************

C                              RETURN PICK
      CALL GPOPST(6)
      CALL GPADCN(1,ACLASS)
      CALL GPPKID(101)
      CALL GPTXCI(2)
      CALL GPAHSC(1.0)
      CALL GPAN2(POSN,LENG,RETURN)
      CALL GPCLST
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
      CALL GPUPWS(IWSID,PRFORM)
C                              GREY VALUE BOX AND SHADOW
      CALL GPOPST(8)
      CALL GPIS(2)
      CALL GPICI(4)
      CALL GPPG2(1,4,2,SHD)
      CALL GPICI(3)
      CALL GPPG2(1,4,2,BOX)
      CALL GPCLST
      CALL GPUPWS(IWSID,PRFORM)

*********************************************************************
*  AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
*********************************************************************
C                              AWAIT AN EVENT
  200 CALL GPAWEV(1000.,IWS,ICLA,IDEV)
      IF(ICLA.EQ.5)THEN
C                              ERASE OLD POSITION ICONS
         CALL GPEST(9)
         CALL GPEST(10)
         CALL GPEST(11)
C                              GET PICK
         CALL GPGTPK(1,IDEP,PPATH)
C                              IF NO PICK, RETURN
         IF(PPATH(2).EQ.101)GO TO 1000
C                              CHOICE IS SECOND ITEM OF PICK PATH
         CHOICE=PPATH(2)
         ISTRUC=PPATH(1)
C                              GET LOCATION
         IF(ISTRUC.EQ.12)THEN
              LOCAT(1)=POINT(CHOICE,1)
              LOCAT(2)=POINT(CHOICE,2)
              LOCAT(3)=POINT(CHOICE,3)
         ELSE
              LOCAT(1)=JOINT(CHOICE,1)
              LOCAT(2)=JOINT(CHOICE,2)
              LOCAT(3)=JOINT(CHOICE,3)
         ENDIF

         REWIND 10
         DO 350 III=1,3
              WRITE(10,300)CLEAR
  300         FORMAT(A20)
  350    CONTINUE
         REWIND 10
         WRITE(10,375)LOCAT(1)
         WRITE(10,375)LOCAT(2)
         WRITE(10,375)LOCAT(3)
  375    FORMAT(F14.6)
         REWIND 10
         READ(10,400)XVAL
         READ(10,400)YVAL
         READ(10,400)ZVAL
  400    FORMAT(A14)

C                              DRAW POSITION ICON

         CALL GPOPST(11)
         CALL ICON(LOCAT,AXISR)
         CALL GPCLST

         CALL GPOPST(9)
         CALL ICON(LOCAT,AXISR)
         CALL GPCLST

         CALL GPOPST(10)
         CALL GPTXCI(1)
         CALL GPAHSC(1.0)
```

```
         CALL GPAN2(LX,20,X)
         CALL GPAN2(LX1,14,XVAL)
         CALL GPAN2(LY,20,Y)
         CALL GPAN2(LY1,14,YVAL)
         CALL GPAN2(LZ,20,Z)
         CALL GPAN2(LZ1,14,ZVAL)
         CALL GPCLST

         CALL GPUPWS(IWSID,PRFORM)
C                          REQUEST A NEW INQUIRY
         GO TO 200
      ELSE
C                          IF NOT A PICK, GO TO AWAIT AN EVENT
         GO TO 200
      ENDIF

C                          MAKE SURE PICK IS DEACTIVATED
 1000 CALL GPPKMO(IWSID,1,1,2)
      CALL GPEST(6)
      CALL GPEST(8)
      CALL GPEST(9)
      CALL GPEST(10)
      CALL GPEST(11)
      CALL GPUPWS(IWSID,PRFORM)

C                          SET PICK FILTER(ALL CLASS 2,3 UNDETECTABLE)
      CALL GPPKF(IWSID,1,0,ACLASS,3,ACLASS)

      RETURN
      END
```

SUBROUTINE INQIC

```
      SUBROUTINE INQIC(IWSID,CSIZE,AXISR,INAM,INUM,IC)
*********************************************************************
* ----- SUBROUTINE INQIC -----                                      *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED TO INQUIRE JOINT I.C.                   *
*                                                                   *
*********************************************************************

      INTEGER IWSID,PRFORM,PPATH(3),ACLASS(3),CHOICE,PNUM(100),LENG,
     &STRU,INUM(20,2)

      REAL CSIZE(3),DATA(35),PAREA(6),L1(4),POINT(100,3),AXISR,POSN(2),
     & BOX(8),SHD(8),LX(2),LY(2),LZ(2),LX1(2),LY1(2),LZ1(2),
     & JOINT(20,21),ORIENT(3),IC(20,4),LX11(2),LY11(2),LZ11(2)

      CHARACTER*7 INAM(20)
      CHARACTER*16 RETURN
      CHARACTER*20 X,Y,Z,CLEAR,V,A
      CHARACTER*14 XVAL,YVAL,ZVAL

      DATA RETURN/'1.  RETURN      '/
      DATA LENG/16/
      DATA POSN/0.05,2.2/

      DATA X    /'POSITION =          '/
      DATA Y    /'INCREMENT=          '/
      DATA Z    /'STEPS    =          '/
      DATA V    /'VELOCITY =          '/
      DATA A    /'ACCELERATION=       '/
      DATA CLEAR/'                    '/

      LX(1)=0.05
      LX(2)=0.68

      LX1(1)=0.14
      LX1(2)=0.68

      LX11(1)=0.26
      LX11(2)=0.68

      LY(1)=0.05
      LY(2)=0.62

      LY1(1)=0.14
      LY1(2)=0.62

      LY11(1)=0.26
      LY11(2)=0.62

      LZ(1)=0.05
      LZ(2)=0.56

      LZ1(1)=0.14
      LZ1(2)=0.56
```

```
      LZ11(1)=0.26
      LZ11(2)=0.56

      DATA DATA/32,0,0,          2,2,1,1,
     &                          1,1,1,1,1,1,
     &                          1,1,1,1,1,1,
     &                          1,1,1,1,1,1,
     &                          1,1,1,1,1,1,
     &                            1,1,1,1/
C                               FLAG FOR UPDATE
      DATA PRFORM/2/
C                               INCLUSION CLASS LIST
      DATA ACLASS/3,4,6/

C                               GREY VALUATOR BOX
      DATA BOX/0.00,0.545,0.48,0.545,0.48,0.73,0.00,0.73/

      DO 10 I=2,8,2
         SHD(I)=BOX(I)-0.025
   10 CONTINUE

      DO 20 I=1,7,2
         SHD(I)=BOX(I)+0.025
   20 CONTINUE


*********************************************************************
*  FIND TYPE OF IC INQUIRY
*********************************************************************

      CALL CMMAND(IWSID,78)
      CALL MENU(IWSID,CSIZE,16,CHOICE)
      KKK=CHOICE


*********************************************************************
*  REQUEST PICK INPUT
*********************************************************************
C                               SET PICK FILTER(ALL CLASS 2,3 DETECTABLE)
      CALL GPPKF(IWSID,1,3,ACLASS,0,ACLASS)
C                               MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                               DEFINE PICK AREA
      PAREA(1)=((-.98+1)/2)*CSIZE(1)
      PAREA(2)=((.98+1)/2)*CSIZE(1)
      PAREA(3)=((-.98+1)/2)*CSIZE(2)
      PAREA(4)=((.48+1)/2)*CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=CSIZE(3)
C                               INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                               PLACE PICK DEVICE IN THE EVENT MODE
      CALL GPPKMO(IWSID,1,3,2)
C                               TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)
C                               GET INQUIRY OR RETURN COMMAND
      CALL CMMAND(IWSID,77)

*********************************************************************
*  INCLUDE A RETURN PICK
*********************************************************************

C                               RETURN PICK
      CALL GPOPST(6)
      CALL GPADCN(1,ACLASS)
      CALL GPPKID(101)
      CALL GPTXCI(2)
      CALL GPAHSC(1.0)
      CALL GPAN2(POSN,LENG,RETURN)
      CALL GPCLST
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
      CALL GPUPWS(IWSID,PRFORM)
C                               GREY VALUE BOX AND SHADOW
      CALL GPOPST(8)
      CALL GPIS(2)
      CALL GPICI(4)
      CALL GPPG2(1,4,2,SHD)
      CALL GPICI(3)
      CALL GPPG2(1,4,2,BOX)
      CALL GPCLST
      CALL GPUPWS(IWSID,PRFORM)

*********************************************************************
*  AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
*********************************************************************
C                               AWAIT AN EVENT
  200 CALL GPAWEV(1000.,IWS,ICLA,IDEV)
      IF(ICLA.EQ.5)THEN
C                               ERASE OLD INQ.
         CALL GPEST(10)
C                               GET PICK
         CALL GPGTPK(1,IDEP,PPATH)
C                               IF NO PICK, RETURN
```

```
      IF(PPATH(2).EQ.101)GO TO 1000
C                              CHOICE IS SECOND ITEM OF PICK PATH
      CHOICE=PPATH(2)
      STRU=PPATH(1)

      IF(KKK.EQ.1)THEN
           POS=IC(CHOICE,1)
           VEL=IC(CHOICE,2)
           INC=INUM(CHOICE,2)
         REWIND 10
         DO 350 III=1,3
              WRITE(10,300)CLEAR
300           FORMAT(A20)
350      CONTINUE
         REWIND 10
         WRITE(10,375)POS
         WRITE(10,375)VEL
         WRITE(10,380)INC
375      FORMAT(F8.3)
380      FORMAT(I8)
         REWIND 10
         READ(10,400)XVAL
         READ(10,400)YVAL
         READ(10,400)ZVAL
400      FORMAT(A8)

         CALL GPOPST(10)
         CALL GPTXCI(1)
         CALL GPAHSC(1.0)
         CALL GPAN2(LX,20,X)
         CALL GPAN2(LX11,8,XVAL)
         CALL GPAN2(LY,20,Y)
         CALL GPAN2(LY11,8,YVAL)
         CALL GPAN2(LZ,20,Z)
         CALL GPAN2(LZ11,8,ZVAL)
         CALL GPCLST

         CALL GPUPWS(IWSID,PRFORM)
C                              REQUEST A NEW INQUIRY
         GO TO 200
      ENDIF

      IF(KKK.EQ.2)THEN
         VEL=IC(CHOICE,3)
         REWIND 10
         WRITE(10,300)CLEAR
         REWIND 10
         WRITE(10,450)VEL
450      FORMAT(F14.3)
         REWIND 10
         READ(10,500)XVAL
500      FORMAT(A14)

         CALL GPOPST(10)
         CALL GPTXCI(1)
         CALL GPAHSC(1.0)
         CALL GPAN2(LX,20,V)
         CALL GPAN2(LY1,14,XVAL)
         CALL GPCLST

         CALL GPUPWS(IWSID,PRFORM)
C                              REQUEST A NEW INQUIRY
         GO TO 200
      ENDIF

      IF(KKK.EQ.3)THEN
         ACC=IC(CHOICE,4)
         REWIND 10
         WRITE(10,300)CLEAR
         REWIND 10
         WRITE(10,550)ACC
550      FORMAT(F14.3)
         REWIND 10
         READ(10,600)XVAL
600      FORMAT(A14)

         CALL GPOPST(10)
         CALL GPTXCI(1)
         CALL GPAHSC(1.0)
         CALL GPAN2(LX,20,A)
         CALL GPAN2(LY1,14,XVAL)
         CALL GPCLST

         CALL GPUPWS(IWSID,PRFORM)
C                              REQUEST A NEW INQUIRY
         GO TO 200
      ENDIF
   ELSE
C                              IF NOT A PICK, GO TO AWAIT AN EVENT
      GO TO 200
   ENDIF


C                              MAKE SURE PICK IS DEACTIVATED
1000 CALL GPPKMO(IWSID,1,1,2)
   CALL GPEST(6)
```

```
      CALL GPEST(8)
      CALL GPEST(16)
      CALL GPEST(10)
      CALL GPUPWS(IWSID,PRFORM)
C                              SET PICK FILTER(ALL CLASS 2,3 UNDETECTABLE)
      CALL GPPKF(IWSID,1,0,ACLASS,3,ACLASS)

      RETURN
      END
```

SUBROUTINE INQL

```
      SUBROUTINE INQL(IWSID,CSIZE,AXISR,LNAM,LNUM,JNAM)
*********************************************************************
* ----- SUBROUTINE INQO -----                                       *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED TO INQUIRE JOINT ORIENTATIONS           *
*                                                                   *
*********************************************************************

      INTEGER IWSID,PRFORM,PPATH(3),ACLASS(1),CHOICE,LENG,STRU,
     &        LNUM(30,14)

      REAL CSIZE(3),DATA(35),PAREA(6),L1(4),POINT(100,3),AXISR,POSN(2),
     & BOX(8),SHD(8),LX(2),LY(2),LZ(2),LX1(2),LY1(2),LZ1(2)

      CHARACTER*7 JNAM(100),LNAM(30),LNAME,JNAME
      CHARACTER*16 RETURN
      CHARACTER*20 X,Y,Z,CLEAR

      DATA RETURN/'1.  RETURN      '/
      DATA LENG/16/
      DATA POSN/0.05,2.2/

      DATA X    /'LINK NAME =          '/
      DATA Y    /'                     '/
      DATA Z    /'                     '/
      DATA CLEAR/'                     '/

      DATA DATA/32,0,0,          2,2,1,1,
     &                         1,1,1,1,1,1,
     &                         1,1,1,1,1,1,
     &                         1,1,1,1,1,1,
     &                         1,1,1,1,1,1,
     &                           1,1,1,1/
C                              FLAG FOR UPDATE
C     DATA PRFORM/2/
C                              INCLUSION CLASS LIST
C     DATA ACLASS/5/

C                              GREY VALUATOR BOX
      DATA BOX/0.00,0.545,0.48,0.545,0.48,0.73,0.00,0.73/


      LY(1)=0.02
      LY(2)=0.62

      LY1(1)=0.31
      LY1(2)=0.62

      DO 10 I=2,8,2
         SHD(I)=BOX(I)-0.025
   10 CONTINUE

      DO 20 I=1,7,2
         SHD(I)=BOX(I)+0.025
   20 CONTINUE


*********************************************************************
*  REQUEST PICK INPUT
*********************************************************************
C                              SET PICK FILTER(ALL CLASS 2,3 DETECTABLE)
      CALL GPPKF(IWSID,1,1,ACLASS,0,ACLASS)
C                                MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                              DEFINE PICK AREA
      PAREA(1)=((-.98+1)/2)*CSIZE(1)
      PAREA(2)=((.98+1)/2)*CSIZE(1)
      PAREA(3)=((-.98+1)/2)*CSIZE(2)
      PAREA(4)=((.48+1)/2)*CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=CSIZE(3)
C                              INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                              PLACE PICK DEVICE IN THE EVENT MODE
```

```
      CALL GPPKMO(IWSID,1,3,2)
C                               TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)
C                               GET INQUIRY OR RETURN COMMAND
      CALL CMMAND(IWSID,59)

*********************************************************************
*  INCLUDE A RETURN PICK
*********************************************************************

C                               RETURN PICK
      CALL GPOPST(6)
      CALL GPADCN(1,ACLASS)
      CALL GPPKID(101)
      CALL GPTXCI(2)
      CALL GPAHSC(1.0)
      CALL GPAN2(POSN,LENG,RETURN)
      CALL GPCLST
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
      CALL GPUPWS(IWSID,PRFORM)
C                               GREY VALUE BOX AND SHADOW
      CALL GPOPST(8)
      CALL GPIS(2)
      CALL GPICI(4)
      CALL GPPG2(1,4,2,SHD)
      CALL GPICI(3)
      CALL GPPG2(1,4,2,BOX)
      CALL GPCLST
      CALL GPUPWS(IWSID,PRFORM)


*********************************************************************
*  AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
*********************************************************************
C                               AWAIT AN EVENT
  200 CALL GPAWEV(1000.,IWS,ICLA,IDEV)
      IF(ICLA.EQ.5)THEN
C                               ERASE OLD PICK
         CALL GPEST(10)
C                               GET PICK
         CALL GPGTPK(1,IDEP,PPATH)
C                               IF NO PICK, RETURN
         IF(PPATH(2).EQ.101)GO TO 1000
C                               CHOICE IS SECOND ITEM OF PICK PATH
         CHOICE=PPATH(2)

         LNAME=LNAM(CHOICE)

         CALL GPOPST(10)
         CALL GPTXCI(1)
         CALL GPAHSC(1.0)
         CALL GPAN2(LY,20,X)
         CALL GPAN2(LY1,7,LNAME)
         CALL GPCLST

         CALL GPUPWS(IWSID,PRFORM)
C                               REQUEST A NEW INQUIRY
         GO TO 200
      ELSE
C                               IF NOT A PICK, GO TO AWAIT AN EVENT
         GO TO 200
      ENDIF


C                               MAKE SURE PICK IS DEACTIVATED
 1000 CALL GPPKMO(IWSID,1,1,2)
      CALL GPEST(10)
      CALL GPEST(6)
      CALL GPEST(8)
      CALL GPUPWS(IWSID,PRFORM)

C                               SET PICK FILTER(ALL CLASS 2,3 UNDETECTABLE)
      CALL GPPKF(IWSID,1,0,ACLASS,1,ACLASS)

      RETURN
      END
```

SUBROUTINE INQO

```
      SUBROUTINE INQO(IWSID,CSIZE,AXISR,JOINT)
*********************************************************************
* ----- SUBROUTINE INQO -----                                       *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED TO INQUIRE JOINT ORIENTATIONS           *
*                                                                   *
*********************************************************************
```

```
      INTEGER IWSID,PRFORM,PPATH(3),ACLASS(2),CHOICE,PNUM(100),LENG,STRU

      REAL CSIZE(3),DATA(35),PAREA(6),L1(4),POINT(100,3),AXISR,POSN(2),
     & BOX(8),SHD(8),LX(2),LY(2),LZ(2),LX1(2),LY1(2),LZ1(2),
     & JOINT(20,21),ORIENT(3)

      CHARACTER*7 PNAM(100)
      CHARACTER*16 RETURN
      CHARACTER*20 X,Y,Z,CLEAR
      CHARACTER*14 XVAL,YVAL,ZVAL

      DATA RETURN/'1.  RETURN      '/
      DATA LENG/16/
      DATA POSN/0.05,2.2/

      DATA X    /'DX =                '/
      DATA Y    /'DY =                '/
      DATA Z    /'DZ =                '/
      DATA CLEAR/'                    '/

      LX(1)=0.05
      LX(2)=0.68

      LX1(1)=0.14
      LX1(2)=0.68

      LY(1)=0.05
      LY(2)=0.62

      LY1(1)=0.14
      LY1(2)=0.62

      LZ(1)=0.05
      LZ(2)=0.56

      LZ1(1)=0.14
      LZ1(2)=0.56

      DATA DATA/32,0,0,          2,2,1,1,
     &                          1,1,1,1,1,1,
     &                          1,1,1,1,1,1,
     &                          1,1,1,1,1,1,
     &                          1,1,1,1,1,1,
     &                            1,1,1,1/
C                               FLAG FOR UPDATE
      DATA PRFORM/2/
C                               INCLUSION CLASS LIST
      DATA ACLASS/3,4/

C                               GREY VALUATOR BOX
      DATA BOX/0.00,0.545,0.48,0.545,0.48,0.73,0.00,0.73/

      DO 10 I=2,8,2
         SHD(I)=BOX(I)-0.025
   10 CONTINUE

      DO 20 I=1,7,2
         SHD(I)=BOX(I)+0.025
   20 CONTINUE


*********************************************************************
*  REQUEST PICK INPUT
*********************************************************************
C                               SET PICK FILTER(ALL CLASS 2,3 DETECTABLE)
      CALL GPPKF(IWSID,1,2,ACLASS,0,ACLASS)
C                               MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                               DEFINE PICK AREA
      PAREA(1)=((-.98+1)/2)*CSIZE(1)
      PAREA(2)=((.98+1)/2)*CSIZE(1)
      PAREA(3)=((-.98+1)/2)*CSIZE(2)
      PAREA(4)=((.48+1)/2)*CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=CSIZE(3)
C                               INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                               PLACE PICK DEVICE IN THE EVENT MODE
      CALL GPPKMO(IWSID,1,3,2)
C                               TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)
C                               GET INQUIRY OR RETURN COMMAND
      CALL CMMAND(IWSID,37)

*********************************************************************
*  INCLUDE A RETURN PICK
*********************************************************************

C                               RETURN PICK
      CALL GPOPST(6)
      CALL GPADCN(1,ACLASS)
      CALL GPPKID(101)
      CALL GPTXCI(2)
      CALL GPAHSC(1.0)
      CALL GPAN2(POSN,LENG,RETURN)
```

```
      CALL GPCLST
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
      CALL GPUPWS(IWSID,PRFORM)
C                              GREY VALUE BOX AND SHADOW
      CALL GPOPST(8)
      CALL GPIS(2)
      CALL GPICI(4)
      CALL GPPG2(1,4,2,SHD)
      CALL GPICI(3)
      CALL GPPG2(1,4,2,BOX)
      CALL GPCLST
      CALL GPUPWS(IWSID,PRFORM)

*********************************************************************
*  AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
*********************************************************************
C                              AWAIT AN EVENT
  200 CALL GPAWEV(1000.,IWS,ICLA,IDEV)
      IF(ICLA.EQ.5)THEN
C                              ERASE OLD ORIENT. ICONS
         CALL GPEST(9)
         CALL GPEST(10)
         CALL GPEST(11)
         CALL GPEST(16)
C                              GET PICK
         CALL GPGTPK(1,IDEP,PPATH)
C                              IF NO PICK, RETURN
         IF(PPATH(2).EQ.101)GO TO 1000
C                              CHOICE IS SECOND ITEM OF PICK PATH
         CHOICE=PPATH(2)
         STRU=PPATH(1)

         IF(STRU.EQ.18)THEN
              ORIENT(1)=JOINT(CHOICE,18)
              ORIENT(2)=JOINT(CHOICE,19)
              ORIENT(3)=JOINT(CHOICE,20)
         ELSE
              ORIENT(1)=JOINT(CHOICE,4)
              ORIENT(2)=JOINT(CHOICE,5)
              ORIENT(3)=JOINT(CHOICE,6)
         ENDIF

         REWIND 10
         DO 350 III=1,3
              WRITE(10,300)CLEAR
  300         FORMAT(A20)
  350    CONTINUE
         REWIND 10
         WRITE(10,375)ORIENT(1)
         WRITE(10,375)ORIENT(2)
         WRITE(10,375)ORIENT(3)
  375    FORMAT(F14.6)
         REWIND 10
         READ(10,400)XVAL
         READ(10,400)YVAL
         READ(10,400)ZVAL
  400    FORMAT(A14)

C                              DRAW POSITION ICON

         CALL GPOPST(16)
         CALL ICONO(ORIENT,AXISR)
         CALL GPCLST

         CALL GPOPST(10)
         CALL GPTXCI(1)
         CALL GPAHSC(1.0)
         CALL GPAN2(LX,20,X)
         CALL GPAN2(LX1,14,XVAL)
         CALL GPAN2(LY,20,Y)
         CALL GPAN2(LY1,14,YVAL)
         CALL GPAN2(LZ,20,Z)
         CALL GPAN2(LZ1,14,ZVAL)
         CALL GPCLST

         CALL GPUPWS(IWSID,PRFORM)
C                              REQUEST A NEW INQUIRY
         GO TO 200
      ELSE
C                              IF NOT A PICK, GO TO AWAIT AN EVENT
         GO TO 200
      ENDIF

C                              MAKE SURE PICK IS DEACTIVATED
 1000 CALL GPPKMO(IWSID,1,1,2)
      CALL GPEST(6)
      CALL GPEST(8)
      CALL GPEST(16)
      CALL GPEST(10)
      CALL GPUPWS(IWSID,PRFORM)
```

```
C                          SET PICK FILTER(ALL CLASS 2,3 UNDETECTABLE)
      CALL GPPKF(IWSID,1,0,ACLASS,2,ACLASS)

      RETURN
      END
```

SUBROUTINE INVJ

```
      SUBROUTINE INVJ(IWSID,JNAM,JNUM,JOINT,AXISR,SCALE,CSIZE)
*********************************************************************
* ----- SUBROUTINE INVJT  -----                                     *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED TO MAKE JOINTS INVISIBLE                *
*                                                                   *
*********************************************************************

      INTEGER IWSID,NUMP,JNUM(20,3),JTYPE,END

      REAL LOCAT(3),JOINT(20,21),AXISR,ORIENT(3),SCALE,ENDP(3),CSIZE(3)

      CHARACTER*7 JNAME,JNAM(20)

      CALL CMMAND(IWSID,86)
      CALL PICKJ2(IWSID,JOINT,AXISR,CSIZE,END,ENDP,IAXIS,IR)
      JNUM(END,3)=1
      IF(IR.EQ.1)GO TO 1000

      CALL RDRWJT(IWSID,JNAM,JNUM,JOINT,AXISR,SCALE)

 1000 RETURN
      END
```

SUBROUTINE INVL

```
      SUBROUTINE INVL(IWSID,JNAM,JNUM,JOINT,AXISR,LNAM,LNUM,LINK,
     &SCALE,CSIZE)
*********************************************************************
* ----- SUBROUTINE INVL   -----                                     *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED TO MAKE LINKS INVISIBLE                 *
*                                                                   *
*********************************************************************

      INTEGER IWSID,NUMP,JNUM(20,3),JTYPE,END,LNUM(30,14)

      REAL LOCAT(3),JOINT(20,21),AXISR,ORIENT(3),SCALE,ENDP(3),CSIZE(3),
     & LINK(30,6)

      CHARACTER*7 JNAME,JNAM(20),LNAM(30)

      CALL CMMAND(IWSID,87)
      CALL PICKL2(IWSID,CSIZE,AXISR,LNAM,LNUM,JNAM,END,IR)
      LNUM(END,14)=1
      IF(IR.EQ.1)GO TO 1000

      CALL RDRWLK(IWSID,JNAM,JNUM,JOINT,AXISR,LNAM,LNUM,LINK,SCALE)

 1000 RETURN
      END
```

SUBROUTINE JICONS

```
      SUBROUTINE JICONS(LOCAT,ORIENT,TYPE,AXISR,JOINT,NUMJ,SCALE,MAT)
*********************************************************************
* ----- SUBROUTINE JICONS  -----                                    *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED DISPLAY THE ICONS FOR THE JOINTS        *
*                                                                   *
*********************************************************************

      INTEGER TYPE

      REAL LOCAT(3),ORIENT(3),AXISR,REV(78),PRI(30),SPH(132),MATPO(4,4),
```

```
     & LINE(6),LINE2(6),SCR(33),FLA(30),CLD(78),GND(48),JOINT(20,21),
     & GEAR(39),GEAR1(39),GEAR2(39),P(6),REVS(78),PRIS(30),MAT(4,4),
     & DIST1(3),DISTN(3),G1(3),G2(3),GTN(3),M(4,4),MS(4,4)

      DATA M/1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./

      DATA REV/ 0.,  .08,   .02,  -.04, .0692,   .02,-.0692,   .04,.02,
     &    -.08,  0.,   .02,-.0692,  -.04,   .02,  -.04,-.0692,.02,
     &      0., -.08,   .02,   .04,-.0692,   .02, .0692,  -.04,.02,
     &     .08,  0.,   .02, .0692,   .04,   .02,   .04, .0692,.02,
     &      .0,  .08,.02,
     &      0.,  .08,  -.02,  -.04, .0692,  -.02,-.0692,   .04,-.02,
     &    -.08,  0.,  -.02,-.0692,  -.04,  -.02,  -.04,-.0692,-.02,
     &      0., -.08,  -.02,   .04,-.0692,  -.02, .0692,  -.04,-.02,
     &     .08,  0.,  -.02, .0692,   .04,  -.02,   .04, .0692,-.02,
     &      .0,  .08, -.02  /
      DATA PRI/.04,  .04,   .05,  -.04,   .04,   .05,  -.04,  -.04, .05,
     &        .04, -.04,   .05,   .04,   .04,.05,
     &        .04,  .04,  -.05,  -.04,   .04,  -.05,  -.04,  -.04,-.05,
     &        .04, -.04,  -.05,   .04,   .04,  -.05  /

      DATA FLA/.07,  .07,   .02,  -.07,   .07,   .02,  -.07,  -.07,.02,
     &        .07, -.07,   .02,   .07,   .07,   .02,
     &        .07,  .07,  -.02,  -.07,   .07,  -.02,  -.07,  -.07,-.02,
     &        .07, -.07,  -.02,   .07,   .07,  -.02 /

      DATA CLD/ .0, .04,   .05,  -.02, .0346,   .05,-.0346,   .02, .05,
     &       -.04,  0.,   .05,-.0346,  -.02,   .05,  -.02,-.0346, .05,
     &         0., -.04,   .05,   .02,-.0346,   .05, .0346,  -.02,.05,
     &        .04,  0.,  .05, .0346,   .02,   .05,   .02, .0346,.05,
     &         .0,  .04, .05,
     &         0.,  .04,  -.05,  -.02, .0346,  -.05,-.0346,   .02,-.05,
     &       -.04,  0.,  -.05,-.0346,  -.02,  -.05,  -.02,-.0346,-.05,
     &         0., -.04,  -.05,   .02,-.0346,  -.05, .0346,  -.02,-.05,
     &        .04,  0.,  -.05, .0346,   .02,  -.05,   .02, .0346,-.05,
     &         .0,  .04, -.05   /


      DATA SPH/.0,  .04,  0.,   .0, .0346,   .02,   .0,   .02, .0346,
     &        .0,  .0,  .04,
     &        .0,  0.,  .04,   .0,  -.02, .0346,   .0,-.0346,   .02,
     &        .0, -.04,  0.,   .0,-.0346,  -.02,   .0,  -.02,-.0346,
     &        .0,  0., -.04,   .0,   .02,-.0346,   .0, .0346,  -.02,
     &        .0,  .04,  0.,
     &        .0,  .04,  0., -.02,   .0346,  0.,-.0346,   .02,0.,
     &      -.04,  0.,   0.,-.0346,  -.02,  0.,  -.02,-.0346,0.,
     &       0., -.04,   0.,   .02,-.0346,   0., .0346,  -.02,0.,
     &      .04,  0.,   0., .0346,   .02,   0.,   .02, .0346,0.,
     &       .0,  .04,  0.,
     &       .0,  .04,  0., -.02,   .0346,  0.,-.0346,   .02,   0.,
     &      -.04,  0.,   0.,
     &      -.04,  0.,  .0,-.0346,     0., -.02,  -.02,   0.,-.0346,
     &       0.,  0.,-.04,   .02,     0.,-.0346,.0346,   0.,  -.02,
     &       .04,  0.,  0., .0346,     0.,   .02,  .02,   0., .0346,
     &       0.,  0., .04, -.02,      0.,.0346,-.0346,   0.,  .02,
     &     -.04, .0,   0.    /

      DATA SCR/ 0.,  .04, .05, -.02, .0346, .04,  -.0346,   .02,   .03,
     &        -.04,  0.,   .02,-.0346,  -.02,   .01,  -.02,-.0346, .00,
     &         0., -.04,  -.01,   .02,-.0346,  -.02, .0346,  -.02,-.03,
     &        .04,  0.,  -.04, .0346,   .02,  -.05 /

C     DATA GEAR/ 0.,  .08,  .0,  -.04, .0692,   .0,-.0692,   .04,  0.,
C    &          -.08,  0.,   .0,-.0692,  -.04,   .0,  -.04,-.0692,  0.,
C    &           0., -.08,   .0,   .04,-.0692,   .0, .0692,  -.04,  0.,
C    &          .08,  0.,   .0, .0692,   .04,   .0,   .04, .0692,  0.,
C    &           .0,  .08,  .0 /
      DATA GEAR/  0.,  .08,  .0,  -.04, .0692,   .0,-.0692,   .04, 0.,
     &          -.08,  0.,   .0,-.0692,  -.04,   .0,  -.04,-.0692,0.,
     &           0., -.08,   .0,   .04,-.0692,   .0, .0692,  -.04,0.,
     &          .08,  0.,   .0, .0692,   .04,   .0,   .04, .0692,0.,
     &           .0,  .08, 0.0/


      IF(TYPE.EQ.1)THEN
         DO 1 I=1,78
            REVS(I)=REV(I)*.6
    1    CONTINUE
C                            SET POLYLINE COLOR
         CALL GPPLCI(5)
C                            SET LINE TYPE SOLID
         CALL GPLT(1)
C                            DRAW JOINT
         CALL GPPL3(26,3,REVS)
      ENDIF

      IF(TYPE.EQ.2)THEN
         DO 2 I=1,30
            PRIS(I)=PRI(I)*1.5
    2    CONTINUE
C                            SET POLYLINE COLOR
         CALL GPPLCI(5)
C                            SET LINE TYPE SOLID
         CALL GPLT(1)
C                            DRAW JOINT
```

```
         CALL GPPL3(10,3,PRIS)
      ENDIF

      IF(TYPE.EQ.3)THEN
C                              SET POLYLINE COLOR
         CALL GPPLCI(5)
C                              SET LINE TYPE SOLID
C        CALL GPLT(1)
C                              DRAW JOINT
         CALL GPPL3(26,3,CLD)
      ENDIF

      IF(TYPE.EQ.4)THEN
C                              SET POLYLINE COLOR
         CALL GPPLCI(5)
C                              SET LINE TYPE SOLID
         CALL GPLT(1)
C                              DRAW JOINT
         CALL GPPL3(44,3,SPH)
      ENDIF

      IF(TYPE.EQ.5)THEN
C                              SET POLYLINE COLOR
         CALL GPPLCI(5)
C                              SET LINE TYPE SOLID
         CALL GPLT(1)
C                              DRAW JOINT
         CALL GPPL3(26,3,CLD)
         CALL GPPL3(11,3,SCR)
      ENDIF

      IF(TYPE.EQ.6)THEN
C                              SET POLYLINE COLOR
         CALL GPPLCI(5)
C                              SET LINE TYPE SOLID
         CALL GPLT(1)
C                              DRAW JOINT
         CALL GPPL3(10,3,FLA)
      ENDIF

      IF(TYPE.EQ.7)THEN
         P(1)=JOINT(NUMJ,1)
         P(2)=JOINT(NUMJ,2)
         P(3)=JOINT(NUMJ,3)
         P(4)=JOINT(NUMJ,15)
         P(5)=JOINT(NUMJ,16)
         P(6)=JOINT(NUMJ,17)
         DIST=((P(1)-P(4))**2+(P(2)-P(5))**2+(P(3)-P(6))**2)**.5
         R=JOINT(NUMJ,21)
         R2=DIST*12/(1+R)
         R1=R*DIST*12/(1+R)
         DO 10 I=1,37,3
              GEAR1(I)=GEAR(I)*R1
              GEAR1(I+1)=GEAR(I+1)*R1
              GEAR1(I+2)=GEAR(I+2)*R1
   10    CONTINUE
         DO 20 I=1,37,3
              GEAR2(I)=(GEAR(I)*R2)
              GEAR2(I+1)=(GEAR(I+1)*R2)
              GEAR2(I+2)=(GEAR(I+2)*R2)
   20    CONTINUE
C                              SET POLYLINE COLOR
         CALL GPPLCI(5)
C                              SET LINE TYPE SOLID
C        CALL GPLT(1)
C                              DRAW JOINT
         CALL GPMLX3(MAT,3)
         CALL GPPL3(13,3,GEAR1)
         MAT(1,4)=P(4)
         MAT(2,4)=P(5)
         MAT(3,4)=P(6)
         CALL GPMLX3(MAT,3)
         CALL GPPL3(13,3,GEAR2)
      ENDIF

      IF(TYPE.EQ.8)THEN

         X=.125

         GND(1)=X
         GND(2)=-0.009
         GND(3)=0.0

         DO 100 I=4,40,9
            GND(I)=X-.02
            GND(I+1)=-.029
            GND(I+2)=0.0
            GND(I+3)=X
            GND(I+4)=-0.009
            GND(I+5)=0.0
            GND(I+6)=X-.05
            GND(I+7)=-0.009
            GND(I+8)=0.0
            X=X-.05
  100    CONTINUE
```

```
C                              SET POLYLINE COLOR
         CALL GPPLCI(5)
C                              SET LINE TYPE SOLID
         CALL GPLT(1)
C                              DRAW JOINT
         CALL GPPL3(16,3,GND)
      ENDIF

      RETURN
      END
```

SUBROUTINE JOINTS

```
      SUBROUTINE JOINTS(IWSID,AXISR,CSIZE,JNAM,JNUM,JOINT,PNAM,PNUM,
     &POINT,SCALE)
************************************************************************
* ----- SUBROUTINE JOINT -----                                         *
*                                                                      *
*                                                                      *
*   THIS SUBROUTINE IS USED TO CREATE JOINT DATA                       *
*                                                                      *
************************************************************************

      REAL AXISR,CSIZE(3),POINT(100,3),JOINT(20,21),SCALE

      INTEGER IWSID,CHOICE,PNUM(100),JNUM(20,3)

      CHARACTER*7 PNAM(100),JNAM(20)

************************************************************************
*  FIND OUT IF USER IS TO CREATE OR DELETE JOINTS OR POINTS
************************************************************************
C                              REQUEST TYPE OF DATA INPUT
   10 CALL CMMAND(IWSID,10)
      CALL MENU(IWSID,CSIZE,3,CHOICE)
      IF(CHOICE.EQ.1)THEN
C                              REQUEST ADD A JOINT
         CALL ADDJ(IWSID,CSIZE,AXISR,JNAM,JNUM,JOINT,POINT,SCALE)
         GO TO 10
      ENDIF
      IF(CHOICE.EQ.2)THEN
C                              REQUEST DELETE A JOINT
         CALL DELJ(IWSID,CSIZE,AXISR,JNAM,JNUM,JOINT,SCALE)
         GO TO 10
      ENDIF
      IF(CHOICE.EQ.3)THEN
C                              REQUEST ADD A POINT
         CALL ADDP(IWSID,CSIZE,AXISR,PNAM,PNUM,POINT)
         GO TO 10
      ENDIF
      IF(CHOICE.EQ.4)THEN
C                              REQUEST DELETE A JOINT
         CALL DELP(IWSID,CSIZE,AXISR,PNAM,PNUM,POINT)
         GO TO 10
      ENDIF
      IF(CHOICE.EQ.5)THEN
C                              INQUIRE A POSITION
         CALL INQ(IWSID,CSIZE,AXISR,PNAM,PNUM,POINT,JOINT)
         GO TO 10
      ENDIF
      IF(CHOICE.EQ.6)THEN
C                              INQUIRE AN ORIENTATION
         CALL INQO(IWSID,CSIZE,AXISR,JOINT)
         GO TO 10
      ENDIF
      IF(CHOICE.EQ.7)THEN
C                              RETURN TO JOINT,LINK,OR IC MENU
         RETURN
      ENDIF

      RETURN
      END
```

SUBROUTINE LINKS

```
      SUBROUTINE LINKS(IWSID,AXISR,CSIZE,JNAM,JNUM,JOINT,PNAM,PNUM,
     &POINT,LNAM,LNUM,LINK,SCALE)
************************************************************************
* ----- SUBROUTINE LINK  -----                                         *
*                                                                      *
*                                                                      *
*   THIS SUBROUTINE IS USED TO CREATE LINK DATA                        *
```

```
*                                                                      *
************************************************************************

      REAL AXISR,CSIZE(3),POINT(100,3),JOINT(20,21),LINK(30,6)
      INTEGER IWSID,CHOICE,PNUM(100),JNUM(20,3),LNUM(30,14)
      CHARACTER*7 PNAM(100),JNAM(20),LNAM(30)

************************************************************************
*  FIND OUT IF USER IS TO CREATE OR DELETE LINKS
************************************************************************

C                              REQUEST TYPE OF DATA INPUT
   10 CALL CMMAND(IWSID,10)
      CALL MENU(IWSID,CSIZE,11,CHOICE)

      IF(CHOICE.EQ.1)THEN
C                              ADD A LINK
         CALL ADDL(IWSID,CSIZE,AXISR,JNAM,JNUM,JOINT,PNAM,PNUM,POINT,
     &LNAM,LNUM,LINK,SCALE)
         GO TO 10
      ENDIF

      IF(CHOICE.EQ.2)THEN
C                              DELETE A LINK
         CALL DELL(IWSID,CSIZE,AXISR,JNAM,JNUM,JOINT,LNAM,LNUM,LINK,
     & SCALE)
         GO TO 10
      ENDIF

      IF(CHOICE.EQ.3)THEN
C                              INQUIRE LINK NAME
         CALL INQL(IWSID,CSIZE,AXISR,LNAM,LNUM,JNAM)
         GO TO 10
      ENDIF

      IF(CHOICE.EQ.4)THEN
C                              RETURN TO JOINT,LINK,OR IC MENU
         RETURN
      ENDIF

      RETURN
      END
```

SUBROUTINE MENU

```
      SUBROUTINE MENU(IWSID,CSIZE,NUM,CHOICE)
************************************************************************
* ----- SUBROUTINE MENU -----                                          *
*                                                                      *
*                                                                      *
*   THIS SUBROUTINE IS USED TO DISPLAY MENU ITEMS IN THE MENU BOX      *
*                                                                      *
*   AND PROCESS THE MENU PICKS.                                        *
*                                                                      *
************************************************************************

      INTEGER IWSID,LENG,STRUCT,PRFORM,NUM,PPATH(3),ACLASS(1),CHOICE
      REAL POSN(2),CSIZE(3),DATA(35),PAREA(6),L1(4)
      CHARACTER*35 TMENU(30,18)

C                              TITLE TEXT, LENGTH
      DATA LENG/19/

      DATA TMENU(1,1)/'1. NEW MODEL        '/
      DATA TMENU(1,10)/'                   '/
      DATA TMENU(1,2)/'2. RESTART MODEL    '/
      DATA TMENU(1,11)/'                   '/
      DATA TMENU(1,3)/'                    '/
      DATA TMENU(1,12)/'                   '/
      DATA TMENU(1,4)/'                    '/
      DATA TMENU(1,13)/'                   '/
      DATA TMENU(1,5)/'                    '/
      DATA TMENU(1,14)/'                   '/
      DATA TMENU(1,6)/'                    '/
      DATA TMENU(1,15)/'                   '/
      DATA TMENU(1,7)/'                    '/
      DATA TMENU(1,16)/'                   '/
      DATA TMENU(1,8)/'                    '/
      DATA TMENU(1,17)/'                   '/
      DATA TMENU(1,9)/'                    '/
      DATA TMENU(1,18)/'                   '/

      DATA TMENU(2,1) /'1. DEFINE           '/
      DATA TMENU(2,10)/'   JOINTS           '/
```

```
      DATA TMENU(2,2) /'2. DEFINE          '/
      DATA TMENU(2,11)/'   LINKS           '/
      DATA TMENU(2,3) /'3. ENTER INITIAL   '/
      DATA TMENU(2,12)/'   CONDITIONS      '/
      DATA TMENU(2,4) /'4. CHANGE          '/
      DATA TMENU(2,13)/'   VIEW            '/
      DATA TMENU(2,5) /'                   '/
      DATA TMENU(2,14)/'                   '/
      DATA TMENU(2,6) /'5. END OF DATA     '/
      DATA TMENU(2,15)/'   INPUT           '/
      DATA TMENU(2,7) /'                   '/
      DATA TMENU(2,16)/'                   '/
      DATA TMENU(2,8) /'                   '/
      DATA TMENU(2,17)/'                   '/
      DATA TMENU(2,9) /'                   '/
      DATA TMENU(2,18)/'                   '/

      DATA TMENU(3,1) /'1. ADD JOINT       '/
      DATA TMENU(3,10) /'                  '/
      DATA TMENU(3,2) /'2. DELETE JOINT    '/
      DATA TMENU(3,11)/'                   '/
      DATA TMENU(3,3) /'3. ADD POINT       '/
      DATA TMENU(3,12)/'                   '/
      DATA TMENU(3,4) /'4. DELETE POINT    '/
      DATA TMENU(3,13)/'                   '/
      DATA TMENU(3,5) /'5. INQUIRE         '/
      DATA TMENU(3,14)/'   POSITION        '/
      DATA TMENU(3,6) /'6. INQUIRE         '/
      DATA TMENU(3,15)/'   ORIENTATION     '/
      DATA TMENU(3,7) /'7. RETURN          '/
      DATA TMENU(3,16)/'                   '/
      DATA TMENU(3,8) /'                   '/
      DATA TMENU(3,17)/'                   '/
      DATA TMENU(3,9) /'                   '/
      DATA TMENU(3,18)/'                   '/

      DATA TMENU(4,1) /'1. RE-ENTER AXIS   '/
      DATA TMENU(4,10)/'   RANGE           '/
      DATA TMENU(4,2) /'2. ENTER NEW POINT '/
      DATA TMENU(4,11)/'                   '/
      DATA TMENU(4,3) /'                   '/
      DATA TMENU(4,12)/'                   '/
      DATA TMENU(4,4) /'                   '/
      DATA TMENU(4,13)/'                   '/
      DATA TMENU(4,5) /'                   '/
      DATA TMENU(4,14)/'                   '/
      DATA TMENU(4,6) /'                   '/
      DATA TMENU(4,15)/'                   '/
      DATA TMENU(4,7) /'                   '/
      DATA TMENU(4,16)/'                   '/
      DATA TMENU(4,8) /'                   '/
      DATA TMENU(4,17)/'                   '/
      DATA TMENU(4,9) /'                   '/
      DATA TMENU(4,18)/'                   '/

      DATA TMENU(5,1) /'1. ENTER POSITION  '/
      DATA TMENU(5,10)/'                   '/
      DATA TMENU(5,2) /'2. RETURN          '/
      DATA TMENU(5,11)/'                   '/
      DATA TMENU(5,3) /'                   '/
      DATA TMENU(5,12)/'                   '/
      DATA TMENU(5,4) /'                   '/
      DATA TMENU(5,13)/'                   '/
      DATA TMENU(5,5) /'                   '/
      DATA TMENU(5,14)/'                   '/
      DATA TMENU(5,6) /'                   '/
      DATA TMENU(5,15)/'                   '/
      DATA TMENU(5,7) /'                   '/
      DATA TMENU(5,16)/'                   '/
      DATA TMENU(5,8) /'                   '/
      DATA TMENU(5,17)/'                   '/
      DATA TMENU(5,9) /'                   '/
      DATA TMENU(5,18)/'                   '/

      DATA TMENU(6,1) /'1. REVOLUTE        '/
      DATA TMENU(6,10)/'                   '/
      DATA TMENU(6,2) /'2. PRISMATIC       '/
      DATA TMENU(6,11)/'                   '/
      DATA TMENU(6,3) /'3. CYLINDRIC       '/
      DATA TMENU(6,12)/'                   '/
      DATA TMENU(6,4) /'4. SPHERICAL       '/
      DATA TMENU(6,13)/'                   '/
      DATA TMENU(6,5) /'5. SCREW           '/
      DATA TMENU(6,14)/'                   '/
      DATA TMENU(6,6) /'6. FLAT            '/
      DATA TMENU(6,15)/'                   '/
      DATA TMENU(6,7) /'7. GEAR            '/
      DATA TMENU(6,16)/'                   '/
      DATA TMENU(6,8) /'8. GROUND          '/
      DATA TMENU(6,17)/'                   '/
      DATA TMENU(6,9) /'9. RETURN          '/
      DATA TMENU(6,18)/'                   '/

      DATA TMENU(7,1) /'1. DIRECT ENTRY    '/
```

```
      DATA TMENU(7,10)/'                    '/
      DATA TMENU(7,2) /'2. VALUATOR ENTRY   '/
      DATA TMENU(7,11)/'                    '/
      DATA TMENU(7,3) /'3. TOWARDS A JOINT  '/
      DATA TMENU(7,12)/'   OR POINT         '/
      DATA TMENU(7,4) /'4. PARALLEL TO AN   '/
      DATA TMENU(7,13)/'   EXISTING JOINT   '/
      DATA TMENU(7,5) /'5. RETURN           '/
      DATA TMENU(7,14)/'                    '/
      DATA TMENU(7,6) /'                    '/
      DATA TMENU(7,15)/'                    '/
      DATA TMENU(7,7) /'                    '/
      DATA TMENU(7,16)/'                    '/
      DATA TMENU(7,8) /'                    '/
      DATA TMENU(7,17)/'                    '/
      DATA TMENU(7,9) /'                    '/
      DATA TMENU(7,18)/'                    '/

      DATA TMENU(8,1) /'1. ENTER            '/
      DATA TMENU(8,10)/'   ORIENTATION      '/
      DATA TMENU(8,2) /'2. RETURN           '/
      DATA TMENU(8,11)/'                    '/
      DATA TMENU(8,3) /'                    '/
      DATA TMENU(8,12)/'                    '/
      DATA TMENU(8,4) /'                    '/
      DATA TMENU(8,13)/'                    '/
      DATA TMENU(8,5) /'                    '/
      DATA TMENU(8,14)/'                    '/
      DATA TMENU(8,6) /'                    '/
      DATA TMENU(8,15)/'                    '/
      DATA TMENU(8,7) /'                    '/
      DATA TMENU(8,16)/'                    '/
      DATA TMENU(8,8) /'                    '/
      DATA TMENU(8,17)/'                    '/
      DATA TMENU(8,9) /'                    '/
      DATA TMENU(8,18)/'                    '/

      DATA TMENU(9,1) /'1. SAME AXIS        '/
      DATA TMENU(9,10)/'   ORIENTATION      '/
      DATA TMENU(9,2) /'2. ENTER NEW AXIS   '/
      DATA TMENU(9,11)/'   ORIENTATION      '/
      DATA TMENU(9,3) /'                    '/
      DATA TMENU(9,12)/'                    '/
      DATA TMENU(9,4) /'                    '/
      DATA TMENU(9,13)/'                    '/
      DATA TMENU(9,5) /'                    '/
      DATA TMENU(9,14)/'                    '/
      DATA TMENU(9,6) /'                    '/
      DATA TMENU(9,15)/'                    '/
      DATA TMENU(9,7) /'                    '/
      DATA TMENU(9,16)/'                    '/
      DATA TMENU(9,8) /'                    '/
      DATA TMENU(9,17)/'                    '/
      DATA TMENU(9,9) /'                    '/
      DATA TMENU(9,18)/'                    '/

      DATA TMENU(10,1) /'1.SAME CENTER      '/
      DATA TMENU(10,10)/'   LOCATION        '/
      DATA TMENU(10,2) /'2.ENTER NEW CENTER '/
      DATA TMENU(10,11)/'   LOCATION        '/
      DATA TMENU(10,3) /'                   '/
      DATA TMENU(10,12)/'                   '/
      DATA TMENU(10,4) /'                   '/
      DATA TMENU(10,13)/'                   '/
      DATA TMENU(10,5) /'                   '/
      DATA TMENU(10,14)/'                   '/
      DATA TMENU(10,6) /'                   '/
      DATA TMENU(10,15)/'                   '/
      DATA TMENU(10,7) /'                   '/
      DATA TMENU(10,16)/'                   '/
      DATA TMENU(10,8) /'                   '/
      DATA TMENU(10,17)/'                   '/
      DATA TMENU(10,9) /'                   '/
      DATA TMENU(10,18)/'                   '/

      DATA TMENU(11,1) /'1. ADD LINK        '/
      DATA TMENU(11,10)/'                   '/
      DATA TMENU(11,2) /'2. DELETE LINK     '/
      DATA TMENU(11,11)/'                   '/
      DATA TMENU(11,3) /'3. INQUIRE LINK    '/
      DATA TMENU(11,12)/'                   '/
      DATA TMENU(11,4) /'4. RETURN          '/
      DATA TMENU(11,13)/'                   '/
      DATA TMENU(11,5) /'                   '/
      DATA TMENU(11,14)/'                   '/
      DATA TMENU(11,6) /'                   '/
      DATA TMENU(11,15)/'                   '/
      DATA TMENU(11,7) /'                   '/
      DATA TMENU(11,16)/'                   '/
      DATA TMENU(11,8) /'                   '/
      DATA TMENU(11,17)/'                   '/
      DATA TMENU(11,9) /'                   '/
      DATA TMENU(11,18)/'                   '/

      DATA TMENU(12,1) /'1. TOWARDS AN      '/
```

```
DATA TMENU(12,10)/'   EXISTING JOINT  '/
DATA TMENU(12,2) /'2. TOWARDS AN      '/
DATA TMENU(12,11)/'   EXISTING POINT  '/
DATA TMENU(12,3) /'                   '/
DATA TMENU(12,12)/'                   '/
DATA TMENU(12,4) /'                   '/
DATA TMENU(12,13)/'                   '/
DATA TMENU(12,5) /'                   '/
DATA TMENU(12,14)/'                   '/
DATA TMENU(12,6) /'                   '/
DATA TMENU(12,15)/'                   '/
DATA TMENU(12,7) /'                   '/
DATA TMENU(12,16)/'                   '/
DATA TMENU(12,8) /'                   '/
DATA TMENU(12,17)/'                   '/
DATA TMENU(12,9) /'                   '/
DATA TMENU(12,18)/'                   '/

DATA TMENU(13,1) /'1. ZOOM IN         '/
DATA TMENU(13,10)/'                   '/
DATA TMENU(13,2) /'2. RESTORE         '/
DATA TMENU(13,11)/'   ORIGINAL VIEW   '/
DATA TMENU(13,3) /'3. CHANGE JOINT    '/
DATA TMENU(13,12)/'   SCALE           '/
DATA TMENU(13,4) /'4. JOINT INVISIBLE '/
DATA TMENU(13,13)/'                   '/
DATA TMENU(13,5) /'5. LINK INVISIBLE  '/
DATA TMENU(13,14)/'                   '/
DATA TMENU(13,6) /'6. RESTORE         '/
DATA TMENU(13,15)/'   INVISIBLE       '/
DATA TMENU(13,7) /'7. RETURN          '/
DATA TMENU(13,16)/'                   '/
DATA TMENU(13,8) /'                   '/
DATA TMENU(13,17)/'                   '/
DATA TMENU(13,9) /'                   '/
DATA TMENU(13,18)/'                   '/

DATA TMENU(14,1) /'1. ADD I.C.        '/
DATA TMENU(14,10)/'                   '/
DATA TMENU(14,2) /'2. DELETE I.C.     '/
DATA TMENU(14,11)/'                   '/
DATA TMENU(14,3) /'3. INQUIRE I.C.    '/
DATA TMENU(14,12)/'                   '/
DATA TMENU(14,4) /'4. RETURN          '/
DATA TMENU(14,13)/'                   '/
DATA TMENU(14,5) /'                   '/
DATA TMENU(14,14)/'                   '/
DATA TMENU(14,6) /'                   '/
DATA TMENU(14,15)/'                   '/
DATA TMENU(14,7) /'                   '/
DATA TMENU(14,16)/'                   '/
DATA TMENU(14,8) /'                   '/
DATA TMENU(14,17)/'                   '/
DATA TMENU(14,9) /'                   '/
DATA TMENU(14,18)/'                   '/

DATA TMENU(15,1) /'1. ENTER I.C.      '/
DATA TMENU(15,10)/'                   '/
DATA TMENU(15,2) /'2. RETURN          '/
DATA TMENU(15,11)/'                   '/
DATA TMENU(15,3) /'                   '/
DATA TMENU(15,12)/'                   '/
DATA TMENU(15,4) /'                   '/
DATA TMENU(15,13)/'                   '/
DATA TMENU(15,5) /'                   '/
DATA TMENU(15,14)/'                   '/
DATA TMENU(15,6) /'                   '/
DATA TMENU(15,15)/'                   '/
DATA TMENU(15,7) /'                   '/
DATA TMENU(15,16)/'                   '/
DATA TMENU(15,8) /'                   '/
DATA TMENU(15,17)/'                   '/
DATA TMENU(15,9) /'                   '/
DATA TMENU(15,18)/'                   '/

DATA TMENU(16,1) /'1. POSITION        '/
DATA TMENU(16,10)/'                   '/
DATA TMENU(16,2) /'2. VELOCITY        '/
DATA TMENU(16,11)/'                   '/
DATA TMENU(16,3) /'3. ACCELERATION    '/
DATA TMENU(16,12)/'                   '/
DATA TMENU(16,4) /'                   '/
DATA TMENU(16,13)/'                   '/
DATA TMENU(16,5) /'                   '/
DATA TMENU(16,14)/'                   '/
DATA TMENU(16,6) /'                   '/
DATA TMENU(16,15)/'                   '/
DATA TMENU(16,7) /'                   '/
DATA TMENU(16,16)/'                   '/
DATA TMENU(16,8) /'                   '/
DATA TMENU(16,17)/'                   '/
DATA TMENU(16,9) /'                   '/
DATA TMENU(16,18)/'                   '/
```

```
      DATA TMENU(17,1) /'1. NEW MECHANISM    '/
      DATA TMENU(17,10)/'                    '/
      DATA TMENU(17,2) /'2. QUIT             '/
      DATA TMENU(17,11)/'                    '/
      DATA TMENU(17,3) /'                    '/
      DATA TMENU(17,12)/'                    '/
      DATA TMENU(17,4) /'                    '/
      DATA TMENU(17,13)/'                    '/
      DATA TMENU(17,5) /'                    '/
      DATA TMENU(17,14)/'                    '/
      DATA TMENU(17,6) /'                    '/
      DATA TMENU(17,15)/'                    '/
      DATA TMENU(17,7) /'                    '/
      DATA TMENU(17,16)/'                    '/
      DATA TMENU(17,8) /'                    '/
      DATA TMENU(17,17)/'                    '/
      DATA TMENU(17,9) /'                    '/
      DATA TMENU(17,18)/'                    '/

      DATA TMENU(18,1) /'1. IMP              '/
      DATA TMENU(18,10)/'                    '/
      DATA TMENU(18,2) /'2. RSCR             '/
      DATA TMENU(18,11)/'                    '/
      DATA TMENU(18,3) /'3. RSSR             '/
      DATA TMENU(18,12)/'                    '/
      DATA TMENU(18,4) /'4. WRITE NO         '/
      DATA TMENU(18,13)/'    ANALYSIS FILE   '/
      DATA TMENU(18,5) /'                    '/
      DATA TMENU(18,14)/'                    '/
      DATA TMENU(18,6) /'                    '/
      DATA TMENU(18,15)/'                    '/
      DATA TMENU(18,7) /'                    '/
      DATA TMENU(18,16)/'                    '/
      DATA TMENU(18,8) /'                    '/
      DATA TMENU(18,17)/'                    '/
      DATA TMENU(18,9) /'                    '/
      DATA TMENU(18,18)/'                    '/


      DATA DATA/32,0,0,        2,2,1,1,
     &                        1,1,1,1,1,1,
     &                        1,1,1,1,1,1,
     &                        1,1,1,1,1,1,
     &                        1,1,1,1,1,1,
     &                          1,1,1,1/
C                             FLAG FOR UPDATE
      DATA PRFORM/2/
C                             INCLUSION CLASS LIST
      DATA ACLASS/1/

C                             DEFINE TEXT STARTING POSITION
      POSN(1)=0.05
      POSN(2)=2.3

*********************************************************************
*  OPEN STRUCTURE FOR MENU ITEMS
*********************************************************************
      CALL GPEST(6)
C                             OPEN STRUCTURE 6
      CALL GPOPST(6)
C                             INSERT CLASS NAME TO ALLOW PICKING MENU ITEM
      CALL GPADCN(1,ACLASS)
C                             SET TEXT COLOR TO YELLOW
      CALL GPTXCI(2)
C                             SET ANNOTATION SIZE SCALE FACTOR
      CALL GPAHSC(1.00)

      DO 100 I=1,9
C                             SET PICK IDENTIFIER
         CALL GPPKID(I)
C                             DRAW TEXT
         CALL GPAN2(POSN,LENG,TMENU(NUM,I))
C                             SET NEXT POSITION
         POSN(2)=POSN(2)-.08
C                             DRAW TEXT
         CALL GPAN2(POSN,LENG,TMENU(NUM,I+9))
C                             SET NEXT POSITION
         POSN(2)=POSN(2)-.18
  100 CONTINUE
C                             CLOSE STRUCTURE
      CALL GPCLST


*********************************************************************
*  SCREEN DISPLAY
*********************************************************************
C                             ASSOCIATE THE STRUCTURE 6 WITH A VIEW 3
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
C                             TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)


*********************************************************************
*  REQUEST PICK INPUT
```

```
C                              SET PICK FILTER(ALL CLASS 1 DETECTABLE)
      CALL GPPKF(IWSID,1,1,ACLASS,0,ACLASS)
C                              MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                              DEFINE PICK AREA
      PAREA(1)=((-.98+1)/2)*CSIZE(1)
      PAREA(2)=((-.525+1)/2)*CSIZE(1)
      PAREA(3)=((-.95+1)/2)*CSIZE(2)
      PAREA(4)=((.33+1)/2)*CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=CSIZE(3)
C                              INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                              PLACE PICK DEVICE IN THE EVENT MODE
      CALL GPPKMO(IWSID,1,3,2)
C                              TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)

C*********************************************************************
*  AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
C*********************************************************************
C                              AWAIT AN EVENT
  200 CALL GPAWEV(1000.,IWS,ICLA,IDEV)
      IF(ICLA.EQ.5)THEN
C                              GET PICK
         CALL GPGTPK(1,1,PPATH)
C                              CHOICE IS SECOND ITEM OF PICK PATH
         CHOICE=PPATH(2)
C                              DEACTIVATE PICK
         CALL GPPKMO(IWSID,1,1,2)
C                              DELETE MENU STRUCTURES
         CALL GPEST(6)
C        CALL GPDLST(6)
C                              UPDATE THE WORKSTATION
         CALL GPUPWS(IWSID,PRFORM)
C                              RETURN SETUP
         GO TO 1000
      ELSE
C                              IF NOT A PICK, GO TO AWAIT AN EVENT
         GO TO 200
      ENDIF

C                              MAKE SURE PICK IS DEACTIVATED
 1000 CALL GPPKMO(IWSID,1,1,2)


      RETURN
      END
```

SUBROUTINE NSTART

```
      SUBROUTINE NSTART(IWSID,FILE,CSIZE,AXISR,POINT,JOINT,LINK,IC,
     &PNUM,JNUM,LNUM,INUM)
C*********************************************************************
* ----- SUBROUTINE NSTART -----                                       *
*                                                                     *
*                                                                     *
*   THIS SUBROUTINE WILL START A MECHIN INPUT FILE.  THE USER WILL    *
*                                                                     *
*   BE ASKED TO INPUT THE NEW FILE NAME AND THE RANGE OF THE          *
*                                                                     *
*   AXES.                                                             *
*                                                                     *
C*********************************************************************

      INTEGER IWSID,PNUM(100),JNUM(20,3),LNUM(30,14),INUM(20,2),
     &        DATA(35),LENG,ACLASS(1)
      REAL AREA(6),CSIZE(3),AXISR,JOINT(20,21),LINK(30,6),IC(20,4),
     &     POINT(100,3),POSN(2),PAREA(6)
      CHARACTER*7 FILE,AXIS,CLEAR,FILEO,INIT
      CHARACTER*19 FLINE
      CHARACTER*16 RETURN
      CHARACTER*50 CLEAR1

      DATA RETURN/'1.  QUIT        '/
      DATA LENG/16/
      DATA POSN/0.05,2.2/


      DATA DATA/32,0,0,        2,2,1,1,
     &                       1,1,1,1,1,1,
     &                       1,1,1,1,1,1,
     &                       1,1,1,1,1,1,
     &                       1,1,1,1,1,1,
     &                         1,1,1,1/
C                            INCLUSION CLASS LIST
```

```
      DATA ACLASS/2/


      DATA INIT/'1.0     '/
      DATA AXIS/'        '/
      DATA CLEAR/'       '/
      DATA CLEAR1/'                                                    '/
      DATA FLINE/'MECHIN RESTART FILE'/

      FILE='       '

      CRET=0

      AREA(1)=0.08*CSIZE(1)
      AREA(2)=0.90*CSIZE(1)
      AREA(3)=0.70*CSIZE(1)
      AREA(4)=0.90*CSIZE(1)
      AREA(5)=0.02*CSIZE(1)
      AREA(6)=0.90*CSIZE(1)

*************************************************************************
*  INITIALIZE PICK
*************************************************************************
C                             SET PICK FILTER(ALL CLASS 2 DETECTABLE)
      CALL GPPKF(IWSID,1,1,ACLASS,0,ACLASS)
C                             MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                             DEFINE PICK AREA
      PAREA(1)=((-.98+1)/2)*CSIZE(1)
      PAREA(2)=((.98+1)/2)*CSIZE(1)
      PAREA(3)=((-.98+1)/2)*CSIZE(2)
      PAREA(4)=((.48+1)/2)*CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=CSIZE(3)
C                             INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                             PLACE PICK DEVICE IN THE EVENT MODE
      CALL GPPKMO(IWSID,1,3,2)
C                             TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,2)

*************************************************************************
*  INCLUDE A RETURN PICK
*************************************************************************

C                             RETURN PICK
      CALL GPOPST(6)
      CALL GPADCN(1,ACLASS)
      CALL GPPKID(101)
      CALL GPTXCI(2)
      CALL GPAHSC(1.0)
      CALL GPAN2(POSN,LENG,RETURN)
      CALL GPCLST
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
      CALL GPUPWS(IWSID,2)

*************************************************************************
*  INITIALIZE THE STRING DEVICE.
*************************************************************************
    2 CALL GPSTMO(IWSID,1,1,2)
C                             INITIALIZE THE STRING DEVICE
      CALL GPINST(IWSID,1,7,CLEAR,1,AREA,7,1,0,CLEAR)
C                             PLACE STRING DEVICE IN THE EVENT MODE
      CALL GPSTMO(IWSID,1,3,2)
C                             REQUEST THE FILE NAME
      CALL CMMAND(IWSID,4)


*************************************************************************
*  GO TO CLOSE IF QUIT IS PICKED
*************************************************************************
C                             AWAIT AN EVENT
    3 CALL GPAWEV(1000.,IWS,ICLA,IDEV)
      IF(ICLA.EQ.5)THEN
C                             CLOSE FOR RESTART
         CALL GPDAST
         CALL GPCLPH
         STOP
      ENDIF

*************************************************************************
*  PROCESS SRTING EVENT. OPEN MECHIN FILE
*************************************************************************
      L=7
C                             GET STRING
      CALL GPGTST(L,LR,FILE)
      IF(FILE.EQ.'       ')GO TO 3
C                             SEE IF FILE ALREADY EXISTS
      CALL FILEXS(FILE,IRET)
C                             IF FILE EXISTS, HAVE USER RETYPE TO USE
      IF(IRET.EQ.0)THEN
         CALL GPSTMO(IWSID,1,1,2)
         FILEO=FILE
         FILE=CLEAR
```

```
         CALL CMMAND(IWSID,8)
         CALL GPSTMO(IWSID,1,1,2)
         CALL GPINST(IWSID,1,7,CLEAR,1,AREA,7,1,0,CLEAR)
         CALL GPRQST(IWSID,1,7,ISTAT,INUM,FILE)
         IF(FILE.EQ.FILEO)THEN
            CONTINUE
         ELSE
            CALL GPINST(IWSID,1,7,CLEAR,1,AREA,7,1,0,CLEAR)
            CALL GPSTMO(IWSID,1,3,2)
            GO TO 3
         ENDIF
      ENDIF
C                              OPEN A NEW FILE
      OPEN(UNIT=15,FILE=FILE,STATUS='NEW',ERR=2)
      CALL GPEST(6)
      CALL GPPKMO(IWSID,1,1,2)

*********************************************************************
*  INITIALIZE THE ARRAYS
*********************************************************************

      DO 11 I=1,20
         DO 10 J=1,21
            JOINT(I,J)=0.0
   10    CONTINUE
   11 CONTINUE

      DO 13 I=1,100
         DO 12 J=1,3
            POINT(I,J)=0.0
   12    CONTINUE
   13 CONTINUE

      DO 15 I=1,30
         DO 14 J=1,6
            LINK(I,J)=0.0
   14    CONTINUE
   15 CONTINUE

      DO 17 I=1,20
         DO 16 J=1,4
            IC(I,J)=0.0
   16    CONTINUE
   17 CONTINUE

      DO 18 I=1,100
         PNUM(I)=0
   18 CONTINUE

      DO 20 I=1,20
         DO 19 J=1,3
            JNUM(I,J)=0
   19    CONTINUE
   20 CONTINUE

      DO 22 I=1,30
         DO 21 J=1,14
            LNUM(I,J)=0
   21    CONTINUE
   22 CONTINUE

      DO 24 I=1,20
         DO 23 J=1,2
            INUM(I,J)=0
   23    CONTINUE
   24 CONTINUE

*********************************************************************
*  WRITE FILE NAME TO THE SCREEN
*********************************************************************
C                              CALL SUBROUTINE TO WRITE FILENAME
      CALL FILENM(IWSID,FILE)


*********************************************************************
*  REQUEST THE COORDINATE AXIS RANGE
*********************************************************************
C                              COMMAND FOR COORDINATE AXIS RANGE
  500 CALL CMMAND(IWSID,5)
      CALL GPSTMO(IWSID,1,1,2)
C                              INITIALIZE THE STRING DEVICE FOR AXIS
      CALL GPINST(IWSID,1,7,INIT,1,AREA,7,1,0,CLEAR)
C                              REQUEST STRING FOR AXIS
      CALL GPRQST(IWSID,1,7,ISTAT,NUM,AXIS)
      WRITE(10,800)CLEAR1
  800 FORMAT(A50)
      REWIND 10
      WRITE(10,1000)AXIS
 1000 FORMAT(A7)
      REWIND 10
      READ(10,*,ERR=500)AXISR
C                              REQUEST THE COORDINATE AXIS RANGE
      CALL AXES(IWSID,AXISR)

 3000 CALL GPSTMO(IWSID,1,1,2)
```

```
      CALL GPPKF(IWSID,1,0,ACLASS,1,ACLASS)
      RETURN
      END
```

SUBROUTINE ORIEN

```
      SUBROUTINE ORIEN(IWSID,JPOSN,POINT,JOINT,AXISR,CSIZE,ORIENT,RFLG)
*************************************************************************
* ----- SUBROUTINE ORIEN -----                                          *
*                                                                       *
*                                                                       *
*   THIS SUBROUTINE IS USED TO ORIENT A VECTOR IN SPACE                 *
*                                                                       *
*************************************************************************

      INTEGER IWSID,ECHOSW,EVENT,DLEN,ECHO,ACLASS,PRFORM,CHOICE,
     & PPATH(3),RFLG,NUMV

      REAL CSIZE(3),AREA1(6),AREA2(6),AREA3(6),VLUE1,LOVAL,HIVAL,AXISR,
     & BOX(8),SHD(8),ORIENT(3),MATRIX(1000,4),MATUSE(4,4),LINE(6),
     & LINEX(6),LINEY(6),LINEZ(6),XBOX(15),JPOSN(3),POINT(100,3),
     & YBOX(15),ZBOX(15),LX(6),LY(6),LZ(6),OL(6),AREA(6),JOINT(20,21)

      CHARACTER*7 CLEAR,INIT

      DATA CLEAR/'       '/
      DATA INIT /'0.     '/

      DATA LX/0.,0.,0.,0.,0.,0./
      DATA LY/0.,0.,0.,0.,0.,0./
      DATA LZ/0.,0.,0.,0.,0.,0./

      DATA OL/0.,0.,0.,0.,0.,0./

C                              FLAG FOR UPDATE
      DATA PRFORM/2/

C                              INCLUSION CLASS LIST
      DATA ACLASS/1/
      DATA VLUE1/0./
      DATA ECHO/3/
      DATA DLEN/0/
      DATA EVENT/3/
      DATA ECHOSW/2/

C                              GREY VALUATOR BOX
      DATA BOX/0.00,0.545,0.48,0.545,0.48,0.73,0.00,0.73/

      NUMV=1

      LOVAL=-1.0
      HIVAL=1.0

      AREA(1)=.08*CSIZE(1)
      AREA(2)=.90*CSIZE(1)
      AREA(3)=.70*CSIZE(2)
      AREA(4)=.90*CSIZE(2)
      AREA(5)=.02*CSIZE(1)
      AREA(6)=.90*CSIZE(3)

      AREA1(1)=.52*CSIZE(1)
      AREA1(2)=CSIZE(1)
      AREA1(3)=.855*CSIZE(2)-.006
      AREA1(4)=CSIZE(2)
      AREA1(5)=0.
      AREA1(6)=CSIZE(3)

      AREA2(1)=.52*CSIZE(1)
      AREA2(2)=CSIZE(1)
      AREA2(3)=.855*CSIZE(2)-.014
      AREA2(4)=CSIZE(2)
      AREA2(5)=0.
      AREA2(6)=CSIZE(3)

      AREA3(1)=.52*CSIZE(1)
      AREA3(2)=CSIZE(1)
      AREA3(3)=.855*CSIZE(2)-.022
      AREA3(4)=CSIZE(2)
      AREA3(5)=0.
      AREA3(6)=CSIZE(3)


      DO 100 I=2,8,2
         SHD(I)=BOX(I)-0.025
  100 CONTINUE

      DO 200 I=1,7,2
         SHD(I)=BOX(I)+0.025
  200 CONTINUE
```

```
*************************************************************************
*  OPEN STRUCTURE FOR THE MENU BOX (VIEW 1)
*************************************************************************
C                               OPEN A STRUCTURE
      CALL GPOPST(8)
C                               SET SOLID INTEROR STYLE FOR GREY BOX
      CALL GPIS(2)
C                               SET COLOR FOR INTERIOR OF SHADOW BOX
      CALL GPICI(4)
C                               DRAW GREY SHADOW BOX
      CALL GPPG2(1,4,2,SHD)
C                               SET COLOR FOR INTERIOR OF GREY BOX
      CALL GPICI(3)
C                               DRAW GREY TITLE BOX
      CALL GPPG2(1,4,2,BOX)
C                               CLOSE STRUCTURE
      CALL GPCLST

*************************************************************************
*  SCREEN DISPLAY
*************************************************************************
C                               UPDATE THE WORKSTATION
      CALL GPUPWS(IWSID,PRFORM)

*************************************************************************
*  CHOOSE METHOD OF JOINT ORIENTATION
*************************************************************************

  300 CALL CMMAND(IWSID,30)
      CALL MENU(IWSID,CSIZE,7,CHOICE)

      IF(CHOICE.EQ.1)THEN
C                               DIRECT ENTRY
         CALL DEO(IWSID,AXISR,CSIZE,ORIENT,RFLG)
         IF(RFLG.EQ.1)THEN
C                               REENTER POINT
              ORIENT(1)=0.
              ORIENT(2)=0.
              ORIENT(3)=0.
              RFLG=0
              GO TO 300
         ENDIF
      ENDIF

      IF(CHOICE.EQ.2)THEN
C                               VALUATOR ENTRY
         CALL VEO(IWSID,AXISR,CSIZE,ORIENT,RFLG)
         IF(RFLG.EQ.1)THEN
C                               REENTER POINT
              ORIENT(1)=0.
              ORIENT(2)=0.
              ORIENT(3)=0.
              RFLG=0
              GO TO 300
         ENDIF
      ENDIF

      IF(CHOICE.EQ.3)THEN
C                               TOWARDS AN EXISTING POINT
         CALL TEPO(IWSID,JPOSN,POINT,JOINT,AXISR,CSIZE,ORIENT,RFLG)
         IF(RFLG.EQ.1)THEN
C                               REENTER POINT
              ORIENT(1)=0.
              ORIENT(2)=0.
              ORIENT(3)=0.
              RFLG=0
              GO TO 300
         ENDIF
      ENDIF

      IF(CHOICE.EQ.4)THEN
C                               PARALLEL WITH AN EXISTING JOINT
         CALL PEJO(IWSID,JPOSN,JOINT,AXISR,CSIZE,ORIENT,RFLG)
         IF(RFLG.EQ.1)THEN
C                               REENTER POINT
              ORIENT(1)=0.
              ORIENT(2)=0.
              ORIENT(3)=0.
              RFLG=0
              GO TO 300
         ENDIF
      ENDIF

      IF(CHOICE.EQ.5)THEN
C                               RETURN WITH NO ORIENTATION
         RFLG=1
      ENDIF

 1000 CALL GPEST(8)
```

```
      CALL GPEST(6)
      CALL GPUPWS(IWSID,2)

      RETURN
      END
```

SUBROUTINE PEJO

```
      SUBROUTINE PEJO(IWSID,JPOSN,JOINT,AXISR,CSIZE,LOCAT,RFLG)
*********************************************************************
* ----- SUBROUTINE PEJO  -----                                      *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE WILL PROMPT FOR THE ENTRY OF JOINT              *
*                                                                   *
*   ORIENTATION COORDINATES PARALLEL WITH AN EXISTING JOINT         *
*                                                                   *
*********************************************************************

      INTEGER IWSID,PRFORM,PPATH(3),ACLASS(2),CHOICE,PNUM(100),LENG,
     &RFLG

      REAL CSIZE(3),DATA(35),PAREA(6),L1(4),JOINT(20,21),AXISR,POSN(2),
     &P(3),P1(3),VEC(3),MAG,LOCAT(3),JPOSN(3),
     &LX(2),LX1(2),LY(2),LY1(2),LZ(2),LZ1(2)

      CHARACTER*7 PNAM(100)
      CHARACTER*20 X,Y,Z,CLEAR
      CHARACTER*16 XVAL,YVAL,ZVAL,RETURN

      DATA RETURN/'1.  RETURN      '/
      DATA LENG/16/
      DATA POSN/0.05,2.2/


      DATA DATA/32,0,0,          2,2,1,1,
     &                         1,1,1,1,1,1,
     &                         1,1,1,1,1,1,
     &                         1,1,1,1,1,1,
     &                         1,1,1,1,1,1,
     &                           1,1,1,1/
C                             FLAG FOR UPDATE
      DATA PRFORM/2/
C                             INCLUSION CLASS LIST(CLASS 3 -- JOINTS)
      DATA ACLASS/3,4/

      DATA X      /'DX=                '/
      DATA Y      /'DY=                '/
      DATA Z      /'DZ=                '/
      DATA CLEAR /'                    '/


      LX(1)=0.05
      LX(2)=0.68

      LX1(1)=0.14
      LX1(2)=0.68

      LY(1)=0.05
      LY(2)=0.62

      LY1(1)=0.14
      LY1(2)=0.62

      LZ(1)=0.05
      LZ(2)=0.56

      LZ1(1)=0.14
      LZ1(2)=0.56


*********************************************************************
*  REQUEST PICK INPUT
*********************************************************************
C                             SET PICK FILTER(ALL CLASS 3 DETECTABLE)
      CALL GPPKF(IWSID,1,2,ACLASS,0,ACLASS)
C                             MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                             DEFINE PICK AREA
      PAREA(1)=((-.98+1)/2)*CSIZE(1)
      PAREA(2)=((.98+1)/2)*CSIZE(1)
      PAREA(3)=((-.98+1)/2)*CSIZE(2)
      PAREA(4)=((.48+1)/2)*CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=CSIZE(3)
C                             INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                             PLACE PICK DEVICE IN THE EVENT MODE
```

```
      CALL GPPKMO(IWSID,1,3,2)
C                              TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)
C                              GET PICK OR RETURN COMMAND
  200 CALL CMMAND(IWSID,35)

*************************************************************************
*  INCLUDE A RETURN PICK
*************************************************************************

C                              RETURN PICK
      CALL GPOPST(6)
      CALL GPADCN(1,ACLASS)
      CALL GPPKID(101)
      CALL GPTXCI(2)
      CALL GPAHSC(1.0)
      CALL GPAN2(POSN,LENG,RETURN)
      CALL GPCLST
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
      CALL GPUPWS(IWSID,PRFORM)

*************************************************************************
*  AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
*************************************************************************
C                              AWAIT AN EVENT
      CALL GPAWEV(1000.,IWS,ICLA,IDEV)
      IF(ICLA.EQ.5)THEN
C                              GET PICK
         CALL GPGTPK(1,1,PPATH)
C                              IF RETURN
         IF(PPATH(2).EQ.101)THEN
              RFLG=1
              GO TO 1000
         ENDIF
C                              ERASE RETURN PROMPT
         CALL GPEST(6)
C                              CHOICE IS SECOND ITEM OF PICK PATH
         CHOICE=PPATH(2)
         ISTRUC=PPATH(1)
C                              GET ORIENT COORDINATES
         IF(ISTRUC.EQ.18)THEN
              LOCAT(1)=JOINT(CHOICE,18)
              LOCAT(2)=JOINT(CHOICE,19)
              LOCAT(3)=JOINT(CHOICE,20)
         ELSE
              LOCAT(1)=JOINT(CHOICE,4)
              LOCAT(2)=JOINT(CHOICE,5)
              LOCAT(3)=JOINT(CHOICE,6)
         ENDIF
C                              DRAW ORIENTATION ICON
         CALL GPEST(16)

         CALL GPOPST(16)
         CALL ICONO(LOCAT,AXISR)
         CALL GPCLST
C                              DISPLAY VALUES

         REWIND 10
         DO 350 III=1,3
              WRITE(10,300)CLEAR
  300         FORMAT(A20)
  350    CONTINUE
         REWIND 10
         WRITE(10,375)LOCAT(1)
         WRITE(10,375)LOCAT(2)
         WRITE(10,375)LOCAT(3)
  375    FORMAT(F16.14)
         REWIND 10
         READ(10,400)XVAL
         READ(10,400)YVAL
         READ(10,400)ZVAL
  400    FORMAT(A14)


         CALL GPOPST(10)
         CALL GPTXCI(1)
         CALL GPAHSC(1.0)
         CALL GPAN2(LX,20,X)
         CALL GPAN2(LX1,16,XVAL)
         CALL GPAN2(LY,20,Y)
         CALL GPAN2(LY1,16,YVAL)
         CALL GPAN2(LZ,20,Z)
         CALL GPAN2(LZ1,16,ZVAL)
         CALL GPCLST
C                              ENTER ORIENTATION OR RETURN
         CALL CMMAND(IWSID,32)
         CALL MENU(IWSID,CSIZE,8,CHOICE)

         IF(CHOICE.EQ.1)THEN
              RFLG=0
              GO TO 1000
         ENDIF
```

```
         IF(CHOICE.EQ.2)THEN
              RFLG=1
              LOCAT(1)=0.
              LOCAT(2)=0.
              LOCAT(3)=0.
              GO TO 1000
         ENDIF

      ELSE
C                              IF NOT A PICK, GO TO AWAIT AN EVENT
         GO TO 200
      ENDIF

C                              MAKE SURE PICK IS DEACTIVATED
 1000 CALL GPPKMO(IWSID,1,1,2)
      CALL GPEST(6)
      CALL GPEST(10)
      CALL GPEST(16)

C                              SET PICK FILTER(ALL CLASS 2 UNDETECTABLE)
      CALL GPPKF(IWSID,1,0,ACLASS,2,ACLASS)

      RETURN
      END
```

SUBROUTINE PICKJ

```
      SUBROUTINE PICKJ(IWSID,JOINT,AXISR,CSIZE,END,ENDP,IAXIS,RFLG)
*************************************************************************
* ----- SUBROUTINE PICKJ -----                                          *
*                                                                       *
*                                                                       *
*   THIS SUBROUTINE WILL PROMPT FOR THE ENTRY OF JOINT                  *
*                                                                       *
*   AT THE END OF A LINK                                                *
*                                                                       *
*************************************************************************

      INTEGER IWSID,PRFORM,PPATH(3),ACLASS(1),CHOICE,LENG,END,IAXIS,
     &RFLG

      REAL CSIZE(3),DATA(35),PAREA(6),L1(4),JOINT(20,21),AXISR,POSN(2),
     &P(3),P1(3),VEC(3),MAG,LOCAT(3),JPOSN(3),ENDP(3),
     &LX(2),LX1(2),LY(2),LY1(2),LZ(2),LZ1(2)

      CHARACTER*20 X,Y,Z,CLEAR
      CHARACTER*16 XVAL,YVAL,ZVAL,RETURN

      DATA RETURN/'1.  RETURN      '/
      DATA LENG/16/
      DATA POSN/0.05,2.2/

      DATA DATA/32,0,0,          2,2,1,1,
     &                         1,1,1,1,1,1,
     &                         1,1,1,1,1,1,
     &                         1,1,1,1,1,1,
     &                         1,1,1,1,1,1,
     &                           1,1,1,1/
C                              FLAG FOR UPDATE
      DATA PRFORM/2/
C                              INCLUSION CLASS LIST(CLASS 4 -- JOINTS AXIS
      DATA ACLASS/4/

*************************************************************************
*  REQUEST PICK INPUT
*************************************************************************
C                              SET PICK FILTER(ALL CLASS 3,4 DETECTABLE)
      CALL GPPKF(IWSID,1,1,ACLASS,0,ACLASS)
C                              MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                              DEFINE PICK AREA
      PAREA(1)=((-.98+1)/2)*CSIZE(1)
      PAREA(2)=((.98+1)/2)*CSIZE(1)
      PAREA(3)=((-.98+1)/2)*CSIZE(2)
      PAREA(4)=((.48+1)/2)*CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=CSIZE(3)
C                              INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                              PLACE PICK DEVICE IN THE EVENT MODE
      CALL GPPKMO(IWSID,1,3,2)
C                              TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)
C                              GET PICK OR RETURN COMMAND
```

```
*********************************************************************
*  INCLUDE A RETURN PICK
*********************************************************************

C                              RETURN PICK
  200 CALL GPOPST(6)
      CALL GPADCN(1,ACLASS)
      CALL GPPKID(101)
      CALL GPTXCI(2)
      CALL GPAHSC(1.0)
      CALL GPAN2(POSN,LENG,RETURN)
      CALL GPCLST
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
      CALL GPUPWS(IWSID,PRFORM)

*********************************************************************
*  AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
*********************************************************************
C                              AWAIT AN EVENT
      CALL GPAWEV(1000.,IWS,ICLA,IDEV)
      IF(ICLA.EQ.5)THEN
C                              GET PICK
         CALL GPGTPK(1,1,PPATH)
C                              IF RETURN
         IF(PPATH(2).EQ.101)THEN
              RFLG=1
              GO TO 1000
         ENDIF
C                              ERASE RETURN PROMPT
         CALL GPEST(6)
C                              CHOICE IS SECOND ITEM OF PICK PATH
         END=PPATH(2)
         ISTRUC=PPATH(1)
         IF(ISTRUC.EQ.17)THEN
              IAXIS=1
              ENDP(1)=JOINT(END,1)
              ENDP(2)=JOINT(END,2)
              ENDP(3)=JOINT(END,3)
         ENDIF
         IF(ISTRUC.EQ.18)THEN
              IAXIS=2
              ENDP(1)=JOINT(END,1)
              ENDP(2)=JOINT(END,2)
              ENDP(3)=JOINT(END,3)
         ENDIF
      ENDIF
C                              MAKE SURE PICK IS DEACTIVATED
 1000 CALL GPPKMO(IWSID,1,1,2)
      CALL GPEST(6)

C                              SET PICK FILTER(ALL CLASS 2 UNDETECTABLE)
      CALL GPPKF(IWSID,1,0,ACLASS,1,ACLASS)

      RETURN
      END
```

SUBROUTINE PICKJ2

```
      SUBROUTINE PICKJ2(IWSID,JOINT,AXISR,CSIZE,END,ENDP,IAXIS,RFLG)
*********************************************************************
* ----- SUBROUTINE PICKJ2 -----                                      *
*                                                                    *
*                                                                    *
*   THIS SUBROUTINE WILL PROMPT FOR THE ENTRY OF JOINT               *
*                                                                    *
*   AT THE END OF A LINK                                             *
*                                                                    *
*********************************************************************

      INTEGER IWSID,PRFORM,PPATH(3),ACLASS(2),CHOICE,LENG,END,IAXIS,
     &RFLG

      REAL CSIZE(3),DATA(35),PAREA(6),L1(4),JOINT(20,21),AXISR,POSN(2),
     &P(3),P1(3),VEC(3),MAG,LOCAT(3),JPOSN(3),ENDP(3),
     &LX(2),LX1(2),LY(2),LY1(2),LZ(2),LZ1(2)

      CHARACTER*20 X,Y,Z,CLEAR
      CHARACTER*16 XVAL,YVAL,ZVAL,RETURN

      DATA RETURN/'1.  RETURN      '/
      DATA LENG/16/
      DATA POSN/0.05,2.2/


      DATA DATA/32,0,0,        2,2,1,1,
     &                        1,1,1,1,1,1,
```

```
     &                          1,1,1,1,1,1,
     &                          1,1,1,1,1,1,
     &                          1,1,1,1,1,1,
     &                            1,1,1,1/
C                               FLAG FOR UPDATE
C     DATA PRFORM/2/
C                               INCLUSION CLASS LIST(CLASS 4 -- JOINTS AXIS
      DATA ACLASS/3,4/

*********************************************************************
*  REQUEST PICK INPUT
*********************************************************************
C                               SET PICK FILTER(ALL CLASS 3,4 DETECTABLE)
      CALL GPPKF(IWSID,1,2,ACLASS,0,ACLASS)
C                               MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                               DEFINE PICK AREA
      PAREA(1)=((-.98+1)/2)*CSIZE(1)
      PAREA(2)=((.98+1)/2)*CSIZE(1)
      PAREA(3)=((-.98+1)/2)*CSIZE(2)
      PAREA(4)=((.48+1)/2)*CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=CSIZE(3)
C                               INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                               PLACE PICK DEVICE IN THE EVENT MODE
      CALL GPPKMO(IWSID,1,3,2)
C                               TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)
C                               GET PICK OR RETURN COMMAND

*********************************************************************
*  INCLUDE A RETURN PICK
*********************************************************************

C                               RETURN PICK
  200 CALL GPOPST(6)
      CALL GPADCN(1,ACLASS)
      CALL GPPKID(101)
      CALL GPTXCI(2)
      CALL GPAHSC(1.0)
      CALL GPAN2(POSN,LENG,RETURN)
      CALL GPCLST
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
      CALL GPUPWS(IWSID,PRFORM)

*********************************************************************
*  AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
*********************************************************************
C                               AWAIT AN EVENT
      CALL GPAWEV(1000.,IWS,ICLA,IDEV)
      IF(ICLA.EQ.5)THEN
C                               GET PICK
         CALL GPGTPK(1,1,PPATH)
C                               IF RETURN
         IF(PPATH(2).EQ.101)THEN
              RFLG=1
              GO TO 1000
         ENDIF
C                               ERASE RETURN PROMPT
         CALL GPEST(6)
C                               CHOICE IS SECOND ITEM OF PICK PATH
         END=PPATH(2)
         ISTRUC=PPATH(1)
         IF(ISTRUC.EQ.17)THEN
              IAXIS=1
              ENDP(1)=JOINT(END,1)
              ENDP(2)=JOINT(END,2)
              ENDP(3)=JOINT(END,3)
         ENDIF
         IF(ISTRUC.EQ.18)THEN
              IAXIS=2
              ENDP(1)=JOINT(END,1)
              ENDP(2)=JOINT(END,2)
              ENDP(3)=JOINT(END,3)
         ENDIF
      ENDIF
C                               MAKE SURE PICK IS DEACTIVATED
 1000 CALL GPPKMO(IWSID,1,1,2)
      CALL GPEST(6)

C                               SET PICK FILTER(ALL CLASS 3,4 UNDETECTABLE)
      CALL GPPKF(IWSID,1,0,ACLASS,2,ACLASS)

      RETURN
      END
```

SUBROUTINE PICKL2

```
      SUBROUTINE PICKL2(IWSID,CSIZE,AXISR,LNAM,LNUM,JNAM,END,IRET)
*********************************************************************
* ----- SUBROUTINE PICKL2----                                       *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED TO PICK LINKS                           *
*                                                                   *
*********************************************************************

      INTEGER IWSID,PRFORM,PPATH(3),ACLASS(1),CHOICE,LENG,STRU,
     &        LNUM(30,14),END

      REAL CSIZE(3),DATA(35),PAREA(6),L1(4),POINT(100,3),AXISR,POSN(2),
     & BOX(8),SHD(8),LX(2),LY(2),LZ(2),LX1(2),LY1(2),LZ1(2)

      CHARACTER*7 JNAM(100),LNAM(30),LNAME,JNAME
      CHARACTER*16 RETURN
      CHARACTER*20 X,Y,Z,CLEAR

      DATA RETURN/'1.  RETURN      '/
      DATA LENG/16/
      DATA POSN/0.05,2.2/

      DATA X    /'LINK NAME =          '/
      DATA Y    /'                     '/
      DATA Z    /'                     '/
      DATA CLEAR/'                     '/

      DATA DATA/32,0,0,          2,2,1,1,
     &                         1,1,1,1,1,1,
     &                         1,1,1,1,1,1,
     &                         1,1,1,1,1,1,
     &                         1,1,1,1,1,1,
     &                           1,1,1,1/
C                              FLAG FOR UPDATE
      DATA PRFORM/2/
C                              INCLUSION CLASS LIST
      DATA ACLASS/5/

C                              GREY VALUATOR BOX
      DATA BOX/0.00,0.545,0.48,0.545,0.48,0.73,0.00,0.73/


      LY(1)=0.02
      LY(2)=0.62

      LY1(1)=0.31
      LY1(2)=0.62

      DO 10 I=2,8,2
         SHD(I)=BOX(I)-0.025
   10 CONTINUE

      DO 20 I=1,7,2
         SHD(I)=BOX(I)+0.025
   20 CONTINUE


*********************************************************************
*  REQUEST PICK INPUT
*********************************************************************
C                              SET PICK FILTER(ALL CLASS 2,3 DETECTABLE)
      CALL GPPKF(IWSID,1,1,ACLASS,0,ACLASS)
C                              MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                              DEFINE PICK AREA
      PAREA(1)=((-.98+1)/2)*CSIZE(1)
      PAREA(2)=((.98+1)/2)*CSIZE(1)
      PAREA(3)=((-.98+1)/2)*CSIZE(2)
      PAREA(4)=((.48+1)/2)*CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=CSIZE(3)
C                              INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                              PLACE PICK DEVICE IN THE EVENT MODE
      CALL GPPKMO(IWSID,1,3,2)
C                              TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)
C                              GET INQUIRY OR RETURN COMMAND

*********************************************************************
*  INCLUDE A RETURN PICK
*********************************************************************

C                              RETURN PICK
      CALL GPOPST(6)
      CALL GPADCN(1,ACLASS)
      CALL GPPKID(101)
      CALL GPTXCI(2)
```

```
      CALL GPAHSC(1.0)
      CALL GPAN2(POSN,LENG,RETURN)
      CALL GPCLST
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
      CALL GPUPWS(IWSID,PRFORM)
C                              GREY VALUE BOX AND SHADOW
      CALL GPOPST(8)
      CALL GPIS(2)
      CALL GPICI(4)
      CALL GPPG2(1,4,2,SHD)
      CALL GPICI(3)
      CALL GPPG2(1,4,2,BOX)
      CALL GPCLST
      CALL GPUPWS(IWSID,PRFORM)

*********************************************************************
*  AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
*********************************************************************
C                              AWAIT AN EVENT
  200 CALL GPAWEV(1000.,IWS,ICLA,IDEV)
      IF(ICLA.EQ.5)THEN
C                              ERASE OLD PICK
         CALL GPEST(10)
C                              GET PICK
         CALL GPGTPK(1,IDEP,PPATH)
C                              IF NO PICK, RETURN
         IF(PPATH(2).EQ.101)THEN
            IRET=1
            GO TO 1000
         ENDIF
C                              CHOICE IS SECOND ITEM OF PICK PATH
         END=PPATH(2)

      ELSE
C                              IF NOT A PICK, GO TO AWAIT AN EVENT
         GO TO 200
      ENDIF


C                              MAKE SURE PICK IS DEACTIVATED
 1000 CALL GPPKMO(IWSID,1,1,2)
      CALL GPEST(10)
      CALL GPEST(6)
      CALL GPEST(8)
      CALL GPUPWS(IWSID,PRFORM)

C                              SET PICK FILTER(ALL CLASS 2,3 UNDETECTABLE)
      CALL GPPKF(IWSID,1,0,ACLASS,1,ACLASS)


      RETURN
      END
```

SUBROUTINE POSIT

```
      SUBROUTINE POSIT(IWSID,AXISR,CSIZE,LOCAT,RFLG)
*********************************************************************
* ----- SUBROUTINE POSIT -----                                      *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED TO POSITION A POINT IN SPACE            *
*                                                                   *
*********************************************************************

      INTEGER IWSID,ECHOSW,EVENT,DLEN,ECHO,ACLASS,PRFORM,CHOICE,
     & PPATH(3),RFLG,NUMV

      REAL CSIZE(3),AREA1(6),AREA2(6),AREA3(6),VLUE1,LOVAL,HIVAL,AXISR,
     & BOX(8),SHD(8),LOCAT(3),MATRIX(1000,4),MATUSE(4,4),LINE(6),
     & LINEX(6),LINEY(6),LINEZ(6),XBOX(15),
     & YBOX(15),ZBOX(15),LX(6),LY(6),LZ(6),AREA(6)

      CHARACTER*7 CLEAR,INIT

      DATA CLEAR/'       '/
      DATA INIT /'0.     '/

      DATA LX/0.,0.,0.,0.,0.,0./
      DATA LY/0.,0.,0.,0.,0.,0./
      DATA LZ/0.,0.,0.,0.,0.,0./

C                              FLAG FOR UPDATE
      DATA PRFORM/2/
C                              INCLUSION CLASS LIST
      DATA ACLASS/1/
```

```
      DATA VLUE1/0./
      DATA ECHO/3/
      DATA DLEN/0/
      DATA EVENT/3/
      DATA ECHOSW/2/

C                              GREY VALUATOR BOX
      DATA BOX/0.00,0.545,0.48,0.545,0.48,0.73,0.00,0.73/

      NUMV=1

      LOVAL=-AXISR
      HIVAL=AXISR

      AREA(1)=.08*CSIZE(1)
      AREA(2)=.90*CSIZE(1)
      AREA(3)=.70*CSIZE(2)
      AREA(4)=.90*CSIZE(2)
      AREA(5)=.02*CSIZE(1)
      AREA(6)=.90*CSIZE(3)

      AREA1(1)=.52*CSIZE(1)
      AREA1(2)=CSIZE(1)
      AREA1(3)=.855*CSIZE(2)-.006
      AREA1(4)=CSIZE(2)
      AREA1(5)=0.
      AREA1(6)=CSIZE(3)

      AREA2(1)=.52*CSIZE(1)
      AREA2(2)=CSIZE(1)
      AREA2(3)=.855*CSIZE(2)-.014
      AREA2(4)=CSIZE(2)
      AREA2(5)=0.
      AREA2(6)=CSIZE(3)

      AREA3(1)=.52*CSIZE(1)
      AREA3(2)=CSIZE(1)
      AREA3(3)=.855*CSIZE(2)-.022
      AREA3(4)=CSIZE(2)
      AREA3(5)=0.
      AREA3(6)=CSIZE(3)


      DO 100 I=2,8,2
         SHD(I)=BOX(I)-0.025
  100 CONTINUE

      DO 200 I=1,7,2
         SHD(I)=BOX(I)+0.025
  200 CONTINUE

*****************************************************************************
*  OPEN STRUCTURE FOR THE VAL. BOX (VIEW 1)
*****************************************************************************
C                              OPEN A STRUCTURE
      CALL GPOPST(8)
C                              SET SOLID INTEROR STYLE FOR GREY BOX
      CALL GPIS(2)
C                              SET COLOR FOR INTERIOR OF SHADOW BOX
      CALL GPICI(4)
C                              DRAW GREY SHADOW BOX
      CALL GPPG2(1,4,2,SHD)
C                              SET COLOR FOR INTERIOR OF GREY BOX
      CALL GPICI(3)
C                              DRAW GREY TITLE BOX
      CALL GPPG2(1,4,2,BOX)
C                              CLOSE STRUCTURE
      CALL GPCLST


*****************************************************************************
*  SCREEN DISPLAY
*****************************************************************************
C                              UPDATE THE WORKSTATION
      CALL GPUPWS(IWSID,PRFORM)


*****************************************************************************
*  INITIALIZE AND ACTIVATE DIALS 1,2,3 AND STRING DEVICE
*****************************************************************************
C                              PLACE VALUATORS IN THE REQUEST MODE
  300 CALL GPVLMO(IWSID,1,1,ECHOSW)
      CALL GPVLMO(IWSID,2,1,ECHOSW)
      CALL GPVLMO(IWSID,3,1,ECHOSW)
C                              INITIALIZE DIALS FOR X , Y , Z INPUT
      CALL GPINVL(IWSID,1,VLUE1,ECHO,AREA1,LOVAL,HIVAL,DLEN,DATA)
      CALL GPINVL(IWSID,2,VLUE1,ECHO,AREA2,LOVAL,HIVAL,DLEN,DATA)
      CALL GPINVL(IWSID,3,VLUE1,ECHO,AREA3,LOVAL,HIVAL,DLEN,DATA)
C                              ACTIVATE DIALS FOR X , Y , Z INPUT
      CALL GPVLMO(IWSID,1,EVENT,ECHOSW)
      CALL GPVLMO(IWSID,2,EVENT,ECHOSW)
      CALL GPVLMO(IWSID,3,EVENT,ECHOSW)

      CALL GPINST(IWSID,1,12,INIT,1,AREA,12,1,0,CLEAR)
      CALL GPSTMO(IWSID,1,EVENT,ECHOSW)
```

```
*************************************************************************
*  SCREEN DISPLAY
*************************************************************************
C                              UPDATE THE WORKSTATION
      CALL GPUPWS(IWSID,PRFORM)

*************************************************************************
*  DISPLAY MENU AND GET CHOICE OF ENTER, RETURN OR VALUATOR
*************************************************************************

      LOCAT(1)=0.
      LOCAT(2)=0.
      LOCAT(3)=0.

  500 CALL CMMAND(IWSID,11)
      CALL ENTER(IWSID,CHOICE,CSIZE)

  600 CALL GPAWEV(1000.,IIWSID,ICLASS,IDEV)
      IF(ICLASS.EQ.3)THEN

C-                            PROCESS VALUATORS (NOT IN SUBROUTINE TO
C                                               SPEED EXECUTION)

         CALL GPSTMO(IWSID,1,1,ECHOSW)
         CALL GPGTVL(VALUE)

         IF(IDEV.EQ.1)LOCAT(1)=VALUE
         IF(IDEV.EQ.2)LOCAT(2)=VALUE
         IF(IDEV.EQ.3)LOCAT(3)=VALUE

*************************************************************************
* GET POSITION ICON                                                     *
*************************************************************************

         LX(4)=LOCAT(1)
         LY(5)=LOCAT(2)
         LZ(6)=LOCAT(3)

         LINEX(1)=LOCAT(1)
         LINEX(2)=LOCAT(2)
         LINEX(3)=LOCAT(3)
         LINEX(4)=0.
         LINEX(5)=LOCAT(2)
         LINEX(6)=LOCAT(3)

         LINEY(1)=LOCAT(1)
         LINEY(2)=LOCAT(2)
         LINEY(3)=LOCAT(3)
         LINEY(4)=LOCAT(1)
         LINEY(5)=0.
         LINEY(6)=LOCAT(3)

         LINEZ(1)=LOCAT(1)
         LINEZ(2)=LOCAT(2)
         LINEZ(3)=LOCAT(3)
         LINEZ(4)=LOCAT(1)
         LINEZ(5)=LOCAT(2)
         LINEZ(6)=0.

         XBOX(1)=0.00
         XBOX(2)=LOCAT(2)+(.06*AXISR)
         XBOX(3)=LOCAT(3)+(.06*AXISR)
         XBOX(4)=0.00
         XBOX(5)=LOCAT(2)-(.06*AXISR)
         XBOX(6)=LOCAT(3)+(.06*AXISR)
         XBOX(7)=0.00
         XBOX(8)=LOCAT(2)-(.06*AXISR)
         XBOX(9)=LOCAT(3)-(.06*AXISR)
         XBOX(10)=0.00
         XBOX(11)=LOCAT(2)+(.06*AXISR)
         XBOX(12)=LOCAT(3)-(.06*AXISR)
         XBOX(13)=0.00
         XBOX(14)=LOCAT(2)+(.06*AXISR)
         XBOX(15)=LOCAT(3)+(.06*AXISR)

         YBOX(1)=LOCAT(1)+(.06*AXISR)
         YBOX(2)=0.00
         YBOX(3)=LOCAT(3)+(.06*AXISR)
         YBOX(4)=LOCAT(1)-(.06*AXISR)
         YBOX(5)=0.00
         YBOX(6)=LOCAT(3)+(.06*AXISR)
         YBOX(7)=LOCAT(1)-(.06*AXISR)
         YBOX(8)=0.00
         YBOX(9)=LOCAT(3)-(.06*AXISR)
         YBOX(10)=LOCAT(1)+(.06*AXISR)
         YBOX(11)=0.00
         YBOX(12)=LOCAT(3)-(.06*AXISR)
         YBOX(13)=LOCAT(1)+(.06*AXISR)
         YBOX(14)=0.00
         YBOX(15)=LOCAT(3)+(.06*AXISR)

         ZBOX(1)=LOCAT(1)+(.06*AXISR)
         ZBOX(2)=LOCAT(2)+(.06*AXISR)
         ZBOX(3)=0.00
         ZBOX(4)=LOCAT(1)-(.06*AXISR)
```

```
      ZBOX(5)=LOCAT(2)+(.06*AXISR)
      ZBOX(6)=0.00
      ZBOX(7)=LOCAT(1)-(.06*AXISR)
      ZBOX(8)=LOCAT(2)-(.06*AXISR)
      ZBOX(9)=0.00
      ZBOX(10)=LOCAT(1)+(.06*AXISR)
      ZBOX(11)=LOCAT(2)-(.06*AXISR)
      ZBOX(12)=0.00
      ZBOX(13)=LOCAT(1)+(.06*AXISR)
      ZBOX(14)=LOCAT(2)+(.06*AXISR)
      ZBOX(15)=0.00

      CALL GPEST(9)

      CALL GPOPST(9)

C                          SET POLYLINE COLOR
      CALL GPPLCI(1)
C                          SET LINE TYPE DASHED
      CALL GPLT(2)
C                          DRAW POLYLINE
      CALL GPPL3(2,3,LINEX)
      CALL GPPL3(2,3,LINEY)
      CALL GPPL3(2,3,LINEZ)
C                          SET LINE TYPE SOLID
      CALL GPLT(1)
      CALL GPPL3(5,3,XBOX)
      CALL GPPL3(5,3,YBOX)
      CALL GPPL3(5,3,ZBOX)

C                          SET POLYLINE COLOR
      CALL GPPLCI(5)
      CALL GPPL3(2,3,LX)
      CALL GPPL3(2,3,LY)
      CALL GPPL3(2,3,LZ)
      CALL GPCLST
      CALL GPUPWS(IWSID,PRFORM)
      GO TO 600
   ENDIF
   IF(ICLASS.EQ.6)THEN
C                          PROCESS STRING ENTRY
      CALL GPVLMO(IWSID,1,1,ECHOSW)
      CALL GPVLMO(IWSID,2,1,ECHOSW)
      CALL GPVLMO(IWSID,3,1,ECHOSW)
      RFLG=0

      CALL DEP(IWSID,AXISR,CSIZE,LOCAT,RFLG)

      IF(RFLG.EQ.1)THEN
C                          REENTER POINT
         CALL GPVLMO(IWSID,1,3,ECHOSW)
         CALL GPVLMO(IWSID,2,3,ECHOSW)
         CALL GPVLMO(IWSID,3,3,ECHOSW)
         CALL GPSTMO(IWSID,1,1,ECHOSW)
         CALL GPINST(IWSID,1,12,INIT,1,AREA,12,1,0,CLEAR)
         CALL GPSTMO(IWSID,1,3,ECHOSW)
         LOCAT(1)=0.
         LOCAT(2)=0.
         LOCAT(3)=0.
         RFLG=0
         GO TO 500
      ENDIF

      CALL GPSTMO(IWSID,1,1,ECHOSW)
      CALL GPEST(8)
      CALL GPEST(9)
      GO TO 1000
   ENDIF

   IF(ICLASS.EQ.5)THEN
      CALL GPGTPK(1,1,PPATH)
      CHOICE=PPATH(2)
         IF(CHOICE.EQ.1)THEN
C                          DEACTIVATE VALUATORS
            CALL GPVLMO(IWSID,1,1,ECHOSW)
            CALL GPVLMO(IWSID,2,1,ECHOSW)
            CALL GPVLMO(IWSID,3,1,ECHOSW)
            CALL GPSTMO(IWSID,1,1,ECHOSW)
            RFLG=0
            CALL GPEST(6)
            CALL GPEST(8)
            CALL GPEST(9)
            CALL GPUPWS(IWSID,2)
            GO TO 1000
         ENDIF
         IF(CHOICE.EQ.2)THEN
C                          DEACTIVATE VALUATORS
            CALL GPVLMO(IWSID,1,1,ECHOSW)
            CALL GPVLMO(IWSID,2,1,ECHOSW)
            CALL GPVLMO(IWSID,3,1,ECHOSW)
            CALL GPSTMO(IWSID,1,1,ECHOSW)
            RFLG=1
            CALL GPEST(6)
            CALL GPEST(8)
            CALL GPEST(9)
            CALL GPUPWS(IWSID,2)
```

```
                GO TO 1000
            ENDIF
        ENDIF

 1000 RETURN
      END
```

SUBROUTINE RDRWJT

```
      SUBROUTINE RDRWJT(IWSID,JNAM,JNUM,JOINT,AXISR,SCALE)
************************************************************************
* ----- SUBROUTINE RDRWJT -----                                        *
*                                                                      *
*                                                                      *
*   THIS SUBROUTINE IS USED TO REDRAW ALL JOINTS AFTER A DELETION      *
*                                                                      *
*   OR RESTART                                                         *
*                                                                      *
************************************************************************

      INTEGER IWSID,NUMP,JNUM(20,3),JTYPE

      REAL LOCAT(3),JOINT(20,21),AXISR,ORIENT(3),SCALE

      CHARACTER*7 JNAME,JNAM(20)

C                          DELETE ALL JOINTS
         CALL GPEST(13)
         CALL GPEST(14)
         CALL GPEST(17)
         CALL GPEST(18)
         CALL GPUPWS(IWSID,2)

      DO 10 I=1,100
C                          CHECK FOR LAST JOINT
         IF(JNAM(I).EQ.'BBBBBBB')GO TO 1000
C                          CHECK DELETED SPACES IN THE ARRAY
         IF(JNAM(I).EQ.'CCCCCCC')GO TO 10
C                          CHECK FOR AN INVISIBLE JOINT
         IF(JNUM(I,3).EQ.1)GO TO 10
C                          GET JOINT NUMBER
         NUMJ=JNUM(I,1)
C                          GET JOINT TYPE
         JTYPE=JNUM(I,2)
C                          GET JOINT LOCATION
         LOCAT(1)=JOINT(I,1)
         LOCAT(2)=JOINT(I,2)
         LOCAT(3)=JOINT(I,3)
C                          GET JOINT ORIENTATION
         ORIENT(1)=JOINT(I,4)
         ORIENT(2)=JOINT(I,5)
         ORIENT(3)=JOINT(I,6)
C                          GET JOINT NAME
         JNAME=JNAM(I)
C                          DRAW JOINT
         CALL DRAWJT(IWSID,AXISR,JNAME,NUMJ,JTYPE,JOINT,SCALE)
   10 CONTINUE

 1000 RETURN
      END
```

SUBROUTINE RDRWLK

```
      SUBROUTINE RDRWLK(IWSID,JNAM,JNUM,JOINT,AXISR,LNAM,LNUM,LINK,
     &SCALE)
************************************************************************
* ----- SUBROUTINE RDRWLK -----                                        *
*                                                                      *
*                                                                      *
*   THIS SUBROUTINE IS USED TO REDRAW ALL LINKS AFTER A DELETION       *
*                                                                      *
*   OR RESTART                                                         *
*                                                                      *
************************************************************************

      INTEGER IWSID,NUML,JNUM(20,3),LTYPE,LNUM(30,14)

      REAL LOCAT(3),JOINT(20,21),AXISR,ORIENT(3),LINK(30,6)

      CHARACTER*7 LNAME,JNAM(20),LNAM(30)

C                          DELETE ALL LINK
         CALL GPDLST(19)
         CALL GPARV(IWSID,4,19,1.)
```

```
         CALL GPVP(IWSID,4,1,1)
         CALL GPUPWS(IWSID,2)

      DO 10 I=1,30
C                          CHECK FOR LAST JOINT
         IF(LNAM(I).EQ.'BBBBBBB')GO TO 1000
C                          CHECK DELETED SPACES IN THE ARRAY
         IF(LNAM(I).EQ.'CCCCCCC')GO TO 10
C                          CHECK DELETED SPACES IN THE ARRAY
         IF(LNUM(I,14).EQ.1)THEN
            GO TO 10
         ENDIF

C                          GET JOINT NUMBER
         NUML=LNUM(I,1)
C                          DRAW JOINT
         CALL DRAWLK(IWSID,LNAM,LNUM,LINK,JOINT,JNUM,AXISR,NUML,SCALE)
   10 CONTINUE

 1000 RETURN
      END
```

SUBROUTINE RDRWPT

```
      SUBROUTINE RDRWPT(IWSID,PNAM,PNUM,POINT,AXISR)
*************************************************************************
* ----- SUBROUTINE DRAWPT -----                                         *
*                                                                       *
*                                                                       *
*   THIS SUBROUTINE IS USED TO REDRAW ALL POINTS AFTER A DELETION       *
*                                                                       *
*   OR RESTART                                                          *
*                                                                       *
*************************************************************************

      INTEGER IWSID,NUMP,PNUM(100)

      REAL LOCAT(3),POINT(100,3),AXISR

      CHARACTER*7 PNAME,PNAM(100)

C                          DELETE ALL POINTS
         CALL GPEST(12)
         CALL GPUPWS(IWSID,2)

      DO 10 I=1,100
C                          CHECK FOR LAST POINT
         IF(PNAM(I).EQ.'BBBBBBB')GO TO 1000
C                          CHECK DELETED SPACES IN THE ARRAY
         IF(PNAM(I).EQ.'CCCCCCC')GO TO 10
C                          GET POINT NUMBER
         NUMP=PNUM(I)
C                          GET POINT LOCATION
         LOCAT(1)=POINT(I,1)
         LOCAT(2)=POINT(I,2)
         LOCAT(3)=POINT(I,3)
C                          GET POINT NAME
         PNAME=PNAM(I)
C                          DRAW POINT
         CALL DRAWPT(IWSID,LOCAT,AXISR,PNAME,NUMP)
   10 CONTINUE

 1000 RETURN
      END
```

SUBROUTINE REINV

```
      SUBROUTINE REINV(IWSID,JNAM,JNUM,JOINT,AXISR,LNAM,LNUM,LINK,
     &SCALE,CSIZE)
*************************************************************************
* ----- SUBROUTINE REINV  -----                                         *
*                                                                       *
*                                                                       *
*   THIS SUBROUTINE IS USED TO MAKE LINKS AND JOINTS VISIBLE            *
*                                                                       *
*************************************************************************

      INTEGER IWSID,NUMP,JNUM(20,3),JTYPE,END,LNUM(30,14)

      REAL LOCAT(3),JOINT(20,21),AXISR,ORIENT(3),SCALE,ENDP(3),CSIZE(3),
     & LINK(30,6)

      CHARACTER*7 JNAME,JNAM(20),LNAM(30)
```

```
      DO 20 I=1,20
         JNUM(I,3)=0
   20 CONTINUE

      DO 30 I=1,30
         LNUM(I,14)=0
   30 CONTINUE

      CALL RDRWJT(IWSID,JNAM,JNUM,JOINT,AXISR,SCALE)
      CALL RDRWLK(IWSID,JNAM,JNUM,JOINT,AXISR,LNAM,LNUM,LINK,SCALE)

 1000 RETURN
      END
```

SUBROUTINE RESTOR

```
      SUBROUTINE RESTOR(IWSID,AXISR,CSIZE)
*************************************************************************
* ----- SUBROUTINE RESTOR ---                                           *
*                                                                       *
*                                                                       *
*   THIS SUBROUTINE IS USED TO RESTORE THE ORIGINAL                     *
*                                                                       *
*************************************************************************

      INTEGER IWSID,PARELL,PRFORM


      REAL AXISR,XAXIS(24),YAXIS(24),ZAXIS(24),WINDOW(4),VP1(6),VP2(6),
     &BOX1(8),BOX4(8),NEAR,FAR,POINT(3),DIST,UP(3),NORMAL(3),
     &MATRIX(4,4),POINTP(3),PX(3),PY(3),PZ(3),WINDO1(4),NEAR1,FAR1

      CHARACTER*1 X,Y,Z

C                              FLAG FOR UPDATE
      DATA PRFORM/2/
C                              GREY MAIN SCREEN BOX
      DATA BOX1/-0.48,-0.98,0.98,-0.98,0.98,0.48,-0.48,0.48/
      DATA DIST/0./
      DATA PARELL/1/
      DATA POINT/0.,0.,0./
      DATA POINTP/2.0,1.5,5.0/
      DATA NORMAL/1.,0.,0./
      DATA UP/0.,0.,1./
      DATA X/'X'/
      DATA Y/'Y'/
      DATA Z/'Z'/


      VP1(1)=(BOX1(1)+1.)/2.
      VP1(2)=(BOX1(3)+1.)/2.
      VP1(3)=(BOX1(4)+1.)/2.
      VP1(4)=(BOX1(6)+1.)/2.
      VP1(5)=(BOX1(1)+1.)/2.
      VP1(6)=(BOX1(3)+1.)/2.

      NEAR=AXISR+(.64*AXISR)
      FAR=-AXISR-(.64*AXISR)


      WINDOW(1)=-AXISR-(.64*AXISR)
      WINDOW(2)=AXISR+(.64*AXISR)
      WINDOW(3)=-AXISR-(.64*AXISR)
      WINDOW(4)=AXISR+(.64*AXISR)

*************************************************************************
*  SET THE VIEW CHARACTERISTICS FOR AXES VIEW 4
*************************************************************************
C                              SET THE CHARACTERISTICS
      CALL GPVMP3(IWSID,4,WINDOW,VP1,PARELL,POINTP,DIST,NEAR,FAR)

C                              ACTIVATE THE VIEW
      CALL GPVCH(IWSID,4,2,2,2,1,0,2,5,2)


      CALL GPUPWS(IWSID,PRFORM)

      RETURN
      END
```

SUBROUTINE RSCRA

```
      SUBROUTINE RSCRA(PNAM,PNUM,POINT,JNAM,JNUM,JOINT,LNAM,LNUM,LINK,
```

```
     &CSIZE,INAM,INUM,IC,IWSID,IRETT)
*************************************************************************
* ----- SUBROUTINE RSCRA -----                                          *
*                                                                       *
*                                                                       *
*   THIS SUBROUTINE IS USED WRITE AN RSCR ANALYSIS FILE                 *
*                                                                       *
*************************************************************************

      INTEGER PNUM(100),LNUM(30,14),JNUM(20,3),INUM(20,2),NP,NJ,NL,NI,
     &CONN(20,3)

      REAL LINK(30,6),JOINT(20,21),POINT(100,3),AXISR,CSIZE(3),AREA(6),
     &IC(20,4)

      CHARACTER*7 LNAM(30),JNAM(20),PNAM(100),FILE,FILEO,FILE1,INAM(20),
     &           JNAME,LNAME1,LNAME2,FILEN
      CHARACTER*19 FLINE

      DATA FILE/'RSCR   '/
      NP=0
      NJ=0
      NL=0
      NI=0

      DO 2 I=1,20
         DO 1 J=1,3
            CONN(I,J)=0
    1    CONTINUE
    2 CONTINUE

      IRETT=0

      AREA(1)=.08*CSIZE(1)
      AREA(2)=.90*CSIZE(1)
      AREA(3)=.70*CSIZE(2)
      AREA(4)=.90*CSIZE(2)
      AREA(5)=.02*CSIZE(3)
      AREA(6)=.90*CSIZE(3)

      FILE1='       '

      REWIND 15
*************************************************************************
*    SORT OUT LINK CONNECTIVITIES
*************************************************************************
   30 DO 100 J=1,30
         IF(LNAM(J).EQ.'CCCCCCC')THEN
            GO TO 90
         ENDIF
         IF(LNAM(J).EQ.'BBBBBBB')THEN
            GO TO 105
         ENDIF

         LN=J
         DO 50 I=2,13,2
            IF(LNUM(LN,I).EQ.0)GO TO 90
            CONN(LNUM(LN,I),(LNUM(LN,I+1)+1))=LNUM(LN,1)
   50    CONTINUE

   90    CONTINUE
  100 CONTINUE

  105 DO 110 I=1,20
         CONN(I,1)=JNUM(I,2)
  110 CONTINUE

*************************************************************************
*    FILE JOINT TO GROUND WITH ZERO'S
*************************************************************************

      DO 200 I=1,20
         IF(CONN(I,1).EQ.8)THEN
            IGRD=CONN(I,2)
            CONN(I,2)=0
            DO 180 J=1,20
               IF(CONN(J,2).EQ.IGRD)THEN
                  CONN(J,2)=0
               ENDIF
               IF(CONN(J,3).EQ.IGRD)THEN
                  CONN(J,3)=0
               ENDIF
  180       CONTINUE
         ENDIF
  200 CONTINUE

*************************************************************************
*  DETERMINE IF MECHANISM IS AN RSCR
*************************************************************************
      III=0
      DO 300 I=1,20
         IF(CONN(I,1).EQ.1)THEN
            CONN(I,1)=91
            III=III+1
            GO TO 300
         ENDIF
```

```
            IF(CONN(I,1).EQ.4)THEN
                 CONN(I,1)=94
                 III=III+1
                 GO TO 300
            ENDIF
            IF(CONN(I,1).EQ.3)THEN
                 CONN(I,1)=93
                 III=III+1
                 GO TO 300
            ENDIF
            IF(CONN(I,1).EQ.1)THEN
                 CONN(I,1)=91
                 III=III+1
                 GO TO 300
            ENDIF
  300 CONTINUE

C     DO 301 I=1,20
C     WRITE(6,*)'CONN    ',(CONN(I,J),J=1,3)
C 301 CONTINUE
C
C     WRITE(6,*)'III ',III
C
      IF(III.NE.4)THEN
         IRETT=1
         RETURN
      ENDIF

*********************************************************************
*  INITIALIZE THE STING DEVICE. REQUEST NEW RESTART FILE NAME
*********************************************************************
C                              INITIALIZE STRING MODE
      CALL GPSTMO(IWSID,1,1,2)
C                              INITIALIZE THE STRING DEVICE
    3 CALL GPINST(IWSID,1,7,FILE,1,AREA,7,1,0,FILE)
C                              REQUEST THE FILE NAME
      CALL CMMAND(IWSID,91)
      FILEN='                '
C                              REQUEST STRING
      CALL GPRQST(IWSID,1,7,ISTAT,NUM,FILEN)
      IF(FILEN.EQ.'                ')GO TO 5000

      FILE=FILEN
C                              SEE IF FILE ALREADY EXISTS
    4 CALL FILEXS(FILE,IRET)
C                              IF FILE EXISTS, HAVE USER RETYPE TO USE
      IF(IRET.EQ.0)THEN
         CALL GPSTMO(IWSID,1,1,2)
         FILEO=FILE
         FILEN='                '
         CALL CMMAND(IWSID,8)
         CALL GPINST(IWSID,1,7,FILE,1,AREA,7,1,0,FILE)
         CALL GPRQST(IWSID,1,7,ISTAT,NUM,FILEN)
         IF(FILEN.EQ.'                ')GO TO 5000
         IF(FILE.NE.FILEO)THEN
              FILEN='                '
              GO TO 4
         ENDIF
      ENDIF


C                              OPEN A NEW FILE
      OPEN(UNIT=16,FILE=FILE,STATUS='NEW',ERR=2)


*********************************************************************
*     WRITE DATA FILE
*********************************************************************

      WRITE(16,999)'RSCR ANALYSIS DATA'
  999 FORMAT(A18)

      DO 1000 I=1,20
         IF(CONN(I,1).EQ.94)THEN
              L1=CONN(I,2)
              L2=CONN(I,3)
         ENDIF
 1000 CONTINUE

      DO 1010 I=1,20
         IF(CONN(I,1).EQ.91)THEN
            IF(CONN(I,2).EQ.0)THEN
               IF(CONN(I,3).EQ.L1.OR.CONN(I,3).EQ.L2)THEN
                  WRITE(16,*)JOINT(I,1),JOINT(I,2),JOINT(I,3)
                  WRITE(16,*)JOINT(I,4),JOINT(I,5),JOINT(I,6)
                  CONN(I,1)=81
                  POS=IC(I,1)
                  RINC=IC(I,2)
                  ISTEP=INUM(I,2)
                  VEL=IC(I,3)
                  ACC=IC(I,4)
                  GO TO 1011
               ENDIF
            ENDIF
            IF(CONN(I,3).EQ.0)THEN
```

```
                IF(CONN(I,2).EQ.L1.OR.CONN(I,2).EQ.L2)THEN
                   WRITE(16,*)JOINT(I,1),JOINT(I,2),JOINT(I,3)
                   WRITE(16,*)JOINT(I,4),JOINT(I,5),JOINT(I,6)
                   POS=IC(I,1)
                   RINC=IC(I,2)
                   ISTEP=INUM(I,2)
                   VEL=IC(I,3)
                   ACC=IC(I,4)
                   CONN(I,1)=81
                   GO TO 1011
                ENDIF
             ENDIF
          ENDIF
1010 CONTINUE

1011 DO 1020 I=1,20
          IF(CONN(I,1).EQ.94)THEN
             WRITE(16,*)JOINT(I,1),JOINT(I,2),JOINT(I,3)
          ENDIF
1020 CONTINUE

     DO 1030 I=1,20
          IF(CONN(I,1).EQ.93)THEN
             WRITE(16,*)JOINT(I,1),JOINT(I,2),JOINT(I,3)
             WRITE(16,*)JOINT(I,4),JOINT(I,5),JOINT(I,6)
             WRITE(16,*)JOINT(I,14)
          ENDIF
1030 CONTINUE

     DO 1040 I=1,20
          IF(CONN(I,1).EQ.91)THEN
             WRITE(16,*)JOINT(I,1),JOINT(I,2),JOINT(I,3)
             WRITE(16,*)JOINT(I,4),JOINT(I,5),JOINT(I,6)
          ENDIF
1040 CONTINUE

     WRITE(16,*)VEL

     IF(ACC.LE.0.0000001.AND.ACC.GT.-0.0000001)THEN
          WRITE(16,*)'0'
     ELSE
          WRITE(16,*)'1'
          WRITE(16,*)ACC
     ENDIF

     WRITE(16,*)ISTEP

5000 CLOSE(UNIT=16)
     RETURN
     END
```

SUBROUTINE RSCRS

```
      SUBROUTINE RSCRS(PNAM,PNUM,POINT,AXISR,CSIZE,IWSID,IRETN)
*****************************************************************************
* ----- SUBROUTINE RSCRS    -----                                             *
*                                                                             *
*                                                                             *
*   THIS SUBROUTINE IS USED WRITE A RSCR SYNTHESIS FILE                       *
*                                                                             *
*****************************************************************************

      INTEGER PNUM(30,4),NP,IWSID,PRFORM,PPATH(3),ACLASS(1),CHOICE,LENG,
     &        DATA(35)

      REAL POINT(30,14),CSIZE(3),AXISR,AREA(6),PAREA(6),O(3,3),
     &     U(3,3),V(3,3),W(3,3),POSN(2),SLOC(3),JOINT(20,21),OR(3),
     &     MAT(4,4),MATSC(4,4),SCALE(3),MAT3(4,4),RO(3),PPOSN(3)

      CHARACTER*7 PNAM(30),FILE,FILEO,FILE1,FILEN
      CHARACTER*19 FLINE
      CHARACTER*16 RETURN

      DATA RETURN/'1.  RETURN      '/
      DATA LENG/16/
      DATA POSN/0.05,2.2/

      DATA MAT/1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./


      DATA DATA/32,0,0,32*0/
C     DATA PRFORM/2/              FLAG FOR UPDATE
C     DATA ACLASS/4/              INCLUSION CLASS LIST

      DATA FILE/'RSCRDAT'/

      NP=0
      IRETN=0
```

```
      III=1

      AREA(1)=.08*CSIZE(1)
      AREA(2)=.90*CSIZE(1)
      AREA(3)=.70*CSIZE(2)
      AREA(4)=.90*CSIZE(2)
      AREA(5)=.02*CSIZE(3)
      AREA(6)=.90*CSIZE(3)

      FILE1='       '

      REWIND 15

*********************************************************************
*    PICK THE THREE BODY POSITIONS FOR SYNTHESIS
*********************************************************************

C   INITIALIZE PICK

C                             SET PICK FILTER(ALL CLASS 4 DETECTABLE)
      CALL GPPKF(IWSID,1,1,ACLASS,0,ACLASS)
C                             MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                             DEFINE PICK AREA
      PAREA(1)=((-.98+1)/2)*CSIZE(1)
      PAREA(2)=((.98+1)/2)*CSIZE(1)
      PAREA(3)=((-.98+1)/2)*CSIZE(2)
      PAREA(4)=((.48+1)/2)*CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=CSIZE(3)
C                             INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                             PLACE PICK DEVICE IN THE EVENT MODE
      CALL GPPKMO(IWSID,1,3,2)
C                             TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)
C                             GET INQUIRY OR RETURN COMMAND
      CALL SCMMAN(IWSID,25)

C  INCLUDE A RETURN PICK

C                             RETURN PICK
      CALL GPOPST(6)
      CALL GPADCN(1,ACLASS)
      CALL GPPKID(101)
      CALL GPTXCI(2)
      CALL GPAHSC(1.0)
      CALL GPAN2(POSN,LENG,RETURN)
      CALL GPCLST
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
      CALL GPUPWS(IWSID,2)

C  AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT

C                             AWAIT AN EVENT
  200 CALL GPAWEV(1000.,IWS,ICLA,IDEV)
      IF(ICLA.EQ.5)THEN
C                             GET PICK
         CALL GPGTPK(1,IDEP,PPATH)
C                             IF NO PICK, RETURN
         IF(PPATH(2).EQ.101)THEN
              IRETN=1
              CALL GPPKMO(IWSID,1,1,2)
              CALL GPEST(6)
              CALL GPUPWS(IWSID,PRFORM)
              GO TO 5000
         ENDIF
C                             CHOICE IS SECOND ITEM OF PICK PATH
         CHOICE=PPATH(2)
C                             GET DATA FOR SYN. FILE FROM POSITION PICKED
         O(III,1)=POINT(CHOICE,1)
         O(III,2)=POINT(CHOICE,2)
         O(III,3)=POINT(CHOICE,3)
         U(III,1)=POINT(CHOICE,4)
         U(III,2)=POINT(CHOICE,5)
         U(III,3)=POINT(CHOICE,6)
         V(III,1)=POINT(CHOICE,7)
         V(III,2)=POINT(CHOICE,8)
         V(III,3)=POINT(CHOICE,9)
         W(III,1)=POINT(CHOICE,10)
         W(III,2)=POINT(CHOICE,11)
         W(III,3)=POINT(CHOICE,12)

C                             REQUEST A NEXT BODY POSITION
         III=III+1
         IF(III.EQ.2)THEN
            CALL SCMMAN(IWSID,26)
            GO TO 200
         ENDIF
         IF(III.EQ.3)THEN
            CALL SCMMAN(IWSID,34)
```

```
            GO TO 200
         ENDIF
         IF(III.GT.3)THEN
            GO TO 1000
         ENDIF
      ELSE
C                              IF NOT A PICK, GO TO AWAIT AN EVENT
         GO TO 200
      ENDIF


C                              MAKE SURE PICK IS DEACTIVATED
 1000 CALL GPPKMO(IWSID,1,1,2)
      CALL GPEST(6)
      CALL GPUPWS(IWSID,PRFORM)

C                              SET PICK FILTER(ALL CLASS 4 UNDETECTABLE)
      CALL GPPKF(IWSID,1,0,ACLASS,1,ACLASS)

*********************************************************************
*    REQUEST POSITION OF THE SPHERICAL JOINT
*********************************************************************

 1900 CALL SCMMAN(IWSID,35)
      CALL SPOSIT(IWSID,AXISR,CSIZE,SLOC,IRRET)
      IF(IRRET.EQ.1)THEN
         GO TO 1900
      ENDIF
C                    DRAW SPHERICAL JOINT ICON
      OR(1)=0.
      OR(2)=0.
      OR(3)=0.
      SCALE(1)=AXISR
      SCALE(2)=AXISR
      SCALE(3)=AXISR
      CALL GPTRL3(SLOC,MAT3)
      CALL GPSC3(SCALE,MATSC)
      CALL GPOPST(12)
      CALL GPMLX3(MAT3,3)
      CALL GPMLX3(MATSC,1)
      CALL JICONS(SLOC,OR,4,AXISR,JOINT,1,1.0,MAT)
      CALL GPCLST
      CALL GPUPWS(IWSID,2)

*********************************************************************
*    REQUEST ORIENTATION OF THE SECOND REVOLUTE JOINT
*********************************************************************

 1910 CALL SCMMAN(IWSID,40)
      PPOSN(1)=0.0
      PPOSN(2)=0.0
      PPOSN(3)=0.0
      CALL SORIEN(IWSID,PPOSN,POINT,AXISR,CSIZE,RO,IRRRET)
      IF(IRRRET.EQ.1)THEN
         GO TO 1910
      ENDIF

*********************************************************************
*  INITIALIZE THE STRING DEVICE. REQUEST SYNTHEIS FILE NAME
*********************************************************************
C                              INITIALIZE STRING MODE
      CALL GPSTMO(IWSID,1,1,2)
C                              INITIALIZE THE STRING DEVICE
 2003 CALL GPINST(IWSID,1,7,FILE,1,AREA,7,1,0,FILE)
C                              REQUEST THE FILE NAME
      CALL SCMMAN(IWSID,83)
      FILEN='          '
C                              REQUEST STRING
      CALL GPRQST(IWSID,1,7,ISTAT,NUM,FILEN)
      IF(FILEN.EQ.'          ')GO TO 5000

      FILE=FILEN
C                              SEE IF FILE ALREADY EXISTS
 2004 CALL FILEXS(FILE,IRET)
C                              IF FILE EXISTS, HAVE USER RETYPE TO USE
      IF(IRET.EQ.0)THEN
         CALL GPSTMO(IWSID,1,1,2)
         FILEO=FILE
         FILEN='          '
         CALL CMMAND(IWSID,8)
         CALL GPINST(IWSID,1,7,FILE,1,AREA,7,1,0,FILE)
         CALL GPRQST(IWSID,1,7,ISTAT,NUM,FILEN)
         IF(FILEN.EQ.'          ')GO TO 5000
         IF(FILE.NE.FILEO)THEN
              FILEN='          '
              GO TO 2004
         ENDIF
      ENDIF


C                              OPEN A NEW FILE
      OPEN(UNIT=16,FILE=FILE,STATUS='NEW')
```

```
*********************************************************************
*    WRITE SYNTHESIS FILE fOR RSCR
*********************************************************************

      WRITE(16,3000)'RSCR SYNTHESIS DATA'
 3000 FORMAT(A19)
      WRITE(16,*)O(1,1),O(1,2),O(1,3)
      WRITE(16,*)O(2,1),O(2,2),O(2,3)
      WRITE(16,*)O(3,1),O(3,2),O(3,3)
      WRITE(16,*)'2'
      WRITE(16,*)U(1,1),U(1,2),U(1,3)
      WRITE(16,*)V(1,1),V(1,2),V(1,3)
      WRITE(16,*)W(1,1),W(1,2),W(1,3)
      WRITE(16,*)U(2,1),U(2,2),U(2,3)
      WRITE(16,*)V(2,1),V(2,2),V(2,3)
      WRITE(16,*)W(2,1),W(2,2),W(2,3)
      WRITE(16,*)U(3,1),U(3,2),U(3,3)
      WRITE(16,*)V(3,1),V(3,2),V(3,3)
      WRITE(16,*)W(3,1),W(3,2),W(3,3)
      WRITE(16,*)SLOC(1),SLOC(2),SLOC(3)
      WRITE(16,*)RO(1),RO(2),RO(3)


 5000 RETURN
      END
```

SUBROUTINE RSTART

```
      SUBROUTINE RSTART(IWSID,FILE,CSIZE,AXISR,PNAM,PNUM,POINT,
     &JNAM,JNUM,JOINT,LNAM,LNUM,LINK,INAM,INUM,IC,SCALE)
*********************************************************************
* ----- SUBROUTINE RESTRT -----                                     *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE WILL READ THE NAME OF A MECHIN RESTART FILE     *
*                                                                   *
*   AND USE THE FILE TO FILL THE APPROPRIATE ARRAYS TO REUSE        *
*                                                                   *
*   THE FILE.                                                       *
*                                                                   *
*********************************************************************

      INTEGER IWSID,PNUM(100),JNUM(20,3),LNUM(30,14),INUM(20,2),
     &        DATA(35),LENG,ACLASS(1)
      REAL AREA(6),CSIZE(3),AXISR,POINT(100,3),JOINT(20,21),LINK(30,6),
     &     IC(20,4),POSN(2),PAREA(6)
      CHARACTER*7 FILE,CMD,QUIT,CLEAR,PNAM(100),JNAM(20),LNAM(30),
     &            INAM(2)
      CHARACTER*28 FLINE,CLINE
      CHARACTER*16 RETURN
      LOGICAL*4 EXT

      DATA RETURN/'1.  QUIT        '/
      DATA LENG/16/
      DATA POSN/0.05,2.2/


      DATA DATA/32,0,0,         2,2,1,1,
     &                         1,1,1,1,1,1,1,
     &                         1,1,1,1,1,1,1,
     &                         1,1,1,1,1,1,1,
     &                           1,1,1,1/
C                            INCLUSION CLASS LIST
      DATA ACLASS/2/


      DATA CLEAR/'       '/
      DATA CMD/'       '/
      DATA QUIT/'QUIT   '/
      DATA CLINE/'MECHIN RESTART FILE-ANALYSIS'/

      FILE='       '

      IFLG=0
      AXISR=0.0

      AREA(1)=0.08*CSIZE(1)
      AREA(2)=0.90*CSIZE(1)
      AREA(3)=0.70*CSIZE(2)
      AREA(4)=0.90*CSIZE(2)
      AREA(5)=0.02*CSIZE(3)
      AREA(6)=0.90*CSIZE(3)

*********************************************************************
*  INITIALIZE PICK
*********************************************************************
C                              SET PICK FILTER(ALL CLASS 2 DETECTABLE)
```

```
      CALL GPPKF(IWSID,1,1,ACLASS,0,ACLASS)
C                                MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                                DEFINE PICK AREA
      PAREA(1)=((-.98+1)/2)*CSIZE(1)
      PAREA(2)=((.98+1)/2)*CSIZE(1)
      PAREA(3)=((-.98+1)/2)*CSIZE(2)
      PAREA(4)=((.48+1)/2)*CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=CSIZE(3)
C                                INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                                PLACE PICK DEVICE IN THE EVENT MODE
      CALL GPPKMO(IWSID,1,3,2)
C                                TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,2)

*****************************************************************
*  INCLUDE A RETURN PICK
*****************************************************************

C                                RETURN PICK
      CALL GPOPST(6)
      CALL GPADCN(1,ACLASS)
      CALL GPPKID(101)
      CALL GPTXCI(2)
      CALL GPAHSC(1.0)
      CALL GPAN2(POSN,LENG,RETURN)
      CALL GPCLST
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
      CALL GPUPWS(IWSID,2)


*****************************************************************
*  INITIALIZE THE STRING DEVICE
*****************************************************************
C                                REQUEST THE FILE NAME
      CALL CMMAND(IWSID,2)
C                                INITIALIZE THE STRING DEVICE
    5 CALL GPSTMO(IWSID,1,1,2)
      CALL GPINST(IWSID,1,7,CLEAR,1,AREA,7,1,0,CLEAR)
      CALL GPSTMO(IWSID,1,3,2)


*****************************************************************
*  QUIT IF QUIT IS PICKED
*****************************************************************
C                                AWAIT AN EVENT
    6 CALL GPAWEV(1000.,IWS,ICLA,IDEV)
      IF(ICLA.EQ.5)THEN
C                                QUIT PHIGS
         CALL GPDAST
         CALL GPCLPH
         STOP
      ENDIF

*****************************************************************
*  PROCESS STRING EVENT
*****************************************************************
      L=7
C                                GET STRING
      CALL GPGTST(L,LR,FILE)

C                                FIND OUT IF FILE EXISTS
      CALL FILEXS(FILE,IRET)

      IF(IRET.EQ.1)THEN
         CALL CMMAND(IWSID,9)
         FILE=CLEAR
         GO TO 6
      ENDIF
C                                OPEN UNIT 15, FILETYPE 'FILE'
      OPEN(UNIT=15, FILE=FILE, ERR=6)
      REWIND 15
      READ(15,10,ERR=6,END=6)FLINE
   10 FORMAT(A28)

      IF(FLINE.NE.CLINE)THEN
         CALL CMMAND(IWSID,6)
         FILE=CLEAR
         GO TO 6
      ENDIF

      CALL GPPKMO(IWSID,1,1,2)
      CALL GPEST(6)

C                                WRITE THAT ACTION IS CONTINUING
      CALL CMMAND(IWSID,3)
C                                WRITE FILENAME TO THE SCREEN
      CALL FILENM(IWSID,FILE)

*****************************************************************
*  SET UP AXES
*****************************************************************
C                                READ THE AXIS RANGE
```

```
      READ(15,*)AXISR
      CALL AXES(IWSID,AXISR)
C                              READ JOINT SCALE FACTOR
      READ(15,*)SCALE
      CALL SCALNM(IWSID,SCALE)

*********************************************************************
*  SET UP POINT ARRAY
*********************************************************************
C                              READ THE NUMBER OF POINTS
      READ(15,*)NP
C                              READ THE POINT DATA
      DO 100 I=1,NP
         READ(15,99)PNAM(I)
   99    FORMAT(A7)
         READ(15,*)PNUM(I)
         READ(15,*)(POINT(I,J),J=1,3)
  100 CONTINUE

*********************************************************************
*  SET UP JOINT ARRAY
*********************************************************************
C                              READ THE NUMBER OF POINTS
      READ(15,*)NJ
C                              READ THE POINT DATA
      DO 200 I=1,NJ
         READ(15,199)JNAM(I)
  199    FORMAT(A7)
         READ(15,*)JNUM(I,1),JNUM(I,2),JNUM(I,3)
         READ(15,*)(JOINT(I,J),J=1,21)
  200 CONTINUE

*********************************************************************
*  SET UP LINK ARRAY
*********************************************************************
C                              READ THE NUMBER OF POINTS
      READ(15,*)NL
C                              READ THE POINT DATA
      DO 300 I=1,NL
         READ(15,299)LNAM(I)
  299    FORMAT(A7)
         READ(15,*)(LNUM(I,J),J=1,14)
         READ(15,*)(LINK(I,J),J=1,6)
  300 CONTINUE

*********************************************************************
*  SET UP LINK ARRAY
*********************************************************************
C                              READ THE POINT DATA
      DO 400 I=1,20
         READ(15,399)INAM(I)
  399    FORMAT(A7)
         READ(15,*)(INUM(I,J),J=1,2)
         READ(15,*)(IC(I,J),J=1,4)
  400 CONTINUE

*********************************************************************
*  DRAW ALL POINTS, JOINTS, AND LINKS
*********************************************************************

      CALL RDRWPT(IWSID,PNAM,PNUM,POINT,AXISR)
      CALL RDRWJT(IWSID,JNAM,JNUM,JOINT,AXISR,SCALE)
      CALL RDRWLK(IWSID,JNAM,JNUM,JOINT,AXISR,LNAM,LNUM,LINK,SCALE)
      CALL ICICON(IWSID,JOINT,INAM,INUM,IC,SCALE)

      CALL GPSTMO(IWSID,1,1,2)
      CALL GPPKF(IWSID,1,0,ACLASS,1,ACLASS)

      RETURN
      END
```

SUBROUTINE SADDP

```
      SUBROUTINE SADDP(IWSID,AXISR,CSIZE,PNAM,PNUM,POINT,SCALE)
*********************************************************************
* ----- SUBROUTINE SADDP -----
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED TO ADD BODY POSITION POINTS             *
*                                                                   *
*********************************************************************

      COMMON/PDEF/PNDEF(100)

      INTEGER IWSID,NUML,PNUM(30,4),RFLG,CLASS(1),
     &     CHOICE

      REAL AXISR,LOCAT(3),AREA(6),POINT(30,14),CSIZE(3),
```

```
     &  ENDP1(3),ENDP2(3),LNK(6),XVEC(3),YVEC(3),X1(6),X2(6),
     &  POINTT(20,21),SC(3),SCM(4,4),TR(3),TRM(4,4),TRM1(4,4),P(3),
     &  P1(3),P2(3),S(4,4),T(4,4),XAX(3),YAX(3),ZAX(3),NAX(3)

      CHARACTER*7 PNDEF,PNAME,PNAM(30),CLEAR,CLEAR1
      CHARACTER*20 CLEAR2


      DATA CLEAR     /'       '/
      DATA CLEAR1    /'       '/
      DATA CLEAR2    /'                    '/
      DATA SCM/1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
      DATA TRM/1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
      DATA TRM1/1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
      DATA S  /1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
      DATA T  /1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./

      DATA CLASS  /5/

      RFLG=0

      LOCAT(1)=0.0
      LOCAT(2)=0.0
      LOCAT(3)=0.0

      AREA(1)=0.08*CSIZE(1)
      AREA(2)=0.90*CSIZE(1)
      AREA(3)=0.70*CSIZE(1)
      AREA(4)=0.90*CSIZE(1)
      AREA(5)=0.02*CSIZE(1)
      AREA(6)=0.90*CSIZE(1)

      PNAME='       '

      SC(1)=SCALE
      SC(2)=SCALE
      SC(3)=SCALE


      CALL GPSC3(SC,SCM)

*********************************************************************
*     SET POINT NUMBER
*********************************************************************
C                  SEARCH PNAM ARRAY FOR NEXT AVAILABLE SLOT (BBBBBBBB)
      DO 10 NUMP=1,30,1
         IF(PNAM(NUMP).EQ.'BBBBBBB'.OR.PNAM(NUMP).EQ.'CCCCCCC')THEN
            GO TO 20
         ELSE
            CONTINUE
         ENDIF
   10 CONTINUE
   20 IF(NUMP.EQ.30)THEN
C                 NO MORE POINTS AVAILABLE
         CALL SCMMAN(IWSID,46)
C                 GET STROKE
         CALL GPINST(IWSID,1,7,CLEAR,1,AREA,7,1,0,CLEAR)
         CALL GPRQST(IWSID,1,7,ISTAT,NUM,CLEAR1)
         RETURN
      ENDIF

*********************************************************************
*     GET THE POINT NAME OR DEFAULT
*********************************************************************
C                  REQUEST STRING
      CALL SCMMAN(IWSID,47)
C                  INITIALIZE THE STRING DEVICE
      CALL GPSTMO(IWSID,1,1,2)
   25 CALL GPINST(IWSID,1,7,PNDEF(NUMP),1,AREA,7,1,0,PNDEF(NUMP))
C                  INQUIRE POINT NAME
      CALL GPRQST(IWSID,1,7,ISTAT,NUM,PNAME)
      CALL GPSTMO(IWSID,1,1,2)


*********************************************************************
*     POSITION THE POINT
*********************************************************************

      CALL POSIT(IWSID,AXISR,CSIZE,LOCAT,RFLG)
      IF(RFLG.EQ.1)THEN
         GO TO 1000
      ENDIF

      POINT(NUMP,1)=LOCAT(1)
      POINT(NUMP,2)=LOCAT(2)
      POINT(NUMP,3)=LOCAT(3)

      CALL SDRAWP(IWSID,AXISR,PNAME,NUMP,POINT,SCALE)


*********************************************************************
*     ORIENT THE X AXIS
*********************************************************************
      CALL SCMMAN(IWSID,30)
      CALL SORIEN(IWSID,LOCAT,POINT,AXISR,CSIZE,XAX,RFLG)
      IF(RFLG.EQ.1)THEN
```

```
         DO 30 I=1,14
            POINT(NUMP,I)=0.0
   30    CONTINUE
         GO TO 1000
      ENDIF

      POINT(NUMP,4)=XAX(1)
      POINT(NUMP,5)=XAX(2)
      POINT(NUMP,6)=XAX(3)

      CALL SDRAWP(IWSID,AXISR,PNAME,NUMP,POINT,SCALE)

*****************************************************************
*     ORIENT THE A VECTOR IN THE X Y PLANE
*****************************************************************

      CALL SCMMAN(IWSID,36)
      CALL SORIEN(IWSID,LOCAT,POINT,AXISR,CSIZE,NAX,RFLG)
      IF(RFLG.EQ.1)THEN
         DO 40 I=1,14
            POINT(NUMP,I)=0.0
   40    CONTINUE
         GO TO 1000
      ENDIF

      CALL VPROD(XAX,NAX,ZAX)
      CALL VPROD(ZAX,XAX,YAX)
      CALL VUNIT(XAX,XAX)
      CALL VUNIT(YAX,YAX)
      CALL VUNIT(ZAX,ZAX)

      CALL SDRAWP(IWSID,AXISR,PNAME,NUMP,POINT,SCALE)

*****************************************************************
*     ADD POINT DATA TO JOINT ARRAYS
*****************************************************************

      PNAM(NUMP)=PNAME
      PNUM(NUMP,1)=NUMP
      PNUM(NUMP,2)=0
      POINT(NUMP,1)=LOCAT(1)
      POINT(NUMP,2)=LOCAT(2)
      POINT(NUMP,3)=LOCAT(3)
      POINT(NUMP,4)=XAX(1)
      POINT(NUMP,5)=XAX(2)
      POINT(NUMP,6)=XAX(3)
      POINT(NUMP,7)=YAX(1)
      POINT(NUMP,8)=YAX(2)
      POINT(NUMP,9)=YAX(3)
      POINT(NUMP,10)=ZAX(1)
      POINT(NUMP,11)=ZAX(2)
      POINT(NUMP,12)=ZAX(3)

*****************************************************************
*     DRAW THE POINT
*****************************************************************

 1000 CALL SRDRWP(IWSID,AXISR,PNAM,PNUM,POINT,SCALE)


      RETURN
      END
```

SUBROUTINE SASSO

```
      SUBROUTINE SASSO(IWSID,CSIZE)
*****************************************************************
* ----- SUBROUTINE SASSO -----                                  *
*                                                               *
*                                                               *
*   THIS SUBROUTINE IS USED TO ASSOCIATE STRUCTURE WITH THE VIEWS   *
*                                                               *
*****************************************************************
      COMMON/PNT/JOINT(20,21),PNAM(100),JNAM(20),LNAM(30),INAM(20)

      INTEGER IWSID

      REAL CSIZE(3),WIND(4),VIEW(4),WINDOC(4),VPC(4),
     &WINDOM(4),VPM(4),JOINT

      CHARACTER*7 PNAM,JNAM,LNAM,INAM
```

```
C                              WINDOW AND VIEWPORT OF TITLE SCREEN
      DATA WIND/-1.0,1.0,-1.0,1.0/
      DATA VIEW/0.0,1.0,0.0,1.0/
C                              WINDOW AND VIEWPORT OF COMMAND BOX
      DATA WINDOC/0.00,1.00,0.00,1.00/
      DATA VPC/.01,.2375,0.67,0.975/
C                              WINDOW AND VIEWPORT OF MENU BOX
      DATA WINDOM/0.00,1.00,0.00,2.8132/
      DATA VPM/.01,.2375,0.01,0.65/

*********************************************************************
*  FILL POINT AND JOINT ARRAYS
*********************************************************************

      DO 10 I=1,100
         PNAM(I)='BBBBBBB'
   10 CONTINUE
      DO 11 I=1,20
         JNAM(I)='BBBBBBB'
         INAM(I)='BBBBBBB'
   11 CONTINUE
      DO 12 I=1,30
         LNAM(I)='BBBBBBB'
   12 CONTINUE
      DO 14 I=1,20
         DO 13 J=1,21
            JOINT(I,J)=0.0
   13    CONTINUE
   14 CONTINUE

*********************************************************************
*  SET UP VIEWS AND ASSIGN STRUCTURES TO VIEWS
*********************************************************************
C                              SET UP WINDOW FOR VIEW 1
      CALL GPVMP2(IWSID,1,WIND,VIEW)
C                              SET VIEW CHARACTERISTICS
      CALL GPVCH(IWSID,1,1,1,1,1,0,2,1,2)
C                              ASSOCIATE THE STRUCTURE 1 WITH A VIEW 1
      CALL GPARV(IWSID,1,1,0.)
C                              ASSOCIATE THE STRUCTURE 2 WITH A VIEW 1
      CALL GPARV(IWSID,1,2,0.)
C                              ASSOCIATE THE STRUCTURE 3 WITH A VIEW 1
      CALL GPARV(IWSID,1,3,0.)
C                              ASSOCIATE THE STRUCTURE 4 WITH A VIEW 1
      CALL GPARV(IWSID,1,4,0.)
C                              ASSOCIATE THE STRUCTURE 8 WITH A VIEW 1
      CALL GPARV(IWSID,1,8,0.)
C                              ASSOCIATE THE STRUCTURE 10 WITH A VIEW 1
      CALL GPARV(IWSID,1,10,0.)
C                              SET VIEW PRIORITY OF VIEW1 HIGHER THAN VIEW0
      CALL GPVP(IWSID,1,0,1)

C                              SET UP WINDOW FOR COMMAND BOX
      CALL GPVMP2(IWSID,2,WINDOC,VPC)
C                              SET CHARACTERISTICS OF VIEW 2
      CALL GPVCH(IWSID,2,2,2,2,1,0,1,1,2)
C                              ASSOCIATE THE STRUCTURE 5 WITH A VIEW 2
      CALL GPARV(IWSID,2,5,0.)
C                              SET VIEW PRIORITY OF VIEW2 HIGHER THAN VIEW1
      CALL GPVP(IWSID,2,1,1)

C                              SET UP WINDOW FOR MENU BOX
      CALL GPVMP2(IWSID,3,WINDOM,VPM)
C                              SET CHARACTERISTICS OF VIEW 3
      CALL GPVCH(IWSID,3,2,2,2,1,0,1,1,2)
C                              ASSOCIATE THE STRUCTURE 6 WITH A VIEW 3
      CALL GPARV(IWSID,3,6,0.)
C                              SET VIEW PRIORITY OF VIEW3 HIGHER THAN VIEW1
      CALL GPVP(IWSID,3,1,1)

      RETURN
      END
```

SUBROUTINE SCALNM

```
      SUBROUTINE SCALNM(IWSID,SCALE)
*********************************************************************
* ----- SUBROUTINE SCALNM -----                                     *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED TO JOINT SCALE ON THE SCREEN            *
*                                                                   *
*********************************************************************

      INTEGER IWSID,LENGT,LENGF,PRFORM

      REAL POSF(2),BOX1(8),SHD1(8),POS1(2),POS2(2),AXISR
```

```
      CHARACTER*10 FILE
      CHARACTER*22 FILE1,FILE2,CLEAR

C                              FILE NAME LENGTH AND POSITION
      DATA LENGF/10/
      DATA POSF/.70,.63/
C                              FILE IDENT. LENGTH AND POSITION
      DATA LENGT/22/
      DATA POS1/.60,.69/
      DATA POS2/.60,.63/
C                              FLAG FOR UPDATE
      DATA PRFORM/2/
C                              FILE TITLE IDENTIFIER
      DATA FILE1/' SCALE          '/
      DATA FILE2/' FACTOR:        '/
      DATA CLEAR/'                '/
C                              GREY FILE BOX
      DATA BOX1/0.57,0.73,0.97,0.73,0.97,0.61,0.57,0.61/

C                              CALCULATION FOR THE TWO SHADOW BOXES
      DO 100 I=1,7,2
         SHD1(I)=BOX1(I)+0.025
  100 CONTINUE

      DO 200 I=2,8,2
         SHD1(I)=BOX1(I)-0.025
  200 CONTINUE

      REWIND 10
      WRITE(10,225)CLEAR
  225 FORMAT(A22)
      REWIND 10
      WRITE(10,250)SCALE
  250 FORMAT(F10.3)
      REWIND 10
      READ(10,275)FILE
  275 FORMAT(A10)


*********************************************************************
*  OPEN STRUCTURE FOR COMMAND BOX (ALSO PUT IN STRUCTURE 3)
*********************************************************************
C                              OPEN A STRUCTURE
      CALL GPOPST(3)
C                              SET SOLID INTEROR STYLE FOR GREY BOX
      CALL GPIS(2)
C                              SET COLOR FOR INTERIOR OF SHADOW BOX
      CALL GPICI(4)
C                              DRAW GREY SHADOW BOX
      CALL GPPG2(1,4,2,SHD1)
C                              SET COLOR FOR INTERIOR OF GREY BOX
      CALL GPICI(3)
C                              DRAW GREY COMMAND BOX
      CALL GPPG2(1,4,2,BOX1)
C                              SET COLOR FOR WHITE OF FILE NAME
      CALL GPICI(1)
C                              SET TEXT COLOR TO WHITE
      CALL GPTXCI(1)
C                              SET ANNOTATION SIZE SCALE FACTOR
      CALL GPAHSC(1.00)
C                              DRAW TITLE TEXT
      CALL GPAN2(POS1,LENGT,FILE1)
      CALL GPAN2(POS2,LENGT,FILE2)
      CALL GPAN2(POSF,LENGF,FILE)
C                              CLOSE STRUCTURE
      CALL GPCLST

*********************************************************************
*  SCREEN DISPLAY
*********************************************************************
C                              UPDATE THE WORKSTATION
      CALL GPUPWS(IWSID,PRFORM)

      RETURN
      END
```

SUBROUTINE SCHSCA

```
      SUBROUTINE SCHSCA(IWSID,AXISR,CSIZE,SCALE,PNAM,PNUM,POINT)
*********************************************************************
* ----- SUBROUTINE SCHSCA -----
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE WILL PROMPT CHANGE OF BODY POINT SCALE          *
*                                                                   *
*********************************************************************
```

```
      INTEGER IWSID,RFLG,CHOICE,PNUM(30,4)
      REAL LOCAT(3),AXISR,AREA(6),CSIZE(3),LX(2),LY(2),LZ(2),LX1(2),
     & LY1(2),LZ1(2),POINT(30,14)
      CHARACTER*7 CLEAR,AXIS,INIT,INIT1,PNAM(30)
      CHARACTER*20 CLEAR1,X,Y,Z
      CHARACTER*12 STRING,CLEAR2

      DATA CLEAR /'       '/
      DATA INIT  /'0.     '/
      DATA INIT1 /'1.     '/
      DATA STRING/'           '/
      DATA AXIS  /'       '/
      DATA CLEAR1/'                '/
      DATA CLEAR2/'           '/
      DATA Y     /'SCALE =        '/

      AREA(1)=0.08*CSIZE(1)
      AREA(2)=0.90*CSIZE(1)
      AREA(3)=0.70*CSIZE(1)
      AREA(4)=0.90*CSIZE(1)
      AREA(5)=0.02*CSIZE(1)
      AREA(6)=0.90*CSIZE(1)

      LOCAT(1)=0.0
      LOCAT(2)=0.0
      LOCAT(3)=0.0

      LY(1)=0.05
      LY(2)=0.62

      LY1(1)=0.15
      LY1(2)=0.62

      CALL GPEST(6)

*********************************************************************
*  GET SCALE FACTOR
*********************************************************************

      CALL SCMMAN(IWSID,69)

    1 CALL GPSTMO(IWSID,1,1,2)
      CALL GPINST(IWSID,1,12,INIT,1,AREA,12,1,0,CLEAR)
      STRING=CLEAR2

      CALL GPRQST(IWSID,1,12,ISTAT,NUM,STRING)
      REWIND 10
      WRITE(10,8)CLEAR1
    8 FORMAT(A20)
      REWIND 10
      WRITE(10,12)STRING
   12 FORMAT(A12)
      REWIND 10
      READ(10,*,ERR=1000)SCALE

      IF(SCALE.LE.0.)THEN
C                            VALUE IS LESS THAN 0
         CALL SCMMAN(IWSID,70)
         GO TO 1
      ENDIF

      CALL SCALNM(IWSID,SCALE)

      CALL SRDRWP(IWSID,AXISR,PNAM,PNUM,POINT,SCALE)

 1000 CALL GPSTMO(IWSID,1,1,2)
      RETURN
      END
```

SUBROUTINE SCHVIE

```
      SUBROUTINE SCHVIE(IWSID,AXISR,CSIZE,SCALE,PNAM,PNUM,POINT)
*********************************************************************
* ----- SUBROUTINE SCHVIE -----                                     *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED TO CHANGE THE MAIN VIEW                 *
*                                                                   *
*********************************************************************

      INTEGER IWSID,CHOICE,PNUM(30,4)

      REAL CSIZE(3),AXISR,POINT(30,14)

      CHARACTER*7 PNAM(30)

    5 CALL SCMMAN(IWSID,10)
      CALL SMENU(IWSID,CSIZE,3,CHOICE)
```

```
      IF(CHOICE.EQ.1)THEN
         CALL ZOOM(IWSID,AXISR,CSIZE)
         GO TO 5
      ENDIF

      IF(CHOICE.EQ.2)THEN
         CALL RESTOR(IWSID,AXISR,CSIZE)
         GO TO 5
      ENDIF

      IF(CHOICE.EQ.3)THEN
         CALL SCHSCA(IWSID,AXISR,CSIZE,SCALE,PNAM,PNUM,POINT)
         GO TO 5
      ENDIF

      IF(CHOICE.EQ.4)THEN
         CONTINUE
      ENDIF

      RETURN
      END
```

SUBROUTINE SCMMAN

```
      SUBROUTINE SCMMAN(IWSID,NUM)
*************************************************************************
* ----- SUBROUTINE SCMMAN -----                                         *
*                                                                       *
*                                                                       *
*   THIS SUBROUTINE IS USED TO DISPLAY COMMANDS FOR SYNTHSIS            *
*                                                                       *
*************************************************************************

      INTEGER IWSID,LENGT,PRFORM,NUM

      REAL POSL1(2),POSL2(2),POSL3(2),POSL4(2),POSL5(2),WINDOC(4),VPC(4)

      CHARACTER*19 C(90,5)

C                              COMMANDS
      DATA C(1,1)/'                 '/
      DATA C(1,2)/'  PICK EITHER NEW '/
      DATA C(1,3)/' MODEL OR RESTART '/
      DATA C(1,4)/'   OF A PREVIOUS  '/
      DATA C(1,5)/'       MODEL      '/

      DATA C(2,1)/'  ENTER THE NAME  '/
      DATA C(2,2)/'   OF THE MECHIN  '/
      DATA C(2,3)/'   RESTART FILE   '/
      DATA C(2,4)/'   (UPPER CASE)   '/
      DATA C(2,5)/'                  '/

      DATA C(3,1)/'                  '/
      DATA C(3,2)/'                  '/
      DATA C(3,3)/'  READING MECHIN  '/
      DATA C(3,4)/'   RESTART FILE   '/
      DATA C(3,5)/'                  '/

      DATA C(4,1)/'                  '/
      DATA C(4,2)/'   ENTER A NAME   '/
      DATA C(4,3)/'  FOR THIS MECHIN '/
      DATA C(4,4)/'    INPUT FILE    '/
      DATA C(4,5)/'   (UPPER CASE)   '/

      DATA C(5,1)/'                  '/
      DATA C(5,2)/'  ENTER THE RANGE '/
      DATA C(5,3)/' OF THE COORDINATE'/
      DATA C(5,4)/'   AXES FOR THIS  '/
      DATA C(5,5)/'       MODEL      '/

      DATA C(6,1)/' FILE NOT A VALID '/
      DATA C(6,2)/'    RESTART FILE  '/
      DATA C(6,3)/' ENTER VALID FILE '/
      DATA C(6,4)/'    (UPPER CASE)  '/
      DATA C(6,5)/'                  '/

      DATA C(7,1)/'                  '/
      DATA C(7,2)/'     PLEASE PICK  '/
      DATA C(7,3)/'       TYPE OF    '/
      DATA C(7,4)/'      DATA INPUT  '/
      DATA C(7,5)/'                  '/

      DATA C(8,1)/'    FILE ALREADY  '/
      DATA C(8,2)/'       EXISTS     '/
      DATA C(8,3)/'RE-ENTER FILE NAME'/
      DATA C(8,4)/'TO USE IT OR ENTER'/
      DATA C(8,5)/'  A NEW FILE NAME '/

      DATA C(9,1)/'    FILE DOES NOT '/
```

```
DATA C(9,2)/'        EXIST       '/
DATA C(9,3)/' ENTER VALID FILE   '/
DATA C(9,4)/'   (UPPER CASE)     '/
DATA C(9,5)/'                    '/

DATA C(10,1)/'                   '/
DATA C(10,2)/'       SELECT      '/
DATA C(10,3)/'  DESIRED OPTION   '/
DATA C(10,4)/'                   '/
DATA C(10,5)/'                   '/

DATA C(11,1)/'  INPUT LOCATION   '/
DATA C(11,2)/'     DIRECTLY      '/
DATA C(11,3)/'(ENTER X LOCATION)'/
DATA C(11,4)/'        OR         '/
DATA C(11,5)/'  USE VALUATORS    '/

DATA C(12,1)/'  END OF PROGRAM   '/
DATA C(12,2)/'     FOR NOW       '/
DATA C(12,3)/'                   '/
DATA C(12,4)/' PRESS ANY BUTTON  '/
DATA C(12,5)/'     TO QUIT       '/

DATA C(13,1)/' ENTER THE NAME OF '/
DATA C(13,2)/'   THE POINT OR    '/
DATA C(13,3)/'  PRESS ENTER TO   '/
DATA C(13,4)/'    USE DEFAULT    '/
DATA C(13,5)/'                   '/

DATA C(14,1)/'    POINT NAME     '/
DATA C(14,2)/'  ALREADY EXISTS   '/
DATA C(14,3)/'                   '/
DATA C(14,4)/' ENTER NEW NAME OR '/
DATA C(14,5)/'  CHOOSE DEFAULT   '/

DATA C(15,1)/'    POINT NAME     '/
DATA C(15,2)/'  CAN NOT BE USED  '/
DATA C(15,3)/'                   '/
DATA C(15,4)/' ENTER NEW NAME OR '/
DATA C(15,5)/'  CHOOSE DEFAULT   '/

DATA C(16,1)/'NUMBER OF POINTS   '/
DATA C(16,2)/'    EXCEEDS THE    '/
DATA C(16,3)/'MAXIMUM ALLOWABLE  '/
DATA C(16,4)/'                   '/
DATA C(16,5)/'    PRESS ENTER    '/

DATA C(17,1)/' POINT IS OUTSIDE  '/
DATA C(17,2)/'  THE PREDEFINED   '/
DATA C(17,3)/' AXIS RANGE. PICK  '/
DATA C(17,4)/' RE-ENTER AXIS OR  '/
DATA C(17,5)/'  RE-ENTER POINT   '/

DATA C(18,1)/'                   '/
DATA C(18,2)/'      INPUT        '/
DATA C(18,3)/'        Y          '/
DATA C(18,4)/'     LOCATION      '/
DATA C(18,5)/'                   '/

DATA C(19,1)/'                   '/
DATA C(19,2)/'      INPUT        '/
DATA C(19,3)/'        Z          '/
DATA C(19,4)/'     LOCATION      '/
DATA C(19,5)/'                   '/

DATA C(20,1)/'  ENTER POINT OR   '/
DATA C(20,2)/'     RETURN        '/
DATA C(20,3)/'   TO RE-ENTER     '/
DATA C(20,4)/'   POINT DATA      '/
DATA C(20,5)/'                   '/

DATA C(21,1)/'                   '/
DATA C(21,2)/'  PICK POINT TO    '/
DATA C(21,3)/'     DELETE        '/
DATA C(21,4)/'                   '/
DATA C(21,5)/'                   '/

DATA C(22,1)/'  PICK POINT OR    '/
DATA C(22,2)/'     JOINT         '/
DATA C(22,3)/'  FOR POSITION     '/
DATA C(22,4)/'    INQUIRY        '/
DATA C(22,5)/'                   '/

DATA C(23,1)/'   PICK TYPE OF    '/
DATA C(23,2)/'    SYNTHESIS      '/
DATA C(23,3)/'    FILE TO BE     '/
DATA C(23,4)/'     WRITTEN       '/
DATA C(23,5)/'                   '/

DATA C(24,1)/'  OPTION NOT YET   '/
DATA C(24,2)/'    AVAILABLE      '/
DATA C(24,3)/'                   '/
DATA C(24,4)/'  PLEASE SELECT    '/
DATA C(24,5)/'   NEW OPTION      '/

DATA C(25,1)/'   PICK THE POINT  '/
```

```
      DATA C(25,2)/'  THAT REPRESENTS '/
      DATA C(25,3)/'        THE       '/
      DATA C(25,4)/'       FIRST      '/
      DATA C(25,5)/'   BODY POSITION  '/

      DATA C(26,1)/'   PICK THE POINT '/
      DATA C(26,2)/'  THAT REPRESENTS '/
      DATA C(26,3)/'        THE       '/
      DATA C(26,4)/'      SECOND      '/
      DATA C(26,5)/'   BODY POSITION  '/

      DATA C(27,1)/'                  '/
      DATA C(27,2)/'     INPUT        '/
      DATA C(27,3)/'       X          '/
      DATA C(27,4)/' DIRECTION COSINE '/
      DATA C(27,5)/'                  '/

      DATA C(28,1)/'                  '/
      DATA C(23,2)/'     INPUT        '/
      DATA C(28,3)/'       Y          '/
      DATA C(28,4)/' DIRECTION COSINE '/
      DATA C(28,5)/'                  '/

      DATA C(29,1)/'                  '/
      DATA C(29,2)/'     INPUT        '/
      DATA C(29,3)/'       Z          '/
      DATA C(29,4)/' DIRECTION COSINE '/
      DATA C(29,5)/'                  '/

      DATA C(30,1)/'       PICK       '/
      DATA C(30,2)/'     METHOD OF    '/
      DATA C(30,3)/'    ORIENTATION   '/
      DATA C(30,4)/'      OF THE      '/
      DATA C(30,5)/'      X AXIS      '/

      DATA C(31,1)/'      VALUE       '/
      DATA C(31,2)/'   GREATER THAN   '/
      DATA C(31,3)/'       1.0        '/
      DATA C(31,4)/'                  '/
      DATA C(31,5)/'  PLEASE RE-ENTER '/

      DATA C(32,1)/'ENTER ORIENTATION '/
      DATA C(32,2)/'    OR RETURN     '/
      DATA C(32,3)/'   TO RE-ENTER    '/
      DATA C(32,4)/'   ORIENTATION    '/
      DATA C(32,5)/'      DATA        '/

      DATA C(33,1)/'ENTER ORIENTATION '/
      DATA C(33,2)/'  WITH VALUATORS  '/
      DATA C(33,3)/'   OR RETURN TO   '/
      DATA C(33,4)/'     RE-ENTER     '/
      DATA C(33,5)/'      DATA        '/

      DATA C(34,1)/'   PICK THE POINT '/
      DATA C(34,2)/'  THAT REPRESENTS '/
      DATA C(34,3)/'        THE       '/
      DATA C(34,4)/'       THIRD      '/
      DATA C(34,5)/'   BODY POSITION  '/

      DATA C(35,2)/'ENTER THE POSITION'/
      DATA C(35,3)/' OF THE SPHERICAL '/
      DATA C(35,4)/'  JOINT DIRECTLY  '/
      DATA C(35,5)/' OR WITH VALUATORS'/

      DATA C(36,1)/'      PICK        '/
      DATA C(36,2)/'    METHOD OF     '/
      DATA C(36,3)/'   ORIENTATION    '/
      DATA C(36,4)/' VECTOR IN THE    '/
      DATA C(36,5)/'    X Y PLANE     '/

      DATA C(37,1)/'    PICK AXIS     '/
      DATA C(37,2)/'       FOR        '/
      DATA C(37,3)/'   ORIENTATION    '/
      DATA C(37,4)/'     INQUIRY      '/
      DATA C(37,5)/'                  '/

      DATA C(38,1)/'  ZERO MAGNITUDE  '/
      DATA C(38,2)/'     VECTOR       '/
      DATA C(38,3)/'     ENTERED      '/
      DATA C(38,4)/'                  '/
      DATA C(38,5)/'    RE-ENTER X    '/

      DATA C(39,1)/'  ZERO MAGNITUDE  '/
      DATA C(39,2)/'     VECTOR       '/
      DATA C(39,3)/'     ENTERED      '/
      DATA C(39,4)/'                  '/
      DATA C(39,5)/' RE-ENTER VALUES  '/

      DATA C(40,1)/'  PICK METHOD OF  '/
      DATA C(40,2)/'   ORIENTATION    '/
      DATA C(40,3)/'  OF THE SECOND   '/
      DATA C(40,4)/'  REVOLUTE JOINT  '/
      DATA C(40,5)/'       AXIS       '/

      DATA C(41,1)/'     ENTER THE    '/
      DATA C(41,2)/'       LEAD       '/
```

```
DATA C(41,3)/'      FOR THIS      '/
DATA C(41,4)/'       SCREW        '/
DATA C(41,5)/'       JOINT        '/

DATA C(42,1)/'  SECOND Z ORIENT.  '/
DATA C(42,2)/'VECTOR IS NEED FOR  '/
DATA C(42,3)/'  A SPHERICAL JOINT '/
DATA C(42,4)/'                    '/
DATA C(42,5)/'                    '/

DATA C(46,1)/' NUMBER OF POINTS   '/
DATA C(46,2)/'   EXCEEDS THE      '/
DATA C(46,3)/'MAXIMUM ALLOWABLE   '/
DATA C(46,4)/'                    '/
DATA C(46,5)/'    PRESS ENTER     '/

DATA C(47,1)/' ENTER THE NAME OF  '/
DATA C(47,2)/'    THE POINT OR    '/
DATA C(47,3)/'   PRESS ENTER TO   '/
DATA C(47,4)/'     USE DEFAULT    '/
DATA C(47,5)/'                    '/

DATA C(51,1)/'   CHOOSE METHOD    '/
DATA C(51,2)/'     OF LOCAL       '/
DATA C(51,3)/'       X AXIS       '/
DATA C(51,4)/'    ORIENTATION     '/
DATA C(51,5)/'                    '/

DATA C(52,1)/'    PICK A POINT    '/
DATA C(52,2)/'   IN THE POSITIVE  '/
DATA C(52,3)/'      X-Z PLANE     '/
DATA C(52,4)/'   OF THE DESIRED   '/
DATA C(52,5)/'    X ORIENTATION   '/

DATA C(53,1)/'    PICK A JOINT    '/
DATA C(53,2)/'   IN THE POSITIVE  '/
DATA C(53,3)/'      X-Z PLANE     '/
DATA C(53,4)/'   OF THE DESIRED   '/
DATA C(53,5)/'    X ORIENTATION   '/

DATA C(54,1)/'     CENTER OF      '/
DATA C(54,2)/'    JOINT CHOSEN    '/
DATA C(54,3)/' IS ON THE Z AXIS.  '/
DATA C(54,4)/'        PICK        '/
DATA C(54,5)/'    A NEW JOINT     '/

DATA C(55,1)/'     CENTER OF      '/
DATA C(55,2)/'    POINT CHOSEN    '/
DATA C(55,3)/' IS ON THE Z AXIS.  '/
DATA C(55,4)/'        PICK        '/
DATA C(55,5)/'    A NEW POINT     '/

DATA C(57,1)/'                    '/
DATA C(57,2)/'PICK FIRST CORNER   '/
DATA C(57,3)/'       OF THE       '/
DATA C(57,4)/'    ZOOM AREA       '/
DATA C(57,5)/'                    '/

DATA C(58,1)/'PICK THE OPPOSITE   '/
DATA C(58,2)/'       CORNER       '/
DATA C(58,3)/'       OF THE       '/
DATA C(58,4)/'    ZOOM AREA       '/
DATA C(58,5)/'                    '/

DATA C(60,1)/'    ENTER A NAME    '/
DATA C(60,2)/'     FOR MECHIN     '/
DATA C(60,3)/'    RESTART FILE    '/
DATA C(60,4)/'    (UPPER CASE)    '/
DATA C(60,5)/'                    '/

DATA C(69,1)/'     ENTER THE      '/
DATA C(69,2)/'    SCALE FACTOR    '/
DATA C(69,3)/' FOR THE POSITION   '/
DATA C(69,4)/'        ICONS       '/
DATA C(69,5)/'                    '/

DATA C(70,1)/'   SCALE ENTERED    '/
DATA C(70,2)/'     IS LESS THAN   '/
DATA C(70,3)/' OR EQUAL TO ZERO   '/
DATA C(70,4)/'                    '/
DATA C(70,5)/'   PLEASE RE-ENTER  '/

DATA C(83,1)/'    ENTER A NAME    '/
DATA C(83,2)/'      FOR RSCR      '/
DATA C(83,3)/'     DATA FILE      '/
DATA C(83,4)/'    (UPPER CASE)    '/
DATA C(83,5)/'                    '/

DATA C(84,1)/' ORIENTATION POINT  '/
DATA C(84,2)/' AND JOINT ORIGIN   '/
DATA C(84,3)/'    ARE THE SAME    '/
DATA C(84,4)/'                    '/
DATA C(84,5)/' PICK A NEW POINT   '/

DATA C(85,1)/'     ENTER A NEW    '/
DATA C(85,2)/'      MECHANISM     '/
```

```
      DATA C(85,3)/'        OR        '/
      DATA C(85,4)/'       QUIT       '/
      DATA C(85,5)/'                  '/

C                               COMMAND LENGTH AND LINE POSITIONS
      DATA LENGT/19/
      DATA POSL1/.03,.7/
      DATA POSL2/.03,.6/
      DATA POSL3/.03,.5/
      DATA POSL4/.03,.4/
      DATA POSL5/.03,.3/
C                               FLAG FOR UPDATE
C     DATA PRFORM/2/            WINDOW AND VIEWPORT OF COMMAND BOX
      DATA WINDOC/0.00,1.00,0.00,1.00/
      DATA VPC/.01,.2375,0.67,0.975/

*************************************************************************
*  DELETE ALL PREVIOUS COMMENTS
*************************************************************************
C                               DELETE ALL PREVIOUS COMMENTS
      CALL GPEST(5)

*************************************************************************
*  OPEN STRUCTURE FOR COMMAND1
*************************************************************************
C                               OPEN A STRUCTURE
      CALL GPOPST(5)
C                               SET TEXT COLOR TO YELLOW
C     CALL GPTXCI(2)
C                               SET ANNOTATION SIZE SCALE FACTOR
C     CALL GPAHSC(1.0)
C                               DRAW TEXT
      CALL GPAN2(POSL1,LENGT,C(NUM,1))
      CALL GPAN2(POSL2,LENGT,C(NUM,2))
      CALL GPAN2(POSL3,LENGT,C(NUM,3))
      CALL GPAN2(POSL4,LENGT,C(NUM,4))
      CALL GPAN2(POSL5,LENGT,C(NUM,5))
C                               CLOSE STRUCTURE
      CALL GPCLST

*************************************************************************
*  SCREEN DISPLAY---------- VIEW INITIALIZATION GOES TO SETUP
*************************************************************************
C                               UPDATE THE WORKSTATION
      CALL GPUPWS(IWSID,PRFORM)

      RETURN
      END
```

SUBROUTINE SCRN1

```
      SUBROUTINE SCRN1(IWSID,CSIZE)
*************************************************************************
* ----- SUBROUTINE SCRN1 -----                                          *
*                                                                       *
*                                                                       *
*   THIS SUBROUTINE IS USED TO DISPLAY SCREEN 1 --------                *
*                                                                       *
*                  MECHIN TITLE SCREEN                                  *
*                                                                       *
*************************************************************************

      INTEGER IWSID,LNGTH1,STRUCT,STROKE,PRFORM,LD,ACLASS(1)


      REAL HGHT1,POS1(2),BOXPTS(8),SHDPTS(8),MATH1(4,4),ANG,TRAT1(3),
     &MATTT1(4,4),MATTB1(4,4),MATTT2(4,4),MATTB2(4,4),TEMP(4,4),
     &MATR2(4,4),TRAB1(3),TRAT2(3),TRAB2(3),X(3),Y(3),WIND(4),VIEW(6),
     &POIN(3),MATR1(4,4),POD1(2),POD2(2),POD3(2),HGHTD,POSM(2),CSIZE(3),
     &PAREA(6),BX2(8),SH2(8)

      CHARACTER*7 TEXT1,CLEAR
      CHARACTER*30 MSG,DESC1,DESC2,DESC3


C                               TITLE TEXT, LENGTH, HEIGHT, AND POSITION
      DATA TEXT1/'MECHIN '/
      DATA CLEAR/'       '/
      DATA LNGTH1/7/
      DATA HGHT1/.3/
      DATA POS1/-.93,-.162/

C                               DESCRIPTION LENGTH, HEIGHT, AND POSITION
      DATA DESC1/'A GRAPHICAL PREPROCESSOR FOR  '/
      DATA DESC2/'      SPATIAL MECHANISM       '/
      DATA DESC3/'   ANALYSIS AND SYNTHESIS     '/
```

```
      DATA LD/30/
      DATA HGHTD/.05/
      DATA POD1/-.88,-.35/
      DATA POD2/-.88,-.45/
      DATA POD3/-.88,-.55/
C                             MESSAGE FOR CONTINUE AND LENGTH
      DATA MSG  /'  PRESS ENTER TO CONTINUE      '/
      DATA LNGTHM/30/
      DATA POSM/-.97,-.92/

      DATA ACLASS/1/

C                             IDENTITY MATRIX 1
      DATA MATR1/1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
C                             IDENTITY MATRIX 2
      DATA MATR2/1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
C                             IDENTITY MATRIX  TRANS. TO
      DATA MATTT1/1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
C                             IDENTITY MATRIX  TRANS. BACK
      DATA MATTB1/1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
C                             IDENTITY MATRIX  TRANS. TO
      DATA MATTT2/1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
C                             IDENTITY MATRIX  TRANS. BACK
      DATA MATTB2/1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
C                             TITLE STRUCTURE NUMBER
C     DATA STRUCT/1/
C                             STROKE PRECISION VALUE
C     DATA STROKE/3/
C                             FLAG FOR UPDATE
C     DATA PRFORM/2/
C                             GREY TITLE BOX
C     DATA BOXPTS/-.93,-.195,.93,-.195,.93,.150,-.93,.150/
C                             DARK GREY SHADOW BOX
C     DATA SHDPTS/-.88,-.245,0.98,-.245,0.98,.100,-.88,.100/
C                             GREY PICK BOX
C     DATA BX2/-.95,-.94,-.35,-.94,-.35,-.88,-.95,-.88/
C                             GREY PICK SHADOW
C     DATA SH2/-.92,-.97,-.32,-.97,-.32,-.91,-.92,-.91/
C                             ROTATION ANGLE
C     DATA ANG/0.02/
C                             SHADOW TRANSLATION
      DATA TRAT1/0.,0.0725,0./
      DATA TRAB1/0.,-0.0725,0./
      DATA TRAT2/0.,0.0275,0./
      DATA TRAB2/0.,-0.0275,0./
      DATA X/1,0,0/
      DATA Y/0,1,0/
      DATA IDEV/0/

*************************************************************************
*  OPEN STRUCTURE FOR THE TITLE
*************************************************************************
C                             OPEN A STRUCTURE FOR SHADOW
      CALL GPOPST(1)
C                             SET INITIAL TRANSFORM MATRICES
      CALL GPMLX3(MATR1,3)
C                             SET SOLID INTERIOR STYLE SOLID
      CALL GPIS(2)
C                             SET COLOR FOR INTERIOR OF SHADOW BOX
      CALL GPICI(4)
C                             DRAW GREY SHADOW BOX
      CALL GPPG2(1,4,2,SHDPTS)
C                             CLOSE STRUCTURE
      CALL GPCLST


C                             OPEN A STRUCTURE FOR TITLE BOX
      CALL GPOPST(2)
C                             SET INITIAL TRANSFORM MATRICES
      CALL GPMLX3(MATR2,3)
C                             SET SOLID INTERIOR STYLE FOR GREY BOX
      CALL GPIS(2)
C                             SET COLOR FOR INTERIOR OF GREY BOX
      CALL GPICI(3)
C                             DRAW GREY TITLE BOX
      CALL GPPG2(1,4,2,BOXPTS)
C                             SET TEXT PRECISION TO STROKE
      CALL GPTXPR(STROKE)
C                             SET CHARACTER HEIGHT
      CALL GPCHH(HGHT1)
C                             SET TEXT COLOR
      CALL GPTXCI(2)
C                             ACTIVATE TEXT FONT
      CALL GPACFO(IWSID,1,11)
C                             SET TEXT FONT
      CALL GPTXFO(11)
C                             INSERT TEXT PRIMITIVE
      CALL GPTX2(POS1,LNGTH1,TEXT1)
C                             CLOSE STRUCTURE
      CALL GPCLST
```

```
C                               OPEN A STRUCTURE FOR DESCRIPTION
      CALL GPOPST(3)
C                               SET TEXT PRECISION TO STROKE
      CALL GPTXPR(STROKE)
C                               SET CHARACTER HEIGHT
      CALL GPCHH(HGHTD)
C                               SET TEXT COLOR
      CALL GPTXCI(1)
C                               ACTIVATE TEXT FONT
      CALL GPACFO(IWSID,1,3)
C                               SET TEXT FONT
      CALL GPTXFO(3)
C                               INSERT TEXT PRIMITIVE
      CALL GPTX2(POD1,LD,DESC1)
      CALL GPTX2(POD2,LD,DESC2)
      CALL GPTX2(POD3,LD,DESC3)
C                               CLOSE STRUCTURE
      CALL GPCLST

C                               OPEN A STRUCTURE FOR CHOICES
      CALL GPOPST(4)
C                               SET SOLID INTERIOR STYLE FOR GREY BOX
      CALL GPIS(2)
C                               SET COLOR FOR INTERIOR OF SHADOW BOX
      CALL GPICI(4)
C                               DRAW GREY SHADOW BOX
      CALL GPPG2(1,4,2,SH2)
C                               SET COLOR FOR INTERIOR OF GREY BOX
      CALL GPICI(3)
C                               DRAW GREY TITLE BOX
      CALL GPPG2(1,4,2,BX2)
C                               INSERT CLASS NAME TO ALLOW PICKING MENU ITEM
      CALL GPADCN(1,ACLASS)
C                               SET TEXT COLOR
      CALL GPTXCI(2)
C                               SET ANNOTATION SIZE SCALE FACTOR
      CALL GPAHSC(1.0)
C                               SET PICK IDENTIFIER
      CALL GPPKID(1)
C                               DRAW TEXT
      CALL GPAN2(POSM,LNGTHM,MSG)
C                               CLOSE STRUCTURE
      CALL GPCLST

*****************************************************************
*  SET STRING PARAMETERS
*****************************************************************
C                               MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPSTMO(IWSID,1,1,2)
C                               DEFINE PICK AREA
      PAREA(1)=0.
      PAREA(2)=CSIZE(1)
      PAREA(3)=0.
      PAREA(4)=CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=CSIZE(3)
C                               INTIALIZE PICK PATH
      CALL GPINST(IWSID,1,7,CLEAR,1,PAREA,7,7,32,DATA)
C                               PLACE PICK DEVICE IN THE EVENT MODE
      CALL GPSTMO(IWSID,1,3,2)

*****************************************************************
*  SCREEN DISPLAY
*****************************************************************
C                               UPDATE THE WORKSTATION
      CALL GPUPWS(IWSID,PRFORM)

*****************************************************************
*  GO ON OR CONTINUE ROTATION
*****************************************************************
C                               FORMULATE TRANSLATION MATRICES
      CALL GPTRL3(TRAT1,MATTT1)
      CALL GPTRL3(TRAB1,MATTB1)
      CALL GPTRL3(TRAT2,MATTT2)
      CALL GPTRL3(TRAB2,MATTB2)
C                               REQUEST INPUT
  100 CALL GPAWEV(0.0,IIWSID,ICLS,IDEV)
C                               ROTATE OR CONTINUE
      IF(ICLS.EQ.6)THEN
         GO TO 200
      ENDIF
         CALL GPROTX(ANG,MATR1)
         CALL GPCMT3(MATTT1,MATR1,TEMP)
         CALL GPCMT3(TEMP,MATTB1,MATR1)
         CALL GPOPST(1)
         CALL GPEP(1)
         CALL GPDLE
         CALL GPMLX3(MATR1,3)
         CALL GPCLST
```

```
         CALL GPROTX(ANG,MATR2)
         CALL GPCMT3(MATTT2,MATR2,TEMP)
         CALL GPCMT3(TEMP,MATTB2,MATR2)
         CALL GPOPST(2)
         CALL GPEP(1)
         CALL GPDLE
         CALL GPMLX3(MATR2,3)
         CALL GPCLST

C                              TRAVERSE ALL ACTIVE VIEWS
         CALL GPUPWS(IWSID,PRFORM)

         ANG=ANG+0.12
         GO TO 100
C     ENDIF

C                              DELETE ALL STRUCTURES
  200 CALL GPEST(1)
      CALL GPEST(2)
      CALL GPEST(3)
      CALL GPEST(4)
C                              DEACTIVATE PICK
      CALL GPPKMO(IWSID,1,1,2)
      CALL GPUPWS(IWSID,PRFORM)

      RETURN
      END
```

SUBROUTINE SCRN2

```
      SUBROUTINE SCRN2(IWSID,CSIZE,CHOICE)
*********************************************************************
* ----- SUBROUTINE SCRN2 -----                                      *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED TO DISPLAY SCREEN 1 --------            *
*                                                                   *
*                  ANALYSIS OR SYNTHESIS CHOICE                     *
*                                                                   *
*********************************************************************

      INTEGER IWSID,LENG,LENGT,STRUCT,STROKE,PRFORM,PPATH(3),ACLASS(1),
     &CHOICE

      REAL HGHT1,POSN(2),POSNT(2),BOXPTS(8),SHDPTS(8),CSIZE(3),
     &PAREA(6),DATA(35)

      CHARACTER*35 TMENU(2),TITLE

C                              TITLE TEXT, LENGTH, HEIGHT, AND POSITION
      DATA TMENU(1)/'1. INPUT FOR ANALYSIS                '/
      DATA TMENU(2)/'2. INPUT FOR SYNTHESIS               '/
      DATA TITLE/'PLEASE CHOOSE AN INPUT OPTION      '/
      DATA LENG/35/
      DATA LENGT/35/
      DATA POSNT/-.55,.100/
      DATA DATA/32,0,0,          2,2,1,1,
     &                         1,1,1,1,1,1,
     &                         1,1,1,1,1,1,
     &                         1,1,1,1,1,1,
     &                           1,1,1,1/
C                              FLAG FOR UPDATE
      DATA PRFORM/2/
C                              INCLUSION CLASS LIST
      DATA ACLASS/1/
C                              GREY TITLE BOX
      DATA BOXPTS/-.93,-.195,.93,-.195,.93,.150,-.93,.150/
C                              DARK GREY SHADOW BOX
      DATA SHDPTS/-.88,-.245,0.98,-.245,0.98,.100,-.88,.100/

      POSN(1)=-.23
      POSN(2)=-.010

*********************************************************************
*  OPEN STRUCTURE FOR THE CHOICE BOX
*********************************************************************
C                              OPEN A STRUCTURE FOR THE BACKGROUND
      CALL GPOPST(1)
C                              SET SOLID INTEROR STYLE FOR GREY BOX
      CALL GPIS(2)
C                              SET COLOR FOR INTERIOR OF SHADOW BOX
      CALL GPICI(4)
C                              DRAW GREY SHADOW BOX
      CALL GPPG2(1,4,2,SHDPTS)
C                              SET COLOR FOR INTERIOR OF GREY BOX
```

```
      CALL GPICI(3)
C                               DRAW GREY CHOICE BOX
      CALL GPPG2(1,4,2,BOXPTS)
C                               SET TEXT COLOR TO YELLOW
      CALL GPTXCI(2)
C                               SET ANNOTATION SIZE SCALE FACTOR
      CALL GPAHSC(2.0)
C                               DRAW TEXT
      CALL GPAN2(POSNT,LENGT,TITLE)
C                               CLOSE STRUCTURE
      CALL GPCLST


C                               OPEN A STRUCTURE FOR CHOICES
      CALL GPOPST(2)
C                               INSERT CLASS NAME TO ALLOW PICKING MENU ITEM
      CALL GPADCN(1,ACLASS)
C                               SET TEXT COLOR TO YELLOW
      CALL GPTXCI(2)
C                               SET ANNOTATION SIZE SCALE FACTOR
      CALL GPAHSC(1.0)

      DO 100 I=1,2
C                               SET PICK IDENTIFIER
         CALL GPPKID(I)
C                               DRAW TEXT
         CALL GPAN2(POSN,LENG,TMENU(I))
C                               SET NEXT POSITION
         POSN(2)=POSN(2)-.09
  100 CONTINUE
C                               CLOSE STRUCTURE
      CALL GPCLST

*********************************************************************
*  SCREEN DISPLAY
*********************************************************************
C                               UPDATE THE WORKSTATION
      CALL GPUPWS(IWSID,PRFORM)

*********************************************************************
*  REQUEST PICK INPUT
*********************************************************************
C                               SET PICK FILTER(ALL CLASS 1 DETECTABLE)
      CALL GPPKF(IWSID,1,1,ACLASS,0,ACLASS)
C                               MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                               DEFINE PICK AREA
      PAREA(1)=((BOXPTS(1)+1.)/2.)*CSIZE(1)
      PAREA(2)=((BOXPTS(3)+1.)/2.)*CSIZE(1)
      PAREA(3)=((BOXPTS(2)+1.)/2.)*CSIZE(2)
      PAREA(4)=((BOXPTS(8)+1.)/2.)*CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=((BOXPTS(8)+1.)/2.)*CSIZE(3)
C                               INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                               PLACE PICK DEVICE IN THE EVENT MODE
      CALL GPPKMO(IWSID,1,3,2)
C                               TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)

*********************************************************************
*  AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
*********************************************************************
C                               AWAIT AN EVENT
  200 CALL GPAWEV(1000.,IWS,ICLA,IDEV)
      IF(ICLA.EQ.5)THEN
C                               GET PICK
         CALL GPGTPK(1,1,PPATH)
C                               CHOICE IS SECOND ITEM OF PICK PATH
         CHOICE=PPATH(2)
C                               DEACTIVATE PICK
         CALL GPPKMO(IWSID,1,1,2)
C                               DELETE ALL STRUCTURES
         CALL GPEST(1)
         CALL GPEST(2)
C                               RETURN TO SETUP
         RETURN
      ELSE
C                               IF NOT A PICK, GO TO AWAIT AN EVENT
         GO TO 200
      ENDIF



C
CTEMPORARY STOP(REMOVE AND FIX RETURN)
C     CALL GPCHMO(IWSID,1,1,1)
C     CALL GPRQCH(IWSID,1,ISTS,ICHO)
C
```

```
C
C     RETURN
      END
```

SUBROUTINE SCRN3

```
      SUBROUTINE SCRN3(IWSID)
*********************************************************************
* ----- SUBROUTINE SCRN3 -----                                      *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED TO DISPLAY SCREEN 3 --------            *
*                                                                   *
*                  ANALYSIS BACKGROUND SCREEN                       *
*                                                                   *
*********************************************************************

      INTEGER IWSID,LENGT,LENCT,STROKE,PRFORM

      REAL POSNT(2),BOX1(8),BOX2(8),SHD2(8),BOX3(8),SHD3(8),BOX4(8),
     &BOX51(8),BOX52(8),POSCT(2)
      CHARACTER*35 TITLE
      CHARACTER*8 CTITL

C                              TITLE TEXT, LENGTH, HEIGHT, AND POSITION
      DATA CTITL/'ANALYSIS'/
      DATA LENCT/8/
      DATA POSCT/-.882,.89/
C                              FLAG FOR UPDATE
      DATA PRFORM/2/
C                              GREY MAIN SCREEN BOX
      DATA BOX1/-0.48,-0.98,0.98,-0.98,0.98,0.48,-0.48,0.48/
C                              GREY SIDE STRIP
      DATA BOX2/-0.98,-0.95,-0.525,-0.95,-0.525,0.33,-0.98,0.33/
C                              GREY COMMAND BOX
      DATA BOX3/-0.98,0.37,-0.525,0.37,-0.525,0.98,-0.98,0.98/
C                              GREY POSITION BOX
      DATA BOX4/-0.48,0.52,-0.02,0.52,-0.02,0.98,-0.48,0.98/
C                              WHITE TITLE IN COMMAND BOX
      DATA BOX51/-0.96,0.90,-0.90,0.90,-0.90,0.91,-0.96,0.91/
      DATA BOX52/-0.605,0.90,-0.545,0.90,-0.545,0.91,-0.605,0.91/

C                              CALCULATION FOR THE TWO SHADOW BOXES
      DO 100 I=1,7,2
         SHD2(I)=BOX2(I)+0.025
         SHD3(I)=BOX3(I)+0.025
  100 CONTINUE

      DO 200 I=2,8,2
         SHD2(I)=BOX2(I)-0.025
         SHD3(I)=BOX3(I)-0.025
  200 CONTINUE

*********************************************************************
*  OPEN STRUCTURE FOR THE DISPLAY BOX
*********************************************************************
C                              OPEN A STRUCTURE
      CALL GPOPST(1)
C                              SET SOLID INTEROR STYLE FOR GREY BOX
      CALL GPIS(2)
C                              SET COLOR FOR INTERIOR OF GREY BOX
      CALL GPICI(3)
C                              DRAW GREY DISPLAY BOX
      CALL GPPG2(1,4,2,BOX1)
C                              CLOSE STRUCTURE
      CALL GPCLST


*********************************************************************
*  OPEN STRUCTURE FOR THE MENU BOX
*********************************************************************
C                              OPEN A STRUCTURE
      CALL GPOPST(2)
C                              SET SOLID INTEROR STYLE FOR GREY BOX
      CALL GPIS(2)
C                              SET COLOR FOR INTERIOR OF SHADOW BOX
      CALL GPICI(4)
C                              DRAW GREY SHADOW BOX
      CALL GPPG2(1,4,2,SHD2)
C                              SET COLOR FOR INTERIOR OF GREY BOX
      CALL GPICI(3)
C                              DRAW GREY TITLE BOX
      CALL GPPG2(1,4,2,BOX2)
C                              CLOSE STRUCTURE
      CALL GPCLST
```

```
*********************************************************************
*  OPEN STRUCTURE FOR COMMAND BOX
*********************************************************************
C                               OPEN A STRUCTURE
      CALL GPOPST(3)
C                               SET SOLID INTEROR STYLE FOR GREY BOX
      CALL GPIS(2)
C                               SET COLOR FOR INTERIOR OF SHADOW BOX
      CALL GPICI(4)
C                               DRAW GREY SHADOW BOX
      CALL GPPG2(1,4,2,SHD3)
C                               SET COLOR FOR INTERIOR OF GREY BOX
      CALL GPICI(3)
C                               DRAW GREY COMMAND BOX
      CALL GPPG2(1,4,2,BOX3)
C                               SET COLOR FOR WHITE OF COMMAND TITLE
      CALL GPICI(1)
C                               DRAW WHITE BOXES IN TITLE
      CALL GPPG2(1,4,2,BOX51)
      CALL GPPG2(1,4,2,BOX52)
C                               SET TEXT COLOR TO WHITE
      CALL GPTXCI(1)
C                               SET ANNOTATION SIZE SCALE FACTOR
      CALL GPAHSC(1.25)
C                               DRAW TITLE TEXT
      CALL GPAN2(POSCT,LENCT,CTITL)
C                               CLOSE STRUCTURE
      CALL GPCLST


*********************************************************************
*  OPEN STRUCTURE FOR VALUATOR BOX
*********************************************************************
C                               OPEN A STRUCTURE
      CALL GPOPST(4)
C                               SET SOLID INTEROR STYLE FOR GREY BOX
      CALL GPIS(2)
C                               SET COLOR FOR INTERIOR OF GREY BOX
      CALL GPICI(3)
C                               DRAW GREY VALUATOR BOX
      CALL GPPG2(1,4,2,BOX4)
C                               CLOSE STRUCTURE
      CALL GPCLST


*********************************************************************
*  SCREEN DISPLAY
*********************************************************************
C                               UPDATE THE WORKSTATION
      CALL GPUPWS(IWSID,PRFORM)

      RETURN
      END
```

SUBROUTINE SCRN4

```
      SUBROUTINE SCRN4(IWSID)
*********************************************************************
* ----- SUBROUTINE SCRN4 -----                                      *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED TO DISPLAY SCREEN 4 --------            *
*                                                                   *
*                 SYNTHESIS BACKGROUND SCREEN                       *
*                                                                   *
*********************************************************************
      INTEGER IWSID,LENGT,LENCT,STROKE,PRFORM

      REAL POSNT(2),BOX1(8),BOX2(8),SHD2(8),BOX3(8),SHD3(8),BOX4(8),
     &BOX51(8),BOX52(8),POSCT(2)
      CHARACTER*35 TITLE
      CHARACTER*9 CTITL

C                               TITLE TEXT, LENGTH, HEIGHT, AND POSITION
      DATA CTITL/'SYNTHESIS'/
      DATA LENCT/9/
      DATA POSCT/-.850,.89/
C                               FLAG FOR UPDATE
      DATA PRFORM/2/
C                               GREY MAIN SCREEN BOX
      DATA BOX1/-0.48,-0.98,0.98,-0.98,0.98,0.48,-0.48,0.48/
C                               GREY SIDE STRIP
```

```
      DATA BOX2/-0.98,-0.95,-0.525,-0.95,-0.525,0.33,-0.98,0.33/
C                              GREY COMMAND BOX
      DATA BOX3/-0.98,0.37,-0.525,0.37,-0.525,0.98,-0.98,0.98/
C                              GREY POSITION BOX
      DATA BOX4/-0.48,0.52,-0.02,0.52,-0.02,0.98,-0.48,0.98/
C                              WHITE TITLE IN COMMAND BOX
      DATA BOX51/-0.96,0.90,-0.90,0.90,-0.90,0.91,-0.96,0.91/
      DATA BOX52/-0.605,0.90,-0.545,0.90,-0.545,0.91,-0.605,0.91/

C                              CALCULATION FOR THE TWO SHADOW BOXES
      DO 100 I=1,7,2
         SHD2(I)=BOX2(I)+0.025
         SHD3(I)=BOX3(I)+0.025
  100 CONTINUE

      DO 200 I=2,8,2
         SHD2(I)=BOX2(I)-0.025
         SHD3(I)=BOX3(I)-0.025
  200 CONTINUE

*********************************************************************
*  OPEN STRUCTURE FOR THE DISPLAY BOX
*********************************************************************
C                              OPEN A STRUCTURE
      CALL GPOPST(1)
C                              SET SOLID INTEROR STYLE FOR GREY BOX
      CALL GPIS(2)
C                              SET COLOR FOR INTERIOR OF GREY BOX
      CALL GPICI(3)
C                              DRAW GREY DISPLAY BOX
      CALL GPPG2(1,4,2,BOX1)
C                              CLOSE STRUCTURE
      CALL GPCLST


*********************************************************************
*  OPEN STRUCTURE FOR THE MENU BOX
*********************************************************************
C                              OPEN A STRUCTURE
      CALL GPOPST(2)
C                              SET SOLID INTEROR STYLE FOR GREY BOX
      CALL GPIS(2)
C                              SET COLOR FOR INTERIOR OF SHADOW BOX
      CALL GPICI(4)
C                              DRAW GREY SHADOW BOX
      CALL GPPG2(1,4,2,SHD2)
C                              SET COLOR FOR INTERIOR OF GREY BOX
      CALL GPICI(3)
C                              DRAW GREY TITLE BOX
      CALL GPPG2(1,4,2,BOX2)
C                              CLOSE STRUCTURE
      CALL GPCLST


*********************************************************************
*  OPEN STRUCTURE FOR COMMAND BOX
*********************************************************************
C                              OPEN A STRUCTURE
      CALL GPOPST(3)
C                              SET SOLID INTEROR STYLE FOR GREY BOX
      CALL GPIS(2)
C                              SET COLOR FOR INTERIOR OF SHADOW BOX
      CALL GPICI(4)
C                              DRAW GREY SHADOW BOX
      CALL GPPG2(1,4,2,SHD3)
C                              SET COLOR FOR INTERIOR OF GREY BOX
      CALL GPICI(3)
C                              DRAW GREY COMMAND BOX
      CALL GPPG2(1,4,2,BOX3)
C                              SET COLOR FOR WHITE OF COMMAND TITLE
      CALL GPICI(1)
C                              DRAW WHITE BOXES IN TITLE
      CALL GPPG2(1,4,2,BOX51)
      CALL GPPG2(1,4,2,BOX52)
C                              SET TEXT COLOR TO WHITE
      CALL GPTXCI(1)
C                              SET ANNOTATION SIZE SCALE FACTOR
      CALL GPAHSC(1.00)
C                              DRAW TITLE TEXT
      CALL GPAN2(POSCT,LENCT,CTITL)
C                              CLOSE STRUCTURE
      CALL GPCLST


*********************************************************************
*  OPEN STRUCTURE FOR VALUATOR BOX
*********************************************************************
C                              OPEN A STRUCTURE
      CALL GPOPST(4)
C                              SET SOLID INTEROR STYLE FOR GREY BOX
      CALL GPIS(2)
C                              SET COLOR FOR INTERIOR OF GREY BOX
```

```
      CALL GPICI(3)
C                              DRAW GREY VALUATOR BOX
      CALL GPPG2(1,4,2,BOX4)
C                              CLOSE STRUCTURE
      CALL GPCLST


*********************************************************************
*  SCREEN DISPLAY
*********************************************************************
C                              UPDATE THE WORKSTATION
      CALL GPUPWS(IWSID,PRFORM)

      RETURN
      END
```

SUBROUTINE SDATA

```
      SUBROUTINE SDATA(IWSID,AXISR,CSIZE,PNAM,PNUM,POINT,SCALE)
*********************************************************************
* ----- SUBROUTINE SDATA -----                                      *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED FOR DATA ENTRY FOR SYNTHESIS DATA       *
*                                                                   *
*********************************************************************

      INTEGER IWSID,CHOICE,PNUM(30,4)

      CHARACTER*7 PNAM(30)

      REAL AXISR,CSIZE(3),POINT(30,14),SCALE


*********************************************************************
*    ASSOCIATE STRUCTURES WITH VIEWS
*********************************************************************
C                           POSITION ICON
      CALL GPARV(IWSID,5,9,0.)
      CALL GPARV(IWSID,6,16,0.)
      CALL GPARV(IWSID,4,11,0.)
C                           POINTS AND NAMES
      CALL GPARV(IWSID,4,12,0.)
C                           X AXES
      CALL GPARV(IWSID,4,13,0.)
C                           Y AXES
      CALL GPARV(IWSID,4,14,0.)
C                           Z AXES
      CALL GPARV(IWSID,4,15,0.)

*********************************************************************
*    INQUIRE TYPE OF DATA INPUT
*********************************************************************

   10 CALL SCMMAN(IWSID,7)
      CALL SMENU(IWSID,CSIZE,2,CHOICE)

      IF(CHOICE.EQ.1)THEN
C                        BODY POSITIONS
         CALL SPT(IWSID,AXISR,CSIZE,PNAM,PNUM,POINT,SCALE)
         GO TO 10
      ENDIF

      IF(CHOICE.EQ.2)THEN
C                      CHANGE VIEW
         CALL SCHVIE(IWSID,AXISR,CSIZE,SCALE,PNAM,PNUM,POINT)
         GO TO 10
      ENDIF

      IF(CHOICE.EQ.4)THEN
C                    END OF DATA INPUT
         RETURN
      ENDIF


      RETURN
      END
```

SUBROUTINE SDELP

```
      SUBROUTINE SDELP(IWSID,CSIZE,AXISR,PNAM,PNUM,POINT,SCALE)
*********************************************************************
```

```
* ----- SUBROUTINE SDELP -----                                              *
*                                                                           *
*                                                                           *
*   THIS SUBROUTINE IS USED TO DELETE A BODY POSITION                       *
*                                                                           *
*****************************************************************************
      INTEGER IWSID,PRFORM,PPATH(3),ACLASS(1),CHOICE,PNUM(30,4),LENG
      REAL CSIZE(3),DATA(35),PAREA(6),L1(4),POINT(30,14),AXISR,POSN(2)
      CHARACTER*7 PNAM(30),C
      CHARACTER*16 RETURN
      DATA RETURN/'1.  RETURN      '/
      DATA LENG/16/
      DATA POSN/0.05,2.2/
      DATA C/'CCCCCCC'/
      DATA DATA/32,0,0,        2,2,1,1,
     &                       1,1,1,1,1,1,
     &                       1,1,1,1,1,1,
     &                       1,1,1,1,1,1,
     &                       1,1,1,1,1,1,
     &                         1,1,1,1/
C                            FLAG FOR UPDATE
C     DATA PRFORM/2/
C                            INCLUSION CLASS LIST
      DATA ACLASS/4/

*****************************************************************************
*  REQUEST PICK INPUT
*****************************************************************************
C                            SET PICK FILTER(ALL CLASS 4 DETECTABLE)
      CALL GPPKF(IWSID,1,1,ACLASS,0,ACLASS)
C                            MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                            DEFINE PICK AREA
      PAREA(1)=((-.98+1)/2)*CSIZE(1)
      PAREA(2)=((.98+1)/2)*CSIZE(1)
      PAREA(3)=((-.98+1)/2)*CSIZE(2)
      PAREA(4)=((.48+1)/2)*CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=CSIZE(3)
C                            INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                            PLACE PICK DEVICE IN THE EVENT MODE
      CALL GPPKMO(IWSID,1,3,2)
C                            TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)
C                            GET PICK OR RETURN COMMAND
      CALL CMMAND(IWSID,21)

*****************************************************************************
*  INCLUDE A RETURN PICK
*****************************************************************************

      CALL GPOPST(6)
      CALL GPADCN(1,ACLASS)
      CALL GPPKID(101)
      CALL GPTXCI(2)
      CALL GPAHSC(1.0)
      CALL GPAN2(POSN,LENG,RETURN)
      CALL GPCLST
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
      CALL GPUPWS(IWSID,PRFORM)

*****************************************************************************
*  AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
*****************************************************************************
C                            AWAIT AN EVENT
  200 CALL GPAWEV(1000.,IWS,ICLA,IDEV)
      IF(ICLA.EQ.5)THEN
C                            GET PICK
         CALL GPGTPK(1,1,PPATH)
C                            IF NO PICK, RETURN
         IF(PPATH(2).EQ.101)GO TO 1000
C                            CHOICE IS SECOND ITEM OF PICK PATH
         CHOICE=PPATH(2)
C                            DEACTIVATE POINT
         PNAM(CHOICE)=C
         DO 300 I=1,14
              POINT(CHOICE,I)=0.
  300    POINT(CHOICE,I)=0.
C                            REDRAW ALL POINTS
         CALL SRDRWP(IWSID,AXISR,PNAM,PNUM,POINT,SCALE)
C                            AWAIT ANOTHER EVENT
         GO TO 200
      ELSE
C                            IF NOT A PICK, GO TO AWAIT AN EVENT
         GO TO 200
      ENDIF
```

```
C                              MAKE SURE PICK IS DEACTIVATED
 1000 CALL GPPKMO(IWSID,1,1,2)
      CALL GPEST(6)

C                              SET PICK FILTER(ALL CLASS 2 UNDETECTABLE)
      CALL GPPKF(IWSID,1,0,ACLASS,1,ACLASS)

      RETURN
      END
```

SUBROUTINE SDRAWP

```
      SUBROUTINE SDRAWP(IWSID,AXISR,PNAME,NUMP,POINT,SC)
*********************************************************************
* ----- SUBROUTINE DRAWJT -----                                     *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED TO DRAW A JOINT IN THE MAIN WINDOW      *
*                                                                   *
*********************************************************************

      INTEGER IWSID,NUMP,CLASS(1),CLASS1(1),JTYPE

      REAL LOCAT(3),LOCAT1(3),ORIENT(3),AXISR,SCALE(3),ORIGIN(3),
     &  MATSC(4,4),MATRIX(4,4),UP(3),Z(3),MAT1(4,4),MAT2(4,4),MAT3(4,4),
     &  MAT4(4,4),MAT5(4,4),Y(3),NEWX(3),NEWYU(3),LZ1(3),NEWY(3),
     &  LZ2(3),Z1(6),Z2(6),POINT(30,14),ZO(3),LOCA1(3),LOCA2(3),
     &  ORIEN2(3),LOCAT2(3),SC,SC2(3),MSC2(4,4),NEWXU(3),P(24),
     &  XAX(3),YAX(3),ZAX(3),X1(6),Y1(6),LX1(3),LY1(3),SC1(6),SC3(6)

      CHARACTER*7 JNAME
      CHARACTER*2 XN,YN,ZN

      DATA XN /'X '/
      DATA YN /'Y '/
      DATA ZN /'Z '/
      DATA CLASS/4/
      DATA CLASS1/4/
      DATA MAT1 /1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
      DATA MAT2 /1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
      DATA MAT3 /1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
      DATA MAT4 /1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
      DATA MATSC/1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
      DATA MSC2 /1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
      DATA MATRIX/1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1.,0.,0.,0.,0.,1./
      DATA P/1.,0.,0.,-1.,0.,0.,0.,0.,0.,0.,1.,0.,0.,-1.,0.,0.,0.,0.,
     &       0.,0.,1., 0.,0.,-1./

      LOCAT(1)=POINT(NUMP,1)
      LOCAT(2)=POINT(NUMP,2)
      LOCAT(3)=POINT(NUMP,3)

      LOCAT1(1)=-.08*AXISR
      LOCAT1(2)=-.24*AXISR
      LOCAT1(3)=0.0

      XAX(1)=POINT(NUMP,4)
      XAX(2)=POINT(NUMP,5)
      XAX(3)=POINT(NUMP,6)

      YAX(1)=POINT(NUMP,7)
      YAX(2)=POINT(NUMP,8)
      YAX(3)=POINT(NUMP,9)

      ZAX(1)=POINT(NUMP,10)
      ZAX(2)=POINT(NUMP,11)
      ZAX(3)=POINT(NUMP,12)

      X1(1)=0.0
      X1(2)=0.0
      X1(3)=0.0
      X1(4)=(XAX(1)*AXISR/10.)*2.4
      X1(5)=(XAX(2)*AXISR/10.)*2.4
      X1(6)=(XAX(3)*AXISR/10.)*2.4

      Y1(1)=0.0
      Y1(2)=0.0
      Y1(3)=0.0
      Y1(4)=(YAX(1)*AXISR/10.)*2.4
      Y1(5)=(YAX(2)*AXISR/10.)*2.4
      Y1(6)=(YAX(3)*AXISR/10.)*2.4

      Z1(1)=0.0
      Z1(2)=0.0
      Z1(3)=0.0
      Z1(4)=(ZAX(1)*AXISR/10.)*2.4
      Z1(5)=(ZAX(2)*AXISR/10.)*2.4
      Z1(6)=(ZAX(3)*AXISR/10.)*2.4
```

```
      LX1(1)=(XAX(1)*AXISR/10.)*2.4+(AXISR*.02)
      LX1(2)=(XAX(2)*AXISR/10.)*2.4+(AXISR*.02)
      LX1(3)=(XAX(3)*AXISR/10.)*2.4+(AXISR*.08)

      LY1(1)=(YAX(1)*AXISR/10.)*2.4+(AXISR*.02)
      LY1(2)=(YAX(2)*AXISR/10.)*2.4+(AXISR*.02)
      LY1(3)=(YAX(3)*AXISR/10.)*2.4+(AXISR*.08)

      LZ1(1)=(ZAX(1)*AXISR/10.)*2.4+(AXISR*.02)
      LZ1(2)=(ZAX(2)*AXISR/10.)*2.4+(AXISR*.02)
      LZ1(3)=(ZAX(3)*AXISR/10.)*2.4+(AXISR*.08)

C                     CALCULATE AXIS TRANSLATION MATRIX
      CALL GPTRL3(LOCAT,MAT1)

      SCALE(1)=AXISR*SC
      SCALE(2)=AXISR*SC
      SCALE(3)=AXISR*SC


      SC1(1)=SC*.9
      SC1(2)=SC*.9
      SC1(3)=SC*.9

      SC2(1)=SC*.5
      SC2(2)=SC*.5
      SC2(3)=SC*.5

      SC3(1)=SC*.5*AXISR/10.
      SC3(2)=SC*.5*AXISR/10.
      SC3(3)=SC*.5*AXISR/10.

  108 CALL GPSC3(SC2,MSC2)
      CALL GPSC3(SC1,MATSC)
      CALL GPSC3(SC3,MATRIX)

*********************************************************************
*    OPEN STRUCTURE AND DRAW POINT  (MAKE PICKABLE)
*********************************************************************
C                     OPEN STRUCTURE 12
      CALL GPOPST(12)
C                     INSERT CLASS NAME TO ALLOW PICKING JOINTS (CLASS 2)
      CALL GPADCN(1,CLASS)
C                     SET PICK I.D.
      CALL GPPKID(NUMP)
C                     SET TRANSFORMATION
      CALL GPMLX3(MAT1,3)
      CALL GPMLX3(MSC2,1)
C                     SET CHARACTER HEIGHT SCALE FACTOR
      CALL GPAHSC(.25)
C                     TEXT COLOR INDEX
      CALL GPTXCI(2)
C                     ASSOCIATE TEXT
      CALL GPAN3(LOCAT1,7,PNAME)
C                     SET TRANSFORMATION
      CALL GPMLX3(MAT1,3)
      CALL GPMLX3(MATRIX,1)
C                     SET LINE TYPE SOLID
      CALL GPLT(1)
C                     SET LINE COLOR
      CALL GPPLCI(2)
C                     ASSOCIATE FIRST POINT ICON
      CALL GPPL3(8,3,P)
C                     CLOSE STRUCTURE
      CALL GPCLST

      IF(XAX(1).EQ.0.0.AND.XAX(2).EQ.0.0.AND.XAX(3).EQ.0.)GO TO 1000

C                     OPEN STRUCTURE 13 (X AXIS)
      CALL GPOPST(13)
C                     INSERT CLASS NAME TO ALLOW PICKING AXES   (CLASS 4)
      CALL GPADCN(1,CLASS)
C                     SET PICK I.D.
      CALL GPPKID(NUMP)
C                     SET TRANSFORMATION
      CALL GPMLX3(MAT1,3)
      CALL GPMLX3(MATSC,1)
C                     SET LINE TYPE SOLID
      CALL GPLT(1)
C                     SET LINE COLOR
      CALL GPPLCI(5)
C                     ASSOCIATE X AXIS
      CALL GPPL3(2,3,X1)
C                     SET CHARACTER HEIGHT SCALE FACTOR
      CALL GPAHSC(.1)
C                     TEXT COLOR INDEX
      CALL GPTXCI(5)
C                     ASSOCIATE TEXT
      CALL GPAN3(LX1,2,XN)
C                     CLOSE STRUCTURE
      CALL GPCLST

      IF(YAX(1).EQ.0.0.AND.YAX(2).EQ.0.0.AND.YAX(3).EQ.0.)GO TO 1000

C                     OPEN STRUCTURE 14 (Y AXIS)
```

```
      CALL GPOPST(14)
C                   INSERT CLASS NAME TO ALLOW PICKING AXES   (CLASS 4)
      CALL GPADCN(1,CLASS)
C                   SET PICK I.D.
      CALL GPPKID(NUMP)
C                   SET TRANSFORMATION
      CALL GPMLX3(MAT1,3)
      CALL GPMLX3(MATSC,1)
C                   SET LINE TYPE SOLID
      CALL GPLT(1)
C                   SET LINE COLOR
      CALL GPPLCI(5)
C                   ASSOCIATE Y AXIS
      CALL GPPL3(2,3,Y1)
C                   SET CHARACTER HEIGHT SCALE FACTOR
      CALL GPAHSC(.1)
C                   TEXT COLOR INDEX
      CALL GPTXCI(5)
C                   ASSOCIATE TEXT
      CALL GPAN3(LY1,2,YN)
C                   CLOSE STRUCTURE
      CALL GPCLST

C                   OPEN STRUCTURE 15 (Z AXIS)
      CALL GPOPST(15)
C                   INSERT CLASS NAME TO ALLOW PICKING AXES   (CLASS 4)
      CALL GPADCN(1,CLASS)
C                   SET PICK I.D.
      CALL GPPKID(NUMP)
C                   SET TRANSFORMATION
      CALL GPMLX3(MAT1,3)
      CALL GPMLX3(MATSC,1)
C                   SET LINE TYPE SOLID
      CALL GPLT(1)
C                   SET LINE COLOR
      CALL GPPLCI(5)
C                   ASSOCIATE Z AXIS
      CALL GPPL3(2,3,Z1)
C                   SET CHARACTER HEIGHT SCALE FACTOR
      CALL GPAHSC(.1)
C                   TEXT COLOR INDEX
      CALL GPTXCI(5)
C                   ASSOCIATE TEXT
      CALL GPAN3(LZ1,2,ZN)
C                   CLOSE STRUCTURE
      CALL GPCLST

 1000 CALL GPUPWS(IWSID,2)

      RETURN
      END
```

SUBROUTINE SETUP

```
      SUBROUTINE SETUP(IWSID,CHOICE,CSIZE)
*********************************************************************
* ----- SUBROUTINE SETUP -----                                      *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED TO OPEN PHIGS, SET INITIAL PARAMETERS,  *
*                                                                   *
*   INITIALIZE ARRAYS, SET REALS AND INTEGERS, SET THE INITIAL      *
*                                                                   *
*   COLOR TABLE, INQUIRE SCREEN SIZE, SET UP INPUT DEVICES,  AND    *
*                                                                   *
*   OTHER GENERAL SETUP PROCEDURES.                                 *
*                                                                   *
*********************************************************************

      COMMON/PNT/JOINT(20,21),PNAM(100),JNAM(20),LNAM(30),INAM(20)
      COMMON/SYN/SPNAM(30)
      COMMON/PDEF/PNDEF(100)
      COMMON/JDEF/JNDEF(20)
      COMMON/LDEF/LNDEF(30)

      INTEGER IWSID,CSTART,CNUM,ASIZE(3),ERRIND,CHOICE

      REAL COLORS(27),CSIZE(3),UNITS,WIND(4),VIEW(4),WINDOC(4),VPC(4),
     &WINDOM(4),VPM(4),JOINT

      CHARACTER*8 ERFILE,WSTYPE,CONNID

      CHARACTER*7 PNAM,PNDEF,JNAM,JNDEF,LNAM,LNDEF,INAM,SPNAM

C                         ERROR FILE
```

```
      DATA ERFILE/'SYSPRINT'/
C                                WORKSTATION TYPE
C     DATA WSTYPE/'5080    '/
C                                CONNECTION IDENTIFIER
C     DATA CONNID/'IBM5080 '/
C                                WORKSTATION TYPE
C     DATA WSTYPE/'3250    '/
C                                CONNECTION IDENTIFIER
C     DATA CONNID/'IBM3250 '/
C                                WORKSTATION TYPE
C     DATA WSTYPE/'GDDM    '/
C                                CONNECTION IDENTIFIER
C     DATA CONNID/'*       '/
C                                START INDEX INTO COLOR TABLE
      DATA CSTART/0/
C                                NUMBER OF COLORS IN TABLE (BESIDES BACKGRD)
      DATA CNUM/9/
C                                COLOR TABLE
C                                     BACKGROUND BLUE
      DATA COLORS/  0.,   0.,   1.,
C                                     MAXIMUM WHITE
     &              1.,   1.,   1.,
C                                     MAXIMUM YELLOW
     &              1.,   1.,   0.,
C                                     BACKGROUND GREY
     &             .40,  .40,  .40,
C                                     DARK GREY
     &             .25,  .25,  .25,
C                                     MAXIMUM BLACK
     &              0.,   0.,   0.,
C                                     TURQUOISE
     &              0.,   1.,   1.,
C                                     RED
     &              1.,   0.,   0.,
C                                     MAGENTA
     &              1.,   0.,   1./
C                                WINDOW AND VIEWPORT OF TITLE SCREEN
      DATA WIND/-1.0,1.0,-1.0,1.0/
      DATA VIEW/0.0,1.0,0.0,1.0/
C                                WINDOW AND VIEWPORT OF COMMAND BOX
      DATA WINDOC/0.00,1.00,0.00,1.00/
      DATA VPC/.01,.2375,0.67,0.975/
C                                WINDOW AND VIEWPORT OF MENU BOX
      DATA WINDOM/0.00,1.00,0.00,2.8132/
      DATA VPM/.01,.2375,0.01,0.65/
C                                WORKSTATION IDENTIFIER
C     IWSID=1
C                                DEFAULT POINT NAMES
      PNDEF(1)= 'PT1     '
      PNDEF(2)= 'PT2     '
      PNDEF(3)= 'PT3     '
      PNDEF(4)= 'PT4     '
      PNDEF(5)= 'PT5     '
      PNDEF(6)= 'PT6     '
      PNDEF(7)= 'PT7     '
      PNDEF(8)= 'PT8     '
      PNDEF(9)= 'PT9     '
      PNDEF(10)='PT10    '
      PNDEF(11)='PT11    '
      PNDEF(12)='PT12    '
      PNDEF(13)='PT13    '
      PNDEF(14)='PT14    '
      PNDEF(15)='PT15    '
      PNDEF(16)='PT16    '
      PNDEF(17)='PT17    '
      PNDEF(18)='PT18    '
      PNDEF(19)='PT19    '
      PNDEF(20)='PT20    '
      PNDEF(21)='PT21    '
      PNDEF(22)='PT22    '
      PNDEF(23)='PT23    '
      PNDEF(24)='PT24    '
      PNDEF(25)='PT25    '
      PNDEF(26)='PT26    '
      PNDEF(27)='PT27    '
      PNDEF(28)='PT28    '
      PNDEF(29)='PT29    '
      PNDEF(30)='PT30    '
      PNDEF(31)='PT31    '
      PNDEF(32)='PT32    '
      PNDEF(33)='PT33    '
      PNDEF(34)='PT34    '
      PNDEF(35)='PT35    '
      PNDEF(36)='PT36    '
      PNDEF(37)='PT37    '
      PNDEF(38)='PT38    '
      PNDEF(39)='PT39    '
      PNDEF(40)='PT40    '
      PNDEF(41)='PT41    '
      PNDEF(42)='PT42    '
      PNDEF(43)='PT43    '
      PNDEF(44)='PT44    '
      PNDEF(45)='PT45    '
      PNDEF(46)='PT46    '
      PNDEF(47)='PT47    '
```

```
PNDEF(48)='PT48    '
PNDEF(49)='PT49    '
PNDEF(50)='PT50    '
PNDEF(51)='PT51    '
PNDEF(52)='PT52    '
PNDEF(53)='PT53    '
PNDEF(54)='PT54    '
PNDEF(55)='PT55    '
PNDEF(56)='PT56    '
PNDEF(57)='PT57    '
PNDEF(58)='PT58    '
PNDEF(59)='PT59    '
PNDEF(60)='PT60    '
PNDEF(61)='PT61    '
PNDEF(62)='PT62    '
PNDEF(63)='PT63    '
PNDEF(64)='PT64    '
PNDEF(65)='PT65    '
PNDEF(66)='PT66    '
PNDEF(67)='PT67    '
PNDEF(68)='PT68    '
PNDEF(69)='PT69    '
PNDEF(70)='PT70    '
PNDEF(71)='PT71    '
PNDEF(72)='PT72    '
PNDEF(73)='PT73    '
PNDEF(74)='PT74    '
PNDEF(75)='PT75    '
PNDEF(76)='PT76    '
PNDEF(77)='PT77    '
PNDEF(78)='PT78    '
PNDEF(79)='PT79    '
PNDEF(80)='PT80    '
PNDEF(81)='PT81    '
PNDEF(82)='PT82    '
PNDEF(83)='PT83    '
PNDEF(84)='PT84    '
PNDEF(85)='PT85    '
PNDEF(86)='PT86    '
PNDEF(87)='PT87    '
PNDEF(88)='PT88    '
PNDEF(89)='PT89    '
PNDEF(90)='PT90    '
PNDEF(91)='PT91    '
PNDEF(92)='PT92    '
PNDEF(93)='PT93    '
PNDEF(94)='PT94    '
PNDEF(95)='PT95    '
PNDEF(96)='PT96    '
PNDEF(97)='PT97    '
PNDEF(98)='PT98    '
PNDEF(99)='PT99    '
PNDEF(100)='PT100   '

JNDEF(1)= 'JT1     '
JNDEF(2)= 'JT2     '
JNDEF(3)= 'JT3     '
JNDEF(4)= 'JT4     '
JNDEF(5)= 'JT5     '
JNDEF(6)= 'JT6     '
JNDEF(7)= 'JT7     '
JNDEF(8)= 'JT8     '
JNDEF(9)= 'JT9     '
JNDEF(10)='JT10    '
JNDEF(11)='JT11    '
JNDEF(12)='JT12    '
JNDEF(13)='JT13    '
JNDEF(14)='JT14    '
JNDEF(15)='JT15    '
JNDEF(16)='JT16    '
JNDEF(17)='JT17    '
JNDEF(18)='JT18    '
JNDEF(19)='JT19    '
JNDEF(20)='JT20    '

LNDEF(1)= 'LK1     '
LNDEF(2)= 'LK2     '
LNDEF(3)= 'LK3     '
LNDEF(4)= 'LK4     '
LNDEF(5)= 'LK5     '
LNDEF(6)= 'LK6     '
LNDEF(7)= 'LK7     '
LNDEF(8)= 'LK8     '
LNDEF(9)= 'LK9     '
LNDEF(10)='LK10    '
LNDEF(11)='LK11    '
LNDEF(12)='LK12    '
LNDEF(13)='LK13    '
LNDEF(14)='LK14    '
LNDEF(15)='LK15    '
LNDEF(16)='LK16    '
LNDEF(17)='LK17    '
LNDEF(18)='LK18    '
LNDEF(19)='LK19    '
LNDEF(20)='LK20    '
LNDEF(21)='LK21    '
```

```
      LNDEF(22)='LK22    '
      LNDEF(23)='LK23    '
      LNDEF(24)='LK24    '
      LNDEF(25)='LK25    '
      LNDEF(26)='LK26    '
      LNDEF(27)='LK27    '
      LNDEF(28)='LK28    '
      LNDEF(29)='LK29    '
      LNDEF(30)='LK30    '

*************************************************************************
*  OPEN AND INITIATE PHIGS
*************************************************************************
C                             OPEN PHIGS
      CALL GPOPPH(ERFILE,0)
C                             OPEN 5080 WORKSTATION
      CALL GPOPWS(IWSID,CONNID,WSTYPE)
C                             LOAD COLOR TABLE ON THE WORKSTATION
      CALL GPCR(IWSID,CSTART,CNUM,COLORS)

*************************************************************************
*  FILL POINT AND JOINT ARRAYS
*************************************************************************

      DO 10 I=1,100
         PNAM(I)='BBBBBBB'
   10 CONTINUE
      DO 11 I=1,20
         JNAM(I)='BBBBBBB'
         INAM(I)='BBBBBBB'
   11 CONTINUE
      DO 12 I=1,30
         LNAM(I)='BBBBBBB'
         SPNAM(I)='BBBBBBB'
   12 CONTINUE
      DO 14 I=1,20
         DO 13 J=1,21
            JOINT(I,J)=0.0
   13    CONTINUE
   14 CONTINUE

*************************************************************************
*  GET DISPLAY SURFACE SIZE
*************************************************************************
C                             INQUIRE DISPLAY SURFACE SIZE
      CALL GPQADS(IWSID,ERRIND,UNITS,CSIZE,ASIZE)

*************************************************************************
*  SET UP VIEWS AND ASSIGN STRUCTURES TO VIEWS
*************************************************************************
C                             SET UP WINDOW FOR VIEW 1
      CALL GPVMP2(IWSID,1,WIND,VIEW)
C                             SET VIEW CHARACTERISTICS
      CALL GPVCH(IWSID,1,1,1,1,1,0,2,1,2)
C                             ASSOCIATE THE STRUCTURE 1 WITH A VIEW 1
      CALL GPARV(IWSID,1,1,0.)
C                             ASSOCIATE THE STRUCTURE 2 WITH A VIEW 1
      CALL GPARV(IWSID,1,2,0.)
C                             ASSOCIATE THE STRUCTURE 3 WITH A VIEW 1
      CALL GPARV(IWSID,1,3,0.)
C                             ASSOCIATE THE STRUCTURE 4 WITH A VIEW 1
      CALL GPARV(IWSID,1,4,0.)
C                             ASSOCIATE THE STRUCTURE 8 WITH A VIEW 1
      CALL GPARV(IWSID,1,8,0.)
C                             ASSOCIATE THE STRUCTURE 10 WITH A VIEW 1
      CALL GPARV(IWSID,1,10,0.)
C                             SET VIEW PRIORITY OF VIEW1 HIGHER THAN VIEW0
      CALL GPVP(IWSID,1,0,1)

C                             SET UP WINDOW FOR COMMAND BOX
      CALL GPVMP2(IWSID,2,WINDOC,VPC)
C                             SET CHARACTERISTICS OF VIEW 2
      CALL GPVCH(IWSID,2,2,2,2,1,0,1,1,2)
C                             ASSOCIATE THE STRUCTURE 5 WITH A VIEW 2
      CALL GPARV(IWSID,2,5,0.)
C                             SET VIEW PRIORITY OF VIEW2 HIGHER THAN VIEW1
      CALL GPVP(IWSID,2,1,1)

C                             SET UP WINDOW FOR MENU BOX
      CALL GPVMP2(IWSID,3,WINDOM,VPM)
C                             SET CHARACTERISTICS OF VIEW 3
      CALL GPVCH(IWSID,3,2,2,2,1,0,1,1,2)
C                             ASSOCIATE THE STRUCTURE 6 WITH A VIEW 3
      CALL GPARV(IWSID,3,6,0.)
C                             SET VIEW PRIORITY OF VIEW3 HIGHER THAN VIEW1
      CALL GPVP(IWSID,3,1,1)

*************************************************************************
*  CALL TITLE SCREEN
*************************************************************************
C                             CALL FILE WITH TITLE SCREEN
      CALL SCRN1(IWSID,CSIZE)
```

```
*********************************************************************
*  CALL ANALYSIS OR SYTHESIS CHOICE SCREEN
*********************************************************************
C                              CALL ANALYSIS OR SYNTHESIS CHOICE SCREEN
      CALL SCRN2(IWSID,CSIZE,CHOICE)


      RETURN
      END
```

SUBROUTINE SINQ

```
      SUBROUTINE SINQ(IWSID,CSIZE,AXISR,PNAM,PNUM,POINT)
*********************************************************************
* ----- SUBROUTINE SINQ -----                                       *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED TO INQUIRE BODY POSITIONS               *
*                                                                   *
*********************************************************************

      INTEGER IWSID,PRFORM,PPATH(3),ACLASS(1),CHOICE,PNUM(30,4),LENG

      REAL CSIZE(3),DATA(35),PAREA(6),L1(4),POINT(30,14),AXISR,POSN(2),
     & LOCAT(3),BOX(8),SHD(8),LX(2),LY(2),LZ(2),LX1(2),LY1(2),LZ1(2)

      CHARACTER*7 PNAM(30)
      CHARACTER*16 RETURN
      CHARACTER*20 X,Y,Z,CLEAR
      CHARACTER*14 XVAL,YVAL,ZVAL

      DATA RETURN/'1.  RETURN      '/
      DATA LENG/16/
      DATA POSN/0.05,2.2/

      DATA X    /'X =                '/
      DATA Y    /'Y =                '/
      DATA Z    /'Z =                '/
      DATA CLEAR/'                   '/

      LX(1)=0.05
      LX(2)=0.68

      LX1(1)=0.14
      LX1(2)=0.68

      LY(1)=0.05
      LY(2)=0.62

      LY1(1)=0.14
      LY1(2)=0.62

      LZ(1)=0.05
      LZ(2)=0.56

      LZ1(1)=0.14
      LZ1(2)=0.56

      DATA DATA/32,0,0,          2,2,1,1,
     &                          1,1,1,1,1,1,
     &                          1,1,1,1,1,1,
     &                          1,1,1,1,1,1,
     &                          1,1,1,1,1,1,
     &                            1,1,1,1/
C                              FLAG FOR UPDATE
      DATA PRFORM/2/
C                              INCLUSION CLASS LIST
      DATA ACLASS/4/

C                              GREY VALUATOR BOX
      DATA BOX/0.00,0.545,0.48,0.545,0.48,0.73,0.00,0.73/

      DO 10 I=2,8,2
         SHD(I)=BOX(I)-0.025
   10 CONTINUE

      DO 20 I=1,7,2
         SHD(I)=BOX(I)+0.025
   20 CONTINUE


*********************************************************************
*  REQUEST PICK INPUT
*********************************************************************
C                              SET PICK FILTER(ALL CLASS 4 DETECTABLE)
      CALL GPPKF(IWSID,1,1,ACLASS,0,ACLASS)
C                              MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
```

```
      CALL GPPKMO(IWSID,1,1,2)
C                               DEFINE PICK AREA
      PAREA(1)=((-.98+1)/2)*CSIZE(1)
      PAREA(2)=((.98+1)/2)*CSIZE(1)
      PAREA(3)=((-.98+1)/2)*CSIZE(2)
      PAREA(4)=((.48+1)/2)*CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=CSIZE(3)
C                               INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                               PLACE PICK DEVICE IN THE EVENT MODE
      CALL GPPKMO(IWSID,1,3,2)
C                               TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)
C                               GET INQUIRY OR RETURN COMMAND
      CALL CMMAND(IWSID,22)

*********************************************************************
*  INCLUDE A RETURN PICK
*********************************************************************

C                               RETURN PICK
      CALL GPOPST(6)
      CALL GPADCN(1,ACLASS)
      CALL GPPKID(101)
      CALL GPTXCI(2)
      CALL GPAHSC(1.0)
      CALL GPAN2(POSN,LENG,RETURN)
      CALL GPCLST
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
      CALL GPUPWS(IWSID,PRFORM)
C                               GREY VALUE BOX AND SHADOW
      CALL GPOPST(8)
      CALL GPIS(2)
      CALL GPICI(4)
      CALL GPPG2(1,4,2,SHD)
      CALL GPICI(3)
      CALL GPPG2(1,4,2,BOX)
      CALL GPCLST
      CALL GPUPWS(IWSID,PRFORM)

*********************************************************************
*  AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
*********************************************************************
C                               AWAIT AN EVENT
  200 CALL GPAWEV(1000.,IWS,ICLA,IDEV)
      IF(ICLA.EQ.5)THEN
C                               ERASE OLD POSITION ICONS
         CALL GPEST(9)
         CALL GPEST(10)
         CALL GPEST(11)
C                               GET PICK
         CALL GPGTPK(1,IDEP,PPATH)
C                               IF NO PICK, RETURN
         IF(PPATH(2).EQ.101)GO TO 1000
C                               CHOICE IS SECOND ITEM OF PICK PATH
         CHOICE=PPATH(2)
         ISTRUC=PPATH(1)
C                               GET LOCATION
         LOCAT(1)=POINT(CHOICE,1)
         LOCAT(2)=POINT(CHOICE,2)
         LOCAT(3)=POINT(CHOICE,3)

         REWIND 10
         DO 350 III=1,3
            WRITE(10,300)CLEAR
  300       FORMAT(A20)
  350    CONTINUE
         REWIND 10
         WRITE(10,375)LOCAT(1)
         WRITE(10,375)LOCAT(2)
         WRITE(10,375)LOCAT(3)
  375    FORMAT(F14.6)
         REWIND 10
         READ(10,400)XVAL
         READ(10,400)YVAL
         READ(10,400)ZVAL
  400    FORMAT(A14)

C                               DRAW POSITION ICON

         CALL GPOPST(11)
         CALL ICON(LOCAT,AXISR)
         CALL GPCLST

         CALL GPOPST(9)
         CALL ICON(LOCAT,AXISR)
         CALL GPCLST

         CALL GPOPST(10)
         CALL GPTXCI(1)
         CALL GPAHSC(1.0)
         CALL GPAN2(LX,20,X)
```

```
         CALL GPAN2(LX1,14,XVAL)
         CALL GPAN2(LY,20,Y)
         CALL GPAN2(LY1,14,YVAL)
         CALL GPAN2(LZ,20,Z)
         CALL GPAN2(LZ1,14,ZVAL)
         CALL GPCLST

         CALL GPUPWS(IWSID,PRFORM)
C                          REQUEST A NEW INQUIRY
         GO TO 200
      ELSE
C                          IF NOT A PICK, GO TO AWAIT AN EVENT
         GO TO 200
      ENDIF

C                          MAKE SURE PICK IS DEACTIVATED
 1000 CALL GPPKMO(IWSID,1,1,2)
      CALL GPEST(6)
      CALL GPEST(8)
      CALL GPEST(9)
      CALL GPEST(10)
      CALL GPEST(11)
      CALL GPUPWS(IWSID,PRFORM)

C                          SET PICK FILTER(ALL CLASS 4 UNDETECTABLE)
      CALL GPPKF(IWSID,1,0,ACLASS,1,ACLASS)

      RETURN
      END
```

SUBROUTINE SINQO

```
      SUBROUTINE SINQO(IWSID,CSIZE,AXISR,POINT)
*********************************************************************
* ----- SUBROUTINE SINQO -----                                      *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED TO INQUIRE BODY POSITION ORIENTATIONS   *
*                                                                   *
*********************************************************************

      INTEGER IWSID,PRFORM,PPATH(3),ACLASS(1),CHOICE,PNUM(30,4),LENG,
     &STRUC

      REAL CSIZE(3),DATA(35),PAREA(6),L1(4),POINT(30,14),AXISR,POSN(2),
     & BOX(8),SHD(8),LX(2),LY(2),LZ(2),LX1(2),LY1(2),LZ1(2),
     & ORIENT(3)

      CHARACTER*7 PNAM(30)
      CHARACTER*16 RETURN
      CHARACTER*20 X,Y,Z,CLEAR
      CHARACTER*14 XVAL,YVAL,ZVAL

      DATA RETURN/'1.  RETURN      '/
      DATA LENG/16/
      DATA POSN/0.05,2.2/

      DATA X    /'DX =                '/
      DATA Y    /'DY =                '/
      DATA Z    /'DZ =                '/
      DATA CLEAR/'                    '/

      LX(1)=0.05
      LX(2)=0.68

      LX1(1)=0.14
      LX1(2)=0.68

      LY(1)=0.05
      LY(2)=0.62

      LY1(1)=0.14
      LY1(2)=0.62

      LZ(1)=0.05
      LZ(2)=0.56

      LZ1(1)=0.14
      LZ1(2)=0.56

      DATA DATA/32,0,0,        2,2,1,1,
     &                       1,1,1,1,1,1,
     &                       1,1,1,1,1,1,
     &                       1,1,1,1,1,1,
     &                       1,1,1,1,1,1,
     &                         1,1,1,1/
C                            FLAG FOR UPDATE
```

```
      DATA PRFORM/2/
C                               INCLUSION CLASS LIST
      DATA ACLASS/4/
C                               GREY VALUATOR BOX
      DATA BOX/0.00,0.545,0.48,0.545,0.48,0.73,0.00,0.73/

      DO 10 I=2,8,2
         SHD(I)=BOX(I)-0.025
   10 CONTINUE

      DO 20 I=1,7,2
         SHD(I)=BOX(I)+0.025
   20 CONTINUE


*********************************************************************
*  REQUEST PICK INPUT
*********************************************************************
C                               SET PICK FILTER(ALL CLASS 2,3 DETECTABLE)
      CALL GPPKF(IWSID,1,1,ACLASS,0,ACLASS)
C                               MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                               DEFINE PICK AREA
      PAREA(1)=((-.98+1)/2)*CSIZE(1)
      PAREA(2)=((.98+1)/2)*CSIZE(1)
      PAREA(3)=((-.98+1)/2)*CSIZE(2)
      PAREA(4)=((.48+1)/2)*CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=CSIZE(3)
C                               INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                               PLACE PICK DEVICE IN THE EVENT MODE
      CALL GPPKMO(IWSID,1,3,2)
C                               TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)
C                               GET INQUIRY OR RETURN COMMAND
      CALL SCMMAN(IWSID,37)

*********************************************************************
*  INCLUDE A RETURN PICK
*********************************************************************

C                               RETURN PICK
      CALL GPOPST(6)
      CALL GPADCN(1,ACLASS)
      CALL GPPKID(101)
      CALL GPTXCI(2)
      CALL GPAHSC(1.0)
      CALL GPAN2(POSN,LENG,RETURN)
      CALL GPCLST
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
      CALL GPUPWS(IWSID,PRFORM)
C                               GREY VALUE BOX AND SHADOW
      CALL GPOPST(8)
      CALL GPIS(2)
      CALL GPICI(4)
      CALL GPPG2(1,4,2,SHD)
      CALL GPICI(3)
      CALL GPPG2(1,4,2,BOX)
      CALL GPCLST
      CALL GPUPWS(IWSID,PRFORM)


*********************************************************************
*  AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
*********************************************************************
C                               AWAIT AN EVENT
  200 CALL GPAWEV(1000.,IWS,ICLA,IDEV)
      IF(ICLA.EQ.5)THEN
C                               ERASE OLD ORIENT. ICONS
         CALL GPEST(9)
         CALL GPEST(10)
         CALL GPEST(11)
         CALL GPEST(16)
C                               GET PICK
         CALL GPGTPK(1,IDEP,PPATH)
C                               IF NO PICK, RETURN
         IF(PPATH(2).EQ.101)GO TO 1000
C                               CHOICE IS SECOND ITEM OF PICK PATH
         CHOICE=PPATH(2)
         STRU=PPATH(1)

         IF(STRU.EQ.13)THEN
              ORIENT(1)=POINT(CHOICE,4)
              ORIENT(2)=POINT(CHOICE,5)
              ORIENT(3)=POINT(CHOICE,6)
         ENDIF
         IF(STRU.EQ.14)THEN
              ORIENT(1)=POINT(CHOICE,7)
              ORIENT(2)=POINT(CHOICE,8)
              ORIENT(3)=POINT(CHOICE,9)
         ENDIF
         IF(STRU.EQ.15)THEN
```

```
            ORIENT(1)=POINT(CHOICE,10)
            ORIENT(2)=POINT(CHOICE,11)
            ORIENT(3)=POINT(CHOICE,12)
         ENDIF

         REWIND 10
         DO 350 III=1,3
            WRITE(10,300)CLEAR
300         FORMAT(A20)
350      CONTINUE
         REWIND 10
         WRITE(10,375)ORIENT(1)
         WRITE(10,375)ORIENT(2)
         WRITE(10,375)ORIENT(3)
375      FORMAT(F14.6)
         REWIND 10
         READ(10,400)XVAL
         READ(10,400)YVAL
         READ(10,400)ZVAL
400      FORMAT(A14)

C                              DRAW POSITION ICON

         CALL GPOPST(16)
         CALL ICONO(ORIENT,AXISR)
         CALL GPCLST

         CALL GPOPST(10)
         CALL GPTXCI(1)
         CALL GPAHSC(1.0)
         CALL GPAN2(LX,20,X)
         CALL GPAN2(LX1,14,XVAL)
         CALL GPAN2(LY,20,Y)
         CALL GPAN2(LY1,14,YVAL)
         CALL GPAN2(LZ,20,Z)
         CALL GPAN2(LZ1,14,ZVAL)
         CALL GPCLST

         CALL GPUPWS(IWSID,PRFORM)
C                              REQUEST A NEW INQUIRY
         GO TO 200
      ELSE
C                              IF NOT A PICK, GO TO AWAIT AN EVENT
         GO TO 200
      ENDIF

C                              MAKE SURE PICK IS DEACTIVATED
 1000 CALL GPPKMO(IWSID,1,1,2)
      CALL GPEST(6)
      CALL GPEST(8)
      CALL GPEST(16)
      CALL GPEST(10)
      CALL GPUPWS(IWSID,PRFORM)
C                              SET PICK FILTER(ALL CLASS 2,3 UNDETECTABLE)
      CALL GPPKF(IWSID,1,0,ACLASS,2,ACLASS)

      RETURN
      END
```

SUBROUTINE SMENU

```
      SUBROUTINE SMENU(IWSID,CSIZE,NUM,CHOICE)
*************************************************************************
* ----- SUBROUTINE SMENU -----                                          *
*                                                                       *
*                                                                       *
*   THIS SUBROUTINE IS USED TO DISPLAY MENU ITEMS IN THE MENU BOX       *
*                                                                       *
*   AND PROCESS THE MENU PICKS.                                         *
*                                                                       *
*************************************************************************
      INTEGER IWSID,LENG,STRUCT,PRFORM,NUM,PPATH(3),ACLASS(1),CHOICE
      REAL POSN(2),CSIZE(3),DATA(35),PAREA(6),L1(4)
      CHARACTER*35 TMENU(30,18)

C                              TITLE TEXT, LENGTH
      DATA LENG/19/
      DATA TMENU(1,1)/'1. NEW MODEL       '/
      DATA TMENU(1,10)/'                  '/
      DATA TMENU(1,2)/'2. RESTART MODEL   '/
```

```
DATA TMENU(1,11)/'                    '/
DATA TMENU(1,3)/'                     '/
DATA TMENU(1,12)/'                    '/
DATA TMENU(1,4)/'                     '/
DATA TMENU(1,13)/'                    '/
DATA TMENU(1,5)/'                     '/
DATA TMENU(1,14)/'                    '/
DATA TMENU(1,6)/'                     '/
DATA TMENU(1,15)/'                    '/
DATA TMENU(1,7)/'                     '/
DATA TMENU(1,16)/'                    '/
DATA TMENU(1,8)/'                     '/
DATA TMENU(1,17)/'                    '/
DATA TMENU(1,9)/'                     '/
DATA TMENU(1,18)/'                    '/

DATA TMENU(2,1) /'1. ENTER BODY       '/
DATA TMENU(2,10) /'   POSITIONS       '/
DATA TMENU(2,2) /'2. CHANGE VIEW      '/
DATA TMENU(2,11)/'                    '/
DATA TMENU(2,3) /'                    '/
DATA TMENU(2,12)/'                    '/
DATA TMENU(2,4) /'3. END OF DATA      '/
DATA TMENU(2,13)/'   INPUT            '/
DATA TMENU(2,5) /'                    '/
DATA TMENU(2,14)/'                    '/
DATA TMENU(2,6) /'                    '/
DATA TMENU(2,15)/'                    '/
DATA TMENU(2,7) /'                    '/
DATA TMENU(2,16)/'                    '/
DATA TMENU(2,8) /'                    '/
DATA TMENU(2,17)/'                    '/
DATA TMENU(2,9) /'                    '/
DATA TMENU(2,18)/'                    '/

DATA TMENU(3,1) /'1. ZOOM             '/
DATA TMENU(3,10) /'                   '/
DATA TMENU(3,2) /'2. RESTORE          '/
DATA TMENU(3,11)/'   ORIGINAL VIEW    '/
DATA TMENU(3,3) /'3. SCALE POINTS     '/
DATA TMENU(3,12)/'                    '/
DATA TMENU(3,4) /'4. RETURN           '/
DATA TMENU(3,13)/'                    '/
DATA TMENU(3,5) /'                    '/
DATA TMENU(3,14)/'                    '/
DATA TMENU(3,6) /'                    '/
DATA TMENU(3,15)/'                    '/
DATA TMENU(3,7) /'                    '/
DATA TMENU(3,16)/'                    '/
DATA TMENU(3,8) /'                    '/
DATA TMENU(3,17)/'                    '/
DATA TMENU(3,9) /'                    '/
DATA TMENU(3,18)/'                    '/

DATA TMENU(4,1) /'1. ADD POINT        '/
DATA TMENU(4,10) /'                   '/
DATA TMENU(4,2) /'2. DELETE POINT     '/
DATA TMENU(4,11)/'                    '/
DATA TMENU(4,3) /'3. INQUIRE          '/
DATA TMENU(4,12)/'   POSITION         '/
DATA TMENU(4,4) /'4. INQUIRE          '/
DATA TMENU(4,13)/'   ORIENTATION      '/
DATA TMENU(4,5) /'5. RETURN           '/
DATA TMENU(4,14)/'                    '/
DATA TMENU(4,6) /'                    '/
DATA TMENU(4,15)/'                    '/
DATA TMENU(4,7) /'                    '/
DATA TMENU(4,16)/'                    '/
DATA TMENU(4,8) /'                    '/
DATA TMENU(4,17)/'                    '/
DATA TMENU(4,9) /'                    '/
DATA TMENU(4,18)/'                    '/

DATA TMENU(5,1) /'1. DIRECT ENTRY     '/
DATA TMENU(5,10) /'                   '/
DATA TMENU(5,2) /'2. VALUATOR ENTRY   '/
DATA TMENU(5,11)/'                    '/
DATA TMENU(5,3) /'3. TOWARD AN        '/
DATA TMENU(5,12)/'   EXISTING POINT   '/
DATA TMENU(5,4) /'4. RETURN           '/
DATA TMENU(5,13)/'                    '/
DATA TMENU(5,5) /'                    '/
DATA TMENU(5,14)/'                    '/
DATA TMENU(5,6) /'                    '/
DATA TMENU(5,15)/'                    '/
DATA TMENU(5,7) /'                    '/
DATA TMENU(5,16)/'                    '/
DATA TMENU(5,8) /'                    '/
DATA TMENU(5,17)/'                    '/
DATA TMENU(5,9) /'                    '/
DATA TMENU(5,18)/'                    '/

DATA TMENU(6,1) /'1. RSCR             '/
DATA TMENU(6,10) /'                   '/
```

```
      DATA TMENU(6,2) /'2. RCCC             '/
      DATA TMENU(6,11)/'                    '/
      DATA TMENU(6,3) /'3. WRITE NO         '/
      DATA TMENU(6,12)/'   SYNTHESIS FILE   '/
      DATA TMENU(6,4) /'                    '/
      DATA TMENU(6,13)/'                    '/
      DATA TMENU(6,5) /'                    '/
      DATA TMENU(6,14)/'                    '/
      DATA TMENU(6,6) /'                    '/
      DATA TMENU(6,15)/'                    '/
      DATA TMENU(6,7) /'                    '/
      DATA TMENU(6,16)/'                    '/
      DATA TMENU(6,8) /'                    '/
      DATA TMENU(6,17)/'                    '/
      DATA TMENU(6,9) /'                    '/
      DATA TMENU(6,18)/'                    '/


      DATA DATA/32,0,0,        2,2,1,1,
     &                        1,1,1,1,1,1,
     &                        1,1,1,1,1,1,
     &                        1,1,1,1,1,1,
     &                        1,1,1,1,1,1,
     &                          1,1,1,1/
C                             FLAG FOR UPDATE
C     DATA PRFORM/2/
C                             INCLUSION CLASS LIST
      DATA ACLASS/1/

C                             DEFINE TEXT STARTING POSITION
      POSN(1)=0.05
      POSN(2)=2.3

*****************************************************************************
*  OPEN STRUCTURE FOR MENU ITEMS
*****************************************************************************

C                             DELETE PREVIOUS MENUS
C     CALL GPEST(6)
C                             OPEN STRUCTURE 6
      CALL GPOPST(6)
C                             INSERT CLASS NAME TO ALLOW PICKING MENU ITEM
      CALL GPADCN(1,ACLASS)
C                             SET TEXT COLOR TO YELLOW
      CALL GPTXCI(2)
C                             SET ANNOTATION SIZE SCALE FACTOR
      CALL GPAHSC(1.00)

      DO 100 I=1,9
C                             SET PICK IDENTIFIER
         CALL GPPKID(I)
C                             DRAW TEXT
         CALL GPAN2(POSN,LENG,TMENU(NUM,I))
C                             SET NEXT POSITION
         POSN(2)=POSN(2)-.08
C                             DRAW TEXT
         CALL GPAN2(POSN,LENG,TMENU(NUM,I+9))
C                             SET NEXT POSITION
         POSN(2)=POSN(2)-.18
  100 CONTINUE
C                             CLOSE STRUCTURE
      CALL GPCLST

*****************************************************************************
*  SCREEN DISPLAY
*****************************************************************************
C                             ASSOCIATE THE STRUCTURE 6 WITH A VIEW 3
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
C                             TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)

*****************************************************************************
*  REQUEST PICK INPUT
*****************************************************************************
C                             SET PICK FILTER(ALL CLASS 1 DETECTABLE)
      CALL GPPKF(IWSID,1,1,ACLASS,0,ACLASS)
C                             MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                             DEFINE PICK AREA
      PAREA(1)=((-.98+1)/2)*CSIZE(1)
      PAREA(2)=((-.525+1)/2)*CSIZE(1)
      PAREA(3)=((-.95+1)/2)*CSIZE(2)
      PAREA(4)=((.33+1)/2)*CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=CSIZE(3)
C                             INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                             PLACE PICK DEVICE IN THE EVENT MODE
      CALL GPPKMO(IWSID,1,3,2)
C                             TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)
```

```
*********************************************************************
*  AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
*********************************************************************
C                           AWAIT AN EVENT
  200 CALL GPAWEV(1000.,IWS,ICLA,IDEV)
      IF(ICLA.EQ.5)THEN
C                           GET PICK
         CALL GPGTPK(1,1,PPATH)
C                           CHOICE IS SECOND ITEM OF PICK PATH
         CHOICE=PPATH(2)
C                           DEACTIVATE PICK
         CALL GPPKMO(IWSID,1,1,2)
C                           DELETE MENU STRUCTURES
         CALL GPEST(6)
C                           UPDATE THE WORKSTATION
         CALL GPUPWS(IWSID,PRFORM)
C                           RETURN WITH CHOICE
         GO TO 1000
      ELSE
C                           IF NOT A PICK, GO TO AWAIT AN EVENT
         GO TO 200
      ENDIF

C                           MAKE SURE PICK IS DEACTIVATED
 1000 CALL GPPKMO(IWSID,1,1,2)


      RETURN
      END
```

SUBROUTINE SNSTAR

```
      SUBROUTINE SNSTAR(IWSID,FILE,CSIZE,AXISR,PNAM,PNUM,POINT)
*********************************************************************
* ----- SUBROUTINE SNSTAR -----                                     *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE WILL START A MECHIN INPUT FILE.  THE USER WILL  *
*                                                                   *
*   BE ASKED TO INPUT THE NEW FILE NAME AND THE RANGE OF THE        *
*                                                                   *
*   AXES.                                                           *
*                                                                   *
*********************************************************************

      INTEGER IWSID,PNUM(30,4),
     &        DATA(35),LENG,ACLASS(1)
      REAL AREA(6),CSIZE(3),AXISR,
     &     POINT(30,14),POSN(2),PAREA(6)
      CHARACTER*7 FILE,AXIS,CLEAR,FILEO,INIT,PNAM(30)
      CHARACTER*29 FLINE
      CHARACTER*16 RETURN
      CHARACTER*50 CLEAR1

      DATA RETURN/'1.  QUIT        '/
      DATA LENG/16/
      DATA POSN/0.05,2.2/


      DATA DATA/32,0,0,         2,2,1,1,
     &                        1,1,1,1,1,1,
     &                        1,1,1,1,1,1,
     &                        1,1,1,1,1,1,
     &                        1,1,1,1,1,1,
     &                          1,1,1,1/
C                           INCLUSION CLASS LIST
      DATA ACLASS/2/


      DATA INIT/'1.0    '/
      DATA AXIS/'       '/
      DATA CLEAR/'      '/
      DATA CLEAR1/'                                                  '/
      DATA FLINE/'MECHIN RESTART FILE-SYNTHESIS'/

      FILE='       '

      CRET=0

      AREA(1)=0.08*CSIZE(1)
      AREA(2)=0.90*CSIZE(1)
      AREA(3)=0.70*CSIZE(1)
      AREA(4)=0.90*CSIZE(1)
      AREA(5)=0.02*CSIZE(1)
      AREA(6)=0.90*CSIZE(1)
```

```
*********************************************************************
*  INITIALIZE PICK
*********************************************************************
C                               SET PICK FILTER(ALL CLASS 2 DETECTABLE)
      CALL GPPKF(IWSID,1,1,ACLASS,0,ACLASS)
C                               MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                               DEFINE PICK AREA
      PAREA(1)=((-.98+1)/2)*CSIZE(1)
      PAREA(2)=((.98+1)/2)*CSIZE(1)
      PAREA(3)=((-.98+1)/2)*CSIZE(2)
      PAREA(4)=((.48+1)/2)*CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=CSIZE(3)
C                               INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                               PLACE PICK DEVICE IN THE EVENT MODE
      CALL GPPKMO(IWSID,1,3,2)
C                               TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,2)

*********************************************************************
*  INCLUDE A RETURN PICK
*********************************************************************

C                               RETURN PICK
      CALL GPOPST(6)
      CALL GPADCN(1,ACLASS)
      CALL GPPKID(101)
      CALL GPTXCI(2)
      CALL GPAHSC(1.0)
      CALL GPAN2(POSN,LENG,RETURN)
      CALL GPCLST
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
      CALL GPUPWS(IWSID,2)

*********************************************************************
*  INITIALIZE THE STRING DEVICE.
*********************************************************************
    2 CALL GPSTMO(IWSID,1,1,2)
C                               INITIALIZE THE STRING DEVICE
      CALL GPINST(IWSID,1,7,CLEAR,1,AREA,7,1,0,CLEAR)
C                               PLACE STRING DEVICE IN THE EVENT MODE
      CALL GPSTMO(IWSID,1,3,2)
C                               REQUEST THE FILE NAME
      CALL CMMAND(IWSID,4)


*********************************************************************
*  GO TO CLOSE IF QUIT IS PICKED
*********************************************************************
C                               AWAIT AN EVENT
    3 CALL GPAWEV(1000.,IWS,ICLA,IDEV)
      IF(ICLA.EQ.5)THEN
C                               CLOSE FOR RESTART
         CALL GPDAST
         CALL GPCLPH
         STOP
      ENDIF

*********************************************************************
*  PROCESS STRING EVENT. OPEN MECHIN FILE
*********************************************************************
      L=7
C                               GET STRING
      CALL GPGTST(L,LR,FILE)
      IF(FILE.EQ.'          ')GO TO 3
C                               SEE IF FILE ALREADY EXISTS
      CALL FILEXS(FILE,IRET)
C                               IF FILE EXISTS, HAVE USER RETYPE TO USE
      IF(IRET.EQ.0)THEN
         CALL GPSTMO(IWSID,1,1,2)
         FILEO=FILE
         FILE='          '
         CLEAR='          '
         CALL CMMAND(IWSID,8)
         CALL GPSTMO(IWSID,1,1,2)
         CALL GPINST(IWSID,1,7,CLEAR,1,AREA,7,1,0,CLEAR)
         CALL GPRQST(IWSID,1,7,ISTAT,INUM,FILE)
         IF(FILE.EQ.FILEO)THEN
            CONTINUE
         ELSE
            CALL GPINST(IWSID,1,7,CLEAR,1,AREA,7,1,0,CLEAR)
            CALL GPSTMO(IWSID,1,3,2)
            GO TO 3
         ENDIF
      ENDIF

C                               OPEN A NEW FILE
      OPEN(UNIT=15,FILE=FILE,STATUS='NEW',ERR=2)
      CALL GPEST(6)
      CALL GPPKMO(IWSID,1,1,2)
```

```
*************************************************************************
*  INITIALIZE THE ARRAYS
*************************************************************************

      DO 11 I=1,30
         DO 10 J=1,14
            POINT(I,J)=0.0
   10    CONTINUE
   11 CONTINUE

      DO 13 I=1,30
         DO 12 J=1,4
            PNUM(I,J)=0
   12    CONTINUE
   13 CONTINUE

      DO 14 I=1,30
         PNAM(I)='BBBBBBB'
   14 CONTINUE

*************************************************************************
*  WRITE FILE NAME TO THE SCREEN
*************************************************************************
C                              CALL SUBROUTINE TO WRITE FILENAME
      CALL FILENM(IWSID,FILE)


*************************************************************************
*  REQUEST THE COORDINATE AXIS RANGE
*************************************************************************
C                              COMMAND FOR COORDINATE AXIS RANGE
  500 CALL CMMAND(IWSID,5)
      CALL GPSTMO(IWSID,1,1,2)
C                              INITIALIZE THE STRING DEVICE FOR AXIS
      CALL GPINST(IWSID,1,7,INIT,1,AREA,7,1,0,CLEAR)
C                              REQUEST STRING FOR AXIS
      CALL GPRQST(IWSID,1,7,ISTAT,NUM,AXIS)
      WRITE(10,800)CLEAR1
  800 FORMAT(A50)
      REWIND 10
      WRITE(10,1000)AXIS
 1000 FORMAT(A7)
      REWIND 10
      READ(10,*,ERR=500)AXISR
C                              REQUEST THE COORDINATE AXIS RANGE
      CALL AXES(IWSID,AXISR)

 3000 CALL GPSTMO(IWSID,1,1,2)
      CALL GPPKF(IWSID,1,0,ACLASS,1,ACLASS)

      RETURN
      END
```

SUBROUTINE SORIEN

```
      SUBROUTINE SORIEN(IWSID,PPOSN,POINT,AXISR,CSIZE,ORIENT,RFLG)
*************************************************************************
* ----- SUBROUTINE SORIEN-----                                          *
*                                                                       *
*                                                                       *
*   THIS SUBROUTINE IS USED TO ORIENT A VECTOR (COOR. AXIS FOR SYN.)    *
*                                                                       *
*************************************************************************

      INTEGER IWSID,ECHOSW,EVENT,DLEN,ECHO,ACLASS,PRFORM,CHOICE,
     & PPATH(3),RFLG,NUMV

      REAL CSIZE(3),AREA1(6),AREA2(6),AREA3(6),VLUE1,LOVAL,HIVAL,AXISR,
     & BOX(8),SHD(8),ORIENT(3),MATRIX(1000,4),MATUSE(4,4),LINE(6),
     & LINEX(6),LINEY(6),LINEZ(6),XBOX(15),PPOSN(3),POINT(30,14),
     & YBOX(15),ZBOX(15),LX(6),LY(6),LZ(6),OL(6),AREA(6)

      CHARACTER*7 CLEAR,INIT

      DATA CLEAR/'       '/
      DATA INIT /'0.     '/

      DATA LX/0.,0.,0.,0.,0.,0./
      DATA LY/0.,0.,0.,0.,0.,0./
      DATA LZ/0.,0.,0.,0.,0.,0./

      DATA OL/0.,0.,0.,0.,0.,0./

C                              FLAG FOR UPDATE
      DATA PRFORM/2/

C                              INCLUSION CLASS LIST
      DATA ACLASS/1/
      DATA VLUE1/0./
      DATA ECHO/3/
```

```
      DATA DLEN/0/
      DATA EVENT/3/
      DATA ECHOSW/2/

C                                GREY VALUATOR BOX
      DATA BOX/0.00,0.545,0.48,0.545,0.48,0.73,0.00,0.73/

      NUMV=1

      LOVAL=-1.0
      HIVAL=1.0

      AREA(1)=.08*CSIZE(1)
      AREA(2)=.90*CSIZE(1)
      AREA(3)=.70*CSIZE(2)
      AREA(4)=.90*CSIZE(2)
      AREA(5)=.02*CSIZE(1)
      AREA(6)=.90*CSIZE(3)

      AREA1(1)=.52*CSIZE(1)
      AREA1(2)=CSIZE(1)
      AREA1(3)=.855*CSIZE(2)-.006
      AREA1(4)=CSIZE(2)
      AREA1(5)=0.
      AREA1(6)=CSIZE(3)

      AREA2(1)=.52*CSIZE(1)
      AREA2(2)=CSIZE(1)
      AREA2(3)=.855*CSIZE(2)-.014
      AREA2(4)=CSIZE(2)
      AREA2(5)=0.
      AREA2(6)=CSIZE(3)

      AREA3(1)=.52*CSIZE(1)
      AREA3(2)=CSIZE(1)
      AREA3(3)=.855*CSIZE(2)-.022
      AREA3(4)=CSIZE(2)
      AREA3(5)=0.
      AREA3(6)=CSIZE(3)


      DO 100 I=2,8,2
         SHD(I)=BOX(I)-0.025
  100 CONTINUE

      DO 200 I=1,7,2
         SHD(I)=BOX(I)+0.025
  200 CONTINUE

*************************************************************************
*  OPEN STRUCTURE FOR THE MENU BOX (VIEW 1)
*************************************************************************
C                                OPEN A STRUCTURE
      CALL GPOPST(8)
C                                SET SOLID INTEROR STYLE FOR GREY BOX
      CALL GPIS(2)
C                                SET COLOR FOR INTERIOR OF SHADOW BOX
      CALL GPICI(4)
C                                DRAW GREY SHADOW BOX
      CALL GPPG2(1,4,2,SHD)
C                                SET COLOR FOR INTERIOR OF GREY BOX
      CALL GPICI(3)
C                                DRAW GREY TITLE BOX
      CALL GPPG2(1,4,2,BOX)
C                                CLOSE STRUCTURE
      CALL GPCLST

*************************************************************************
*  SCREEN DISPLAY
*************************************************************************
C                                UPDATE THE WORKSTATION
      CALL GPUPWS(IWSID,PRFORM)

*************************************************************************
*  CHOOSE METHOD OF JOINT ORIENTATION
*************************************************************************

  300 CALL SMENU(IWSID,CSIZE,5,CHOICE)

      IF(CHOICE.EQ.1)THEN
C                                DIRECT ENTRY
         CALL DEO(IWSID,AXISR,CSIZE,ORIENT,RFLG)
         IF(RFLG.EQ.1)THEN
C                                REENTER POINT
              ORIENT(1)=0.
              ORIENT(2)=0.
              ORIENT(3)=0.
              RFLG=0
              GO TO 300
         ENDIF
      ENDIF


      IF(CHOICE.EQ.2)THEN
C                                VALUATOR ENTRY
```

```
         CALL VEO(IWSID,AXISR,CSIZE,ORIENT,RFLG)
         IF(RFLG.EQ.1)THEN
C                                REENTER POINT
              ORIENT(1)=0.
              ORIENT(2)=0.
              ORIENT(3)=0.
              RFLG=0
              GO TO 300
         ENDIF
      ENDIF

      IF(CHOICE.EQ.3)THEN
C                                TOWARDS AN EXISTING POINT
         CALL STEPO(IWSID,PPOSN,POINT,AXISR,CSIZE,ORIENT,RFLG)
         IF(RFLG.EQ.1)THEN
C                                REENTER POINT
              ORIENT(1)=0.
              ORIENT(2)=0.
              ORIENT(3)=0.
              RFLG=0
              GO TO 300
         ENDIF
      ENDIF

      IF(CHOICE.EQ.4)THEN
C                                RETURN WITH NO ORIENTATION
         RFLG=1
      ENDIF


 1000 CALL GPEST(8)
      CALL GPEST(6)
      CALL GPUPWS(IWSID,2)

      RETURN
      END
```

SUBROUTINE SPOSIT

```
      SUBROUTINE SPOSIT(IWSID,AXISR,CSIZE,LOCAT,RFLG)
*****************************************************************************
* ----- SUBROUTINE SPOSIT -----                                             *
*                                                                           *
*                                                                           *
*   THIS SUBROUTINE IS USED TO POSITION A BODY POSITION                     *
*                                                                           *
*****************************************************************************

      INTEGER IWSID,ECHOSW,EVENT,DLEN,ECHO,ACLASS,PRFORM,CHOICE,
     & PPATH(3),RFLG,NUMV

      REAL CSIZE(3),AREA1(6),AREA2(6),AREA3(6),VLUE1,LOVAL,HIVAL,AXISR,
     & BOX(8),SHD(8),LOCAT(3),MATRIX(1000,4),MATUSE(4,4),LINE(6),
     & LINEX(6),LINEY(6),LINEZ(6),XBOX(15),
     & YBOX(15),ZBOX(15),LX(6),LY(6),LZ(6),AREA(6)

      CHARACTER*7 CLEAR,INIT

      DATA CLEAR/'      '/
      DATA INIT /'0.    '/

      DATA LX/0.,0.,0.,0.,0.,0./
      DATA LY/0.,0.,0.,0.,0.,0./
      DATA LZ/0.,0.,0.,0.,0.,0./

C                                FLAG FOR UPDATE
      DATA PRFORM/2/
C                                INCLUSION CLASS LIST
      DATA ACLASS/1/
      DATA VLUE1/0./
      DATA ECHO/3/
      DATA DLEN/0/
      DATA EVENT/3/
      DATA ECHOSW/2/
C                                GREY VALUATOR BOX
      DATA BOX/0.00,0.545,0.48,0.545,0.48,0.73,0.00,0.73/

      NUMV=1

      LOVAL=-AXISR
      HIVAL=AXISR

      AREA(1)=.08*CSIZE(1)
      AREA(2)=.90*CSIZE(1)
      AREA(3)=.70*CSIZE(2)
```

```
      AREA(4)=.90*CSIZE(2)
      AREA(5)=.02*CSIZE(1)
      AREA(6)=.90*CSIZE(3)

      AREA1(1)=.52*CSIZE(1)
      AREA1(2)=CSIZE(1)
      AREA1(3)=.855*CSIZE(2)-.006
      AREA1(4)=CSIZE(2)
      AREA1(5)=0.
      AREA1(6)=CSIZE(3)

      AREA2(1)=.52*CSIZE(1)
      AREA2(2)=CSIZE(1)
      AREA2(3)=.855*CSIZE(2)-.014
      AREA2(4)=CSIZE(2)
      AREA2(5)=0.
      AREA2(6)=CSIZE(3)

      AREA3(1)=.52*CSIZE(1)
      AREA3(2)=CSIZE(1)
      AREA3(3)=.855*CSIZE(2)-.022
      AREA3(4)=CSIZE(2)
      AREA3(5)=0.
      AREA3(6)=CSIZE(3)


      DO 100 I=2,8,2
         SHD(I)=BOX(I)-0.025
  100 CONTINUE

      DO 200 I=1,7,2
         SHD(I)=BOX(I)+0.025
  200 CONTINUE

*************************************************************************
*  OPEN STRUCTURE FOR THE VAL. BOX (VIEW 1)
*************************************************************************
C                              OPEN A STRUCTURE
      CALL GPOPST(8)
C                              SET SOLID INTEROR STYLE FOR GREY BOX
      CALL GPIS(2)
C                              SET COLOR FOR INTERIOR OF SHADOW BOX
      CALL GPICI(4)
C                              DRAW GREY SHADOW BOX
      CALL GPPG2(1,4,2,SHD)
C                              SET COLOR FOR INTERIOR OF GREY BOX
      CALL GPICI(3)
C                              DRAW GREY TITLE BOX
      CALL GPPG2(1,4,2,BOX)
C                              CLOSE STRUCTURE
      CALL GPCLST

*************************************************************************
*  SCREEN DISPLAY
*************************************************************************
C                              UPDATE THE WORKSTATION
      CALL GPUPWS(IWSID,PRFORM)

*************************************************************************
*  INITIALIZE AND ACTIVATE DIALS 1,2,3 AND STRING DEVICE
*************************************************************************
C                              PLACE VALUATORS IN THE REQUEST MODE
  300 CALL GPVLMO(IWSID,1,1,ECHOSW)
      CALL GPVLMO(IWSID,2,1,ECHOSW)
      CALL GPVLMO(IWSID,3,1,ECHOSW)
C                              INITIALIZE DIALS FOR X , Y , Z INPUT
      CALL GPINVL(IWSID,1,VLUE1,ECHO,AREA1,LOVAL,HIVAL,DLEN,DATA)
      CALL GPINVL(IWSID,2,VLUE1,ECHO,AREA2,LOVAL,HIVAL,DLEN,DATA)
      CALL GPINVL(IWSID,3,VLUE1,ECHO,AREA3,LOVAL,HIVAL,DLEN,DATA)
C                              ACTIVATE DIALS FOR X , Y , Z INPUT
      CALL GPVLMO(IWSID,1,EVENT,ECHOSW)
      CALL GPVLMO(IWSID,2,EVENT,ECHOSW)
      CALL GPVLMO(IWSID,3,EVENT,ECHOSW)

      CALL GPINST(IWSID,1,12,INIT,1,AREA,12,1,0,CLEAR)
      CALL GPSTMO(IWSID,1,EVENT,ECHOSW)

*************************************************************************
*  SCREEN DISPLAY
*************************************************************************
C                              UPDATE THE WORKSTATION
      CALL GPUPWS(IWSID,PRFORM)

*************************************************************************
*  DISPLAY MENU AND GET CHOICE OF ENTER, RETURN OR VALUATOR
*************************************************************************

      LOCAT(1)=0.
      LOCAT(2)=0.
      LOCAT(3)=0.

  500 CONTINUE
      CALL ENTER(IWSID,CHOICE,CSIZE)
```

```
  600 CALL GPAWEV(1000.,IIWSID,ICLASS,IDEV)
      IF(ICLASS.EQ.3)THEN
C                              PROCESS VALUATORS (NOT IN SUBROUTINE TO
C                                                 SPEED EXECUTION)
          CALL GPSTMO(IWSID,1,1,ECHOSW)
          CALL GPGTVL(VALUE)

          IF(IDEV.EQ.1)LOCAT(1)=VALUE
          IF(IDEV.EQ.2)LOCAT(2)=VALUE
          IF(IDEV.EQ.3)LOCAT(3)=VALUE

**************************************************************************
* GET POSITION ICON                                                      *
**************************************************************************

          LX(4)=LOCAT(1)
          LY(5)=LOCAT(2)
          LZ(6)=LOCAT(3)

          LINEX(1)=LOCAT(1)
          LINEX(2)=LOCAT(2)
          LINEX(3)=LOCAT(3)
          LINEX(4)=0.
          LINEX(5)=LOCAT(2)
          LINEX(6)=LOCAT(3)

          LINEY(1)=LOCAT(1)
          LINEY(2)=LOCAT(2)
          LINEY(3)=LOCAT(3)
          LINEY(4)=LOCAT(1)
          LINEY(5)=0.
          LINEY(6)=LOCAT(3)

          LINEZ(1)=LOCAT(1)
          LINEZ(2)=LOCAT(2)
          LINEZ(3)=LOCAT(3)
          LINEZ(4)=LOCAT(1)
          LINEZ(5)=LOCAT(2)
          LINEZ(6)=0.

          XBOX(1)=0.00
          XBOX(2)=LOCAT(2)+(.06*AXISR)
          XBOX(3)=LOCAT(3)+(.06*AXISR)
          XBOX(4)=0.00
          XBOX(5)=LOCAT(2)-(.06*AXISR)
          XBOX(6)=LOCAT(3)+(.06*AXISR)
          XBOX(7)=0.00
          XBOX(8)=LOCAT(2)-(.06*AXISR)
          XBOX(9)=LOCAT(3)-(.06*AXISR)
          XBOX(10)=0.00
          XBOX(11)=LOCAT(2)+(.06*AXISR)
          XBOX(12)=LOCAT(3)-(.06*AXISR)
          XBOX(13)=0.00
          XBOX(14)=LOCAT(2)+(.06*AXISR)
          XBOX(15)=LOCAT(3)+(.06*AXISR)

          YBOX(1)=LOCAT(1)+(.06*AXISR)
          YBOX(2)=0.00
          YBOX(3)=LOCAT(3)+(.06*AXISR)
          YBOX(4)=LOCAT(1)-(.06*AXISR)
          YBOX(5)=0.00
          YBOX(6)=LOCAT(3)+(.06*AXISR)
          YBOX(7)=LOCAT(1)-(.06*AXISR)
          YBOX(8)=0.00
          YBOX(9)=LOCAT(3)-(.06*AXISR)
          YBOX(10)=LOCAT(1)+(.06*AXISR)
          YBOX(11)=0.00
          YBOX(12)=LOCAT(3)-(.06*AXISR)
          YBOX(13)=LOCAT(1)+(.06*AXISR)
          YBOX(14)=0.00
          YBOX(15)=LOCAT(3)+(.06*AXISR)

          ZBOX(1)=LOCAT(1)+(.06*AXISR)
          ZBOX(2)=LOCAT(2)+(.06*AXISR)
          ZBOX(3)=0.00
          ZBOX(4)=LOCAT(1)-(.06*AXISR)
          ZBOX(5)=LOCAT(2)+(.06*AXISR)
          ZBOX(6)=0.00
          ZBOX(7)=LOCAT(1)-(.06*AXISR)
          ZBOX(8)=LOCAT(2)-(.06*AXISR)
          ZBOX(9)=0.00
          ZBOX(10)=LOCAT(1)+(.06*AXISR)
          ZBOX(11)=LOCAT(2)-(.06*AXISR)
          ZBOX(12)=0.00
          ZBOX(13)=LOCAT(1)+(.06*AXISR)
          ZBOX(14)=LOCAT(2)+(.06*AXISR)
          ZBOX(15)=0.00

          CALL GPEST(9)

          CALL GPOPST(9)

C                              SET POLYLINE COLOR
```

```
         CALL GPPLCI(1)
C                              SET LINE TYPE DASHED
         CALL GPLT(2)
C                              DRAW POLYLINE
         CALL GPPL3(2,3,LINEX)
         CALL GPPL3(2,3,LINEY)
         CALL GPPL3(2,3,LINEZ)
C                              SET LINE TYPE SOLID
         CALL GPLT(1)
         CALL GPPL3(5,3,XBOX)
         CALL GPPL3(5,3,YBOX)
         CALL GPPL3(5,3,ZBOX)

C                              SET POLYLINE COLOR
         CALL GPPLCI(5)
         CALL GPPL3(2,3,LX)
         CALL GPPL3(2,3,LY)
         CALL GPPL3(2,3,LZ)
         CALL GPCLST
         CALL GPUPWS(IWSID,PRFORM)
         GO TO 600
      ENDIF
      IF(ICLASS.EQ.6)THEN
C                              PROCESS STRING ENTRY
         CALL GPVLMO(IWSID,1,1,ECHOSW)
         CALL GPVLMO(IWSID,2,1,ECHOSW)
         CALL GPVLMO(IWSID,3,1,ECHOSW)
         RFLG=0

         CALL DEP(IWSID,AXISR,CSIZE,LOCAT,RFLG)

         IF(RFLG.EQ.1)THEN
C                              REENTER POINT
            CALL GPVLMO(IWSID,1,3,ECHOSW)
            CALL GPVLMO(IWSID,2,3,ECHOSW)
            CALL GPVLMO(IWSID,3,3,ECHOSW)
            CALL GPSTMO(IWSID,1,1,ECHOSW)
            CALL GPINST(IWSID,1,12,INIT,1,AREA,12,1,0,CLEAR)
            CALL GPSTMO(IWSID,1,3,ECHOSW)
            LOCAT(1)=0.
            LOCAT(2)=0.
            LOCAT(3)=0.
            RFLG=0
            GO TO 500
         ENDIF

         CALL GPSTMO(IWSID,1,1,ECHOSW)
         CALL GPEST(8)
         CALL GPEST(9)
         GO TO 1000
      ENDIF

      IF(ICLASS.EQ.5)THEN
         CALL GPGTPK(1,1,PPATH)
         CHOICE=PPATH(2)
            IF(CHOICE.EQ.1)THEN
C                              DEACTIVATE VALUATORS
               CALL GPVLMO(IWSID,1,1,ECHOSW)
               CALL GPVLMO(IWSID,2,1,ECHOSW)
               CALL GPVLMO(IWSID,3,1,ECHOSW)
               CALL GPSTMO(IWSID,1,1,ECHOSW)
               RFLG=0
               CALL GPEST(6)
               CALL GPEST(8)
               CALL GPEST(9)
               CALL GPUPWS(IWSID,2)
               GO TO 1000
            ENDIF
            IF(CHOICE.EQ.2)THEN
C                              DEACTIVATE VALUATORS
               CALL GPVLMO(IWSID,1,1,ECHOSW)
               CALL GPVLMO(IWSID,2,1,ECHOSW)
               CALL GPVLMO(IWSID,3,1,ECHOSW)
               CALL GPSTMO(IWSID,1,1,ECHOSW)
               RFLG=1
               CALL GPEST(6)
               CALL GPEST(8)
               CALL GPEST(9)
               CALL GPUPWS(IWSID,2)
               GO TO 1000
            ENDIF
      ENDIF

 1000 RETURN
      END
```

SUBROUTINE SPT

```
      SUBROUTINE SPT(IWSID,AXISR,CSIZE,PNAM,PNUM,POINT,SCALE)
*********************************************************************
```

```
* ----- SUBROUTINE SPT -----                                                *
*                                                                          *
*                                                                          *
*   THIS SUBROUTINE IS USED TO CREATE BODY POSITION DATA                   *
*                                                                          *
****************************************************************************

      REAL AXISR,CSIZE(3),POINT(30,14),SCALE
      INTEGER IWSID,CHOICE,PNUM(30,4)
      CHARACTER*7 PNAM(30)
****************************************************************************
*  FIND OUT IF USER IS TO CREATE OR DELETE BODY POSITIONS
****************************************************************************
C                           REQUEST TYPE OF DATA INPUT
   10 CALL CMMAND(IWSID,10)
      CALL SMENU(IWSID,CSIZE,4,CHOICE)
      IF(CHOICE.EQ.1)THEN
C                           REQUEST ADD A POINT
         CALL SADDP(IWSID,AXISR,CSIZE,PNAM,PNUM,POINT,SCALE)
         GO TO 10
      ENDIF
      IF(CHOICE.EQ.2)THEN
C                           REQUEST DELETE A POINT
         CALL SDELP(IWSID,CSIZE,AXISR,PNAM,PNUM,POINT,SCALE)
         GO TO 10
      ENDIF
      IF(CHOICE.EQ.3)THEN
C                           INQUIRE A POSITION
         CALL SINQ(IWSID,CSIZE,AXISR,PNAM,PNUM,POINT)
         GO TO 10
      ENDIF
      IF(CHOICE.EQ.4)THEN
C                           INQUIRE AN ORIENTATION
         CALL SINQO(IWSID,CSIZE,AXISR,POINT)
         GO TO 10
      ENDIF
      IF(CHOICE.EQ.5)THEN
C                           RETURN TO JOINT,LINK,OR IC MENU
         RETURN
      ENDIF

      RETURN
      END
```

SUBROUTINE SRDRWP

```
      SUBROUTINE SRDRWP(IWSID,AXISR,PNAM,PNUM,POINT,SCALE)
****************************************************************************
* ----- SUBROUTINE SREDRW -----                                            *
*                                                                          *
*                                                                          *
*   THIS SUBROUTINE IS USED TO REDRAW ALL POINTS BODY POSITIONS            *
*                                                                          *
*   AFTER A DELETION OR RESTART                                            *
*                                                                          *
*                                                                          *
****************************************************************************
      INTEGER IWSID,NUMP,PNUM(30,4)
      REAL LOCAT(3),POINT(30,14),AXISR,SCALE
      CHARACTER*7 PNAME,PNAM(30)
C                           DELETE ALL POINTS
         CALL GPEST(12)
         CALL GPEST(13)
         CALL GPEST(14)
         CALL GPEST(15)
         CALL GPUPWS(IWSID,2)

      DO 10 I=1,30
C                           CHECK FOR LAST POINT
         IF(PNAM(I).EQ.'BBBBBBB')GO TO 1000
C                           CHECK DELETED SPACES IN THE ARRAY
         IF(PNAM(I).EQ.'CCCCCCC')GO TO 10
C                           GET POINT NUMBER
         NUMP=PNUM(I,1)
C                           GET POINT LOCATION
         LOCAT(1)=POINT(I,1)
         LOCAT(2)=POINT(I,2)
         LOCAT(3)=POINT(I,3)
C                           GET POINT NAME
         PNAME=PNAM(I)
C                           DRAW POINT
         CALL SDRAWP(IWSID,AXISR,PNAME,NUMP,POINT,SCALE)
```

```
   10 CONTINUE
 1000 RETURN
      END
```

SUBROUTINE SRSTAR

```
      SUBROUTINE SRSTAR(IWSID,FILE,CSIZE,AXISR,PNAM,PNUM,POINT,SCALE)
*********************************************************************
* ----- SUBROUTINE SRESTR -----                                     *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE WILL READ THE NAME OF A MECHIN RESTART FILE     *
*                                                                   *
*   AND USE THE FILE TO FILL THE APPROPRIATE ARRAYS TO REUSE        *
*                                                                   *
*   THE FILE.                                                       *
*                                                                   *
*********************************************************************

      INTEGER IWSID,PNUM(30,4),
     &        DATA(35),LENG,ACLASS(1)
      REAL AREA(6),CSIZE(3),AXISR,POINT(30,14),
     &     IC(20,4),POSN(2),PAREA(6)
      CHARACTER*7 FILE,CMD,QUIT,CLEAR,PNAM(30)
      CHARACTER*29 FLINE,CLINE
      CHARACTER*16 RETURN
      LOGICAL*4 EXT

      DATA RETURN/'1.  QUIT        '/
      DATA LENG/16/
      DATA POSN/0.05,2.2/


      DATA DATA/32,0,0,         2,2,1,1,
     &                        1,1,1,1,1,1,
     &                        1,1,1,1,1,1,
     &                        1,1,1,1,1,1,
     &                        1,1,1,1,1,1,
     &                          1,1,1,1/
C                            INCLUSION CLASS LIST
      DATA ACLASS/2/


      DATA CLEAR/'       '/
      DATA CMD/'       '/
      DATA QUIT/'QUIT    '/
      DATA CLINE/'MECHIN RESTART FILE-SYNTHESIS'/

      FILE='       '

      IFLG=0
      AXISR=0.0

      AREA(1)=0.08*CSIZE(1)
      AREA(2)=0.90*CSIZE(1)
      AREA(3)=0.70*CSIZE(2)
      AREA(4)=0.90*CSIZE(2)
      AREA(5)=0.02*CSIZE(3)
      AREA(6)=0.90*CSIZE(3)

*********************************************************************
*  INITIALIZE PICK
*********************************************************************
C                            SET PICK FILTER(ALL CLASS 2 DETECTABLE)
      CALL GPPKF(IWSID,1,1,ACLASS,0,ACLASS)
C                            MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                            DEFINE PICK AREA
      PAREA(1)=((-.98+1)/2)*CSIZE(1)
      PAREA(2)=((.98+1)/2)*CSIZE(1)
      PAREA(3)=((-.98+1)/2)*CSIZE(2)
      PAREA(4)=((.48+1)/2)*CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=CSIZE(3)
C                            INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                            PLACE PICK DEVICE IN THE EVENT MODE
      CALL GPPKMO(IWSID,1,3,2)
C                            TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,2)

*********************************************************************
*  INCLUDE A RETURN PICK
*********************************************************************

C                            RETURN PICK
      CALL GPOPST(6)
      CALL GPADCN(1,ACLASS)
      CALL GPPKID(101)
```

```
      CALL GPTXCI(2)
      CALL GPAHSC(1.0)
      CALL GPAN2(POSN,LENG,RETURN)
      CALL GPCLST
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
      CALL GPUPWS(IWSID,2)

*************************************************************************
*  INITIALIZE THE STRING DEVICE
*************************************************************************
C                               REQUEST THE FILE NAME
      CALL CMMAND(IWSID,2)
C                               INITIALIZE THE STRING DEVICE
    5 CALL GPSTMO(IWSID,1,1,2)
      CALL GPINST(IWSID,1,7,CLEAR,1,AREA,7,1,0,CLEAR)
      CALL GPSTMO(IWSID,1,3,2)

*************************************************************************
*  QUIT IF QUIT IS PICKED
*************************************************************************
C                               AWAIT AN EVENT
    6 CALL GPAWEV(1000.,IWS,ICLA,IDEV)
      IF(ICLA.EQ.5)THEN
C                               QUIT PHIGS
         CALL GPDAST
         CALL GPCLPH
         STOP
      ENDIF

*************************************************************************
*  PROCESS STRING EVEVT
*************************************************************************
      L=7
C                               GET STRING
      CALL GPGTST(L,LR,FILE)

C                               FIND OUT IF FILE EXISTS
      CALL FILEXS(FILE,IRET)

      IF(IRET.EQ.1)THEN
         CALL CMMAND(IWSID,9)
         FILE=CLEAR
         GO TO 6
      ENDIF
C                               OPEN UNIT 15, FILETYPE 'FILE'
      OPEN(UNIT=15, FILE=FILE, ERR=6)
      REWIND 15
      READ(15,10,ERR=6,END=6)FLINE
   10 FORMAT(A29)

      IF(FLINE.NE.CLINE)THEN
         CALL CMMAND(IWSID,6)
         FILE=CLEAR
         GO TO 6
      ENDIF

      CALL GPPKMO(IWSID,1,1,2)
      CALL GPEST(6)

C                               WRITE THAT ACTION IS CONTINUING
      CALL CMMAND(IWSID,3)
C                               WRITE FILENAME TO THE SCREEN
      CALL FILENM(IWSID,FILE)

*************************************************************************
*  SET UP AXES
*************************************************************************
C                               READ THE AXIS RANGE
      READ(15,*)AXISR
      CALL AXES(IWSID,AXISR)
C                               READ JOINT SCALE FACTOR
      READ(15,*)SCALE
      CALL SCALNM(IWSID,SCALE)

*************************************************************************
*  SET UP POINT ARRAY
*************************************************************************
C                               READ THE NUMBER OF POINTS
      READ(15,*)NP
C                               READ THE POINT DATA
      DO 100 I=1,NP
         READ(15,99)PNAM(I)
   99    FORMAT(A7)
         READ(15,*)(PNUM(I,J),J=1,4)
         READ(15,*)(POINT(I,J),J=1,14)
  100 CONTINUE

*************************************************************************
*  DRAW ALL POINTS
*************************************************************************

      CALL SRDRWP(IWSID,AXISR,PNAM,PNUM,POINT,SCALE)
```

```
      CALL GPSTMO(IWSID,1,1,2)
      CALL GPPKF(IWSID,1,0,ACLASS,1,ACLASS)

      RETURN
      END
```

SUBROUTINE STEPO

```
      SUBROUTINE STEPO(IWSID,PPOSN,POINT,AXISR,CSIZE,LOCAT,RFLG)
*********************************************************************
* ----- SUBROUTINE STEPO  -----                                     *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE WILL PROMPT FOR THE ENTRY OF                    *
*                                                                   *
*   ORIENTATION COORDINATES TOWARD AN EXISTING POINT                *
*                                                                   *
*********************************************************************

      INTEGER IWSID,PRFORM,PPATH(3),ACLASS(1),CHOICE,PNUM(30,4),LENG,
     &RFLG

      REAL CSIZE(3),DATA(35),PAREA(6),L1(4),AXISR,POSN(2),
     &P(3),P1(3),VEC(3),MAG,LOCAT(3),PPOSN(3),POINT(30,14),
     &LX(2),LX1(2),LY(2),LY1(2),LZ(2),LZ1(2)

      CHARACTER*7 PNAM(30)
      CHARACTER*20 X,Y,Z,CLEAR
      CHARACTER*16 XVAL,YVAL,ZVAL,RETURN

      DATA RETURN/'1.  RETURN      '/
      DATA LENG/16/
      DATA POSN/0.05,2.2/


      DATA DATA/32,0,0,          2,2,1,1,
     &                         1,1,1,1,1,1,
     &                         1,1,1,1,1,1,
     &                         1,1,1,1,1,1,
     &                         1,1,1,1,1,1,
     &                           1,1,1,1/
C                             FLAG FOR UPDATE
      DATA PRFORM/2/
C                             INCLUSION CLASS LIST
      DATA ACLASS/4/

      DATA X      /'DX=                '/
      DATA Y      /'DY=                '/
      DATA Z      /'DZ=                '/
      DATA CLEAR /'                    '/


      LX(1)=0.05
      LX(2)=0.68

      LX1(1)=0.14
      LX1(2)=0.68

      LY(1)=0.05
      LY(2)=0.62

      LY1(1)=0.14
      LY1(2)=0.62

      LZ(1)=0.05
      LZ(2)=0.56

      LZ1(1)=0.14
      LZ1(2)=0.56

*********************************************************************
*  REQUEST PICK INPUT
*********************************************************************
C                             SET PICK FILTER(ALL CLASS 2 DETECTABLE)
      CALL GPPKF(IWSID,1,1,ACLASS,0,ACLASS)
C                             MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                             DEFINE PICK AREA
      PAREA(1)=((-.98+1)/2)*CSIZE(1)
      PAREA(2)=((.98+1)/2)*CSIZE(1)
      PAREA(3)=((-.98+1)/2)*CSIZE(2)
      PAREA(4)=((.48+1)/2)*CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=CSIZE(3)
C                             INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                             PLACE PICK DEVICE IN THE EVENT MODE
```

```
      CALL GPPKMO(IWSID,1,3,2)
C                              TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)
C                              GET PICK OR RETURN COMMAND
  200 CALL SCMMAN(IWSID,34)

*********************************************************************
*  INCLUDE A RETURN PICK
*********************************************************************

C                              RETURN PICK
      CALL GPOPST(6)
      CALL GPADCN(1,ACLASS)
      CALL GPPKID(101)
      CALL GPTXCI(2)
      CALL GPAHSC(1.0)
      CALL GPAN2(POSN,LENG,RETURN)
      CALL GPCLST
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
      CALL GPUPWS(IWSID,PRFORM)


*********************************************************************
*  AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
*********************************************************************
C                              AWAIT AN EVENT
  210 CALL GPAWEV(1000.,IWS,ICLA,IDEV)
      IF(ICLA.EQ.5)THEN
C                              GET PICK
         CALL GPGTPK(3,I,PPATH)
C                              IF RETURN PICKED
         IF(PPATH(2).EQ.101)THEN
              RFLG=1
              GO TO 1000
         ENDIF
C                              ERASE RETURN PROMPT
         CALL GPEST(6)
C                              CHOICE IS SECOND ITEM OF PICK PATH
         CHOICE=PPATH(2)
C                              GET POINT COORDINATES
         P(1)=POINT(CHOICE,1)
         P(2)=POINT(CHOICE,2)
         P(3)=POINT(CHOICE,3)

         P1(1)=PPOSN(1)
         P1(2)=PPOSN(2)
         P1(3)=PPOSN(3)

         VEC(1)=P(1)-P1(1)
         VEC(2)=P(2)-P1(2)
         VEC(3)=P(3)-P1(3)

         IF(VEC(1).EQ.0.0.AND.VEC(2).EQ.0.0.AND.VEC(3).EQ.0.0)THEN
C                              POINT AND ORIGIN ARE THE SAME -- RE-ENTER
            CALL CMMAND(IWSID,84)
            GO TO 210
         ENDIF

         MAG=((VEC(1)**2)+(VEC(2)**2)+(VEC(3)**2))**.5

         LOCAT(1)=VEC(1)/MAG
         LOCAT(2)=VEC(2)/MAG
         LOCAT(3)=VEC(3)/MAG
C                              DRAW ORIENTATION ICON
         CALL GPEST(16)

         CALL GPOPST(16)
         CALL ICONO(LOCAT,AXISR)
         CALL GPCLST
C                              DISPLAY VALUES

         REWIND 10
         DO 350 III=1,3
              WRITE(10,300)CLEAR
  300         FORMAT(A20)
  350    CONTINUE
         REWIND 10
         WRITE(10,375)LOCAT(1)
         WRITE(10,375)LOCAT(2)
         WRITE(10,375)LOCAT(3)
  375    FORMAT(F16.14)
         REWIND 10
         READ(10,400)XVAL
         READ(10,400)YVAL
         READ(10,400)ZVAL
  400    FORMAT(A14)

         CALL GPOPST(10)
         CALL GPTXCI(1)
         CALL GPAHSC(1.0)
         CALL GPAN2(LX,20,X)
         CALL GPAN2(LX1,16,XVAL)
         CALL GPAN2(LY,20,Y)
         CALL GPAN2(LY1,16,YVAL)
         CALL GPAN2(LZ,20,Z)
```

```
         CALL GPAN2(LZ1,16,ZVAL)
         CALL GPCLST
C                              ENTER ORIENTATION OR RETURN
         CALL CMMAND(IWSID,32)
         CALL MENU(IWSID,CSIZE,8,CHOICE)

         IF(CHOICE.EQ.1)THEN
              RFLG=0
              GO TO 1000
         ENDIF

         IF(CHOICE.EQ.2)THEN
              RFLG=1
              LOCAT(1)=0.
              LOCAT(2)=0.
              LOCAT(3)=0.
              GO TO 1000
         ENDIF

      ELSE
C                              IF NOT A PICK, GO TO AWAIT AN EVENT
         GO TO 200
      ENDIF


C                              MAKE SURE PICK IS DEACTIVATED
 1000 CALL GPPKMO(IWSID,1,1,2)
      CALL GPEST(6)
      CALL GPEST(10)
      CALL GPEST(16)

C                              SET PICK FILTER(ALL CLASS 2 UNDETECTABLE)
      CALL GPPKF(IWSID,1,0,ACLASS,1,ACLASS)


      RETURN
      END
```

SUBROUTINE SWRRES

```
      SUBROUTINE SWRRES(PNAM,PNUM,POINT,AXISR,FILE,IWSID,CSIZE,SCALE)
*************************************************************************
* ----- SUBROUTINE SWRRES -----                                         *
*                                                                       *
*                                                                       *
*   THIS SUBROUTINE IS USED WRITE THE SYNTHESIS RESTART FILE            *
*                                                                       *
*************************************************************************

      INTEGER PNUM(30,4),NP

      REAL POINT(30,14),AXISR,CSIZE(3),AREA(6)

      CHARACTER*7 PNAM(30),FILE,FILEO,FILE1
      CHARACTER*29 FLINE

      DATA FLINE/'MECHIN RESTART FILE-SYNTHESIS'/

      NP=0

      AREA(1)=.08*CSIZE(1)
      AREA(2)=.90*CSIZE(1)
      AREA(3)=.70*CSIZE(2)
      AREA(4)=.90*CSIZE(2)
      AREA(5)=.02*CSIZE(3)
      AREA(6)=.90*CSIZE(3)

      FILE1='       '

      REWIND 15
*************************************************************************
*  INITIALIZE THE STRING DEVICE. REQUEST NEW RESTART FILE NAME          *
*************************************************************************
C                              INITIALIZE STRING MODE
      CALL GPSTMO(IWSID,1,1,2)
C                              INITIALIZE THE STRING DEVICE
    1 CALL GPINST(IWSID,1,7,FILE,1,AREA,7,1,0,FILE)
C                              REQUEST THE FILE NAME
      CALL CMMAND(IWSID,60)
C                              REQUEST STRING
    2 CALL GPRQST(IWSID,1,7,ISTAT,NUM,FILE1)

      IF(FILE1.EQ.'       ')THEN
         GO TO 2
      ENDIF

      IF(FILE1.EQ.FILE)THEN
         GO TO 9
      ENDIF
```

```
      FILE=FILE1
C                              SEE IF FILE ALREADY EXISTS
    3 CALL FILEXS(FILE,IRET)
C                              IF FILE EXISTS, HAVE USER RETYPE TO USE
      IF(IRET.EQ.0)THEN
         CALL GPSTMO(IWSID,1,1,2)
         FILEO=FILE
         FILE='          '
         CALL CMMAND(IWSID,8)
         CALL GPINST(IWSID,1,7,FILE,1,AREA,7,1,0,FILE)
         CALL GPRQST(IWSID,1,7,ISTAT,NUM,FILE)
         IF(FILE1.EQ.'          ')THEN
            GO TO 2
         ENDIF
         IF(FILE.NE.FILEO)THEN
            GO TO 3
         ENDIF
      ENDIF

      CLOSE(UNIT=15)
C                              OPEN A NEW FILE
      OPEN(UNIT=15,FILE=FILE,STATUS='NEW',ERR=2)

*****************************************************************
*    WRITE FIRST LINE OF THE RESTART FILE AND AXIS RANGE
*****************************************************************

    9 WRITE(15,10)FLINE
   10 FORMAT(A29)
      WRITE(15,*)AXISR
      WRITE(15,*)SCALE


*****************************************************************
*    WRITE POINT DATA TO THE RESTART FILE
*****************************************************************

      DO 80 I=1,30
         IF(PNAM(I).EQ.'BBBBBBB')THEN
            GO TO 90
         ENDIF
         NP=NP+1
   80 CONTINUE

   90 WRITE(15,*)NP

      DO 100 I=1,30
         IF(PNAM(I).EQ.'BBBBBBB')THEN
            GO TO 110
         ELSE
            WRITE(15,91)PNAM(I)
   91       FORMAT(A7)
            WRITE(15,*)(PNUM(I,J),J=1,4)
            WRITE(15,*)(POINT(I,J),J=1,14)
         ENDIF
  100 CONTINUE

  110 RETURN
      END
```

SUBROUTINE SYNIN

```
      SUBROUTINE SYNIN(IWSID,CSIZE)
*****************************************************************
* ----- SUBROUTINE SYNIN -----                                  *
*                                                               *
*                                                               *
*   THIS IS THE SYNTHESIS PORTION OF MECHIN.  THIS ROUTINE WILL *
*                                                               *
*   ALLOW THE INPUT OF SPATIAL MECHANISM PARAMETERS AND WILL    *
*                                                               *
*   WRITE AN INPUT FILE FOR AN SYNTHESIS PACKAGE BASED ON THESE *
*                                                               *
*   PARAMETERS.                                                 *
*                                                               *
*****************************************************************
      COMMON/SYN/PNAM(30)

      INTEGER IWSID,CHOICE,PNUM(30,4)
      REAL CSIZE(3),AXISR,POINT(30,14),SCALE
      CHARACTER*7 PNAM,FILE


*****************************************************************
*  CALL BACKGROUND SCREEN
*****************************************************************
C                              CALL FILE WITH SYNTHESIS BACKGROUND
      CALL SCRN4(IWSID)
```

```
*********************************************************************
*  FIND OUT IF A NEW OR OLD MODEL IS TO BE PROCESSED AND SET UP FOR DATA
*********************************************************************
C                              FIND IF NEW OR OLD MODEL
      CALL CMMAND(IWSID,1)
      CALL MENU(IWSID,CSIZE,1,CHOICE)
      IF(CHOICE.EQ.1)THEN
C                              NEW MODEL  --- REQUEST SIZE OF MODEL
         CALL SNSTAR(IWSID,FILE,CSIZE,AXISR,PNAM,PNUM,POINT)
         SCALE=.8
         CALL SCALNM(IWSID,SCALE)
      ELSE
C                              RESTART MODEL  --- READ RESTART FILE
         CALL SRSTAR(IWSID,FILE,CSIZE,AXISR,PNAM,PNUM,POINT,SCALE)
      ENDIF

*********************************************************************
*  INPUT MODEL DATA
*********************************************************************

      CALL SDATA(IWSID,AXISR,CSIZE,PNAM,PNUM,POINT,SCALE)

*********************************************************************
*  WRITE THE RESTART FILE
*********************************************************************

      CALL SWRRES(PNAM,PNUM,POINT,AXISR,FILE,IWSID,CSIZE,SCALE)

*********************************************************************
*  WRITE THE SYNTHESIS FILE
*********************************************************************

      CALL WRSYN(PNAM,PNUM,POINT,AXISR,CSIZE,IWSID)

      RETURN
      END
```

SUBROUTINE TEJOX

```
      SUBROUTINE TEJOX(IWSID,NUMJ,IAXIS,JOINT,AXISR,CSIZE,XVEC,YVEC,
     &RFLG)
*********************************************************************
* ----- SUBROUTINE TEJOX -----                                      *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE WILL PROMPT FOR THE ENTRY OF JOINT X AXIS       *
*                                                                   *
*   IN THE ORIENTATION OF AN EXISTING JOINT                         *
*                                                                   *
*********************************************************************

      INTEGER IWSID,PRFORM,PPATH(3),ACLASS(2),CHOICE,LENG,IAXIS,
     &RFLG

      REAL CSIZE(3),DATA(35),PAREA(6),L1(4),JOINT(20,21),AXISR,POSN(2),
     &P(3),P1(3),VEC(3),MAG,LOCAT(3),XVEC(3),YVEC(3),ZVEC(3)

      CHARACTER*20 CLEAR
      CHARACTER*16 RETURN

      DATA RETURN/'1.  RETURN      '/
      DATA LENG/16/
      .

      DATA DATA/32,0,0,          2,2,1,1,
     &                         1,1,1,1,1,1,1,
     &                         1,1,1,1,1,1,1,
     &                         1,1,1,1,1,1,1,
     &                         1,1,1,1,1,1,1,
     &                           1,1,1,1/
C                              FLAG FOR UPDATE
      DATA PRFORM/2/
C                              INCLUSION CLASS LIST(CLASS 3,4  JOINT AXES)
      DATA ACLASS/3,4/

*********************************************************************
*  REQUEST PICK INPUT
*********************************************************************
C                              SET PICK FILTER(ALL CLASS 3,4 DETECTABLE)
      CALL GPPKF(IWSID,1,2,ACLASS,0,ACLASS)
C                              MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                              DEFINE PICK AREA
      PAREA(1)=((-.98+1)/2)*CSIZE(1)
      PAREA(2)=((.98+1)/2)*CSIZE(1)
```

```
      PAREA(3)=((-.98+1)/2)*CSIZE(2)
      PAREA(4)=((.48+1)/2)*CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=CSIZE(3)
C                              INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                              PLACE PICK DEVICE IN THE EVENT MODE
      CALL GPPKMO(IWSID,1,3,2)
C                              TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)
C                              GET PICK OR RETURN COMMAND

*********************************************************************
*  INCLUDE A RETURN PICK
*********************************************************************

C                              RETURN PICK
  200 CALL GPOPST(6)
      CALL GPADCN(1,ACLASS)
      CALL GPPKID(101)
      CALL GPTXCI(2)
      CALL GPAHSC(1.0)
      CALL GPAN2(POSN,LENG,RETURN)
      CALL GPCLST
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
      CALL GPUPWS(IWSID,PRFORM)

*********************************************************************
*  AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
*********************************************************************
C                              AWAIT AN EVENT
  300 CALL GPAWEV(1000.,IWS,ICLA,IDEV)
      IF(ICLA.EQ.5)THEN
C                              GET PICK
         CALL GPGTPK(1,1,PPATH)
C                              IF RETURN
         IF(PPATH(2).EQ.101)THEN
              RFLG=1
              GO TO 1000
         ENDIF
C                              CHOICE IS SECOND ITEM OF PICK PATH
         END=PPATH(2)
         ISTRUC=PPATH(1)

         VEC(1)=JOINT(END,1)-JOINT(NUMJ,1)
         VEC(2)=JOINT(END,2)-JOINT(NUMJ,2)
         VEC(3)=JOINT(END,3)-JOINT(NUMJ,3)

         IF(IAXIS.EQ.1)THEN
              ZVEC(1)=JOINT(NUMJ,4)
              ZVEC(2)=JOINT(NUMJ,5)
              ZVEC(3)=JOINT(NUMJ,6)
         ENDIF
         IF(IAXIS.EQ.2)THEN
              ZVEC(1)=JOINT(NUMJ,18)
              ZVEC(2)=JOINT(NUMJ,19)
              ZVEC(3)=JOINT(NUMJ,20)
         ENDIF

         CALL VPROD(ZVEC,VEC,YVEC)
         IF(YVEC(1).EQ.0.0.AND.YVEC(2).EQ.0.0.AND.YVEC(3).EQ.0.0)THEN
C                              X POINT ON THE Z VECTOR
              CALL CMMAND(IWSID,54)
              GO TO 300
         ENDIF

         CALL VPROD(YVEC,ZVEC,XVEC)

         XMAG=((XVEC(1)**2)+(XVEC(2)**2)+(XVEC(3)**2))**.5
         YMAG=((YVEC(1)**2)+(YVEC(2)**2)+(YVEC(3)**2))**.5

         XVEC(1)=XVEC(1)/XMAG
         XVEC(2)=XVEC(2)/XMAG
         XVEC(3)=XVEC(3)/XMAG

         YVEC(1)=YVEC(1)/YMAG
         YVEC(2)=YVEC(2)/YMAG
         YVEC(3)=YVEC(3)/YMAG

      ENDIF
C                              MAKE SURE PICK IS DEACTIVATED
 1000 CALL GPPKMO(IWSID,1,1,2)
      CALL GPEST(6)

C                              SET PICK FILTER(ALL CLASS 2 UNDETECTABLE)
      CALL GPPKF(IWSID,1,0,ACLASS,2,ACLASS)


      RETURN
      END
```

```
      SUBROUTINE TEPO(IWSID,JPOSN,POINT,JOINT,AXISR,CSIZE,LOCAT,RFLG)
*********************************************************************
* ----- SUBROUTINE TEPO  -----                                      *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE WILL PROMPT FOR THE ENTRY OF JOINT              *
*                                                                   *
*   ORIENTATION COORDINATES TOWARD AN EXISTING POINT                *
*                                                                   *
*********************************************************************

      INTEGER IWSID,PRFORM,PPATH(3),ACLASS(3),CHOICE,PNUM(100),LENG,
     &RFLG

      REAL CSIZE(3),DATA(35),PAREA(6),L1(4),JOINT(20,21),AXISR,POSN(2),
     &P(3),P1(3),VEC(3),MAG,LOCAT(3),JPOSN(3),POINT(100,3),
     &LX(2),LX1(2),LY(2),LY1(2),LZ(2),LZ1(2)

      CHARACTER*7 PNAM(100)
      CHARACTER*20 X,Y,Z,CLEAR
      CHARACTER*16 XVAL,YVAL,ZVAL,RETURN

      DATA RETURN/'1.  RETURN      '/
      DATA LENG/16/
      DATA POSN/0.05,2.2/

      DATA DATA/32,0,0,          2,2,1,1,
     &                         1,1,1,1,1,1,
     &                         1,1,1,1,1,1,
     &                         1,1,1,1,1,1,
     &                         1,1,1,1,1,1,
     &                           1,1,1,1/
C                             FLAG FOR UPDATE
      DATA PRFORM/2/
C                             INCLUSION CLASS LIST
      DATA ACLASS/2,3,4/

      DATA X     /'DX=                '/
      DATA Y     /'DY=                '/
      DATA Z     /'DZ=                '/
      DATA CLEAR /'                   '/


      LX(1)=0.05
      LX(2)=0.68

      LX1(1)=0.14
      LX1(2)=0.68

      LY(1)=0.05
      LY(2)=0.62

      LY1(1)=0.14
      LY1(2)=0.62

      LZ(1)=0.05
      LZ(2)=0.56

      LZ1(1)=0.14
      LZ1(2)=0.56

*********************************************************************
*  REQUEST PICK INPUT
*********************************************************************
C                             SET PICK FILTER(ALL CLASS 2 DETECTABLE)
      CALL GPPKF(IWSID,1,3,ACLASS,0,ACLASS)
C                             MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                             DEFINE PICK AREA
      PAREA(1)=((-.98+1)/2)*CSIZE(1)
      PAREA(2)=((.98+1)/2)*CSIZE(1)
      PAREA(3)=((-.98+1)/2)*CSIZE(2)
      PAREA(4)=((.48+1)/2)*CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=CSIZE(3)
C                             INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                             PLACE PICK DEVICE IN THE EVENT MODE
      CALL GPPKMO(IWSID,1,3,2)
C                             TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)
C                             GET PICK OR RETURN COMMAND
  200 CALL CMMAND(IWSID,34)

*********************************************************************
*  INCLUDE A RETURN PICK
*********************************************************************
```

```
C                               RETURN PICK
      CALL GPOPST(6)
      CALL GPADCN(1,ACLASS)
      CALL GPPKID(101)
      CALL GPTXCI(2)
      CALL GPAHSC(1.0)
      CALL GPAN2(POSN,LENG,RETURN)
      CALL GPCLST
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
      CALL GPUPWS(IWSID,PRFORM)

*************************************************************************
*  AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
*************************************************************************
C                               AWAIT AN EVENT
  210 CALL GPAWEV(1000.,IWS,ICLA,IDEV)
      IF(ICLA.EQ.5)THEN
C                               GET PICK
         CALL GPGTPK(3,I,PPATH)
C                               IF RETURN PICKED
         IF(PPATH(2).EQ.101)THEN
              RFLG=1
              GO TO 1000
         ENDIF
C                               ERASE RETURN PROMPT
         CALL GPEST(6)
C                               CHOICE IS SECOND ITEM OF PICK PATH
         CHOICE=PPATH(2)
         IS=PPATH(1)
         IF(IS.EQ.12)THEN
C                               GET POINT COORDINATES
            P(1)=POINT(CHOICE,1)
            P(2)=POINT(CHOICE,2)
            P(3)=POINT(CHOICE,3)
         ENDIF
         IF(IS.EQ.13.OR.IS.EQ.14.OR.IS.EQ.17.OR.IS.EQ.18)THEN
C                               GET POINT COORDINATES
            P(1)=JOINT(CHOICE,1)
            P(2)=JOINT(CHOICE,2)
            P(3)=JOINT(CHOICE,3)
         ENDIF

         P1(1)=JPOSN(1)
         P1(2)=JPOSN(2)
         P1(3)=JPOSN(3)

         VEC(1)=P(1)-P1(1)
         VEC(2)=P(2)-P1(2)
         VEC(3)=P(3)-P1(3)

         IF(VEC(1).EQ.0.0.AND.VEC(2).EQ.0.0.AND.VEC(3).EQ.0.0)THEN
C                               POINT AND ORIGIN ARE THE SAME -- RE-ENTER
            CALL CMMAND(IWSID,84)
            GO TO 210
         ENDIF

         MAG=((VEC(1)**2)+(VEC(2)**2)+(VEC(3)**2))**.5

         LOCAT(1)=VEC(1)/MAG
         LOCAT(2)=VEC(2)/MAG
         LOCAT(3)=VEC(3)/MAG
C                               DRAW ORIENTATION ICON
         CALL GPEST(16)

         CALL GPOPST(16)
         CALL ICONO(LOCAT,AXISR)
         CALL GPCLST
C                               DISPLAY VALUES

         REWIND 10
         DO 350 III=1,3
              WRITE(10,300)CLEAR
  300         FORMAT(A20)
  350    CONTINUE
         REWIND 10
         WRITE(10,375)LOCAT(1)
         WRITE(10,375)LOCAT(2)
         WRITE(10,375)LOCAT(3)
  375    FORMAT(F16.14)
         REWIND 10
         READ(10,400)XVAL
         READ(10,400)YVAL
         READ(10,400)ZVAL
  400    FORMAT(A14)

         CALL GPOPST(10)
         CALL GPTXCI(1)
         CALL GPAHSC(1.0)
         CALL GPAN2(LX,20,X)
         CALL GPAN2(LX1,16,XVAL)
         CALL GPAN2(LY,20,Y)
         CALL GPAN2(LY1,16,YVAL)
         CALL GPAN2(LZ,20,Z)
```

```
         CALL GPAN2(LZ1,16,ZVAL)
         CALL GPCLST
C                              ENTER ORIENTATION OR RETURN
         CALL CMMAND(IWSID,32)
         CALL MENU(IWSID,CSIZE,8,CHOICE)

         IF(CHOICE.EQ.1)THEN
              RFLG=0
              GO TO 1000
         ENDIF

         IF(CHOICE.EQ.2)THEN
              RFLG=1
              LOCAT(1)=0.
              LOCAT(2)=0.
              LOCAT(3)=0.
              GO TO 1000
         ENDIF

      ELSE
C                              IF NOT A PICK, GO TO AWAIT AN EVENT
         GO TO 200
      ENDIF


C                              MAKE SURE PICK IS DEACTIVATED
 1000 CALL GPPKMO(IWSID,1,1,2)
      CALL GPEST(6)
      CALL GPEST(10)
      CALL GPEST(16)

C                              SET PICK FILTER(ALL CLASS 2 UNDETECTABLE)
      CALL GPPKF(IWSID,1,0,ACLASS,3,ACLASS)

      RETURN
      END
```

SUBROUTINE TEPOX

```
      SUBROUTINE TEPOX(IWSID,NUMJ,IAXIS,POINT,AXISR,CSIZE,XVEC,YVEC,
     &JOINT,RFLG)
*************************************************************************
* ----- SUBROUTINE TEPOX -----                                          *
*                                                                       *
*                                                                       *
*   THIS SUBROUTINE WILL PROMPT FOR THE ENTRY OF JOINT X AXIS           *
*                                                                       *
*   IN THE ORIENTATION OF AN EXISTING POINT                             *
*                                                                       *
*************************************************************************

      INTEGER IWSID,PRFORM,PPATH(3),ACLASS(1),CHOICE,LENG,IAXIS,
     &RFLG

      REAL CSIZE(3),DATA(35),PAREA(6),L1(4),POINT(100,3),AXISR,POSN(2),
     &P(3),P1(3),VEC(3),MAG,LOCAT(3),XVEC(3),YVEC(3),ZVEC(3),
     &JOINT(20,21)

      CHARACTER*20 CLEAR
      CHARACTER*16 RETURN

      DATA RETURN/'1.  RETURN      '/
      DATA LENG/16/
      DATA POSN/0.05,2.2/

      DATA DATA/32,0,0,          2,2,1,1,
     &                         1,1,1,1,1,1,
     &                         1,1,1,1,1,1,
     &                         1,1,1,1,1,1,
     &                         1,1,1,1,1,1,
     &                           1,1,1,1/
C                              FLAG FOR UPDATE
      DATA PRFORM/2/
C                              INCLUSION CLASS LIST(CLASS 4 -- JOINTS AXIS
      DATA ACLASS/2/

*************************************************************************
*  REQUEST PICK INPUT
*************************************************************************
C                              SET PICK FILTER(ALL CLASS 3,4 DETECTABLE)
      CALL GPPKF(IWSID,1,1,ACLASS,0,ACLASS)
C                              MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
C                              DEFINE PICK AREA
      PAREA(1)=((-.98+1)/2)*CSIZE(1)
```

```
      PAREA(2)=((.98+1)/2)*CSIZE(1)
      PAREA(3)=((-.98+1)/2)*CSIZE(2)
      PAREA(4)=((.48+1)/2)*CSIZE(2)
      PAREA(5)=0.
      PAREA(6)=CSIZE(3)
C                             INTIALIZE PICK PATH
      CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)
C                             PLACE PICK DEVICE IN THE EVENT MODE
      CALL GPPKMO(IWSID,1,3,2)
C                             TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)
C                             GET PICK OR RETURN COMMAND

*********************************************************************
*  INCLUDE A RETURN PICK
*********************************************************************

C                             RETURN PICK
  200 CALL GPOPST(6)
      CALL GPADCN(1,ACLASS)
      CALL GPPKID(101)
      CALL GPTXCI(2)
      CALL GPAHSC(1.0)
      CALL GPAN2(POSN,LENG,RETURN)
      CALL GPCLST
      CALL GPARV(IWSID,3,6,0.)
      CALL GPVP(IWSID,3,1,1)
      CALL GPUPWS(IWSID,PRFORM)

*********************************************************************
*  AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
*********************************************************************
C                             AWAIT AN EVENT
  300 CALL GPAWEV(1000.,IWS,ICLA,IDEV)
      IF(ICLA.EQ.5)THEN
C                             GET PICK
         CALL GPGTPK(1,1,PPATH)
C                             IF RETURN
         IF(PPATH(2).EQ.101)THEN
              RFLG=1
              GO TO 1000
         ENDIF
C                             CHOICE IS SECOND ITEM OF PICK PATH
         END=PPATH(2)
         ISTRUC=PPATH(1)

         VEC(1)=POINT(END,1)-JOINT(NUMJ,1)
         VEC(2)=POINT(END,2)-JOINT(NUMJ,2)
         VEC(3)=POINT(END,3)-JOINT(NUMJ,3)

         IF(IAXIS.EQ.1)THEN
              ZVEC(1)=JOINT(NUMJ,4)
              ZVEC(2)=JOINT(NUMJ,5)
              ZVEC(3)=JOINT(NUMJ,6)
         ENDIF
         IF(IAXIS.EQ.2)THEN
              ZVEC(1)=JOINT(NUMJ,18)
              ZVEC(2)=JOINT(NUMJ,19)
              ZVEC(3)=JOINT(NUMJ,20)
         ENDIF

         CALL VPROD(ZVEC,VEC,YVEC)
         IF(YVEC(1).EQ.0.0.AND.YVEC(2).EQ.0.0.AND.YVEC(3).EQ.0.0)THEN
C                                  X POINT ON Z VECTOR
              CALL CMMAND(IWSID,55)
              GO TO 300
         ENDIF
         CALL VPROD(YVEC,ZVEC,XVEC)

         XMAG=((XVEC(1)**2)+(XVEC(2)**2)+(XVEC(3)**2))**.5
         YMAG=((YVEC(1)**2)+(YVEC(2)**2)+(YVEC(3)**2))**.5

         XVEC(1)=XVEC(1)/XMAG
         XVEC(2)=XVEC(2)/XMAG
         XVEC(3)=XVEC(3)/XMAG

         YVEC(1)=YVEC(1)/YMAG
         YVEC(2)=YVEC(2)/YMAG
         YVEC(3)=YVEC(3)/YMAG

      ENDIF
C                             MAKE SURE PICK IS DEACTIVATED
 1000 CALL GPPKMO(IWSID,1,1,2)
      CALL GPEST(6)

C                             SET PICK FILTER(ALL CLASS 2 UNDETECTABLE)
      CALL GPPKF(IWSID,1,0,ACLASS,1,ACLASS)


      RETURN
      END
```

```
      SUBROUTINE VEO(IWSID,AXISR,CSIZE,ORIENT,RFLG)
*********************************************************************
* ----- SUBROUTINE VEO  -----                                       *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE WILL PROMPT FOR THE VALUATOR ENTRY OF           *
*                                                                   *
*   THE JOINT ORIENTATION                                           *
*                                                                   *
*********************************************************************

      INTEGER IWSID,ECHOSW,EVENT,DLEN,ECHO,ACLASS,PRFORM,CHOICE,
     & PPATH(3),RFLG,NUMV

      REAL CSIZE(3),AREA1(6),AREA2(6),AREA3(6),VLUE1,LOVAL,HIVAL,AXISR,
     & BOX(8),SHD(8),ORIENT(3),MATRIX(1000,4),MATUSE(4,4),LINE(6),
     & LINEX(6),LINEY(6),LINEZ(6),XBOX(15),
     & YBOX(15),ZBOX(15),LX(6),LY(6),LZ(6),OL(6),AREA(6)

      CHARACTER*7 CLEAR,INIT

      DATA CLEAR/'       '/
      DATA INIT /'0.     '/

      DATA LX/0.,0.,0.,0.,0.,0./
      DATA LY/0.,0.,0.,0.,0.,0./
      DATA LZ/0.,0.,0.,0.,0.,0./

C                             FLAG FOR UPDATE
      DATA PRFORM/2/

C                             INCLUSION CLASS LIST
      DATA ACLASS/1/
      DATA VLUE1/0./
      DATA ECHO/3/
      DATA DLEN/0/
      DATA EVENT/3/
      DATA ECHOSW/2/

C                             GREY VALUATOR BOX
      DATA BOX/0.00,0.545,0.48,0.545,0.48,0.73,0.00,0.73/

      AREA(1)=.08*CSIZE(1)
      AREA(2)=.90*CSIZE(1)
      AREA(3)=.70*CSIZE(2)
      AREA(4)=.90*CSIZE(2)
      AREA(5)=.02*CSIZE(1)
      AREA(6)=.90*CSIZE(3)

      AREA1(1)=.52*CSIZE(1)
      AREA1(2)=CSIZE(1)
      AREA1(3)=.855*CSIZE(2)-.006
      AREA1(4)=CSIZE(2)
      AREA1(5)=0.
      AREA1(6)=CSIZE(3)

      AREA2(1)=.52*CSIZE(1)
      AREA2(2)=CSIZE(1)
      AREA2(3)=.855*CSIZE(2)-.014
      AREA2(4)=CSIZE(2)
      AREA2(5)=0.
      AREA2(6)=CSIZE(3)

      AREA3(1)=.52*CSIZE(1)
      AREA3(2)=CSIZE(1)
      AREA3(3)=.855*CSIZE(2)-.022
      AREA3(4)=CSIZE(2)
      AREA3(5)=0.
      AREA3(6)=CSIZE(3)

      NUMV=1

      LOVAL=-1.0
      HIVAL=1.0

*********************************************************************
*  INITIALIZE AND ACTIVATE DIALS 1,2,3 AND STRING DEVICE
*********************************************************************
C                             PLACE VALUATORS IN THE REQUEST MODE
  300 CALL GPVLMO(IWSID,1,1,ECHOSW)
      CALL GPVLMO(IWSID,2,1,ECHOSW)
      CALL GPVLMO(IWSID,3,1,ECHOSW)
C                             INITIALIZE DIALS FOR X , Y , Z INPUT
      CALL GPINVL(IWSID,1,VLUE1,ECHO,AREA1,LOVAL,HIVAL,DLEN,DATA)
      CALL GPINVL(IWSID,2,VLUE1,ECHO,AREA2,LOVAL,HIVAL,DLEN,DATA)
      CALL GPINVL(IWSID,3,VLUE1,ECHO,AREA3,LOVAL,HIVAL,DLEN,DATA)
C                             ACTIVATE DIALS FOR X , Y , Z INPUT
      CALL GPVLMO(IWSID,1,EVENT,ECHOSW)
      CALL GPVLMO(IWSID,2,EVENT,ECHOSW)
      CALL GPVLMO(IWSID,3,EVENT,ECHOSW)
```

```
*************************************************************************
*  SCREEN DISPLAY
*************************************************************************
C                              UPDATE THE WORKSTATION
      CALL GPUPWS(IWSID,PRFORM)
C     CALL GPARW(IWSID,16)

*************************************************************************
*  DISPLAY MENU AND GET CHOICE OF ENTER, RETURN OR VALUATOR
*************************************************************************


  500 CALL CMMAND(IWSID,33)
      CALL ENTERO(IWSID,CHOICE,CSIZE)

  550 ORIENT(1)=0.
      ORIENT(2)=0.
      ORIENT(3)=0.

  600 CALL GPAWEV(1000.,IIWSID,ICLASS,IDEV)
      IF(ICLASS.EQ.3)THEN

C                                PROCESS VALUATORS (NOT IN SUBROUTINE TO
C                                                    SPEED EXECUTION)

         CALL GPGTVL(VALUE)

         IF(IDEV.EQ.1)ORIENT(1)=VALUE
         IF(IDEV.EQ.2)ORIENT(2)=VALUE
         IF(IDEV.EQ.3)ORIENT(3)=VALUE

*************************************************************************
* GET ORIENT. ICON                                                      *
*************************************************************************

         LX(4)=ORIENT(1)
         LY(5)=ORIENT(2)
         LZ(6)=ORIENT(3)

         OL(4)=ORIENT(1)
         OL(5)=ORIENT(2)
         OL(6)=ORIENT(3)


         CALL GPEST(16)

         CALL GPOPST(16)

C                                SET POLYLINE COLOR
         CALL GPPLCI(1)
C                                SET LINE SOLID
C        CALL GPLT(1)
C                                SET POLYLINE COLOR
         CALL GPPLCI(5)
         CALL GPPL3(2,3,LX)
         CALL GPPL3(2,3,LY)
         CALL GPPL3(2,3,LZ)
         CALL GPPLCI(2)
         CALL GPPL3(2,3,OL)
         CALL GPCLST
         CALL GPARV(IWSID,6,16,0.)
         CALL GPUPWS(IWSID,PRFORM)
         GO TO 600
      ENDIF

      IF(ICLASS.EQ.5)THEN
         CALL GPGTPK(1,1,PPATH)
         CHOICE=PPATH(2)
              IF(CHOICE.EQ.1)THEN
                RMAG=((ORIENT(1)**2)+(ORIENT(2)**2)+(ORIENT(3)**2))**.5
                   IF(RMAG.EQ.0.0)THEN
                        CALL CMMAND(IWSID,39)
                        GO TO 550
                   ENDIF
                   ORIENT(1)=ORIENT(1)/RMAG
                   ORIENT(2)=ORIENT(2)/RMAG
                   ORIENT(3)=ORIENT(3)/RMAG
C                             DEACTIVATE VALUATORS
                   CALL GPVLMO(IWSID,1,1,1)
                   CALL GPVLMO(IWSID,2,1,1)
                   CALL GPVLMO(IWSID,3,1,1)
                   RFLG=0
                   CALL GPEST(16)
                   GO TO 1000
              ENDIF
              IF(CHOICE.EQ.2)THEN
C                             DEACTIVATE VALUATORS
                   CALL GPVLMO(IWSID,1,1,1)
                   CALL GPVLMO(IWSID,2,1,1)
                   CALL GPVLMO(IWSID,3,1,1)
                   RFLG=1
                   CALL GPEST(16)
```

```
                GO TO 1000
            ENDIF
      ENDIF

 1000 RETURN
      END
```

SUBROUTINE VUNIT

```
      SUBROUTINE VUNIT(V,U)
*************************************************************************
*     ----- SUBROUTINE VUNIT -----                                       *
*                                                                         *
*     THIS SUBROUTINE IS USED TO DETERMINE A UNIT VECTOR                  *
*                                                                         *
*************************************************************************

      DIMENSION V(3),U(3)
      REAL MAG

      IF(V(1).EQ.0..AND.V(2).EQ.0..AND.V(3).EQ.0.)THEN
         V(1)=.0000001
      ENDIF


      MAG=((V(1)**2)+(V(2)**2)+(V(3)**2))**.5

      DO 20 I=1,3
         U(I)=V(I)/MAG
   20 CONTINUE

      RETURN
      END
```

SUBROUTINE WRANAL

```
      SUBROUTINE WRANAL(PNAM,PNUM,POINT,JNAM,JNUM,JOINT,LNAM,LNUM,LINK,
     &CSIZE,INAM,INUM,IC,IWSID)
*************************************************************************
* ----- SUBROUTINE WRANAL -----                                         *
*                                                                       *
*                                                                       *
*   THIS SUBROUTINE IS USED WRITE THE ANALYSIS FILE                     *
*                                                                       *
*************************************************************************

      INTEGER PNUM(100),LNUM(30,14),JNUM(20,3),INUM(20,2),NP,NJ,NL,NI,
     &CONN(20,3)

      REAL LINK(30,6),JOINT(20,21),POINT(100,3),AXISR,CSIZE(3),AREA(6),
     &IC(20,4)

      CHARACTER*7 LNAM(30),JNAM(20),PNAM(100),FILE,FILEO,FILE1,INAM(20),
     &            JNAME,LNAME1,LNAME2,FILEN

*************************************************************************
*  FIND OUT TYPE OF ANALYSIS FILE TO BE WRITTEN
*************************************************************************
      CALL CMMAND(IWSID,88)
   10 CALL MENU(IWSID,CSIZE,18,ICH)

      IF(ICH.EQ.1)THEN
C                          IMP
         CALL IMP(PNAM,PNUM,POINT,JNAM,JNUM,JOINT,LNAM,LNUM,LINK,
     &            CSIZE,INAM,INUM,IC,IWSID)
      ENDIF

      IF(ICH.EQ.2)THEN
C                          RSCR
         CALL RSCRA(PNAM,PNUM,POINT,JNAM,JNUM,JOINT,LNAM,LNUM,LINK,
     &           CSIZE,INAM,INUM,IC,IWSID,IRETT)
         IF(IRETT.EQ.1)THEN
            CALL CMMAND(IWSID,90)
            GO TO 10
         ENDIF
      ENDIF

      IF(ICH.EQ.3)THEN
C                          RSSR (NOT YET AVAILABLE)
         CALL CMMAND(IWSID,89)
         GO TO 10
      ENDIF
```

```
      IF(ICH.EQ.4)THEN
C                              RETURN
         CONTINUE
      ENDIF

      RETURN
      END
```

SUBROUTINE WRREST

```
      SUBROUTINE WRREST(PNAM,PNUM,POINT,JNAM,JNUM,JOINT,LNAM,LNUM,LINK,
     &AXISR,FILE,IWSID,CSIZE,INAM,INUM,IC,SCALE)
*********************************************************************
* ----- SUBROUTINE WRREST -----                                     *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED WRITE THE ANALYSIS RESTART FILE         *
*                                                                   *
*********************************************************************

      INTEGER PNUM(100),LNUM(30,14),JNUM(20,3),INUM(20,2),NP,NJ,NL,NI

      REAL LINK(30,6),JOINT(20,21),POINT(100,3),AXISR,CSIZE(3),AREA(6),
     &IC(20,4)

      CHARACTER*7 LNAM(30),JNAM(20),PNAM(100),FILE,FILEO,FILE1,INAM(20)
      CHARACTER*28 FLINE

      DATA FLINE/'MECHIN RESTART FILE-ANALYSIS'/

      NP=0
      NJ=0
      NL=0
      NI=0

      AREA(1)=.08*CSIZE(1)
      AREA(2)=.90*CSIZE(1)
      AREA(3)=.70*CSIZE(2)
      AREA(4)=.90*CSIZE(2)
      AREA(5)=.02*CSIZE(3)
      AREA(6)=.90*CSIZE(3)

      FILE1='       '

      REWIND 15
*********************************************************************
*  INITIALIZE THE STRING DEVICE. REQUEST NEW RESTART FILE NAME
*********************************************************************
C                              INITIALIZE STRING MODE
      CALL GPSTMO(IWSID,1,1,2)
C                              INITIALIZE THE STRING DEVICE
    1 CALL GPINST(IWSID,1,7,FILE,1,AREA,7,1,0,FILE)
C                              REQUEST THE FILE NAME
      CALL CMMAND(IWSID,60)
C                              REQUEST STRING
    2 CALL GPRQST(IWSID,1,7,ISTAT,NUM,FILE1)

      IF(FILE1.EQ.'       ')THEN
         GO TO 2
      ENDIF

      IF(FILE1.EQ.FILE)THEN
         GO TO 9
      ENDIF

      FILE=FILE1
C                              SEE IF FILE ALREADY EXISTS
    3 CALL FILEXS(FILE,IRET)
C     IF(IRET.EQ.0)THEN        IF FILE EXISTS, HAVE USER RETYPE TO USE
         CALL GPSTMO(IWSID,1,1,2)
         FILEO=FILE
         FILE='       '
         CALL CMMAND(IWSID,8)
         CALL GPINST(IWSID,1,7,FILE,1,AREA,7,1,0,FILE)
         CALL GPRQST(IWSID,1,7,ISTAT,NUM,FILE)
         IF(FILE1.EQ.'       ')THEN
            GO TO 2
         ENDIF
         IF(FILE.NE.FILEO)THEN
            GO TO 3
         ENDIF
      ENDIF

      CLOSE(UNIT=15)
C                              OPEN A NEW FILE
      OPEN(UNIT=15,FILE=FILE,STATUS='NEW',ERR=2)
```

```
*************************************************************************
*    WRITE FIRST LINE OF THE RESTART FILE AND AXIS RANGE
*************************************************************************

    9 WRITE(15,10)FLINE
   10 FORMAT(A28)
      WRITE(15,*)AXISR
      WRITE(15,*)SCALE


*************************************************************************
*    WRITE POINT DATA TO THE RESTART FILE
*************************************************************************

      DO 80 I=1,100
         IF(PNAM(I).EQ.'BBBBBBB')THEN
            GO TO 90
         ENDIF
         NP=NP+1
   80 CONTINUE

   90 WRITE(15,*)NP

      DO 100 I=1,100
         IF(PNAM(I).EQ.'BBBBBBB')THEN
            GO TO 110
         ELSE
              WRITE(15,91)PNAM(I)
   91         FORMAT(A7)
              WRITE(15,*)PNUM(I)
              WRITE(15,*)(POINT(I,J),J=1,3)
         ENDIF
  100 CONTINUE

*************************************************************************
*    WRITE JOINT DATA TO THE RESTART FILE
*************************************************************************

  110 DO 180 I=1,20
         IF(JNAM(I).EQ.'BBBBBBB')THEN
            GO TO 190
         ENDIF
         NJ=NJ+1
  180 CONTINUE

  190 WRITE(15,*)NJ

      DO 200 I=1,20
         IF(JNAM(I).EQ.'BBBBBBB')THEN
            GO TO 210
         ELSE
              WRITE(15,191)JNAM(I)
  191         FORMAT(A7)
              WRITE(15,*)JNUM(I,1),JNUM(I,2),JNUM(I,3)
              WRITE(15,*)(JOINT(I,J),J=1,21)
         ENDIF
  200 CONTINUE

*************************************************************************
*    WRITE LINK DATA TO THE RESTART FILE
*************************************************************************

  210 DO 280 I=1,30
         IF(LNAM(I).EQ.'BBBBBBB')THEN
            GO TO 290
         ENDIF
         NL=NL+1
  280 CONTINUE

  290 WRITE(15,*)NL

      DO 300 I=1,20
         IF(LNAM(I).EQ.'BBBBBBB')THEN
            GO TO 310
         ELSE
              WRITE(15,291)LNAM(I)
  291         FORMAT(A7)
              WRITE(15,*)(LNUM(I,J),J=1,14)
              WRITE(15,*)(LINK(I,J),J=1,6)
         ENDIF
  300 CONTINUE

*************************************************************************
*    WRITE JOINT DATA TO THE RESTART FILE
*************************************************************************

  310 DO 400 I=1,20
              WRITE(15,391)INAM(I)
  391         FORMAT(A7)
              WRITE(15,*)INUM(I,1),INUM(I,2)
              WRITE(15,*)(IC(I,J),J=1,4)
  400 CONTINUE
```

```
  410 RETURN
      END
```

SUBROUTINE WRSYN

```
      SUBROUTINE WRSYN(PNAM,PNUM,POINT,AXISR,CSIZE,IWSID)
*********************************************************************
* ----- SUBROUTINE WRSYN  -----                                     *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED WRITE THE SYNTHESIS DATA FILE           *
*                                                                   *
*********************************************************************

      INTEGER PNUM(30,4),NP

      REAL POINT(30,14),AXISR,CSIZE(3),AREA(6)

      CHARACTER*7 PNAM(30),FILE,FILEO,FILE1,FILEN

*********************************************************************
*  FIND OUT TYPE OF SYNTHESIS FILE TO BE WRITTEN
*********************************************************************
      CALL SCMMAN(IWSID,23)
   10 CALL SMENU(IWSID,CSIZE,6,ICH)

      IF(ICH.EQ.1)THEN
C                             RSCR
         CALL RSCRS(PNAM,PNUM,POINT,AXISR,CSIZE,IWSID,IRET)
         IF(IRET.EQ.1)THEN
            GO TO 10
         ENDIF
      ENDIF

      IF(ICH.EQ.2)THEN
C                             RCCC (NOT YET AVAILABLE)
         CALL SCMMAN(IWSID,24)
         GO TO 10
      ENDIF

      IF(ICH.EQ.3)THEN
C                             RETURN
         CONTINUE
      ENDIF

      RETURN
      END
```

SUBROUTINE ZOOM

```
      SUBROUTINE ZOOM(IWSID,AXISR,CSIZE)
*********************************************************************
* ----- SUBROUTINE ZOOM -----                                       *
*                                                                   *
*                                                                   *
*   THIS SUBROUTINE IS USED TO ZOOM IN ON AN AREA ON THE SCREEN     *
*                                                                   *
*********************************************************************

      INTEGER IWSID,PARELL,PRFORM,ILOC(3),DATA(76)


      REAL AXISR,XAXIS(24),YAXIS(24),ZAXIS(24),WINDOW(4),VP1(6),VP2(6),
     &BOX1(8),BOX4(8),NEAR,FAR,POINT(3),DIST,UP(3),NORMAL(3),
     &MATRIX(4,4),POINTP(3),PAREA(6),POS1(3),POS2(3),CSIZE(3),W1(4)

C                               FLAG FOR UPDATE
      DATA PRFORM/2/
      DATA DIST/0./
      DATA PARELL/1/
      DATA POINT/0.,0.,0./
      DATA POINTP/2.0,1.5,5.0/
      DATA NORMAL/1.,0.,0./
      DATA UP/0.,0.,1./
      DATA ILOC/0.,0.,0./
      DATA BOX1/-.48,-.98,.98,-.98,.98,.48,-.48,.48/
      DATA DATA/1,75*0/

      NEAR=AXISR+(.64*AXISR)
      FAR=-AXISR-(.64*AXISR)

      VP1(1)=(BOX1(1)+1.)/2.
      VP1(2)=(BOX1(3)+1.)/2.
      VP1(3)=(BOX1(4)+1.)/2.
```

```
      VP1(4)=(BOX1(6)+1.)/2.
      VP1(5)=(BOX1(1)+1.)/2.
      VP1(6)=(BOX1(3)+1.)/2.

*************************************************************************
*  ACTIVATE THE LOCATOR TO GET NEW VIEW PORT
*************************************************************************
C                              MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
      CALL GPPKMO(IWSID,1,1,2)
      CALL GPLCMO(IWSID,1,1,2)
C                              DEFINE PICK AREA
      PAREA(1)=VP1(1)*CSIZE(1)
      PAREA(2)=VP1(2)*CSIZE(1)
      PAREA(3)=VP1(3)*CSIZE(2)
      PAREA(4)=VP1(4)*CSIZE(2)
      PAREA(5)=VP1(5)*CSIZE(3)
      PAREA(6)=VP1(6)*CSIZE(3)
C                              INTIALIZE LOCATOR
      CALL GPINLC(IWSID,1,4,ILOC,1,PAREA,0,DATA)
C                              TRAVERSE ALL ACTIVE VIEWS
      CALL GPUPWS(IWSID,PRFORM)
      CALL CMMAND(IWSID,57)
      CALL GPRQLC(IWSID,1,ISTAT,IV,POS1)
C                              INTIALIZE LOCATOR FOR BOX
      CALL GPINLC(IWSID,1,4,POS1,5,PAREA,72,DATA)
      CALL CMMAND(IWSID,58)
      CALL GPRQLC(IWSID,1,ISTAT,IV,POS2)

      POS1(1)=POS1(1)-(.6550*AXISR)
      POS1(2)=POS1(2)-(.4912*AXISR)
      POS2(1)=POS2(1)-(.6550*AXISR)
      POS2(2)=POS2(2)-(.4912*AXISR)

      XRANG=ABS(POS1(1)-POS2(1))
      YRANG=ABS(POS1(2)-POS2(2))
      IF(YRANG.GE.XRANG)THEN
         WINDOW(1)=((POS2(1)+POS1(1))/2.)-(YRANG/2.)
         WINDOW(2)=((POS2(1)+POS1(1))/2.)+(YRANG/2.)
         WINDOW(3)=((POS1(2)+POS2(2))/2.)-(YRANG/2.)
         WINDOW(4)=((POS1(2)+POS2(2))/2.)+(YRANG/2.)
         POINTP(1)=((POS1(1)+POS2(1))/2.)+2.0
         POINTP(2)=((POS1(2)+POS2(2))/2.)+1.5
      ENDIF
      IF(YRANG.LT.XRANG)THEN
         WINDOW(1)=((POS1(1)+POS2(1))/2.)-(XRANG/2.)
         WINDOW(2)=((POS1(1)+POS2(1))/2.)+(XRANG/2.)
         WINDOW(3)=((POS1(2)+POS2(2))/2.)-(XRANG/2.)
         WINDOW(4)=((POS1(2)+POS2(2))/2.)+(XRANG/2.)
         POINTP(1)=((POS1(1)+POS2(1))/2.)+2.0
         POINTP(2)=((POS1(2)+POS2(2))/2.)+1.5
      ENDIF

*************************************************************************
*  SET THE VIEW CHARACTERISTICS FOR AXES VIEW
*************************************************************************
C                              SET THE CHARACTERISTICS
      CALL GPVMP3(IWSID,4,WINDOW,VP1,PARELL,POINTP,DIST,NEAR,FAR)

C                              ACTIVATE THE VIEW
      CALL GPVCH(IWSID,4,2,2,2,1,0,2,5,2)


      CALL GPUPWS(IWSID,PRFORM)

      RETURN
      END
```

The vita has been removed from the scanned document