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

INTRODUCTION ............................................................... 1

LITERATURE REVIEW ......................................................... 3
  Computer-Aided Mechanism Design ........................................ 3
  Interactive Graphics Software .......................................... 5

MECHIN PROGRAM DEVELOPMENT ......................................... 7
  Introduction to PHIGS .................................................... 7
  Three-Dimensional Data Display ......................................... 9
  Screen Design .................................................................... 11
  Analysis Data Input .......................................................... 15
  Synthesis Data Input ......................................................... 17

MECHIN PROGRAM STRUCTURE ............................................ 19
  Program Organization ....................................................... 19
  Data Storage ..................................................................... 28

Table of Contents iv
<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBROUTINE DELL</td>
<td>95</td>
</tr>
<tr>
<td>SUBROUTINE DELP</td>
<td>96</td>
</tr>
<tr>
<td>SUBROUTINE DEO</td>
<td>97</td>
</tr>
<tr>
<td>SUBROUTINE DEP</td>
<td>99</td>
</tr>
<tr>
<td>SUBROUTINE DRAWJT</td>
<td>102</td>
</tr>
<tr>
<td>SUBROUTINE DRAWK</td>
<td>105</td>
</tr>
<tr>
<td>SUBROUTINE DRAWPT</td>
<td>108</td>
</tr>
<tr>
<td>SUBROUTINE ENTER</td>
<td>108</td>
</tr>
<tr>
<td>SUBROUTINE ENTERO</td>
<td>110</td>
</tr>
<tr>
<td>SUBROUTINE FILENM</td>
<td>111</td>
</tr>
<tr>
<td>SUBROUTINE FILEXS</td>
<td>112</td>
</tr>
<tr>
<td>SUBROUTINE ICICON</td>
<td>112</td>
</tr>
<tr>
<td>SUBROUTINE ICON</td>
<td>113</td>
</tr>
<tr>
<td>SUBROUTINE ICOND</td>
<td>114</td>
</tr>
<tr>
<td>SUBROUTINE ICONO</td>
<td>115</td>
</tr>
<tr>
<td>SUBROUTINE IMP</td>
<td>115</td>
</tr>
<tr>
<td>SUBROUTINE INQ</td>
<td>119</td>
</tr>
<tr>
<td>SUBROUTINE INQIC</td>
<td>121</td>
</tr>
<tr>
<td>SUBROUTINE INQL</td>
<td>124</td>
</tr>
<tr>
<td>SUBROUTINE INQO</td>
<td>125</td>
</tr>
<tr>
<td>SUBROUTINE INVJ</td>
<td>128</td>
</tr>
<tr>
<td>SUBROUTINE INVL</td>
<td>128</td>
</tr>
<tr>
<td>SUBROUTINE JICONS</td>
<td>128</td>
</tr>
<tr>
<td>SUBROUTINE JOINTS</td>
<td>131</td>
</tr>
<tr>
<td>SUBROUTINE LINKS</td>
<td>131</td>
</tr>
<tr>
<td>SUBROUTINE MENU</td>
<td>132</td>
</tr>
<tr>
<td>SUBROUTINE NSTART</td>
<td>137</td>
</tr>
<tr>
<td>SUBROUTINE ORIEN</td>
<td>140</td>
</tr>
<tr>
<td>Subroutine</td>
<td>Page</td>
</tr>
<tr>
<td>------------------</td>
<td>------</td>
</tr>
<tr>
<td>SUBROUTINE PEJO</td>
<td>142</td>
</tr>
<tr>
<td>SUBROUTINE PICKJ</td>
<td>144</td>
</tr>
<tr>
<td>SUBROUTINE PICKJ2</td>
<td>145</td>
</tr>
<tr>
<td>SUBROUTINE PICKL2</td>
<td>147</td>
</tr>
<tr>
<td>SUBROUTINE POSIT</td>
<td>148</td>
</tr>
<tr>
<td>SUBROUTINE RDRWJT</td>
<td>152</td>
</tr>
<tr>
<td>SUBROUTINE RDRWLK</td>
<td>152</td>
</tr>
<tr>
<td>SUBROUTINE RDRWPT</td>
<td>153</td>
</tr>
<tr>
<td>SUBROUTINE REINV</td>
<td>153</td>
</tr>
<tr>
<td>SUBROUTINE RESTOR</td>
<td>154</td>
</tr>
<tr>
<td>SUBROUTINE RSCRA</td>
<td>154</td>
</tr>
<tr>
<td>SUBROUTINE RSCRS</td>
<td>157</td>
</tr>
<tr>
<td>SUBROUTINE RSTART</td>
<td>160</td>
</tr>
<tr>
<td>SUBROUTINE SADDP</td>
<td>162</td>
</tr>
<tr>
<td>SUBROUTINE SASSO</td>
<td>164</td>
</tr>
<tr>
<td>SUBROUTINE SCALNM</td>
<td>165</td>
</tr>
<tr>
<td>SUBROUTINE SCHSCA</td>
<td>166</td>
</tr>
<tr>
<td>SUBROUTINE SCHVIE</td>
<td>167</td>
</tr>
<tr>
<td>SUBROUTINE SCMMAN</td>
<td>168</td>
</tr>
<tr>
<td>SUBROUTINE SCRN1</td>
<td>172</td>
</tr>
<tr>
<td>SUBROUTINE SCRN2</td>
<td>175</td>
</tr>
<tr>
<td>SUBROUTINE SCRN3</td>
<td>177</td>
</tr>
<tr>
<td>SUBROUTINE SCRN4</td>
<td>178</td>
</tr>
<tr>
<td>SUBROUTINE SDATA</td>
<td>180</td>
</tr>
<tr>
<td>SUBROUTINE SDELP</td>
<td>180</td>
</tr>
<tr>
<td>SUBROUTINE SDRAWP</td>
<td>182</td>
</tr>
<tr>
<td>SUBROUTINE SETUP</td>
<td>184</td>
</tr>
<tr>
<td>SUBROUTINE SINQ</td>
<td>188</td>
</tr>
<tr>
<td>Subroutine</td>
<td>Page</td>
</tr>
<tr>
<td>--------------</td>
<td>------</td>
</tr>
<tr>
<td>SUBROUTINE SINQO</td>
<td>190</td>
</tr>
<tr>
<td>SUBROUTINE SMENU</td>
<td>192</td>
</tr>
<tr>
<td>SUBROUTINE SNSTAR</td>
<td>195</td>
</tr>
<tr>
<td>SUBROUTINE SORIEN</td>
<td>197</td>
</tr>
<tr>
<td>SUBROUTINE SPOSIT</td>
<td>199</td>
</tr>
<tr>
<td>SUBROUTINE SPT</td>
<td>202</td>
</tr>
<tr>
<td>SUBROUTINE SRDRWP</td>
<td>203</td>
</tr>
<tr>
<td>SUBROUTINE SRSTAR</td>
<td>204</td>
</tr>
<tr>
<td>SUBROUTINE STEPO</td>
<td>206</td>
</tr>
<tr>
<td>SUBROUTINE SWRRES</td>
<td>208</td>
</tr>
<tr>
<td>SUBROUTINE SYNIN</td>
<td>209</td>
</tr>
<tr>
<td>SUBROUTINE TEJOX</td>
<td>210</td>
</tr>
<tr>
<td>SUBROUTINE TEPO</td>
<td>212</td>
</tr>
<tr>
<td>SUBROUTINE TEPOX</td>
<td>214</td>
</tr>
<tr>
<td>SUBROUTINE VEO</td>
<td>216</td>
</tr>
<tr>
<td>SUBROUTINE VUNIT</td>
<td>218</td>
</tr>
<tr>
<td>SUBROUTINE WRANAL</td>
<td>218</td>
</tr>
<tr>
<td>SUBROUTINE WRREST</td>
<td>219</td>
</tr>
<tr>
<td>SUBROUTINE WRSYN</td>
<td>221</td>
</tr>
<tr>
<td>SUBROUTINE ZOOM</td>
<td>221</td>
</tr>
</tbody>
</table>

Vita .............................................. 223

Table of Contents viii
List of Illustrations

Figure 1. Three-Dimensional Positioning Cues ........................................ 10
Figure 2. Three-Dimensional Orientation Cues ......................................... 12
Figure 3. Data Entry Screen Design ....................................................... 13
Figure 4. Joint Icons ............................................................................. 16
Figure 5. Flowchart of PROGRAM MECHIN ........................................... 20
Figure 6. Flowchart of SUBROUTINE SETUP ......................................... 22
Figure 7. Title Screen ............................................................................ 23
Figure 8. Analysis and Synthesis Choice Screen ........................................ 24
Figure 9. Flowchart of SUBROUTINE ANIN ............................................. 25
Figure 10. Analysis Background Screen ................................................... 26
Figure 11. Flowchart for SUBROUTINE DATA .......................................... 27
Figure 12. Flowchart of SUBROUTINE SYNIN ........................................ 29
Figure 13. Flowchart of SUBROUTINE SDATA ......................................... 30
Figure 14. Data Entry Menu Hierarchy for Analysis ..................................... 40
Figure 15. Joint Icons and Orientations .................................................... 42
Figure 16. Data Entry Menu Hierarchy for Synthesis .................................. 44
Figure 17. RSCR Mechanism .................................................................. 47
Figure 18. MECHIN Representation of an RSCR Mechanism ....................... 48
Figure 19. MECHIN Generated IMP Code for an RSCR Mechanism .............. 49
Figure 20. Slider-Crank Mechanism ........................................ 50
Figure 21. Geared Piston Machine Mechanism .......................... 51
Figure 22. Body Positions for Mechanism Synthesis ....................... 53
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.
SUBROUTINE SETUP

1. OPEN PHIGS
   LOAD COLOR TABLE

2. INITIALIZE DATA ARRAYS

3. GET DISPLAY SURFACE SIZE

4. SET-UP WINDOW AND VIEWPORT DATA

5. CALL SCRNI
   DISPLAY MECHIN TITLE SCREEN

6. CALL SCRN2
   DISPLAY ANALYSIS OR SYNTHESIS CHOICE SCREEN

RETURN

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
SUBROUTINE SYNIN

CALL SCRN4
SET-UP SYNTHESIS BACKGROUND

CALL SCMMAN
ASK USER IF NEW OR RESTART MODEL

CALL SMENU
NEW OR RESTART MENU

CALL SRSTAR
RETRIEVE DATA FOR RESTART

CALL SNSTAR
SET-UP DATA FOR NEW START

CALL SDATA
INPUT AND MODIFY MECHANISM SYNTHESIS DATA

CALL SWRRES
WRITE MECHIN RESTART FILE

CALL WRSYN
WRITE SYNTHESIS FILE

RETURN

Figure 12. Flowchart of SUBROUTINE SYNIN
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.

<table>
<thead>
<tr>
<th>JNAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

where

\[
\text{JNAME is the name of the joint (up to 7 characters)}
\]

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.

<table>
<thead>
<tr>
<th>JNUM</th>
<th>TYPE</th>
<th>IFLG</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>POS1</th>
<th>ZAX1</th>
<th>XAX1</th>
<th>XAX2</th>
<th>SCRL</th>
<th>IOS</th>
<th>POS2</th>
<th>ZAX2</th>
<th>GR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>4-6</td>
<td>7-9</td>
<td>10-12</td>
<td>13</td>
<td>14</td>
<td>15-17</td>
<td>18-20</td>
<td>21</td>
</tr>
</tbody>
</table>

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
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.

<table>
<thead>
<tr>
<th>PNAME</th>
<th>1</th>
</tr>
</thead>
</table>

where

PNAM 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.

<table>
<thead>
<tr>
<th>PNUM</th>
<th>1</th>
</tr>
</thead>
</table>

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.

<table>
<thead>
<tr>
<th>XCOORD</th>
<th>YCOORD</th>
<th>ZCOORD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

where
XCOORD is the x coordinate of the point position

YCOORD is the y coordinate of the point position

ZCOORD 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.

```
<table>
<thead>
<tr>
<th>LNAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
</tbody>
</table>
```

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.

```
<table>
<thead>
<tr>
<th>LNUM</th>
<th>JEN1</th>
<th>JAX1</th>
<th>JEN2</th>
<th>JAX2</th>
<th>.....</th>
<th>JEN6</th>
<th>JAX6</th>
<th>IFLG</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6-11</td>
<td>12</td>
<td>13</td>
<td>14</td>
</tr>
</tbody>
</table>
```

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.

<table>
<thead>
<tr>
<th>END1</th>
<th>END2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>4-6</td>
</tr>
</tbody>
</table>

where

<table>
<thead>
<tr>
<th>END1</th>
<th>END2</th>
</tr>
</thead>
<tbody>
<tr>
<td>are the coordinates of the first end of the link</td>
<td>are the coordinates of the second end of the link</td>
</tr>
</tbody>
</table>

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 DDDDDDDD, this signifies that initial condition data has been applied to the specified joint. If the array is filed with BBBBBBBB or CCCCCCCC, 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.
where

<table>
<thead>
<tr>
<th>INAME</th>
<th>is the initial conditional specification flag</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDDDDD = initial conditions are specified</td>
<td></td>
</tr>
<tr>
<td>BBBB BBBB = initial conditions are not specified</td>
<td></td>
</tr>
<tr>
<td>CCCCCCC = initial conditions are not specified</td>
<td></td>
</tr>
</tbody>
</table>

INU M 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.

<table>
<thead>
<tr>
<th>JNUM</th>
<th>INCR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

where

<table>
<thead>
<tr>
<th>JNUM</th>
<th>is the joint to which the initial conditions are applied</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEP</td>
<td>is the number of increments to be stepped through</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>POSI</th>
<th>INCR</th>
<th>VELO</th>
<th>ACCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

where

<table>
<thead>
<tr>
<th>POSI</th>
<th>is the specified initial position</th>
</tr>
</thead>
<tbody>
<tr>
<td>INCR</td>
<td>is the increment of the position</td>
</tr>
</tbody>
</table>
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 than 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.

<table>
<thead>
<tr>
<th>BNUM</th>
<th>U</th>
<th>U</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>POSN</th>
<th>XAXE</th>
<th>YAXE</th>
<th>ZAXE</th>
<th>U</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>4-6</td>
<td>7-9</td>
<td>10-12</td>
<td>13</td>
<td>14</td>
</tr>
</tbody>
</table>

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 preprocessor, a step-by-step data entry procedure is not required. It is therefore the goal of this chapter to present an overview of 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 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 as 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 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 , -41.5279999
DATA: LINK ( LNK2 , A0 ) = 7.21298027 , -41.0877991
DATA: LINK ( LNK3 , A0 ) = 6.734947982 , -40.5964050
DATA: LINK ( LNK4 , A0 ) = 9.46165943
DATA: LINK ( LNK1 , LNK2 ) = 7.21298027 , 8.06142998
DATA: LINK ( LNK3 , LNK2 ) = 7.21298027 , 8.06142998
DATA: LINK ( LNK4 , LNK3 ) = 6.73476982
DATA: LINK ( LNK5 , LNK4 ) = 6.73476982
DATA: LINK ( LNK6 , LNK5 ) = 5.93500000
DATA: LINK ( LNK7 , LNK6 ) = 5.93500000
DATA: LINK ( LNK8 , LNK7 ) = 5.93500000
DATA: POINT ( P22 , PT22 ) = 0 , 0 , 0
DATA: POINT ( P23 , PT23 ) = 0 , 0 , 0
DATA: POSITION ( A0 ) = 0 , 0 , 0
STORE: POSITION ( A0 )
STORE: VELOCITY ( A0 )
STORE: ACCELER ( A0 )
POINT ( LNK2 ) = PT22
POINT ( LNK3 ) = PT23
DATA: POINT ( PT22 , P1 ) = 0 , 0 , 0
DATA: POINT ( PT23 , P1 ) = 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.


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.
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(INSID,ICHO,CSIZE)

******************************************************************************
* ENTER ANALYSIS OR SYNTHESIS PORTION OF MECHIN. WRITE RESTART
* AND DATA FILES
******************************************************************************
CALL CLOSE(INSID,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(IHSID,CSIZE,AXISR,JNAM,JNUM,JOINT,INAM,INUM,IC,
& SCALE)
******************************************************************************
* ----- SUBROUTINE ADDIC -----
* THIS SUBROUTINE IS USED TO ADD INITIAL CONDITIONS TO A MECHIN
* FILE
******************************************************************************
INTEGER IHSID,JNUM,JFLG,JNAM(JNUM(2,3)),CHOICE,CLASS(2),INUM(JNUM(20,2))
REAL AXISR,LOCAT(3),AREA(6),CSIZE(3),JOINT(JNUM(20,21)),
& ENDP(3),ENDP(3),INK(6),XVEC(3),YVEC(3),XL(6),X(6),
& JOINTY(20,21),BOX(8),SHP(8),LXI(2),LY(2),LZ(2),LXIL(2),LYL(2),
& LZIL(2),LZIL(2,4)

MECHIN PROGRAM LISTING 61
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
LX(1)=0.30
LX(2)=0.68
LY(1)=0.05
LY(2)=0.62
LY(1)=0.30
LY(2)=0.62
LZ(1)=0.05
LZ(2)=0.56
LZ(1)=0.30
LZ(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 (BBBBBBBBB)
DO 21 NUMI=1,20,1
      IF(INAM(NUMI).EQ.'BBBBBBBB'.OR.INAM(NUMI).EQ.'CCCCCCCC')THEN
         GO TO 22
      ELSE CONTINUE
21 CONTINUE
22 CONTINUE
********************************************************************
* OPEN STRUCTURE FOR THE VAL. BOX (VIEW 1)
********************************************************************
C CALL GPOPST(8)      SET SOLID INTERIOR STYLE FOR GREY BOX
CALL GPIS(2)
C
CALL GPICI(4)
C
CALL GPCG(1,4,2,SHD)
C
CALL GPICI(3)
C
CALL GPCG(1,4,2,BX)
C
CALL GPCLST

*********************************************************************
*  SCREEN DISPLAY  
*********************************************************************

C
CALL GPUPWS(INSID,2)

*********************************************************************
*  REQUEST STRING  
*********************************************************************

C
CALL CMAND(INSID,1,1,1,7,INIT,1,AREA,7,1,0,INIT)

*********************************************************************
*  PICK REVOLUTE JOINT OR PRISMATIC JOINT OR RETURN  
*********************************************************************

C
IF INUM(INSID,2).EQ.1 THEN
50
CALL PICKJ(INSID,JOINT,AXISR,CSIZE,JNUM1,ENDP1,IAXIS1,RFLG)

ENDIF

ENDIF

C
IF INUM(INSID,2).EQ.2 THEN

ENDIF

C
IF INUM(INSID,2).EQ.1 THEN

ENDIF

C
ENDIF

END IF

C
ENDIF

C
ENDIF

MECHIN PROGRAM LISTING
CALL GPRGST(IHSID,1,7,ISTAT,NUM,VAL1)
WRITE INPUT TO THE SCREEN
CALL GPOPST(10)
CALL GPTXCI(1)
CALL GPANSC(1,0)
CALL GPAN2(LX,20,X)
CALL GPAN2(LY,17,VAL1)
CALL GPCLST
CALL CMDANIHISID,68)
VAL2=CLEAR
CALL GPRGST(IHSID,1,7,ISTAT,NUM,VAL2)
WRITE INPUT TO THE SCREEN
CALL GPOPST(10)
CALL GPTXCI(1)
CALL GPANSC(1,0)
CALL GPAN2(LY,20,Y)
CALL GPAN2(LY,17,VAL2)
CALL GPCLST
INQUIRE THE NO. OF STEPS
CALL CMDANIHISID,65)
CALL GPINST(IHSID,1,7,INIT2,1,AREA,7,1,0,INIT2)
VAL3=CLEAR
CALL GPRGST(IHSID,1,7,ISTAT,NUM,VAL3)
WRITE INPUT TO THE SCREEN
CALL GPOPST(10)
CALL GPTXCI(1)
CALL GPANSC(1,0)
CALL GPAN2(LZ,20,Z)
CALL GPAN2(LZ,17,VAL3)
CALL GPCLST
GO TO 80
ENDIF

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

ENDIF

IF(CHOICE.EQ.2) THEN
CALL CMDANIHISID,61)
CALL PICKJ2(IHSID,JOINT,AXISR,CSIZE,JNUM1,ENDP1,IAXIS1,RFLG)
ENDIF

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

IF(INAM(JNUM1).EQ.'BBBBBBBB' .OR. INAM(JNUM1).EQ.'CCCCCCCC') THEN
POSITION MUST BE SPECIFIED BEFORE ACC AND VELOCITY
CALL CMDANIHISID,79)
GO TO 150
ENDIF

IF(JNUM(JNUM1,2).EQ.1) THEN
INQUIRE THE VELOCITY
CALL CMDANIHISID,73)
VAL4=CLEAR
CALL GPRGST(IHSID,1,7,ISTAT,NUM,VAL4)
WRITE INPUT TO THE SCREEN
CALL GPOPST(10)
CALL GPTXCI(1)
CALL GPANSC(1,0)
CALL GPAN2(LX,20,V)
CALL GPAN2(LY,17,VAL4)
CALL GPCLST
GO TO 80
ENDIF

IF(JNUM(JNUM1,2).EQ.2) THEN
PRISMATIC JOINT
INQUIRE THE VELOCITY
CALL CMDANIHISID,74)
VAL4=CLEAR
CALL GPRGST(IHSID,1,7,ISTAT,NUM,VAL4)
WRITE INPUT TO THE SCREEN
CALL GPOPST(10)
CALL GPTXCI(1)
CALL GPANSC(1,0)
CALL GPAN2(LX,20,V)
CALL GPAN2(LY,17,VAL4)
CALL GPCLST
ENDIF

IF(JNUM(JNUM1,2).NE.1 .AND. JNUM(JNUM1,2).NE.2) THEN
INVALID JOINT
CALL CMDANIHISID,62)
GO TO 150
ENDIF
END IF
IF(CHOICE.EQ.3) THEN

CALL COMMAND(INSID,61)
CALL PICK(INSID,JOINT,AXISR,CSIZE,JNUM1,ENDP1,IAXIS1,RFLG)

ENDIF
IF(INAMJ.JNUM1.EQ. 'BBBBBBB' OR INAMJ.JNUM1.EQ. 'CCCCCCCC') THEN
POSITION MUST BE SPECIFIED BEFORE ACC AND VELOCITY
CALL COMMAND(INSID,79)
GO TO 150
ENDIF

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

IF(JNUMJ.JNUM1,2).EQ.1) THEN
INQUIRE THE ACCEL
CALL COMMAND(INSID,75)
VAL5=CLEAR CALL GPROST(INSID,1,7,ISTAT,NUM,VAL5)
WRITE INPUT TO THE SCREEN
CALL GPROST(10)
CALL GPTXC(1)
CALL GPANSC(1,0)
CALL GPAN2(LX,20,A)
CALL GPAN2(LY1,7,VAL5)
CALL GPCLS1
GO TO 80
ENDIF

IF(JNUMJ.JNUM1,2).EQ.2) THEN
PRISMATIC JOINT
INQUIRE THE ACCEL
CALL COMMAND(INSID,76)
VAL5=CLEAR CALL GPROST(INSID,1,7,ISTAT,NUM,VAL5)
WRITE INPUT TO THE SCREEN
CALL GPROST(10)
CALL GPTXC(1)
CALL GPANSC(1,0)
CALL GPAN2(LX,20,A)
CALL GPAN2(LY1,7,VAL5)
CALL GPCLS1
ENDIF

IF(JNUMJ.JNUM1,2).NE.1 AND JNUMJ.JNUM1,2).NE.2) THEN
INVALID JOINT
CALL COMMAND(INSID,62)
GO TO 250
ENDIF

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

80 CALL COMMAND(INSID,66)
CALL MENU(INSID,CSIZE,15,CHOICE)
IF(CHOICE.EQ.1) THEN

FILL IC ARRAYS
IF(KK, .EQ. 1) THEN
JNUMJ.JNUM1,1='DDDDDD'
INAMJ.JNUM1,1='DDDDDD'
REWIND 10
WRITE(10,81) CLEAR2
WRITE(10,81) CLEAR2
FORMAT(A20)
REWIND 10
WRITE(10,82) VAL1
WRITE(10,82) VAL2
WRITE(10,82) VAL3
FORMAT(A7)
REWIND 10
READ(10,84) IC(JNUM1,1)
READ(10,84) IC(JNUM1,2)
READ(10,84) IC(JNUM1,3)
END IF

WRITE AN ICON
call ICICON(INSID,JOINT,INAM,INUM,IC,SCALE)

ENDIF
IF(KK, .EQ. 2) THEN
REWIND 10
WRITE(10,83) CLEAR2
FORMAT(A20)
REWIND 10
WRITE(10,84) VAL4
FORMAT(A7)
REWIND 10
READ(10,85) IC(JNUM1,3)
ENDIF

IF(KK, .EQ. 3) THEN
REWIND 10
WRITE(10,85)CLEAR2
85 FORMAT(1X)
REWIND 10
WRITE(10,86)IVAL5
86 FORMAT(1X)
REWIND 10
READ(10,*)IC(JNUM1,4)
ENDIF
ENDIF
100 CALL GPEST(8)
CALL GPEST(10)
RETURN
END

**SUBROUTINE ADDJ**

SUBROUTINE ADDJ(INSID, 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 INSID,NUMJ,JNUM,JNAM,JNUM,JOINT,POINT,SCALE
REAL AXISR,LOCAT(3),ORIENT(3),AREA(6),JOINT(20,21),CSIZE(3), &POINT(100,3),ORIENT(3),LOCAT(3),SCALE
CHARACTER*7 JNDEF,JNAME,JNAM,JNUM,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 CMDAND(INSID,26)
CALL MENUSID,CSIZE,6,CHOICE
C FIND OUT IF RETURN
C IF(CHOICE.EQ.9)THEN
GO TO 1000
ELSE
C TYPE=CHOICE
ENDIF
*********************************************************************
** SET JOINT NUMBER **
*********************************************************************
C DO 10 NUMJ=1,20,1
IF(JNAM(NUMJ).EQ."BBBBBBB".OR.JNAM(NUMJ).EQ."CCccc")THEN
GO TO 20
ELSE
CONTINUE
ENDIF
10 CONTINUE
C 20 IF(INUMJ.EQ.20)THEN
C NO MORE JOINTS AVAILABLE
C CALL CMNAND(INSID,23)
C GET STROKE
C CALL GPINST(INSID,17,CLEAR,1,AREA,7,1,0,CLEAR)
C CALL GPQST(INSID,17,ISTAT,NUM,CLEAR)
GO TO 1000
ENDIF
MECHIN PROGRAM LISTING

66
* GET THE JOINT NAME OR DEFAULT

IF(TYPE.EQ.8)THEN
    JNAME='FRAME'
ENDIF

GET GROUND LINK

REQUEST STRING

CALL COMMAND(ISID,24)

INITIALIZE THE STRING DEVICE

CALL GPSTM(ISID,1,1,2)

CALL GPSTM(ISID,1,1,7,1,JDEF(NUMJ),1,AREA,7,1,0,JDEF(NUMJ))

INQUIRE JOINT NAME

CALL GPSTM(ISID,1,1,7,1,ISTAT,NUM,JNAME)

CALL GPSTM(ISID,1,1,11)

* CHECK TO SEE IF NAME CAN BE USED

DO 30 J=1,20
    IF(JNAME.EQ.JNAM(J))THEN
        NAME ALREADY EXISTS
        CALL COMMAND(ISID,25)
        JNAME=CLEAR
    ENDIF
30 CONTINUE

POSITION THE JOINT

CALL POSITION(ISID,AXISR,CSIZE,LOCAT,RFLG)

IF(RFLG.EQ.1)THEN
    GO TO 1000
ENDIF

* ORIENT THE JOINT

CALL ORIENT(ISID,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
    REVOLUTE
    CALL CCOMMAND(ISID,40)
    CALL CDATA(ISID,CSIZE,2,RFLG)
    JOINT(NUMJ,15)=LOCAT(1)
    JOINT(NUMJ,16)=LOCAT(2)
    JOINT(NUMJ,17)=LOCAT(3)
    ENDIF

IF(TYPE.EQ.2 OR TYPE.EQ.3 OR TYPE.EQ.5)THEN
    SCREW
    CALL CCOMMAND(ISID,40)
    CALL CDATA(ISID,CSIZE,2,RFLG)
    JOINT(NUMJ,15)=LOCAT(1)
    JOINT(NUMJ,16)=LOCAT(2)
    JOINT(NUMJ,17)=LOCAT(3)
    ENDIF

SPECIAL CASE

IF(TYPE.EQ.4)THEN
    SPHERICAL
    CALL CCOMMAND(ISID,42)
    CALL MENU(ISID,CSIZE,9,CHOICE)
    IF(CHOICE.EQ.1)THEN
        ENDIF
    ENDIF

ORIENT THE JOINT

CALL ORIENT(ISID,LOCAT,POINT,JOINT,AXISR,CSIZE,ORIENT2,RFLG)
IF(RFLG.EQ.1) THEN
    GO TO 1000
ENDIF

IF(ITYPE.EQ.5) THEN
    SCREW JOINT(INQUIRE SCREW LEAD)
    CALL CMAND(IHSID,41)
    CALL CDATA(IHSID,CSIZE,Z,RFLG)
    JOINT(NUMJ,15)=Z
ENDIF

IF(ITYPE.EQ.6) THEN
    FLAT JOINT(INQUIRE POSITION OF THE SECOND LOCAL SYS.
    CALL CMAND(IHSID,43)
    CALL MENU(IHSID,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
ENDIF

IF(ITYPE.EQ.7) THEN
    GEAR PAIR(INQUIRE POSITION OF THE SECOND LOCAL SYS.
    CALL CMAND(IHSID,44)
    CALL MENU(IHSID,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
ENDIF

500 IF(ITYPE.EQ.8) THEN
    GROUND LINK (INQUIRE POSITION)
    CALL POSIT(IHSID,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)
SUBROUTINE ADDL

SUBROUTINE ADDL(IHSID, CSIZE, AXISR, JNAM, JNUM, JOINT, PNAM, PNUM, POINT, 
& LNAM, LNUM, LINK, SCALE)

COMMON/LDEF/LNDEF(30)
INTEGER IHSID, NULM, 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), ENDP(3), ENDP2(3), LNKL6, XVEC(3), YVEC(3), X(6), 
& JOINT(20,21), SCI(3), SCH(4,4), TR(3), TRM(4,4), TRM(14,4), P(3), 
& PI(3), PI2(10,4), T(4,4)
CHARACTER*7 LNDEF, LNAM, PNAM(100), JNAM(20), LNAM(30), CLEAR, CLEAR1
CHARACTER*20 CLEAR2

DATA CLEAR /11/ DATA CLEAR1 /' '/ DATA CLEAR2 /' '/ DATA SC/1.,0.,0.,0.,1.,0.,0.,0.,0.,0.,0./ DATA SCH/1,0.,0.,0.,1.,0.,0.,0.,0.,0.,0./ DATA TR/1,0.,0.,0.,1.,0.,0.,0.,0.,0.,0./ DATA TRM/1,0.,0.,0.,1.,0.,0.,0.,0.,0.,0./ DATA T /1.,0.,0.,0.,1.,0.,0.,0.,0.,0.,0./ DATA CLASS /5/
RFLG=0
LOCAT1=0.0 LOCAT2=0.0 LOCAT3=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)
LNAM=CLEAR
SCI1=SCALE
SCI3=SCALE

CALL GPSC3(SC, SCH)

SET LINK NUMBER
SEARCH LNAM ARRAY FOR NEXT AVAILABLE SLOT (BBBBBBBB)
DO 10 NUML=1,30,1
IF(LNAM(NUML).EQ.'BBBBBBBB' .OR. LNAM(NUML).EQ.'CCCCCCCC') THEN
GO TO 20
ELSE CONTINUE
ENDIF
10 CONTINUE
20 IF(NUML.EQ.30) THEN
CALL COMMAND(IHSID,46)
GET STROKE
CALL GPINST(IHSID,1,7,CLEAR,1,AREA,7,1,0,CLEAR)
CALL GPRQSTI(INSID,1,7,ISTAT,NUM,CLEAR)
RETURN
ENDIF
******************************************************************************
* GET THE LINK NAME OR DEFAULT
******************************************************************************
CALL CMMANDI(INSID,47)
CALL GPSTMOI(INSID,1,1,2)
C 25 CALL GPINSTI(INSID,1,7,NDIF(INUM),1,AREA,7,1,0,NDIF(NUML))
C CALL GPRQSTI(INSID,1,7,ISTAT,NUM,LNAME)
CALL GPSTMOI(INSID,1,1,2)
******************************************************************************
* IF AN EXISTING LINK NAME, COUPLER JOINTS ON THIS LINK
******************************************************************************
DO 26 I=1,30 IF(LNAME.EQ.LNAME(1)) THEN 
  NUM=1
  CALL COUPLE(INSID,LNAME,LNUM,LINK,JNUM,JNUM,JOINT,NUML,
  CSIZE)
  CALL DRAHLK(INSID,LNAME,LNUM,LINK,JOINT,JNUM,AXISR,NUML,
  SCALE)
  GO TO 100
ENDIF
26 CONTINUE
******************************************************************************
* PICK FIRST END OF THE LINK
******************************************************************************
CALL CMMANDI(INSID,49)
CALL PICKJ(INSID,JOINT,AXISR,CSIZE,JNUM1,ENDP1,IAXIS1,RFLG)
IF(JNUM.JNUM1(1).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)
JOINT(JNUM1,7)=1.0
JOINT(JNUM1,8)=0.0
JOINT(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 GPOPSTI(19)
CALL GPADCNI1,CLASS)
CALL GPPKIN(JNUM1)
CALL GPPLC(6)
CALL GPMHXC(SCHM1)
CALL GPLY(1)
CALL GPPD(2,3,X1)
CALL GPCLST
CALL GPUPHSI(INSID,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 CMMANDI(INSID,51)
CALL MENU INSID,CSIZE,12,CHOICE)
IF(CHOICE.EQ.1)THEN
CALL CMMANDI(INSID,53)
CALL TEJOXI(INSID,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(JNUM1,7)=XVEC(1)
JOINT(JNUM1,11)=XVEC(2)
MECHIN PROGRAM LISTING

70
JOINT(JNUM1,12)=XVEC(3)
ENDIF
IF(RFLG.EQ.1)THEN
GO TO 100
ENDIF
ENDIF
IF(CHOICE.EQ.2)THEN
CALL COMMAND(INSID,50)
CALL TEPOX(INSID,JNUM1,IAxis1,POINT,AXISR,CSIZE,XVEC,YVEC,
&JOINT,RFLG)
ENDIF
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.3)THEN
GO TO 100
ENDIF
ENDIF
********************************************************************
* DRAH 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)
X(1)=0.
X(2)=0.
X(3)=0.
X(4)=XVEC(1)*AXISR/10.+2.4
X(5)=XVEC(2)*AXISR/10.+2.4
X(6)=XVEC(3)*AXISR/10.+2.4
CALL GPPOST(19)
CALL GAPCHN(1,CLASS)
CALL GAPKID(NUML)
CALL GPPLOC(16)
CALL GPMX3(TRM1,3)
CALL GPMX3(SCH,1)
CALL GPMT(11)
CALL GPPL3(2,3,X1)
CALL GPCLS
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)
X(1)=X(1)+XVEC(1)*AXISR/10.
X(2)=X(2)+XVEC(2)*AXISR/10.
X(3)=X(3)+XVEC(3)*AXISR/10.
CALL GPPOST(19)
CALL GAPCHN(1,CLASS)
CALL GAPKID(NUML)
CALL GPPLOC(16)
CALL GPMX3(TRM1,3)
CALL GPMX3(SCH,1)
CALL GPMT(11)
CALL GPPL3(2,3,X1)
CALL GPCLS
ENDIF
END IF
CALL GPUPHS(INSID,2)
********************************************************************
* PICK SECOND END OF THE LINK
********************************************************************
50 CALL COMMAND(INSID,50)
CALL PICKJ(INSID,JOINT,AXISR,CSIZE,JNUM2,ENDP2,IAxis2,RFLG)
IF(RFLG.EQ.1)THEN
LNAM(JNUM1)='CCCCCCCC'
CALL RURLK(INSID,JNAM,JNUM,JOINT,AXISR,LNAM,LNUM,LINK,SCALE)
ENDIF
GO TO 100
ENDIF

CIF(JNUM1,JNUM2,2).EQ.8)THEN
  GROUND LINK
  TR(1)=JOINT(JNUM1,1)
  TR(2)=JOINT(JNUM1,2)
  TR(3)=JOINT(JNUM1,3)
  CALL GPTR3(TR,TRM)
  JOINT(JNUM2,7)=1.0
  JOINT(JNUM2,6)=0.0
  JOINT(JNUM2,9)=0.0
  XVEC(1)=1.0
  XVEC(2)=0.0
  XVEC(3)=0.0
  X2(1)=0.
  X2(5)=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 GPWSTR(19)
  CALL GPADCN1(CLASS)
  CALL GPPK1D(1)
  CALL GPMLX3(TRM,3)
  CALL GPMLX3(SCM,11)
  CALL GPF(11)
  CALL GPPL312,3,X2)
  CALL GPCL3
  CALL GPUPWS(INSID,2)
GO TO 60
ENDIF

IF(RFLG.EQ.1)THEN
  LNAM(INUM)='CCCCCCC'
  CALL RDRNLK(INSID,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 CMAND(INSID,51)
CALL MENU(INSID,CSIZE,12,CHOICE)
IF(CHOICE.EQ.1)THEN
  CALL CMAND(INSID,53)
  CALL TEPOX(INSID,JNUM2,AXIS2,JOINT,AXISR,CSIZE,XVEC,YVEC,RFLG)
  IF(AXIS2.EQ.1)THEN
    JOINT(JNUM2,8)=XVEC(1)
    JOINT(JNUM2,9)=XVEC(3)
  ENDIF
  IF(AXIS2.EQ.2)THEN
    JOINT(JNUM2,10)=XVEC(1)
    JOINT(JNUM2,11)=XVEC(2)
    JOINT(JNUM2,12)=XVEC(3)
  ENDIF
ENDIF
IF(RFLG.EQ.1)THEN
  LNAM(INUM)='CCCCCCC'
  CALL RDRNLK(INSID,JNAM,JNUM,JOINT,AXISR,LNAM,LNUM,LINK,SCALE)
  GO TO 100
ENDIF

ENDIF
IF(CHOICE.EQ.2)THEN
  CALL CMAND(INSID,52)
  CALL TEPOX(INSID,JNUM2,AXIS2,POINT,AXISR,CSIZE,XVEC,YVEC,
 &JOINT,RFLG)
  IF(AXIS2.EQ.1)THEN
    JOINT(JNUM2,8)=XVEC(1)
    JOINT(JNUM2,9)=XVEC(3)
  ENDIF
  IF(AXIS2.EQ.2)THEN
    JOINT(JNUM2,10)=XVEC(1)
    JOINT(JNUM2,11)=XVEC(2)
    JOINT(JNUM2,12)=XVEC(3)
  ENDIF
ENDIF
IF(RFLG.EQ.1)THEN
  LNAM(INUM)='CCCCCCC'
  CALL RDRNLK(INSID,JNAM,JNUM,JOINT,AXISR,LNAM,LNUM,LINK,SCALE)
  GO TO 100
ENDIF

ENDIF

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

MECHIN PROGRAM LISTING 72
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 GPPOPST19
CALL GPADCN1,CLASS)
CALL GPPKID(JNUM2, J)
CALL GPMLX3(TRM, 3)
CALL GPMLX3(SCM, 1)
CALL GPPLCI6)
CALL GPML32, 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, 4))
X2(3)=.01*AXISR+(JOINT(JNUM2, 14)*JOINT(JNUM2, 4))
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 GPPOPST19
CALL GPADCN1, CLASS)
CALL GPPKID(JNUM2, J)
CALL GPMLX3(TRM, 3)
CALL GPMLX3(SCM, 1)
CALL GPPLCI6)
CALL GPML32, 3, X2)
CALL GPCLST
ENDIF
CALL GPUPHS(INSIO, 2)

********************************************************************
THE LINK
********************************************************************
60 P1(X1, 4)*SCALE
P2(X1, 5)*SCALE
P3(X1, 6)*SCALE
CALL GPXF3(P, TRM, P1)
P1(X1, 4)*SCALE
P2(X1, 5)*SCALE
P3(X1, 6)*SCALE
CALL GPXF3(P, TRM, P2)
LNK1=P1(1)
LNK2=P1(2)
LNK3=P1(3)
LNK4=P2(1)
LNK5=P2(2)
LNK6=P2(3)

IF(JNUM(JNUM1, 2).EQ.8)THEN
P1=0.
P2=0.
P3=0.
CALL GPXF3(P, TRM, P1)
LNK1=P1(1)
LNK2=P1(2)
LNK3=P1(3)
ENDIF

MECHIN PROGRAM LISTING 73
IF(JNUM1(JNUM2,2).EQ.8)THEN
  P1(1)=0.
P1(2)=0.
P1(3)=0.
  CALL GPXF3(P,TRM,P2)
  LNK(4)=P2(1)
  LNK(5)=P2(2)
  LNK(6)=P2(3)
ENDIF

CALL GPP1MO(19)
CALL GPP3I(1,CLASS)
CALL GPMX3I(1,2)
CALL GPPLC(4)
CALL GPLT(2)
CALL GPPL3I(2,3,LNK)
CALL GPCLST

CALL GPUPMS(IHSID,2)

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

LNAM1(JNUM1)=LNAME
LNAM1(JNUM1,1)=JNAME
LNAM1(JNUM1,2)=JNUMI
LNAM1(JNUM1,3)=IAXIS1
LNAM1(JNUM1,4)=JNUM2
LNAM1(JNUM1,5)=IAXIS2
LNAM1(JNUM1,14)=0
LNK1(JNUM1,1)=LNK(1)
LNK1(JNUM1,2)=LNK(2)
LNK1(JNUM1,3)=LNK(3)
LNK1(JNUM1,4)=LNK(4)
LNK1(JNUM1,5)=LNK(5)
LNK1(JNUM1,6)=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(IHSID,CSIZE,AXISR,PNAME,PNUM,POINT)

*********************************************************************
* ----- SUBROUTINE ADDP ----- *
*********************************************************************

COMMON/PDEF/PNDEF(100)
INTEGER IHSID,NUMP,PNUM(100),RFLG
REAL AXISR,LOCAT(3),AREA(6),POINT(100,3),CSIZE(3)
CHARACTER*7 PNDEF,PNAME,PNUM(100),CLEAR,CLEAR1
CHARACTER*20 CLEAR2

MECHIN PROGRAM LISTING 74
DATA CLEAR
DATA CLEAR1
DATA CLEAR2
RFLG=0
LOCAT1(1)=0.0
LOCAT1(2)=0.0
LOCAT1(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)

PNAMES=CLEAR

********************************************************************
* SET POINT NUMBER
********************************************************************
C DO 10 NUMP=1,100,1
IF(PNAMES(NUMP).EQ.'BBBBBBB'.OR.PNAMES(NUMP).EQ.'CCCCCCC')THEN
ELSE CONTINUE
ENDIF
10 CONTINUE
C 20 IF(NUMP.EQ.100)THEN
C NO MORE POINTS AVAILABLE
CALL CMMAND(HSID,16)
C GET STROKE
CALL GPINST(HSID,1,7,CLEAR,1,AREA,7,1,0,CLEAR)
CALL GPRQST(HSID,1,7,STAT,NUM,CLEAR,1)
RETURN
ENDIF

********************************************************************
* GET THE POINT NAME OR DEFAULT
********************************************************************
C REQUEST STRING
CALL CMMAND(HSID,13)
C INITIALIZE THE STRING DEVICE
CALL GPMSTO(HSID,1,1,2)
25 CALL GPMST(HSID,1,7,PNAMES(NUMP),1,AREA,7,1,0,PNAMES(NUMP))
C INQUIRE POINT NAME
CALL GPMST(HSID,1,7,ISTAT,NUM,PNAME)
CALL GPMSTO(HSID,1,1,2)

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

********************************************************************
* POSITION THE POINT
********************************************************************
CALL POSIT(HSID,AXISR,CSIZE,LOCAT,RFLG)
IF(RFLG.EQ.1)THEN
GO TO 100
ENDIF

********************************************************************
* DRAW THE POINT
********************************************************************
CALL DRAPTI(HSID,LOCAT,AXISR,PNAME,NUMP)

********************************************************************
* ADD POINT DATA TO POINT ARRAYS
********************************************************************
PNAMES(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 ----- **
******************************************************************************

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 IHSID,CHOICE,JNUM(20,3),LNUM(30,4),PNUM(100),INUM(20,2)
REAL CSIZE(3),AXISR,POINT(100,3),JOINT,LINK(50,6),SCALE,IC(20,4)
CHARACTER*77 PNAM,JNAM,LNAM,FILE,INAM

******************************************************************************
** CALL BACKGROUND SCREEN **
******************************************************************************

CALL SCRNS(IHSID)

******************************************************************************
** FIND OUT IF A NEW OR OLD MODEL IS TO BE PROCESSED AND SET UP FOR DATA **
******************************************************************************

CALL CMDAND(IHSID,1)
CALL MENU(IHSID,CSIZE,1,CHOICE)
IF(CHOICE.EQ.1) THEN
    CALL NSTART(IHSID,FILE,CSIZE,AXISR,POINT,JOINT,LINK,IC,
PNUM,JNUM,LNUM,INUM)
    SCALE=0.0
    CALL SCALNM(IHSID,SCALE)
ELSE
    CALL RSTART(IHSID,FILE,CSIZE,AXISR,PNUM,PNUM,POINT,JNAM,JNUM,
& JOINT,INAM,LNUM,LINK,INUM,IC,SCALE)
ENDIF

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

CALL DATA(IHSID,AXISR,CSIZE,PNUM,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,IHSID,CSIZE,INAM,INUM,IC,SCALE)

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

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

RETURN
END

SUBROUTINE ANOTH

******************************************************************************
** ----- SUBROUTINE ANOTH ----- **
******************************************************************************

THIS SUBROUTINE ASKS IF ANOTHER MECHANISM IS TO BE INPUT ELSE
QUIT

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

MECHIN PROGRAM LISTING
REAL CSIZE(3)
INTEGER IHSID,CHOICE,CHO

CALL CMDAND(IHSID,85)
CALL MENU(IHSID,CSIZE,17,CHO)

IF(CHO.EQ.1) THEN
  CALL GPDAST
  CALL GPDAVI(IHSID)
  CALL GPDAWIN(IHSID)
  CALL GPVCH(IHSID,0,0,2,2,1,1,1,1,1,1,1,1,1,1,1,1,1)
  CALL GPVCH(IHSID,1,1,2,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1)
  CALL GPVCH(IHSID,2,2,2,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1)
  CALL GPUPSI(IHSID,2)
  CALL SASSO(IHSID,CSIZE)
  CALL SCRT2(IHSID,CSIZE,CHOICE)
ENDIF

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

RETURN
END

SUBROUTINE AXES

SUBROUTINE AXES(IHSID,AXISR)
*********************************************************************
* ----- SUBROUTINE AXES ----- *
** *
* THIS SUBROUTINE IS USED TO DISPLAY THE COORDINATE AXES *
*********************************************************************

INTEGER INSID,PARELL,PRFORM
REAL AXISR,XAXIS1(24),YAXIS1(24),ZAXIS1(24),WIND0H(4),VP1(6),VP2(6),
& BOX1(8),BOXX(8),NEAR,FAR,POINT(3),DIST,UP(3),NORMAL(3),
& MATRIX(4,4),POINTP(3),PX(3),PY(3),PZ(3),WIND01(4),NEAR1,FAR1
CHARACTER*1 X,Y,Z

C DATA PRFORM/2/ FLAG FOR UPDATE
C DATA BOX1/-0.48,-0.98,0.98,-0.98,0.98,0.48,-0.48,0.48/ GREY MAIN SCREEN BOX
C DATA BOX2/-0.48,0.52,-0.02,0.52,-0.02,0.98,-0.48,0.98/ GREY POSITION BOX
C DATA DIST/0./
C DATA PARELL/1/
C DATA POINT*/0.0,1.5,5.0/
C DATA NORMAL/1.,0.0,0./
C DATA UP/0.,0.,0./
C DATA X'X'/
C DATA Y'Y'/
C DATA Z'Z'/

XAXIS1(1)=-AXISR
XAXIS1(2)=0.0
XAXIS1(3)=0.0
XAXIS1(4)=AXISR+.08*AXISR
XAXIS1(5)=-.02*AXISR
XAXIS1(6)=0.0
XAXIS1(7)=AXISR+.08*AXISR
XAXIS1(8)=.02*AXISR
XAXIS1(9)=0.0
XAXIS1(10)=AXISR
XAXIS1(11)=0.0
XAXIS1(12)=0.0
XAXIS1(13)=AXISR
XAXIS1(14)=0.0
XAXIS1(15)=0.0
XAXIS1(16)=AXISR-.08*AXISR
XAXIS1(17)=-.02*AXISR
XAXIS1(18)=0.0
XAXIS1(19)=AXISR-.08*AXISR
XAXIS1(20)=.02*AXISR
XAXIS1(21)=0.0
XAXIS1(22)=AXISR
XAXIS1(23)=0.0
XAXIS1(24)=0.0

MECHIN PROGRAM LISTING 77
YAXIS1 = 0.0
YAXIS2 = AXISR
YAXIS3 = 0.0
YAXIS4 = 0.02*AXISR
YAXIS5 = AXISR + (0.08*AXISR)
YAXIS6 = 0.0
YAXIS7 = 0.02*AXISR
YAXIS8 = AXISR + (0.08*AXISR)
YAXIS9 = 0.0
YAXIS10 = 0.0
YAXIS11 = AXISR
YAXIS12 = 0.0
YAXIS13 = 0.0
YAXIS14 = AXISR
YAXIS15 = 0.0
YAXIS16 = -0.02*AXISR
YAXIS17 = AXISR - (0.08*AXISR)
YAXIS18 = 0.0
YAXIS19 = 0.02*AXISR
YAXIS20 = AXISR - (0.08*AXISR)
YAXIS21 = 0.0
YAXIS22 = 0.0
YAXIS23 = AXISR
YAXIS24 = 0.0

ZAXIS1 = 0.0
ZAXIS2 = 0.0
ZAXIS3 = AXISR
ZAXIS4 = -0.02*AXISR
ZAXIS5 = 0.0
ZAXIS6 = AXISR + (0.08*AXISR)
ZAXIS7 = 0.02*AXISR
ZAXIS8 = 0.0
ZAXIS9 = AXISR + (0.08*AXISR)
ZAXIS10 = 0.0
ZAXIS11 = 0.0
ZAXIS12 = AXISR
ZAXIS13 = 0.0
ZAXIS14 = 0.0
ZAXIS15 = AXISR
ZAXIS16 = -0.02*AXISR
ZAXIS17 = 0.0
ZAXIS18 = AXISR - (0.08*AXISR)
ZAXIS19 = 0.02*AXISR
ZAXIS20 = 0.0
ZAXIS21 = AXISR - (0.08*AXISR)
ZAXIS22 = 0.0
ZAXIS23 = 0.0
ZAXIS24 = AXISR

PX1 = AXISR + (0.06*AXISR)
PX2 = -0.01*AXISR
PX3 = 0.0

PY1 = -0.01*AXISR
PY2 = AXISR + (0.04*AXISR)
PY3 = 0.0

PZ1 = -0.02*AXISR
PZ2 = 0.0
PZ3 = AXISR + (0.07*AXISR)

VP11 = BOX111 + 1.1/2.
VP12 = BOX113 + 1.1/2.
VP13 = BOX114 + 1.1/2.
VP14 = BOX116 + 1.1/2.
VP15 = BOX111 + 1.1/2.
VP16 = BOX113 + 1.1/2.

VP21 = BOX411 + 1.1/2.
VP22 = BOX413 + 1.1/2.
VP23 = BOX414 + 1.1/2.
VP24 = BOX416 + 1.1/2.
VP25 = BOX411 + 1.1/2.
VP26 = BOX413 + 1.1/2.

NEAR = AXISR + (0.64*AXISR)
FAR = AXISR - (0.64*AXISR)

NEAR = 1.64
FAR = 1.64

WINDOW1 = -AXISR - (0.64*AXISR)
WINDOW2 = AXISR + (0.64*AXISR)
WINDOW3 = -AXISR - (0.64*AXISR)
WINDOW4 = AXISR + (0.64*AXISR)

WINDOW1 = 1.64
WINDOW2 = 1.64
WINDOW3 = 1.64
WINDOW4 = 1.64

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

```plaintext
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*32 FILE1,FILE2,CLEAR

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

  C                                 

  MECHIN PROGRAM LISTING

```
**SUBROUTINE CDATA**

SUBROUTINE CDATA(IWSID,CSIZE,Z,RFLG)

*********************************************************************
* ----- SUBROUTINE CDATA ------ *
*********************************************************************

* THIS SUBROUTINE WILL PROMPT FOR DATA ON THE COMMAND BOX 
*********************************************************************

INTEGER IWSID
CHARACTER*7 CLEAR,INIT
CHARACTER*2 STRING,CLEAR2
DATA CLEAR//" DATA INIT /"0. 
DATA STRING//" DATA CLEAR2/
DATA 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)
SUBROUTINE CHSCAL

SUBROUTINE CHSCAL(IHSID,AXISR,CSIZE,SCALE,JNAM,JNUM,JOINT, & LNAM,LNUM,LINK)

* THIS SUBROUTINE WILL PROMPT CHANGE OF JOINT ICON SCALE *

DATA CLEAR/'I I'/ DATA INIT/'O.'/ DATA INITI/'I.'/ DATA STRING/'I /
DATA CLEARI'.'/ DATA CLEARZ'.'/
AREAl=0.08*CSIZE(1) AREAl2=0.90*CSIZE(1) AREAl3=0.70*CSIZE(1) AREAl4=0.90*CSIZE(1) AREAl5=0.02*CSIZE(1) AREAl6=0.90*CSIZE(1) LOCATI=0.0 LOCATI2=0.0 LOCATI3=0.0 LYI(1)=0.05 LYI(2)=0.62 LYI(11)=0.15 LYI(12)=0.62

CALL GPEST(6)

* GET SCALE FACTOR *

CALL CMANDI(IHSID,69)

1 CALL GPSTM(IHSID,1,1,2)
CALL GPINST(IHSID,1,12,INIT,1,AREA,12,1,0,CLEAR)
STRING=CLEAR2
CALL GPGRST(IHSID,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

MECHIN PROGRAM LISTING
SUBROUTINE CHVIEW

SUBROUTINE CHVIEW(IHSID, AXISR, CSIZE, SCALE, JNAM, JNUM, JOINT, LNAM, LNUM, LINK)

INTEGER IHSID, CHOICE, JNUM(20,3), LNUM(30,14)
REAL CSIZE(3), AXISR, JOINT(20,21), LINK(30,6)
CHARACTER*7 JNAM(20), LNAM(30)

C IF(SCALE.LE.0.)THEN VALUE IS GREATER THAN 1
C CALL CMAND(IHSID,70)
C GO TO 1
ENDIF
CALL SCALNM(IHSID,SCALE)
CALL RDRHJTIHSID,JNAM,JNUM,JOINT,AXISR,SCALE)
CALL RDRHJIIHSID,JNAM,JNUM,JOINT,AXISR,LNAM,LNUM,LINK,SCALE)
1000 CALL GPSTMO(IHSID,1,1,2)
RETURN END

SUBROUTINE CLOSE

SUBROUTINE CLOSE(IHSID, CSIZE)

INTEGER IHSID, CSIZE

ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF

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

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

INTEGER CHOICE,INSID 
REAL CSIZE(3)  

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

* SEE IF ANOTHER MECHANISM IS TO BE INPUT ELSE CLOSE  

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

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

* CLOSE PHIGS AND WORKSTATIONS  
CALL GPDAST  
CALL GPCLPH  
RETURN  
END  

** SUBROUTINE COMMAND **  

SUBROUTINE COMMAND(INSID,NUM)  
**************************************************************************  

* ----- SUBROUTINE COMMAND -----  

*** THIS SUBROUTINE IS USED TO DISPLAY COMMANDS IN THE COMMAND BOX ***  
**************************************************************************  

INTEGER INSID,LENGT,PRFORM,NUM  
REAL POSL1(2),POSL2(2),POSL3(2),POSL4(2),POSL5(2),HINDO(4),VPC(4)  
CHARACTER*19 Cl9,5)  

C COMMANDS  
DATA Cl1,1)/' /  
DATA Cl1,2)/' PICK EITHER NEW /  
DATA Cl1,3)/' MODEL OR RESTART /  
DATA Cl1,4)/' OF A PREVIOUS /  
DATA Cl1,5)/' MODEL /  
DATA Cl2,1)/' ENTER THE NAME /  
DATA Cl2,2)/' OF THE MECHIN /  
DATA Cl2,3)/' RESTART FILE /  
DATA Cl2,4)/' (UPPER CASE) /  
DATA Cl2,5)/' /  
DATA Cl3,1)/' /  
DATA Cl3,2)/' ENTERING THE NAME /  
DATA Cl3,3)/' RESTART FILE /  
DATA Cl3,4)/' (UPPER CASE) /  
DATA Cl3,5)/' /  
DATA Cl4,1)/' ENTER A NAME /  
DATA Cl4,2)/' FOR THIS MECHIN /  
DATA Cl4,3)/' INPUT FILE /  
DATA Cl4,4)/' (UPPER CASE) /  
DATA Cl4,5)/' /  
DATA Cl5,1)/' ENTER THE RANGE /  
DATA Cl5,2)/' OF THE COORDINATE /  
DATA Cl5,3)/' AXES FOR THIS MODEL /  
DATA Cl5,4)/' /  
DATA Cl6,1)/' FILE NOT A VALID /  
DATA Cl6,2)/' RESTART FILE /  

MECHIN PROGRAM LISTING
DATA C(6,3) / ENTER VALID FILE / (UPPER CASE) \
DATA C(6,4) / \
DATA C(7,1) / PLEASE PICK \
DATA C(7,2) / TYPE OF \
DATA C(7,3) / DATA INPUT \
DATA C(8,1) / FILE ALREADY EXISTS \
DATA C(8,2) / RE-ENTER FILE NAME \
DATA C(8,3) / TO USE IT OR ENTER \
DATA C(8,4) / A NEW FILE NAME \
DATA C(9,1) / FILE DOES NOT EXIST \
DATA C(9,2) / ENTER VALID FILE \
DATA C(9,3) / (UPPER CASE) \
DATA C(10,1) / SELECT \
DATA C(10,2) / DESIRED OPTION \
DATA C(10,3) / \
DATA C(10,4) / \
DATA C(10,5) / \
DATA C(11,1) / INPUT LOCATION DIRECTLY \
DATA C(11,2) / (ENTER X LOCATION) OR \
DATA C(11,3) / USE VALUATORS \
DATA C(12,1) / END OF PROGRAM FOR NOW \
DATA C(12,2) / PRESS ANY BUTTON TO QUIT \
DATA C(13,1) / ENTER THE NAME OF THE POINT OR PRESS ENTER TO USE DEFAULT NAME \
DATA C(13,2) / \
DATA C(13,3) / \
DATA C(13,4) / \
DATA C(13,5) / \
DATA C(14,1) / POINT NAME ALREADY EXISTS \
DATA C(14,2) / ENTER NEW NAME OR CHOOSE DEFAULT \
DATA C(15,1) / POINT NAME CAN NOT BE USED \
DATA C(15,2) / ENTER NEW NAME OR CHOOSE DEFAULT \
DATA C(16,1) / NUMBER OF POINTS EXCEEDS THE MAXIMUM ALLOWABLE \
DATA C(16,2) / PRESS ENTER \
DATA C(17,1) / POINT IS OUTSIDE THE PREDEFINED AXIS RANGE. PICK RE-ENTER AXIS OR RE-ENTER POINT \
DATA C(18,1) / INPUT V LOCATION \
DATA C(18,2) / \
DATA C(18,3) / \
DATA C(18,4) / \
DATA C(18,5) / \
DATA C(19,1) / INPUT Z LOCATION \
DATA C(19,2) / \
DATA C(19,3) / \
DATA C(19,4) / \
DATA C(19,5) / \
DATA C(20,1) / ENTER POINT OR RETURN TO RE-ENTER POINT DATA \
DATA C(20,2) / \
DATA C(20,3) / \
DATA C(20,4) / \
DATA C(20,5) / \
DATA C(21,1) / PICK POINT TO DELETE \
DATA C(21,2) / \
DATA C(21,3) / \
DATA C(21,4) / \
DATA C(21,5) / \
DATA C(22,1) / PICK POINT OR JOINT \
DATA C(22,2) / \
DATA C(22,3) / \
DATA C(22,4) / \
DATA C(22,5) / \

MECHIN PROGRAM LISTING
DATA C(22,3) FOR POSITION INQUIRY
DATA C(22,4) FOR POSITION
DATA C(23,2) NUMBER OF JOINTS EXCEEDS THE MAXIMUM ALLOWABLE
DATA C(23,4) PRESS ENTER
DATA C(24,1) ENTER THE NAME OF THE JOINT OR USE DEFAULT
DATA C(24,3) ENTER NEW NAME OR CHOOSE DEFAULT
DATA C(26,1) CHOOSE JOINT TYPE
DATA C(27,1) INPUT X DIRECTION COSINE
DATA C(27,4) DIRECTION COSINE
DATA C(28,1) INPUT Y DIRECTION COSINE
DATA C(28,4) DIRECTION COSINE
DATA C(29,1) INPUT Z DIRECTION COSINE
DATA C(29,4) DIRECTION COSINE
DATA C(30,1) PICK METHOD OF ORIENTATION OF JOINT Z AXIS
DATA C(30,5) PLEASE RE-ENTER
DATA C(32,1) ENTER ORIENTATION OR RETURN TO REENTER DATA
DATA C(33,1) ENTER ORIENTATION WITH VALUATORS OR RETURN TO REENTER DATA
DATA C(34,1) PICK A POINT OR JOINT TOWARD WHICH THIS JOINT IS ORIENTED
DATA C(35,1) PICK A JOINT TO WHICH THIS JOINT IS PARALLEL
DATA C(36,1) PICK JOINT TO DELETE
DATA C(37,1) PICK JOINT FOR INQUIRY OF JOINT Z AXIS ORIENTATION
DATA C(38,1) ZERO MAGNITUDE VECTOR
DATA C(38,3)/ ENTERED
DATA C(38,4)/ RE-ENTER X
DATA C(39,1)/ ZERO MAGNITUDE VECTOR
DATA C(39,2)/ ENTERED
DATA C(39,3)/ RE-ENTER X
DATA C(39,4)/ ZERO MAGNITUDE VECTOR
DATA C(39,5)/ ENTERED
DATA C(40,1)/ ENTER AN INITIAL
DATA C(40,2)/ DISPLACEMENT FOR
DATA C(40,3)/ THIS JOINT
DATA C(40,4)/ RE-ENTER VALUES
DATA C(40,5)/ ENTERED
DATA C(41,1)/ ENTER THE LEAD FOR THIS JOINT
DATA C(41,2)/ SCREW JOINT
DATA C(41,3)/ SECOND Z ORIENT
DATA C(41,4)/ SECOND CENTER OF LOCAL COORDINATE SYSTEM NEEDED FOR A FLAT JOINT
DATA C(41,5)/ SECOND CENTER OF LOCAL COORDINATE SYSTEM NEEDED FOR A GEAR PAIR
DATA C(42,1)/ SECOND Z ORIENT
DATA C(42,2)/ VECTOR IS NEEDED FOR A SPHERICAL JOINT
DATA C(42,3)/ ENTERED
DATA C(42,4)/ ENTERED
DATA C(42,5)/ ENTERED
DATA C(43,1)/ NUMBER OF LINKS EXCEEDS THE MAXIMUM ALLOWABLE
DATA C(43,2)/ PRESS ENTER
DATA C(43,3)/ ENTER THE NAME OF THE LINK OR PRESS ENTER TO USE DEFAULT
DATA C(43,4)/_LINK NAME
DATA C(43,5)/ ALREADY EXISTS
DATA C(44,1)/ ENTERED
DATA C(44,2)/ THE GEAR RATIO FOR THIS GEAR PAIR
DATA C(44,3)/ ENTERED
DATA C(44,4)/ ENTERED
DATA C(44,5)/ ENTERED
DATA C(45,1)/ PICK THE Z AXIS ON THE JOINT AT THE FIRST END OF THE LINK OR RETURN
DATA C(45,2)/ PICK THE Z AXIS ON THE JOINT AT THE OTHER END OF THE LINK OR RETURN
DATA C(45,3)/ PICK THE Z AXIS ON THE JOINT AT THE FIRST END OF THE LINK OR RETURN
DATA C(45,4)/ PICK THE Z AXIS ON THE JOINT AT THE OTHER END OF THE LINK OR RETURN
DATA C(45,5)/ PICK THE Z AXIS ON THE JOINT AT THE FIRST END OF THE LINK OR RETURN
DATA C(46,1)/ CENTER OF .JOINT CHOSEN IS ON THE Z AXIS.
DATA C(54,1) = PICK  
DATA C(54,5) = A NEW POINT  
DATA C(55,1) = CENTER OF  
DATA C(55,2) = POINT CHOSEN  
DATA C(55,4) = IS ON THE Z AXIS.  
DATA C(55,5) = PICK  
DATA C(56,1) = A NEW POINT  
DATA C(56,2) = PICK LINK TO  
DATA C(56,3) = DELETE  
DATA C(56,4) =  
DATA C(57,1) = PICK FIRST CORNER  
DATA C(57,2) = OF THE  
DATA C(57,4) = ZOOM AREA  
DATA C(58,1) = PICK THE OPPOSITE  
DATA C(58,2) = CORNER  
DATA C(58,4) = ZOOM AREA  
DATA C(59,1) = PICK LINK  
DATA C(59,3) = FOR  
DATA C(59,4) = NAME INQUIRY  
DATA C(60,1) = ENTER A NAME  
DATA C(60,2) = FOR MECHIN  
DATA C(60,3) = RESTART FILE (UPPER CASE)  
DATA C(60,5) =  
DATA C(61,1) = PICK JOINT FOR  
DATA C(61,3) = INITIAL CONDITION  
DATA C(61,5) = SPECIFICATION  
DATA C(62,1) = INPUT JOINT  
DATA C(62,2) = MUST BE A REVOLUTE  
DATA C(62,4) = OR A PRISMATIC  
DATA C(62,5) = JOINT  
DATA C(63,1) = ENTER THE INITIAL  
DATA C(63,3) = POSITION OF 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 JOINT  
DATA C(64,5) = IN DEGREES  
DATA C(65,1) = ENTER THE  
DATA C(65,2) = NUMBER OF  
DATA C(65,4) = INCREASES TO BE MADE BY THIS JOINT  
DATA C(65,5) = VARIABLE  
DATA C(66,1) = ENTER INITIAL  
DATA C(66,2) = CONDITION OF THE PRISMATIC JOINT  
DATA C(66,5) = RETURN  
DATA C(67,1) = ENTER THE INITIAL  
DATA C(67,2) = POSITION OF THE PRISMATIC JOINT  
DATA C(67,4) =  
DATA C(68,1) = ENTER THE  
DATA C(68,2) = INCREMENT FOR  
DATA C(68,3) = THIS PRISMATIC JOINT  
DATA C(68,5) =  
DATA C(69,1) = ENTER THE  
DATA C(69,3) = SCALE FACTOR 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  

MECHIN PROGRAM LISTING
DATA C(70,4)'/ PLEASE RE-ENTER '/
DATA C(70,5)'/
DATA C(71,1)'/
DATA C(71,2)'/ PICK JOINT TO
DATA C(71,3)'/
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)'/
DATA C(73,2)'/ ENTER THE
DATA C(73,3)'/ VELOCITY OF
DATA C(73,4)' THE REVOLUTE
DATA C(73,5)'/ JOINT'
DATA C(74,1)'/
DATA C(74,2)'/ ENTER THE
DATA C(74,3)'/ VELOCITY OF
DATA C(74,4)'/ THE PRISMATIC
DATA C(74,5)'/ JOINT'
DATA C(75,1)'/
DATA C(75,2)'/ ENTER THE
DATA C(75,3)'/ ACCELERATION OF
DATA C(75,4)'/ THE REVOLUTE
DATA C(75,5)'/ JOINT'
DATA C(76,1)'/
DATA C(76,2)'/ ENTER THE
DATA C(76,3)'/ ACCELERATION OF
DATA C(76,4)'/ THE PRISMATIC
DATA C(76,5)'/ JOINT'
DATA C(77,1)'/
DATA C(77,2)'/ PICK JOINT
DATA C(77,3)'/ FOR
DATA C(77,4)' INQUIRY
DATA C(77,5)' OF JOINT INITIAL
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)'/
DATA C(79,2)'/ POSITION VALUES
DATA C(79,3)'/ MUST BE SPECIFIED
DATA C(79,4)' BEFORE
DATA C(79,5)' VELOCITIES AND
DATA C(79,6)' ACCELERATIONS'
DATA C(80,1)'/
DATA C(80,2)'/ PICK THE Z AXIS
DATA C(80,3)'/ ON THE JOINT
DATA C(80,4)'/ ALSO ON THE
DATA C(80,5)'/ EXISTING LINK'
DATA C(81,1)'/
DATA C(81,2)'/ GROUND LINK
DATA C(81,3)' INVALID
DATA C(81,4)'/ PICK A NEW JOINT
DATA C(81,5)'/
DATA C(82,1)'/
DATA C(82,2)'/ NUMBER OF
DATA C(82,3)'/ JOINTS ON LINK
DATA C(82,4)' EXCEEDS
DATA C(82,5)'/ 6'
DATA C(83,1)'/
DATA C(83,2)'/ ENTER A NAME
DATA C(83,3)'/ FOR IMP
DATA C(83,4)'/ DATA FILE
DATA C(83,5)' (UPPER CASE)'
DATA C(84,1)'/
DATA C(84,2)'/ ORIENTATION POINT'
DATA C(84,3)'/ AND JOINT ORIGIN
DATA C(84,4)'/ ARE THE SAME
DATA C(84,5)'/ PICK A NEW POINT'
DATA C(85,1)'/
DATA C(85,2)' MECHANISM OR
DATA C(85,3)'/ QUIT'
DATA C(85,4)'/
DATA C(86,1)'/
DATA C(86,2)'/ PICK JOINT TO
DATA C(86,3)' MAKE INVISIBLE OR
SUBROUTINE COUPLE

SUBROUTINE COUPLE(IHSID,LNAM,LNUM,LINK,JNAM,JNUM,JOINT,NUML,CSIZE)

*********************************************************************
* DELETE ALL PREVIOUS COMMENTS
*********************************************************************
CALL GPSTI(5)

*********************************************************************
* OPEN STRUCTURE FOR COMMAND1
*********************************************************************
CALL GPIPSI(5)
CALL GPPTXCI(2)
CALL GPANSC(1.0)
DRAG TEXT
CALL GPAN2(POS1,LENGTH,C(NUM,1))
CALL GPAN2(POS2,LENGTH,C(NUM,2))
CALL GPAN2(POS3,LENGTH,C(NUM,3))
CALL GPAN2(POS4,LENGTH,C(NUM,4))
CALL GPAN2(POS5,LENGTH,C(NUM,5))

CALL GPCLST

*********************************************************************
* SCREEN DISPLAY----------- VIEW INITIALIZATION GOES TO SETUP
*********************************************************************
CALL GPUPHSI(HSID,PRFORM)
RETURN
END

SUBROUTINE COUPLE

This subroutine is used to add a coupling link.
COMMON/LDEF/LNDEF(30)
INTEGER IWSID,NUML,PNUM(100),RFLG,JNUM(20,2),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),LIN(6),XVEC(3),YVEC(3),X1(6),X2(6), &
& JOINT(20,2),SCI(3),SCM(4,4),TRM(4,4),TRMI(4,4),P(2), &
& R1(3),P2(3),S(4,4),T(4,4),VEC(3),ZVEC(3)
CHARACTER*7 LNDEF,LNAME,PNAM(100),JNUM(20),LNUM(30),CLEAR,CLEAR1
CHARACTER*20 CLEAR2

DATA CLEAR/" "/ DATA CLEAR1/" "/
DATA CLEAR2/" "/
DATA TRM/1.0,0,0,0,0,0,1/ DATA TRM1/1.0,0,0,0,0,0,0/ DATA T/1.0,0,0,0,0,0,0/ DATA S/1.0,0,0,0,0,0,0/ 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)=0.90*CSIZE(1)
AREA(6)=0.90*CSIZE(1)
LNAME=CLEAR
SCI(1)=SCALE
SCI(2)=SCALE
SCI(3)=SCALE

CALL GPSC3(SC,SCM)

********************************************************************
* PICK FIRST END OF THE LINK
********************************************************************
10 CALL COMMAND(IWSID,80)
CALL PICK(JINSID,JOINT,AXISR,CSIZE,JNUM1,IAXIS1,RFLG)
C IF(JNUM1.EQ.0)THEN
CALL COMMAND(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 COMMAND(IWSID,51)
CALL MENU INSID,CSIZE,12,CHOICE
IF(CHOICE.EQ.1)THEN
CALL COMMAND(IWSID,53)
CALL TEOXI INSID,JNUM1,IAXIS1,JOINT,AXISR,CSIZE,XVEC,YVEC,RFLG
IF(IAXIS1.EQ.1)THEN
JOINT(JNUM1,7)=XVEC(1)
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 COMMAND(IWSID,52)
CALL TEOXI INSID,JNUM1,IAXIS1,POINT,AXISR,CSIZE,XVEC,YVEC,
&JOINT,RFLG
IF(IAXIS1.EQ.1)THEN
JOINT(JNUM1,7)=XVEC(1)
JOINT(JNUM1,9)=XVEC(3)
ENDIF
MECHIN PROGRAM LISTING 90
**SUBROUTINE DATA**

**SUBROUTINE DATA(IHSID,AWSZ,CSIZE,PNAM,PNUM,POINT,JNAM,JNUM, &JOINT,LNAM,LNUM,LINK,INAM,INUM,IC,SCALE)
*****************************************************/
*  POSITION ICON
CALL GPARV(IHSID,5,9,0.)
CALL GPARV(IHSID,4,11,0.)
*  POINTS
CALL GPARV(IHSID,4,12,0.)
*  JOINT NAMES
CALL GPARV(IHSID,4,13,0.)
CALL GPARV(IHSID,4,14,0.)
*  JOINTS
CALL GPARV(IHSID,3,15,0.)
CALL GPARV(IHSID,4,16,0.)
*  LINKS
CALL GPARV(IHSID,4,17,0.)
CALL GPARV(IHSID,4,18,0.)
*  LINK
CALL GPARV(IHSID,4,19,0.)
CALL GPARV(IHSID,4,20,0.)
SUBROUTINE DELIC(IHSID, CSIZE, AXISR, INAM, INUM, IC, JOINT, SCALE)

INTEGER IHSID, PRFORM, PPATH, ACLASS, CHOICE, INUM, JNUM, JOINT, SCALE
REAL CSIZE, DATA, PAREA, L1, JOINT, AXISR, POSN
CHARACTER*7 INAM, C, O

DATA RETURN/1. RETURN /
DATA LENG/16/
DATA POSN/0.05, 2.2/
DATA C/'CCCCCCC'/
DATA D/'DDDDDD'/
DATA DATA/32, 0, 0,
&2, 2, 1, 1, 
&4
&4
&4
C DATA PRFORM/2/
C DATA ACLASS/3, 4, 6/

* REQUEST PICK INPUT
* REQUEST PICK FILTER (ALL CLASS 2 DETECTABLE)
C CALL GPPK(IHSID, 1, 3, ACLASS, 0, ACLASS)
C CALL GPPK(IHSID, 1, 1, 2)

C SET PICK FILTER (ALL CLASS 2 DETECTABLE)
C MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
C DEFINE PICK AREA
PAREA(1)=((-98+1/2)*CSIZE[1])
PAREA(2)=(98+1/2)*CSIZE[1]

MECHIN PROGRAM LISTING
PAREA(3) = ((.98 + 1)/2) * CSIZE(2)
PAREA(4) = ((.48 + 1)/2) * CSIZE(2)
PAREA(5) = 0
PAREA(6) = CSIZE(3)

CALL GPNPK(IHSID,1,0,PPATH,1,PAREA,0,DATA,1)  
INITIALIZE PICK PATH
CALL GPPKMD(IHSID,1,3,2)  
PLACE PICK DEVICE IN THE EVENT MODE
CALL GPUPMS(IHSID,PRFORM)  
TRaverse ALL ACTIVE VIEWS
CALL CMDMI(IHSID,71)  
GET PICK OR RETURN COMMAND

******************************************************************************
* INCLUDE A RETURN PICK
******************************************************************************
CALL GPPOST(6)
CALL GPADCN(1,ACLASS)
CALL GPKDID(ID1)
CALL GPHAICI(1,0)
CALL GPAN2(POSN,LENG,RETURN)
CALL GPCLST
CALL GPARY(IHSID,3,6,0)
CALL GPUP(IHSID,3,1,1)
CALL GPUPMS(IHSID,PRFORM)

******************************************************************************
* AWAIt A PICK EVENT TO OCCUR AND RETRIEVE IT
******************************************************************************

200 CALL GPANEV(IHSID,15,CLS,ICL,IEV)
IF CL.L.EQ.5 THEN
GET PICK
CALL GPSTPK(1,1,PPATH)
IF NO PICK, RETURN
IF (PPATH(2),EQ.101) GO TO 1000
CHOICE IS SECOND ITEM OF PICK PATH

INAM(CHOICE) = C
INM(CHOICE,1) = 0
INM(CHOICE,2) = 0
ICIC(CHOICE,1) = 0.
ICIC(CHOICE,2) = 0.
ICIC(CHOICE,3) = 0.
ICIC(CHOICE,4) = 0.

CALL ICICON(IHSID,JOINT,INAM,INUM,IC,SCALE)
DEACTIVATE POINT
REDRAW ALL POINTS

AHAIT ANOTHER EVENT
GO TO 200
ELSE
IF NOT A PICK, GO TO AHAIT AN EVENT
ENDIF

1000 CALL GPKMD(IHSID,1,1,2)
CALL GPEST(6)
MAKE SURE PICK IS DEACTIVATED
CALL GPKF(IHSID,1,0,ACLASS,3,ACLASS)

RETURN

END

SUBROUTINE DELJ

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

INTEGER IHSID,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 '/

MECHIN PROGRAM LISTING
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/
DATA PRFORM/2/   FLAG FOR UPDATE
DATA ACLASS/3,4/   INCLUSION CLASS LIST

******************************************************************************
* REQUEST PICK INPUT
******************************************************************************
* SET PICK FILTER (ALL CLASS 3 DETECTABLE)
* MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
CALL GPPKF(INSID,1,2,ACLASS,0,ACLASS)
CALL GPPKMO(INSID,1,1,2)
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)=((1.48+1/2)*CSIZE(2))
PAREA(5)=0.
PAREA(6)=CSIZE(3)

INITIALIZE PICK PATH
CALL GPINPK(INSID,1,1,PPATH,1,PAREA,0,DATA,1)
PLACE PICK DEVICE IN THE EVENT MODE
CALL GPPKMO(INSID,1,3,2)
TRAVERSE ALL ACTIVE VIEWS
CALL GPUPWS(INSID,PRFORM)
CALL COMMAND(INSID,36)

******************************************************************************
* INCLUDE A RETURN PICK
******************************************************************************
* AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
******************************************************************************
AWAIT AN EVENT
200 CALL GPAHEV(1000.,1WS,ICLA,IDEV)
   IF(ICALA.EQ.5)THEN
      GET PICK
      CALL GPGTKP(1,1,PPATH)
      IF(PPATH(2).EQ.101)GO TO 1000
      CHOICE=PPATH(2)
      DEACTIVATE POINT
      JNAM(CHOICE)=C
      DO 201=1,21
         JOINT(CHOICE,I)=0.0
      CONTINUE
      REDRAW ALL POINTS
      CALL RDRHJT(INSID,JNAM,JNNM,JOINT,AXISR,SCALE)
   AWAIT ANOTHER EVENT
   ELSE IF NOT A PICK, GO TO AWAIT AN EVENT
   GO TO 200
IDF
C

MAKE SURE PICK IS DEACTIVATED
CALL GPPKMO(INSID,1,1,2)
CALL GPPKF(INSID,1,0,ACLASS,2,ACLASS)

RETURN
SUBROUTINE DELL

INTEGER IMSID,CSIZE,AXISR,JNAM,JNUM,JOINT,LNAM,LNUM,LINK,
& SCALE
REAL CSIZE(3),DATA(35),PAREA(6),L1(4),JOINT(20,21),AXISR,POSN(2),
& LINK(30,3)
CHARACTER*8 JNAM(20),C,LNAM(30),RETURN

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

C DATA PRFORM/2/ INCLUSION CLASS LIST
DATA ACLASS/5/

******************************************************************************
* REQUEST PICK INPUT
******************************************************************************
C CALL GPPKF(IMSID,1,1,ACLASS,0,ACLASS) MAKE PICK FILTER (ALL CLASS 3 DETECTABLE)
C CALL GPPKN(IMSID,1,1,2) DEFINE PICK AREA
PAREA1(1)=(-.98+1/2)*CSIZE(1) PAREA2(1)=(-.98+1/2)*CSIZE(1)
PAREA3(1)=(-.98+1/2)*CSIZE(1) PAREA4(1)=(.98+1/2)*CSIZE(2)
PAREA5=0 PAREA6=CSIZE(3)
C CALL GPINPK(IMSID,1,0) PLACIE PICK AREA IN THE EVENT MODE
C CALL GPUPHS(IMSID,PRFORM) GET PICK OR RETURN COMMAND
C CALL CMMAND(IMSID,56)
******************************************************************************
* INCLUDE A RETURN PICK
******************************************************************************
C CALL GPOPST(6)
C CALL GPADDN(1,ACLASS)
C CALL GPADD(101)
C CALL GPTX(1)
C CALL GPAN(POSN,LENG,RETURN)
C CALL GPCLST
C CALL GPARV(IMSID,3,6,0.)
C CALL GPVPI(IMSID,2,1,1)
C CALL GPUPMS(IMSID,PRFORM)
******************************************************************************
* WAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
******************************************************************************
C 200 CALL GPAHEV(1000.,IMS,1,ACL,1000)
C IF(ICAL.A.EQ.5)THEN GET PICK
C CALL GPGTPK(1,1,PPATH) IF NO PICK, RETURN
C IF((PPATH(2).EQ.101).AND.1000 TO 1000
C CHOICE=PPATH(2) DEACTIVATE POINT
C LNUM(CHOICE)=0 DO 250 I=1,12
MECHIN PROGRAM LISTING 95
SUBROUTINE DELP

SUBROUTINE DELP(IWSID,CSIZE,AXISR,PNAM,PNUM,POINT)

*********************************************************************
* ----- SUBROUTINE DELP ----- *
*********************************************************************
INTEGER IWSID,PRFORM,PPATH(3),ACLASS(1),CHOICE,PNUM(100),LENG
REAL CSIZE,PAREA(12),DATA(35),PAREA(6),L1(4),POINT(100,3),AXISR,POSN(2)
CHARACTER*7 PNAM(100),C
CHARACTER*16 RETURN
DATA RETURN/'RETURN'/ DATA LENG/16/ DATA POSN/0.05,2.2/ DATA C/CCCCCCCC'/
DATA DATA/32,0,0, DATA PRFORM/2/ DATA ACLASS/2/ FLAG FOR UPDATE
CHARACTER (16) RETURN INCLUSION CLASS LIST
acciones

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

*************** REQUEST PICK INPUT ****************
C CALL GPOPST(6) CALL GPADCN(1,ACLASS) CALL GPPKDI(101) CALL GPTXEI(2)
C CALL GPAHSCI(1,0) CALL GPAE(POSN,LENG,RETURN)

MECHIN PROGRAM LISTING
CALL GPCLST
CALL GPARV(INSID,3,6,0.)
CALL GPPVI(INSID,6,1.)
CALL GPUPMGS(INSID,PRORM)

********************************************************************* *
* AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
*********************************************************************

C AWAIT AN EVENT
200 CALL GPAHEV(1000.,INS,ICLA,IDEV)
IF(ICLA.EQ.5)THEN
  CALL GPGETP(1,1,PPATH)
  IF(NOT PICK, RETURN
  IF(PPATH(2).EQ.100)GO TO 1000
  CHOICE=PPATH(2)
  PNAM(CHOICE)=C
  CALL RDRWPT(INSID,PNAM,PNUM,POINT,AXISR)
  GO TO 200
ELSE
  GO TO 200
ENDIF

C MAKE SURE PICK IS DEACTIVATED
1000 CALL GPPKMO(INSID,1,1,Z)
CALL GPST(6)
C MAKE PICK FILTER(ALL CLASS 2 UNDETECTABLE)
RETURN
END

SUBROUTINE DEO

SUBROUTINE DEO(INSID,AXISR,CSIZE,LOCAT,RFLG)

********************************************************************* *
------- SUBROUTINE DEO -------
********************************************************************* *

** THIS SUBROUTINE WILL PROMPT FOR THE DIRECT ENTRY OF JOINT ORIENTATION COORDINATES **
********************************************************************* *
** INTEGER INSID,RFLG,CHOICE **
** REAL LOCAT(3),AXISR,AREA(6),CSIZE(3),LX(2),LY(2),LZ(2),LXi(2), **
& LYi(2),LZi(2),AXIS(6),INIT,INITi1 **
** CHARACTER*20 CLEAR,AXIS,INIT,INITi1 **
** CHARACTER*12 STRING,CLEARZ **
** DATA CLEAR/' DATA INIT/'O. **
** DATA INITi1/'O. **
** DATA STRING/' DATA CLEARZ/' DATA AXiS/' DATA CLEARZ/' **
** 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 **
** LX(1)=0.15 **
** LX(2)=0.68 **
** LY(1)=0.05 **
** LY(2)=0.62 **

** MECHIN PROGRAM LISTING **

97
MECHIN PROGRAM LISTING
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)

MECHIN PROGRAM LISTING 99
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;'10' /
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

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

LZ(1)=0.05
LZ(2)=0.62

CALL GPEST(6)

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

CALL GPEST(12, NUM, STRING)
REWRITE 10
WRITE(10, 8) CLEAR1
8 FORMAT(20)
REWRITE 10
WRITE(10, 12) STRING
12 FORMAT(A12) REWRITE 10
READ(10, *), ERR=1000 LOCAT(1)
WRITE X LOCATION TO THE SCREEN
CALL GPOPST(10)
CALL GPTXC(11)
CALL GAPHC(1, 1.0)
CALL GPAN2(LX, 20, X)
CALL GPAN2(LX1, 12, STRING)
CALL GPCLST
CALL GPEST(9)
CALL GPOPST(9)
CALL ICON1 LOCAT, AXISR)
CALL GPCLST

15 IF(ABS(LOCAT(1)).GT.AXISR) THEN
POIN T IS OUTSIDE DEFINED AXES
CALL CMND(INSID, 17)
CALL MENU(INSID, CSIZE, 4, CHOICE)
CALL GPSTMO(INSID, 1, 1, 1)
15 IF(CHOICE.EQ.1) THEN
REDEFINE THE AXIS RANGE
AXIS=CLEAR
CALL CMND(INSID, 5)
CALL GPSND(INSID, 1, 1, 1, 2)
CALL GPINSI(INSID, 1, 1, 1, INIT1, AREA, 1, 1, 0, CLEAR)
CALL GPRSTI(INSID, 1, 1, 1, 1, ISTAT, NUM, AXIS)
REWRITE 10
WRITE(10, 16) CLEAR1
16 FORMAT(20)
REWRITE 10
WRITE(10, 20) AXIS
20 FORMAT(7)
REWRITE 10
READ(10, *, ERR=1000) AXISR
CALL GPEST7)
CALL AXES(INSID, AXISR)
GO TO 15
C IF(CHOICE.EQ.2) THEN RETURN WITH NO POINT
RFLG=1
GO TO 1000
ENDIF
ENDIF

********************************************************************* *
* PROCESS Y COORDINATE
********************************************************************* *
CALL GPSTM0(IHSID,1,1,2) CALL GPMST(IHSID,1,12,INIT,1,AREA,12,1,0,CLEAR) STRING=CLEAR2
CALL CMD0(IHSID,19) CALL GMRST(IHSID,1,12,ISTAT,NUM,STRING) REMIND 10 WRITE(10,108)CLEAR1
108 FORMAT(A20) REMIND 10 WRITE(10,112)STRING
112 FORMAT(A12) REMIND 10 READ(10,*,ERR=1000)LOCAT(2)
C CALL GPPSTI(10) CALL GPTXC(1) CALL GPAHSC(1.0) CALL GPAN2(LY,20,Y) CALL GPAN2(LY1,12,STRING) CALL GPCST
CALL GPEST(9)
CALL GPPST(9) CALL ICON(LOCAT,AXISR) CALL GPCST
115 IF(ABS(LOCAT(2)).GT.AXISR) THEN POINT IS OUTSIDE DEFINED AXES
CALL CMD0(IHSID,17) CALL GPMST(IHSID,1,1,1) CALL MENU(IHSID,CSIZE,*,CHOICE)
C IF(CHOICE.EQ.1) THEN AXIS=CLEAR
CALL CMD0(IHSID,5) CALL GPSTM0(IHSID,1,1,2)
CALL GPMST(IHSID,1,1,7,INIT1,1,AREA,7,1,0,CLEAR)
CALL GPPST(IHSID,1,7,ISTAT,NUM,AXIS) REMIND 10 WRITE(10,116)CLEAR1
116 FORMAT(A20) REMIND 10 WRITE(10,120)AXIS
120 FORMAT(A7) REMIND 10 READ(10,*,ERR=1000)AXISR CALL GPEST(7)
CALL AXES(IHSID,AXISR) GO TO 115
ENDIF
IF(CHOICE.EQ.2) THEN RETURN WITH NO POINT
RFLG=1
GO TO 1000
ENDIF

********************************************************************* *
* PROCESS Z COORDINATE
********************************************************************* *
CALL GPSTM0(IHSID,1,1,2) CALL GPMST(IHSID,1,12,INIT,1,AREA,12,1,0,CLEAR) STRING=CLEAR2
CALL CMD0(IHSID,19) CALL GMRST(IHSID,1,12,ISTAT,NUM,STRING) REMIND 10 WRITE(10,208)CLEAR1
208 FORMAT(A20) REMIND 10 WRITE(10,212)STRING
212 FORMAT(A12) REMIND 10 READ(10,*,ERR=1000)LOCAT(3)
C WRITE Z LOCATION TO THE SCREEN
CALL GPPSTI(10) CALL GPTXC(1) CALL GPAHSC(1.0) CALL GPAN2(LZ,20,Z) CALL GPAN2(LZ1,12,STRING)
SUBROUTINE DRAWJT

SUBROUTINE DRAWJT(IHSID,AXISR,JNAME,NUMJ,JTYPE,JOINT,SC)

THIS SUBROUTINE IS USED TO DRAW A JOINT IN THE MAIN WINDOW

INTEGER IHSID,NUMJ,CLASS1(1),CLASS1(1),JTYPE
REAL LOCAT(3),LOCATI(3),ORIENT(3),AXISR,SCALE(3),ORIGINI(3),
& MATSC(4,4),MATRIX(4,4),UP(3),Z(3),MATI(4,4),MAT2(4,4),MAT3(4,4),
& MAT4(4,4),MAT5(4,4),V(3),NEXI(3),NEVI(3),LZ(3),NEV(3),
& LZ2(3),Z2(6),Z2(6),JOINT(20,21),ZO(3),LOCALI(3),LOCALZ(3),
& ORIENI(3),LOCAT2(3),SC,SC2(4),MCS2(4,4),NEV2(3)
CHARACTER*7 JNAME
CHARACTER*2 ZIN,Z2N

DATA ZIN/'Z1'/
DATA Z2N/'Z2'/
DATA CLASS1/3/ DATA CLASS1/3/
DATA CLASS1/4/ DATA CLASS1/4/
DATA MATI/1.,0.,0.,0.,1.,0.,0.,0.,1.,0.,0.,0.,0.,0.,0.,1./
DATA MAT2/1.,0.,0.,0.,1.,0.,0.,0.,1.,0.,0.,0.,0.,0.,0.,1./
DATA MAT3/1.,0.,0.,0.,1.,0.,0.,0.,1.,0.,0.,0.,0.,0.,0.,1./
DATA MAT4/1.,0.,0.,0.,1.,0.,0.,0.,1.,0.,0.,0.,0.,0.,0.,1./
DATA MATSC/1.0,0.0,0.0,1.0,0.0,0.0,1.0,0.0,0.0,1.0/
DATA MATRIX/1.0,0.0,0.0,1.0,0.0,0.0,1.0,0.0,0.0,1.0/
LOCAT(1)=JOINT(NUMJ,1)
LOCAT(2)=JOINT(NUMJ,2)
LOCAT(3)=JOINT(NUMJ,3)
LOCAT(4)=JOINT(NUMJ,15)
LOCAT(5)=JOINT(NUMJ,16)
LOCAT(6)=JOINT(NUMJ,17)
ORIENT(1)=JOINT(NUMJ,4)
ORIENT(2)=JOINT(NUMJ,5)
ORIENT(3)=JOINT(NUMJ,6)
ORIENT(4)=JOINT(NUMJ,18)
ORIENT(5)=JOINT(NUMJ,19)
ORIENT(6)=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)=LOCAT(1)
LOCAT1(2)=LOCAT(2)
LOCAT1(3)=LOCAT(3)
LOCAT2(1)=LOCAT2(1)*(ZO(1)*SC)
LOCAT2(2)=LOCAT2(1)*(ZO(2)*SC)
LOCAT2(3)=LOCAT2(1)*(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)=0.0
Z2(2)=0.0
Z2(3)=0.0
Z2(4)=ORIENT(1)*AXISR/10.*2.4+(AXISR*.02)
Z2(5)=ORIENT(2)*AXISR/10.*2.4+(AXISR*.02)
Z2(6)=ORIENT(3)*AXISR/10.*2.4+(AXISR*.02)
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*.02)
LZ2(1)=ORIENT(1)*AXISR/10.*2.4+(AXISR*.02)
LZ2(2)=ORIENT(2)*AXISR/10.*2.4+(AXISR*.02)
LZ2(3)=ORIENT(3)*AXISR/10.*2.4+(AXISR*.02)
CALL GPTRL3(LOCAT,MAT1)
CALL GPTRL3(LOCAT,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
Z1(1)=0.0
Z1(2)=0.0
Z1(3)=1.0
Y1(1)=0.0
Y1(2)=0.0
Y1(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 VPRODIUP,ORIENT,NEH
CALL VPRODORIENT,NEHX,NEHY
100 DO 101 I=1,3
MATS(1,I)=NEHX(I)
MATS(1,I)=NEHY(I)
MAT3(I,3)=ORIENT(I)
MAT3(I,4)=LOCAT(I)
101 CONTINUE
108 CALL GPSC3(SCALE,MATSC)
   CALL GPSC3(SC2,MSC2)
********************************************************************
* OPEN STRUCTURE 15 AND DRAW JOINT (MAKE PICKABLE)             *
********************************************************************
C OPEN STRUCTURE 13
C CALL GPOST(13)       INSERT CLASS NAME TO ALLOW PICKING JOINTS (CLASS 2)
C CALL GPADCN(1,CLASS)  SET PICK I.D.
C CALL GPPKID(NUMJ)    IF(JTYPE.EQ.7) GO TO 110
C CALL GPHLX3(MAT3,1)  CALL JOINT ICON
C CALL JICON(LOCAT,ORIENT,JTYPE,AXISR,JOINT,NUMJ,SC,MAT3)
C CLOSE STRUCTURE
C CALL GPCLST
C CALL GPOST(14)       INSERT CLASS NAME TO ALLOW PICKING JOINTS (CLASS 2)
C CALL GPADCN(1,CLASS)  SET PICK I.D.
C CALL GPPKID(NUMJ)    IF(JTYPE.EQ.7) GO TO 110
C CALL GPHLX3(MAT3,3)  CALL JOINT ICON
C CALL JICON(LOCAT,ORIENT,JTYPE,AXISR,JOINT,NUMJ,SC,MAT3)
C CLOSE STRUCTURE
C CALL GPCLST
C OPEN STRUCTURE 14 (JOINT)
C CALL GPOST(14)       INSERT CLASS NAME TO ALLOW PICKING JOINTS (CLASS 2)
C CALL GPADCN(1,CLASS)  SET PICK I.D.
C CALL GPPKID(NUMJ)    IF(JTYPE.EQ.7) GO TO 110
C CALL GPHLX3(MAT3,2)  CALL JOINT ICON
C CALL JICON(LOCAT,ORIENT,JTYPE,AXISR,JOINT,NUMJ,SC,MAT3)
C CLOSE STRUCTURE
C CALL GPCLST
C OPEN STRUCTURE 17 (Z AXIS 1)
C CALL GPOST(17)       INSERT CLASS NAME TO ALLOW PICKING AXES (CLASS 4)
C CALL GPADCN(1,CLASS)  SET PICK I.D.
C CALL GPPKID(NUMJ)    SET TRANSFORMATION
C CALL GPMLX3(MAT3,3)  CALL Z AXIS 1
C CALL GPLT(1)         SET LINE TYPE SOLID
C CALL GPPLCI(6)       SET LINE COLOR
C CALL GPPL3(2,2,Z1)   ASSOCIATE FIRST Z AXIS
C CALL GPAHSC(.1)      SET CHARACTER HEIGHT SCALE FACTOR
C CALL GPTXCI(6)       TEXT COLOR INDEX
C CALL GPANS(LZ1,2,Z1) ASSOCIATE TEXT
C CLOSE STRUCTURE
C CALL GPCLST
C IF(JTYPE.EQ.8)THEN    GROUND LINK (DRAW JUST 1 AXIS)
C GO TO 1000
ENDIF
C OPEN STRUCTURE 18 (Z AXIS 2)
C CALL GPOST(18)       INSERT CLASS NAME TO ALLOW PICKING AXES (CLASS 4)
C CALL GPADCN(1,CLASS)  SET PICK I.D.
C CALL GPPKID(NUMJ)    SET TRANSFORMATION
C CALL GPMLX3(MAT3,2)  CALL Z AXIS 2
C CALL GPLT(1)         SET LINE TYPE SOLID
C CALL GPPLCI(2)       SET LINE COLOR
C CALL GPPL3(2,2,Z2)   ASSOCIATE FIRST Z AXIS
C CALL GPAHSC(.1)      SET CHARACTER HEIGHT SCALE FACTOR
C CALL GPTXCI(2)       TEXT COLOR INDEX
C CALL GPANS(LZ2,2,Z2) ASSOCIATE TEXT
C CLOSE STRUCTURE
C CALL GPCLST
1000 CALL GPUPHS(IHSID,2)
RETURN
SUBROUTINE DRAWLK(IHSID, LNAM, LNUM, LINK, JOINT, JNUM, AXISR, NUML, & SCALE)
*********************************************************************
* ----- SUBROUTINE DRAWLK -----
**
** THIS SUBROUTINE IS USED TO DRAW A LINK IN THE MAIN WINDOW
**
*********************************************************************

INTEGER IHSID, NUML, CLASS, LNAM(30,14), JNUM(6), IAXIS(6), & JNUMJ(20,3)
REAL LINK(30,6), JOINT(20,21), XVEC(6), X1(6), X2(6), LN(6), & SCM(4,4), TRM(4,4), TRM1(6,4,4), F(3), P1(3), P2(3), S(4,4), & T(4,4), X1(6,6), TRM(4,4)

CHARACTER*7 LNAM(30)

DATA CLASS/5/ DATA SCM/1.0, 0.0, 0.0, 0.0, 0.0, 0.0/ DATA TRM/1.0, 0.0, 0.0, 0.0, 0.0, 0.0/ DATA S/1.0, 0.0, 0.0, 0.0, 0.0, 0.0/ DATA T/1.0, 0.0, 0.0, 0.0, 0.0, 0.0/

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 GPSCL(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(I)=JOINT(JNUM(I),7)
      XVEC(2)=JOINT(JNUM(I),8)
      XVEC(3)=JOINT(JNUM(I),9)
   ENDIF
   IF(IAXIS(I).EQ.2) THEN
      XVEC(I)=JOINT(JNUM(I),10)
      XVEC(2)=JOINT(JNUM(I),11)
      XVEC(3)=JOINT(JNUM(I),12)
   ENDIF
   IF(IAXIS(I).EQ.1) THEN
      TRM(I)=JOINT(JNUM(I),1)
      TRM(2)=JOINT(JNUM(I),2)
      TRM(3)=JOINT(JNUM(I),3)
   CALL GPTL3(TRM, TRM1)
   DO 60 J=1,4
      ;DO 60 K=1,4
      ;TRM1(I,J,K)=TRM(I,J,K)
      CONTINUE
   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

50 CONTINUE

60 CONTINUE

END
CALL GPOPST(19)
CALL GPADCN(1,CLASS)
CALL GPPL3(2,3,X1)
CALL GPCLST
ENDIF

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

CALL GPTRL3(TR,TRM)
DO 63 J=1,4
DO 62 K=1,4
TRM(I,J,K)=TRM(J,K)
CONTINUE
CONTINUE

CALL GPXF3CP(Tr,Pl)
MECHIN PROGRAM LISTING

100 IF(LNUM=14.JEQ.1) GO TO 1000
Pi=XX1(I,4)+SCALE
P2=XX1(I,5)+SCALE
P3=XX1(I,6)+SCALE
DO 103 J=1,4
DO 102 K=1,4
TRM(I,J,K)=TRM(I,J,K)
CONTINUE
CONTINUE

CALL GPXF3(P,TRM,P1)
P1=XX1(2,4)+SCALE
PI* SCALE
PI(J) = XX1(2,J)*SCALE
PI(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(2),2), EQ.8) THEN
  P1(1) = 0.
  P1(2) = 0.
  P1(3) = 0.
  CALL GPXF3(P,TRM,P1)
  LNK(1) = P1(1)
  LNK(2) = P1(2)
  LNK(3) = P1(3)
ENDIF

IF(JNUM1(JNUM(1),1), EQ.8) THEN
  P1(1) = 0.
  P1(2) = 0.
  P1(3) = 0.
  CALL GPXF3(P,TRM,P2)
  LNK(1) = P2(1)
  LNK(2) = P2(2)
  LNK(3) = P2(3)
ENDIF

CALL GPOPST19
CALL GPADCN1,CLASS
CALL GPPKIDINUML CALL GPMLX31T,3J CALL GPMLX31S,3J
CALL GPPL31Z,3,LNK) CALL GPCLST
CALL GPUPHS19 J

********************************************************************
* DRAH THE COUPLING LINKS
********************************************************************

DO 100  J=1,6
  IF(JNUM(J), EQ.0) THEN
    GO TO 1000
  ENDIF
  P1(1) = XX1(I,1)*SCALE
  P1(2) = XX1(I,6)*SCALE
  P1(3) = XX1(I,6)*SCALE

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

CALL GPXF3(P,TRM,P1)

  P2(1) = JOINT(JNUM1(1),1) + (JOINT(JNUM2,1),1) - JOINT(JNUM1(1),1))/2.
  P2(2) = JOINT(JNUM1(1),2) + (JOINT(JNUM2,1),2) - JOINT(JNUM1(1),2))/2.
  P2(3) = JOINT(JNUM1(1),3) + (JOINT(JNUM2,1),3) - JOINT(JNUM1(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 GPOPST19
CALL GPADCN1,CLASS
CALL GPPKIDINUML CALL GPMLX31T,3J
SUBROUTINE DRAWPT

SUBROUTINE DRAWPT(IWSID,LOCAT,AXISR,PNAME,NUMPJ

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

SUBROUTINE ENTER(IWSID,CHOICE,CSIZE)

*----- SUBROUTINE ENTER ----- *
**
* THIS SUBROUTINE IS USED TO DETERMINE MENU CHOICE *
**
*********************************************************************
INTEGER IWSID,LENG,PRFORM,PATH(3),AClass(1),CHOICE
REAL POSN(2),CSIZE(3),PAREA(6),DATA(35)
CHARACTER*35 TMENU(10)

DATA TMENU(1)/'1. ENTER POSITION '/
DATA TMENU(2)/'2. RETURN '/
DATA TMENU(3)/'3. OPTIONS '/
DATA TMENU(4)/'4. SCREEN DISPLAY '/
DATA TMENU(5)/'5. REQUEST PICK INPUT '/
DATA TMENU(6)/'6. OPEN STRUCTURE '/
DATA TMENU(7)/'7. CLOSE STRUCTURE '/
DATA TMENU(8)/'8. INSERT A TEST POLYLINE '
DATA TMENU(9)/'9. INSERT CLASS NAME TO ALLOW PICKING MENU ITEM '/
DATA TMENU(10)/'  

DATA /32,0,0,
DATA /32,0,0,
DATA /32,0,0,

DATA PRFORM/2/ FLAG FOR UPDATE
DATA ACLASS/1/ INCLUSION CLASS LIST
POSNI(1)=0.05 POSNI(2)=2.2

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

CALL GPOPST(6) OPEN A STRUCTURE
CALL GPADCN(1,ACLASS) INSERT A TEST POLYLINE
CALL GPTXCI(2) SET TEXT COLOR TO YELLOW
CALL GPAHSC(1.00) SET ANNOTATION SIZE SCALE FACTOR

DO 100 I=1,5
CALL GPPKID(I) SET PICK IDENTIFIER
CALL GPAN2(POSN,LENG,TMENUL(i)) DRAW TEXT
POSN(2)=POSN(2)-.10 SET NEXT POSITION
CALL GPPKID(I) SET PICK IDENTIFIER
CALL GPAN2(POSN,LENG,TMENUL(i+5)) DRAW TEXT
POSN(2)=POSN(2)-.20 SET NEXT POSITION
100 CONTINUE CLOSE STRUCTURE CALL GPC1ST

******************************************************************************
* SCREEN DISPLAY
******************************************************************************

CALL GPARV(INSID,3,6,0.) ASSOCIATE THE STRUCTURE 6 WITH A VIEW 3
CALL GPVP(INSID,5,1,1) TRAVERSE ALL ACTIVE VIEWS
CALL GPUPHS(INSID,PRFORM)

******************************************************************************
* REQUEST PICK INPUT
******************************************************************************

CALL GPPKF(INSID,1,1,ACLASS,0,ACLASS) SET PICK FILTER (ALL CLASS 1 DETECTABLE)
CALL GPPKMO(INSID,1,3,5) MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
CALL GPPKMO(INSID,1,3,6) DEFINE PICK AREA
POREA(1)=0.0
POREA(2)=0.2
POREA(3)=0.0
POREA(4)=0.0
POREA(5)=0.0
POREA(6)=0.2

CALL GPINPK(INSID,1,0,PPATH,1,POREA,0,DATAl) INITIALIZE PICK PATH
CALL GPINPK(INSID,1,3,5) PLACE PICK DEVICE IN THE EVENT MODE
CALL GPUPHS(INSID,PRFORM) TRAVERSE ALL ACTIVE VIEWS
SUBROUTINE ENTERO(HSID, CHOICE, CSIZE)

*********************************************************************
* ----- SUBROUTINE ENTER ----- *
* THIS SUBROUTINE IS USED TO DETERMINE MENU CHOICE *
*********************************************************************

INTEGER HSID, LENG, OPROM, PPATH(3), ACLASS(1), CHOICE
REAL POSN(2), CSIZE(3), PAREA(6), DATA(35)
CHARACTER*35 TMENU

C OPTIONS IN DATA SUBROUTINE
DATA TMENU(1)/'1. ENTER'/
DATA TMENU(6)/'ORIENTATION'/
DATA TMENU(12)/'2. RETURN'/
DATA TMENU(17)/'3. RETURN'/
DATA TMENU(8)/'4. RETURN'/
DATA TMENU(14)/'5. RETURN'/
DATA TMENU(10)'/'
DATA LENG/19/
DATA/32,0,0,
& 1,1,1,1,1,1,
& 1,1,1,1,1,1,
& 1,1,1,1,1,1,

C DATA OPROM/2/ FLAG FOR UPDATE
DATA ACLASS/1/ INCLUSION CLASS LIST
POSN(1)=0.05
POSN(2)=2.2

*********************************************************************
* OPEN STRUCTURE FOR THE MENU OF ENTER POINT OR RETURN *
*********************************************************************
OPEN A STRUCTURE CALL GPOPST
CALL GPADCN(1, ACLASS) INSERT CLASS NAME TO ALLOW PICKING MENU ITEM
CALL GPTXCI(2) SET TEXT COLOR TO YELLOW
CALL GPAHSC(1,00) SET ANNOTATION SIZE SCALE FACTOR
DO 100 I=1,5
CALL GPKID(I) SET PICK IDENTIFIER
CALL GPAN2(POSN, LENG, TMENU(I)) DRAW TEXT
POSN(2)=POSN(2)-.10 SET NEXT POSITION
CALL GPKID(I) SET PICK IDENTIFIER
CALL GPAN2(POSN, LENG, TMENU(I+5)) DRAW TEXT
POSN(2)=POSN(2)-.20 SET NEXT POSITION
100 CONTINUE CLOSE STRUCTURE
CALL GPCLSR

*********************************************************************
* SCREEN DISPLAY *
*********************************************************************
CALL GPARV(HSID, 3, 6, 0.) ASSOCIATE THE STRUCTURE 6 WITH A VIEW 3
CALL GVPV(HSID, 3, 1, 1) TRAVERSE ALL ACTIVE VIEWS
CALL GPUPWS(HSID, OPROM)

MECHIN PROGRAM LISTING
SUBROUTINE FILENM(IWSID, FILE)

* ------ SUBROUTINE FILENM ------ *

* THIS SUBROUTINE IS USED TO DISPLAY FILENAME ON THE SCREEN *

INTEGER IWSID, LENGF, LENGF, PRFORM

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

DATA LENGF/7/ FILE NAME LENGTH AND POSITION
DATA POSF/.79,.94/ FILE IDENT. LENGTH AND POSITION
DATA LENGF/16/ FILE IDENT. LENGTH AND POSITION
DATA POST/.60,.94/ FILE TITLE IDENTIFIER
DATA PRFORM/2/ FLAG FOR UPDATE
DATA FILET/'FILE : '/

C CALCULATION FOR THE TWO SHADOW BOXES
DO 100 I=1,7,2
   SHDII=BOXI(I)+.025
100 CONTINUE
DO 200 I=2,8,2
   SHDII=BOXI(I)-.025
200 CONTINUE

* OPEN STRUCTURE FOR COMMAND BOX (ALSO PUT IN STRUCTURE 3) *

CALL GPOPSTI(3) OPEN A STRUCTURE
CALL GPISI(2) SET SOLID INTERIOR STYLE FOR GREY BOX
CALL GPICII(4) SET COLOR FOR INTERIOR OF SHADOW BOX
CALL GPPG2(1,4,2, SHD1) DRAW GREY SHADOW BOX
CALL GPICII(3) SET COLOR FOR INTERIOR OF GREY BOX
CALL GPPG2(1,4,2, BOX1) DRAW GREY COMMAND BOX
CALL GPICII(1) SET COLOR FOR WHITE OF FILE NAME
CALL GPTXCI(1) SET TEXT COLOR TO WHITE
CALL GPAHSC(1.00) SET ANNOTATION SIZE SCALE FACTOR
CALL GPPG2(1,4,2, BOX1) DRAW TITLE TEXT
SUBROUTINE FILEXS

--- FILEXS

CHECK FOR FILE EXISTENCE (AFILE = FILE TO BE CHECKED)

IAM = FILE TO BE CHECKED
IRET = RETURN FLAG
0 -- FILE EXISTS
1 -- FILE DOES NOT EXIST

SUBROUTINE FILEXS (AFILE, IRET)

CHARACTER*7 AFIL
CHARACTER*20 SYS

DATA SYS / 'STATE FILE A' /

DO 20 I = 1, 7
  SYS(I+11:I+11) = AFILE(I:I)
  CONTINUE

CALL SYSCAL (SYS, 20, IRET)

RETURN

END

SUBROUTINE ICICON

--- SUBROUTINE ICICON

THIS SUBROUTINE IS USED TO DISPLAY THE ICONS FOR THE IC

INTEGER INUM(20,2), INSID, CLASS(1)
REAL IC(20,4), JOIN(20,21), P(3)
CHARACTER*7 INAM(20)
CHARACTER*2 TEXT

DATA TEXT,'IC'/
DATA CLASS,'/'

CALL GPEST(20)
CALL GPUPHS(IHSID,2)
DO 10 III=1,20
C
IF(INAM(III).EQ.'BBB BBB')THEN CONTINUE
ENDIF
C
IF(INAM(III).EQ.'CCCCCCC')THEN CONTINUE
ENDIF
C
IF(INAM(III).EQ.'DDDDDDD')THEN

DRAW ICON
P(1)=JOIN(INUM(III,1,1),1-.60*SCALE)
P(2)=JOIN(INUM(III,1,2),2)
P(3)=JOIN(INUM(III,1,3),3)
CALL GPOPST(20)
CALL GPADCN(1,CLASS)
CALL GPREDI(III)
CALL GPAHSCI(2)
CALL GPTXC17)
CALL GPAN3(2,TEXT)
CALL GPCLST
C
CALL GPUPHS(IHSID,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(1)=LOCAT(1)
LX(2)=LOCAT(2)
LX(3)=LOCAT(3)
LX(4)=LOCAT(4)
LX(5)=LOCAT(5)
LX(6)=LOCAT(6)
LINEY(1)=LOCAT(1)
LINEY(2)=LOCAT(2)
LINEY(3)=LOCAT(3)
LINEY(4)=LOCAT(4)
LINEY(5)=LOCAT(5)
LINEY(6)=LOCAT(6)
LINEZ(1)=LOCAT(1)
LINEZ(2)=LOCAT(2)
LINEZ(3)=LOCAT(3)
LINEZ(4)=LOCAT(4)
LINEZ(5)=LOCAT(5)
LINEZ(6)=LOCAT(6)

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

SUBROUTINE ICOND(INSID,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 INSID,CHOICE,JNUM(20,3),JNAM(20,3),JOINT(20,21),INAM(20,21)
REAL CSIZE(3),AXISR,JOINT(20,21),IC(20,4)
CHARACTER*7 JNAM(20,3),INAM(20,21)

CALL CMMAND(INSID,10)
CALL MENUL(INSID,CSIZE,12,CHOICE)

IF(CHOICE.EQ.1)THEN
  CALL ADDIC(INSID,CSIZE,AXISR,JNAM,JNUM,JOINT,INAM,INUM,IC, &SCALE)
  CALL GPPLC(1)
  CALL GPLT(2)
  CALL GPPL3(2,3,LINEX)
  CALL GPPL3(2,3,LINEY)
  CALL GPPL3(2,3,LINEZ)
  CALL GPPL3(5,3,XBOX)
  CALL GPPL3(5,3,YBOX)
  CALL GPPL3(5,3,ZBOX)
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),
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(1)=LOCAT(1)
LY(1)=LOCAT(2)
LZ(1)=LOCAT(3)
OR(1)=LOCAT(4)
OR(2)=LOCAT(5)
OR(3)=LOCAT(6)

C CALL GPPLCI(1) SET POLYLINE COLOR
C CALL GPPT(1) SET LINE TYPE SOLID
C CALL GPPLCI(2) SET POLYLINE COLOR
CALL GPPL3(2,3,OR)
CALL GPPLCI(5)
CALL GPPL3(2,3,LY)
CALL GPPL3(2,3,LZ)

RETURN END

SUBROUTINE IMP(PNAM,PNUM,POINT,JNAM,JNUM,JOINT,LNAM,LNUM,LINK,
CSIZE,INAM,INUM,IC,WSID)

******************************************************************************
* ----- 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), & C(20,4)
CHARACTER*7 LNAM(30),JNAM(20),PNAM(100),FILE,FILEO,FILE1,INAM(20), & JNAME,LNAME1,LNAME2,FILLEN
CHARACTER*19 FILEN
DATA FILE/'IMP''/
NP=0
NJ=0
NI=0

DO 2 I=1,20
  DO 1 J=1,3
    CONTINUE
  1 CONTINUE
  2 CONTINUE
  AREA(1)=.08*CSIZE(1)
  AREA(2)=.90*CSIZE(1)
  AREA(3)=.70*CSIZE(2)
  AREA(4)=.02*CSIZE(3)
  AREA(6)=.90*CSIZE(3)
  FILEN=' ' 

REHIND 15
*********************************************************************
* INITIALIZE THE STRING DEVICE. REQUEST NEW RESTART FILE NAME
*********************************************************************
C INITIALIZE STRING MODE
CALL GPSTMOIHSID,1,1,2)
C 3 CALL GINST(IHSID,1,7,FILE,1,AREA,/1.0,FILE)
C 4 CALL CMDAID(IHSID,03)
FILEN='' REQUEST STRING
C CALL GPROST(IHSID,1,7,ISTAT,NUM,FILEN)
IF(FILEN.EQ.' ')GO TO 5000
FILE=FILE
SEE IF FILE ALREADY EXISTS
C IF(FILE.EQ.' ')THEN
call GPROST(IHSID,1,7,ISTAT,NUM,FILEN)
IF(FILEN.EQ.' ')GO TO 5000
IF(FILE.NE.FILEO)THEN FILEN=
GO TO 4
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 
  END IF 
  IF(LNAM(J).EQ.'BBBBBBB')THEN
    GO TO 105
  END IF
  LN=J
  DO 50 I=2,13,2
    IF(LNUM(LN,1).EQ.0)GO TO 90
    CONN(LNUM(LN,1),(LNUM(LN,1)+1)+1)=LNUM(LN,1)
  50 CONTINUE
  90 CONTINUE
  105 CONTINUE
110 CONTINUE
*********************************************************************
* FILE, JOINT TO GROUND WITH ZERO'S
*********************************************************************
DO 200 I=1,20
IFICONN(I,1).EQ.8)THEN
  IGRD=CONN(I,8)
ENDIF
DO 20 I=1,20
  IF(CONN(I,2).EQ.IGRD)THEN
    CONN(I,2)=0
  ENDIF
  IF(CONN(I,3).EQ.IGRD)THEN
    CONN(I,3)=0
  ENDIF
ENDIF
20 CONTINUE
200 CONTINUE

********************************************************************
* WRITE FIRST LINE OF THE RESTART FILE AND AXIS RANGE          *
********************************************************************
WRITE(16,*)'THIS WILL BE THE IMP FILE'
WRITE(16,*)'CONNECTION ARRAY'
DO 911 I=1,20
  WRITE(16,*)((CONN(I,J),J=1,3))
911 CONTINUE
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
  ELSE IF(CONN(I,1).EQ.0)THEN
    GO TO 2000
  ENDIF
  JNAME=JNAME(I)
  JNUMB=JNUMB(I,1)
  LNAME1=LNAMCONNI,2)
  IF(CONN(I,2).EQ.0)THEN
    LNAME1='FRAME'
  ENDIF
  LNAME2=LNAMCONNI,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
  ENDIF
  WRITE(16,*)'DATA: SCREW',JNAME,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
  ENDIF
  WRITE(16,*)'DATA: GEAR',JNAME,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
  ELSE IF(CONN(I,1).EQ.0)THEN
    GO TO 2000
  ENDIF
  JNAME=JNAME(I)
  JNUMB=JNUMB(I,1)
  LNAME1=LNAMCONNI,2)
  IF(CONN(I,2).EQ.0)THEN
    LNAME1='FRAME'
  ENDIF
  LNAME2=LNAMCONNI,3)
  IF(CONN(I,3).EQ.0)THEN
    LNAME2='FRAME'
  ENDIF
  3000 CONTINUE
ENDIF
LNAME2=LNAM(CONN(I,3))
IF(CONN(I,3).EQ.0)THEN
LNAME2='FRAME'
ENDIF
JTYPE=CONN(I,1)
WRITE(16,'(A)')'DATA: LINK ('''LNAME1','''JNAME,''') = ''$'
& JOIN(JNUMB,1)'','','
& JOIN(JNUMB,3)'','''$'
WRITE(16,'(A)')
& JOIN(JNUMB,1)+JOIN(JNUMB,4)'','','
& JOIN(JNUMB,2)+JOIN(JNUMB,5)'','=''$'
WRITE(16,'(A)')
& JOIN(JNUMB,1)+JOIN(JNUMB,7)'','','
& JOIN(JNUMB,2)+JOIN(JNUMB,8)'','','
& JOIN(JNUMB,3)+JOIN(JNUMB,9)
WRITE(16,'(A)')'DATA: LINK ('''LNAME2','''JNAME,''') = ''$'
WRITE(16,'(A)')
& JOIN(JNUMB,15)'','','
& JOIN(JNUMB,16)'','','
& JOIN(JNUMB,15)+JOIN(JNUMB,14)'',''''$'
WRITE(16,'(A)')
& JOIN(JNUMB,16)+JOIN(JNUMB,19)'','','
& JOIN(JNUMB,17)+JOIN(JNUMB,20)+JOIN(JNUMB,14),''$'
WRITE(16,'(A)')
& JOIN(JNUMB,15)+JOIN(JNUMB,10)'','','
& JOIN(JNUMB,16)+JOIN(JNUMB,11)'','','
& JOIN(JNUMB,17)+JOIN(JNUMB,12)+JOIN(JNUMB,14)
3000 CONTINUE
3100 DO 3500 I=1,20
IFIINAM(I).EQ.'DDDDDD' THEN
WRITE(16,'(A)')'DATA: POSITION(''JNAM(I),'') = ''$'
WRITE(16,'(A)')
& ICI(I,1),ICI(I,2),ICI(I,3)
WRITE(16,'(A)')'DATA: VELOCITY(''JNAM(I),'') = ''$'
WRITE(16,'(A)')
& ICI(I,3)
WRITE(16,'(A)')'DATA: ACCELER (''JNAM(I),'') = ''$'
& ICI(I,4)
ENDIF
3500 CONTINUE
III=III+1
DO 3600 I=1,20
IFI(CONN(I,2).EQ.0.AND.CONN(I,3).EQ.0)THEN
GO TO 3600
ENDIF
IFI(CONN(I,1).EQ.0)THEN
GO TO 4000
ENDIF
LNAME1=LNAM(CONN(I,2))
IF(CONN(I,2).EQ.0)THEN
WRITE(16,'(A)')'STORE: POSITION (''JNAM(I),'')
WRITE(16,'(A)')'
& STORE: VELOCITY (''JNAM(I),'')
WRITE(16,'(A)')'
& STORE: ACCELER (''JNAM(I),'')
GO TO 3600
ENDIF
LNAME2=LNAM(CONN(I,3))
IF(CONN(I,3).EQ.0)THEN
WRITE(16,'(A)')'STORE: POSITION (''JNAM(I),'')
WRITE(16,'(A)')'
& STORE: VELOCITY (''JNAM(I),'')
WRITE(16,'(A)')'
& STORE: ACCELER (''JNAM(I),'')
GO TO 3600
ENDIF
WRITE(16,'(A)')'POINT(''A7,'') = ''PT',I,I,II)
WRITE(16,'(A)')'
& POINT(''A7,'') = ''PT',I,I,II)
WRITE(16,'(A)')'
& POINT(''A7,'') = ''PT',I,I,II')
WRITE(16,'(A)')'
& POINT(''A7,'') = ''PT',I,I,II')
FORMAT 'STORE: VELOCITY (PT',I,I,II')
WRITE(16,'(A)')'
& STORE: ACCELER (PT',I,I,II')
3550 FORMAT 'POINT(''A7,'') = ''PT',I,I,II')
3551 FORMAT 'DATA: POINT (PT',I,I,II'',''A7,'') = 0., 0., 0.,'')
3552 FORMAT 'STORE: POSITION (PT',I,I,II'',''PT',I,I,II',')
3553 FORMAT 'STORE: VELOCITY (PT',I,I,II'',''PT',I,I,II',')
3554 FORMAT 'STORE: ACCELER (PT',I,I,II'',''PT',I,I,II',')

MECHIN PROGRAM LISTING
SUBROUTINE INQ

SUBROUTINE INQ(IHSID,CSIZE,AXISR,PNAM,PNUM,POINT,JOINT)
*********************************************************************
- ----- SUBROUTINE INQ ----- *
* THIS SUBROUTINE IS USED TO INQUIRE POSITIONS *
*********************************************************************

INTEGER IHSID,PRFDRM,PPATH(3),ACLASS(3),CHOICE,PNUM(100),LENG
REAL CSIZE(3),DATA(35),PAREA(6),LI(4),POINT(100,3),AXISR,POSN(2),
& LOCAT(31),BOX(8),SHD(8),LX(2),LY(2),LZ(2),LXII(2),LYII(2),LZII(2),
& JOINT(20,21)
CHARACTER*7 PNAM(100) CHARACTER*16 RETURN
DATA RETURN/'1. RETURN '/ DATA LENG/16/ DATA PDSN/0.05,2.2/
DATA X/ 'X = DATA Y/ 'Y = DATA Z/ 'Z = DATA CLEAR/ '
LX(1)=0.05 LX(2)=0.68
LY(1)=0.05 LY(2)=0.62
LZ(1)=0.05 LZ(2)=0.56
DATA PFORM/0.025,0.025,0.025,0.025,0.025,0.025/ DATA ACLASS/2,3,4/
DATA BOX/0.00,0.545,0.48,0.545,0.48,0.73,0.00,0.73/ DO 10 I=0,8,2 . SHD(I)=BOX(I)-0.025 10 CONTINUE DO 20 I=0,7,2 SHD(I)=BOX(I)+0.025 20 CONTINUE

*********************************************************************
REQUEST PICK INPUT
****** SUBROUTINE INQ ******
C SET PICK FILTER(ALL CLASS 2,3 DETECTABLE)
C CALL GPPKF(IHSID,1,3,ACLASS,0,ACLASS)
C MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
C CALL GPPKMO(IHSID,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)

MECHIN PROGRAM LISTING

119
PAREA(4)=(1.48+1)/2)*CSIZE(2)
PAREA(5)=0
PAREA(6)=CSIZE(3)

C CALL GPINPK(IINSID,1,PPATH,1,PAREA,0,DATA,1) INITIALIZE PICK PATH
C CALL GPPKMO(IINSID,1,3,21) PLACE PICK DEVICE IN THE EVENT MODE
C CALL GPUPHS(IINSID,1) TRAVERSE ALL ACTIVE VIEWS
C CALL GPUPHS(IINSID,1,PERFORM) GET INQUIRY OR RETURN COMMAND
C CALL CMDM(IINSID,22) * INCLUDE A RETURN PICK
***********************************************************************
INCLUDE A RETURN PICK
***********************************************************************

C CALL GPOPST(6)
C CALL GPADCN(ACLASS)
C CALL GPPKID(I)
C CALL GPANS(I,0)
C CALL GPANC(POSN,LENG,RSTR)
C CALL GPCLST
C CALL GPVAR(IINSID,3,6,0)
C CALL GPPV(IINSID,3,i)
C CALL GPUPHS(IINSID,1,PERFORM)

C CALL GPOPST(8)
C CALL GPIS(2)
C CALL GPIC(I,1,4,2,SHD)
C CALL GPIC(I,3)
C CALL GPIC(I,1,4,2,BOX)
C CALL GPCLST
C CALL GPUPHS(IINSID,1,PERFORM)

***********************************************************************
* AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
***********************************************************************

C CALL GPHEV(1000,INS,ICLASS,IDEV)

C IF(ICLASS.EQ.5) THEN ERASE OLD POSITION ICONS
C CALL GPEST(9)
C CALL GPEST(10)
C CALL GPEST(11)
C CALL GPUPST(1,IDEVP)
C CALL GPUPK(IDEP,PPATH)
C IF(PPATH(2).EQ.10) THEN GO TO 1000
C IF (NO PICK, RETURN
C IF (NO PICK, RETURN
C CHOICE IS SECOND ITEM OF PICK PATH
C ISTRUCC=PPATH(1)
C CHOICE=PPATH(2)
C GET LOCATION
C IF (ISTRUCC.EQ.12) THEN
C LOCAT(1)=POINT(CHOICE,1)
C LOCAT(2)=POINT(CHOICE,2)
C LOCAT(3)=POINT(CHOICE,3)
C ELSE
C LOCAT(1)=JOINT(CHOICE,1)
C LOCAT(2)=JOINT(CHOICE,2)
C LOCAT(3)=JOINT(CHOICE,3)
C ENDIF
C REMIND 10
C DO 350 III=1,3
C WRITE(10,300) CLEAR
C FORMAT(A20)
C CONTINUE
C REMIND 10
C WRITE(10,375)LOCAT(1)
C WRITE(10,375)LOCAT(2)
C WRITE(10,375)LOCAT(3)
C FORMAT(F14.6)
C REMIND 10
C READ(10,400)XVAL
C READ(10,400)YVAL
C READ(10,400)ZVAL
C FORMAT(A14)
C DRAM POSITION ICON
C CALL GPOPST(11)
C CALL ICNIC(LOCAT,AXISR)
C CALL GPCLST
C CALL GPOPST(9)
C CALL ICNIC(LOCAT,AXISR)
C CALL GPCLST
C CALL GPOPST(10)
C CALL GPTXC(I,1)
C CALL GPANS(1,0)

MECHIN PROGRAM LISTING 120
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)
REAL CSIZE(3),DATA(35),PAREA(4),L1(14),POINT(100,3),AXISR,POSN(2),
& BOX(1),SID(17),LMAX,LYMAX,LZMAX,LYLMAX,LZLMAX,LZLMAX,
& ORIENT(3),IC(20,4),LX1L2,LX112,LX1112,LXI1,LX112,LX1112
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 PNUM/0.05,2.2/
DATA X/'POSITION'/ DATA Y/'INCREMENT'/ DATA V/'STEPS'/ DATA A/'ACCELERATION'/
DATA CLEAR'/

LX1=0.05
LX11=0.14
LX111=0.26
LY1=0.05
LY11=0.14
LY111=0.26
LZ1=0.05
LZ11=0.14
LZ111=0.26

MECHIN PROGRAM LISTING
LZ11(1)=0.26
LZ11(2)=0.36
DATA DATA/32,0,0,
& 2.2,1.1,
& 1.1.1,1.1.
& 0.48,1.1,1.1,
& 1.1.1,1.1,1.1,1.1
C DATA PRFORM/2/
C DATA ACLASS/3,4,6/
C 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(IHSID,78)
CALL MENU(IHSID,CSIZE,16,CHOICE)
KKK=CHOICE
********************************************************************* *
REQUEST PICK INPUT
*********************************************************************
CALL GPKF(IHSID,1,3,ACLS,0,ACLS)
MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
CALL GPKM(IHSID,1,1,2)
DEFINE PICK AREA
PARE(1)=(-.98+1/2)*CSIZE(1)
PARE(2)=(-.98+1/2)*CSIZE(2)
PARE(3)=(-.98+1/2)*CSIZE(3)
PARE(4)=(-.98+1/2)*CSIZE(2)
PARE(5)=0
PARE(6)=CSIZE(1)
CALL GPNP(IHSID,1,0,PPATH,1,PARE,0,DATA,1)
PLACE PICK DEVICE IN THE EVENT MODE
CALL GPKNO(IHSID,1,3,2)
TRAVERSE ALL ACTIVE VIEWS
CALL GPUPH(IHSID,PRFORM)
GET INQUIRY OR RETURN COMMAND
CALL CMMAND(IHSID,77)
********************************************************************* *
INCLUDE A RETURN PICK
*********************************************************************
C CALL GPWWST(6)
C CALL GPADCN(1,ACLS)
C CALL GPKK(IHSID,101)
C CALL GPXCL(1)
C CALL GPHSCC(78)
C CALL GPANZ(PSN,LEN,RET)
C CALL GPCLST
C CALL GPARY(IHSID,1,3,6,0)
C CALL GPVP(IHSID,1,1,1)
C CALL GVPMSI(IHSID,PRFORM)
C CALL GPWWST(8)
C CALL GPIS(2)
C CALL GPES(1,4,2,SHD)
C CALL GPTC(3)
C CALL GPP(1,4,2,BOX)
C CALL GPCLST
C CALL GVPMSI(IHSID,PRFORM)
********************************************************************* *
AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
*********************************************************************
200 CALL GPEHEV(1000,1,HS,ICLA,DEV)
 IF(ICALA.EQ.5) THEN
  CALL GPEVST(10)
  ERASE OLD INQ.
  CALL GPGRPK(1,IDEV,PPATH)
  IF NO PICK, RETURN
MECHIN PROGRAM LISTING 122
IF (PATH(2).EQ.10) GO TO 1000
CHOICE=PATH(2)
STRU=PATH(1)

IF (KKK.EQ.1) THEN
   POS=ICICHOICE,1
   VEL=ICICHOICE,2
   INC=INCUMCHOICE + 2
   REWIND 10
   WRITE(10,300) CLEAR
   FORMAT(A20)
   CONTINUE
   WRITE(10,375) POS
   WRITE(10,375) VEL
   WRITE(10,375) INC
   FORMAT(F8.3)
   REWIND 10
   READ(10,400) XVAL
   READ(10,400) YVAL
   READ(10,400) ZVAL
   400 FORMAT(8I)
   CALL GPOPST(10)
   CALL GPXCI(1)
   CALL GPASC(1.0)
   CALL GPAN2(LX,20,X)
   CALL GPAN2(LXI,8,XVAL)
   CALL GPAN2(LY,20,Y)
   CALL GPAN2(LYI,8,YVAL)
   CALL GPAN2(LZ,20,Z)
   CALL GPAN2(LZI,8,ZVAL)
   CALL GPCLST
   CALL GPUPHS(ISID,PRFORM)
   GO TO 200
ENDIF

IF (KKK.EQ.2) THEN
   VEL=ICICHOICE,3
   REWIND 10
   WRITE(10,450) CLEAR
   WRITE(10,450) VEL
   FORMAT(F4.3)
   REWIND 10
   READ(10,500) XVAL
   500 FORMAT(14)
   CALL GPOPST(10)
   CALL GPXCI(1)
   CALL GPASC(1.0)
   CALL GPAN2(LX,20, V)
   CALL GPAN2(LYI,14,XVAL)
   CALL GPCLST
   CALL GPUPHS(ISID,PRFORM)
   GO TO 200
ENDIF

IF (KKK.EQ.3) THEN
   ACC=ICICHOICE,4
   REWIND 10
   WRITE(10,550) CLEAR
   WRITE(10,550) ACC
   FORMAT(F4.3)
   REWIND 10
   READ(10,600) XVAL
   600 FORMAT(14)
   CALL GPOPST(10)
   CALL GPXCI(1)
   CALL GPASC(1.0)
   CALL GPAN2(LX,20, A)
   CALL GPAN2(LYI,14,XVAL)
   CALL GPCLST
   CALL GPUPHS(ISID,PRFORM)
   GO TO 200
ENDIF

ELSE
   IF NOT A PICK, GO TO AWAIT AN EVENT
ENDIF

GO TO 1000
CALL GPPKMO(ISID,1,1,2)
CALL GPST(10)

MECHIN PROGRAM LISTING
CALL GPESTI(8)
CALL GPESTI(16)
CALL GPESTI(10)
CALL GPPKH(ISID,PRFORM)

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

RETURN
END

SUBROUTINE INQL

SUBROUTINE INQL(ISID,CSIZE,AXISR,LNAM,LNUM,JNAM)

*********************************************************************
* ----- SUBROUTINE INQL ----- *
*********************************************************************

** THIS SUBROUTINE IS USED TO INQUIRE JOINT ORIENATIONS **

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

INTEGER ISID,PRFORM,PPATH,ACLASS(1),CHOICE,LENG,STRU,
& LNUM(30,14)
REAL CSIZE(3),DATE(35),PAREA(L),PIST(100,3),AXISR,POSN(2),
& POX(8),SHD(8),LXI(2),LXI(2),LXI(2),LXI(2),LXI(2)
CHARACTER J NAME(100),LNAM(30),LNAME,JNAME
CHARACTER RETURN)

C SET PICK FILTER(ALL CLASS 2,3 DETECTABLE)
CALL GPPKF(ISID,1,0,ACLASS,ACLASS)
C MAKE PICK IS DEACTIVATED REQUEST MODE)
CALL GPPKM(ISID)
C DEFINE PICK AREA
PAREA(1)=(-.98+1/2)*CSIZE(1)
PAREA(2)=(-.98+1/2)*CSIZE(1)
PAREA(3)=(-.98+1/2)*CSIZE(1)
PAREA(4)=(-.98+1/2)*CSIZE(2)
PAREA(5)=0
PAREA(6)=CSIZE(3)
C INITIALIZE PICK PATH
CALL GPPKP(ISID,1,1,PPATH,PAREA,DATA,1)
C PLACE PICK DEVICE IN THE EVENT MODE

MECHN PROGRAM LISTING
SUBROUTINE INQO

SUBROUTINE INQO(IHSID, CSIZE, AXISR, JOINT)

********************************************************************
* ----- SUBROUTINE INQO ----- *
* * *
* THIS SUBROUTINE IS USED TO INQUIRE JOINT ORIENTATIONS *
* * *
********************************************************************

RETURN

MECHIN PROGRAM LISTING
INTEGER IMSID,PRFORM,PPATH(3),ACLASS(2),CHOICE,PNAME(100),LENG,STRU
REAL CSIZE(3),DATA(35),AREA(6),L(11),POINT(100,3),AXIS,POSN(2),&
BOX(8),SHD(16),LX(2),LY(2),LZ(2),LX1(2),LY1(2),LZ1(2),&
JOINT(20,21),ORIENT(3)
CHARACTER*7 PNAME(100)
CHARACTER*16 RETURN
CHARACTER*14 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
LY(1)=0.05
LY(2)=0.68
LZ(1)=0.05
LZ(2)=0.56
LZ1(1)=0.14
LZ1(2)=0.56
DATA DATA/32,0,0,
&
&
&
&
&
C DATA PRFORM/2/
C DATA ACLASS/3,4/
C 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 CALL GPKF(IMSID,1,2,ACLASS,0,ACLASS)
C CALL GPPKMO(IMSID,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(6)=CSIZE(3)
C CALL GPNPK(IMSID,1,0,PPATH,1,PAREA,0,DATA,1)
C PLACE PICK DEVICE IN THE EVENT MODE
C CALL GPPKMO(IMSID,1,3,2)
C TRAVERSE ALL ACTIVE VIEWS
C CALL GPUPH(IMSID,PRFORM)
C GET INQUIRY OR RETURN COMMAND
C CALL CMMAND(IMSID,37)
*********************************************************************
* INCLUDE A RETURN PICK
*********************************************************************
C CALL GPOPST(6)
C RETURN PICK
C CALL GPDICN(1,ACLASS)
C CALL GPPKID(101)
C CALL GPTXCI(2)
C CALL GPAHSCI(1,0)
C CALL GPAN(POSN,LENG,RETURN)

MECHIN PROGRAM LISTING
CALL GPCLST
CALL GPARV(INSID, 3, 6, 0.)
CALL GPPUI(INSID, 1, 1)
CALL GPUPWH(INSID, PFORM)
CALL GPOPST(8)
CALL GPISt(2)
CALL GPICt(4)  
CALL GPVPCIHSID, PFORM
CALL GPUPHSIIHSID, PFORM
CALL GPOPST(16)
CALL GPESTI(16)
CALL GPESTI(10)
CALL GPESTI(6)

GREY VALUE BOX AND SHADOW

*********************************************************************
* AwaIT A PICK EVENT TO OCCUR AND RETRIEVE IT
***************************************************************************

C AwaIT AN EVENT

200 CALL GPAMEV(1000, IMS, ICLA, IDEV)
IF(ICALA .EQ. 5) THEN
   CALL GPESTI(9)
   CALL GPESTI(10)
   CALL GPESTI(16)
   ELSE
   GET PICK
C IF NO PICK, RETURN
C IF(IPPATH(2).EQ.101) GO TO 1000
   CHOICE = IPPATH(2)
   STRU = IPPATH(1)
   IF(STRU .EQ. 18) THEN
      CHOICE = IPPATH(1)
      END IF
   ORIENT1 = JOINTICHOICE, 4
   ORIENT2 = JOINTICHOICE, 5
   ORIENT3 = JOINTICHOICE, 6
   END IF
   IF(ISTR. EQ. 18) THEN
      ORIENT1 = JOINTICHOICE, 18
      ORIENT2 = JOINTICHOICE, 19
      ORIENT3 = JOINTICHOICE, 20
   ELSE
      ORIENT1 = JOINTICHOICE, 9
      ORIENT2 = JOINTICHOICE, 10
      ORIENT3 = JOINTICHOICE, 11
   ENDIF
   REmIND 10
   DO 350 III = 1, 5
      WRITE(10, 300) CLEAR
      Format=(20)
      END
   300 CONTINUE
   REmIND 10
   WRITE(10, 375) ORIENT(1)
   WRITE(10, 375) ORIENT(2)
   WRITE(10, 375) ORIENT(3)
   Format=(14, 6)
   REmIND 10
   READ(10, 400) XVAL
   READ(10, 400) YVAL
   READ(10, 400) ZVAL
   READ(10, 400) IZVAL
   READ(10, 400) II
   END
   WRITE(10, 320)
   Format=(20)
   320 CONTINUE
   WRITE(10, 375) ORIENT(1)
   WRITE(10, 375) ORIENT(2)
   WRITE(10, 375) ORIENT(3)
   Format=(14, 6)
   END
   WRITE(10, 320)
   Format=(20)

C DRAW POSITION ICON

CALL GPOPST(16)
CALL ICONO(ORIENT, AXISR)
CALL GPCLST
CALL GPOPST(10)
CALL GPTXCt(1)
CALL GPAHSCI(1.0)
CALL GPANZILX, 20, XVAL
CALL GPANZILX, 14, XVAL
CALL GPANZILY, 20, YVAL
CALL GPANZILY, 14, YVAL
CALL GPANZILZ, 20, ZVAL
CALL GPANZILZ, 14, ZVAL
CALL GPCLST
CALL GPUPWH(INSID, PFORM)
C GO TO 200
CALL GPPKMO(INSID, 1, 1, 2)
CALL GPESTI(6)
CALL GPESTI(8)
CALL GPESTI(16)
CALL GPESTI(10)
CALL GPUPWH(INSID, PFORM)
C ELSE
GO TO 200
C IF NOT A PICK, GO TO AwaIT AN EVENT
ENDIF
C MAKE SURE PICK IS DEACTIVATED
1000 CALL GPPKMO(INSID, 1, 1, 2)
CALL GPESTI(6)
CALL GPESTI(8)
CALL GPESTI(16)
CALL GPESTI(10)
CALL GPUPWH(INSID, PFORM)
SUBROUTINE INVJ

SUBROUTINE INVJ(IHSID, JNAM, JNUM, JOINT, AXISR, SCALE, CSIZE)

*********************************************************************
* THIS SUBROUTINE IS USED TO MAKE JOINTS INVISIBLE                   *
*********************************************************************

INTEGER IHSID, NUMP, JNUM(20, 3), JTYPE, END
REAL LOCAT(3), JOINT(20, 21), AXISR, ORIENT(3), ENDP(3), CSIZE(3)
REAL REV(78), PRI(30), SPH(132), MATPO(4, 4)

CALL CMMAND(IHSID, 86) CALL PICKJ(IHSID, JOINT, AXISR, CSIZE, END, ENDP, IAXIS, IR)
JNUM = 1 IF(IR .EQ. 1) GO TO 1000
CALL RDRWJT(IHSID, JNAM, JNUM, JOINT, AXISR, SCALE)
1000 RETURN

END

SUBROUTINE INVL

SUBROUTINE INVL(IHSID, JNAM, JNUM, JOINT, AXISR, LNAM, LNUM, LINK,

*********************************************************************
* THIS SUBROUTINE IS USED TO MAKE LINKS INVISIBLE                   *
*********************************************************************

INTEGER IHSID, NUMP, JNUM(20, 3), JTYPE, END, LNUM(30, 14)
REAL LOCAT(3), JOINT(20, 21), AXISR, ORIENT(3), ENDP(3), CSIZE(3),
& LINK(30, 6)
REAL REV(78), PRI(30), SPH(132), MATPO(4, 4)

CALL CMMAND(IHSID, 87) CALL PICKL(IHSID, CSIZE, AXISR, LNAM, LNUM, JNAM, END, IR)
LNUM = 14 IF(IR .EQ. 1) GO TO 1000
CALL RDRWLK(IHSID, JNAM, JNUM, JOINT, AXISR, LNAM, LNUM, LINK, SCALE)
1000 RETURN

END

SUBROUTINE JICONS

SUBROUTINE JICONS(LOCAT, ORIENT, TYPE, AXISR, JOINT, NUMJ, SCALE, MAT)

*********************************************************************
* THIS SUBROUTINE IS USED TO DISPLAY THE ICONS FOR THE JOINTS        *
*********************************************************************

INTEGER TYPE
REAL LOCAT(3), ORIENT(3), AXISR, REV(78), PRI(30), SPH(132), MATPO(4, 4),

CALL GPPL3(10,3,PRIS)
ENDIF

IF(TYPE.EQ.3)THEN
  CALL GPPLC1(5)
  CALL GPLT(1)
  CALL GPPL3(26,3,CLDJ)
ENDIF

IF(TYPE.EQ.4)THEN
  CALL GPPLC1(5)
  CALL GPLT(1)
  CALL GPPL3(44,3,SPH)
ENDIF

IF(TYPE.EQ.5)THEN
  CALL GPPLC1(5)
  CALL GPLT(1)
  CALL GPPL3(26,3,CLDJ)
ENDIF

IF(TYPE.EQ.6)THEN
  CALL GPPLC1(5)
  CALL GPLT(1)
  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,4)
  P(5)=JOINT(NUMJ,5)
  P(6)=JOINT(NUMJ,6)
  DIST=(P(1)-P(4))*(P(2)-P(5))*(P(3)-P(6))***.5
  R=JOINT(NUMJ,21)
  R2=DIST*12/(R)**.5
  R1=R*DIST*12/(R)**.5
  DO 10 I=1,3,1
    GEAR(I)=GEAR(I)*R1
    GEAR(I+1)=GEAR(I+1)*R1
    GEAR(I+2)=GEAR(I+2)*R1
  CONTINUE
  DO 20 I=1,3,1
    GEAR(I)=GEAR(I)*R2
    GEAR(I+1)=GEAR(I+1)*R2
    GEAR(I+2)=GEAR(I+2)*R2
  CONTINUE
CALL GPMLX3(MAT,3)
CALL GPPL3(13,3,GEAR)
MAT(1,4)=P(4)
MAT(2,4)=P(5)
MAT(3,4)=P(6)
CALL GPMLX3(MAT,3)
CALL GPPL3(13,3,GEAR)
ENDIF

IF(TYPE.EQ.8)THEN
  X=.125
  GND(1)=X
  GND(2)=-.009
  GND(3)=.0
  DO 100 I=4,39,1
    GND(I)=GND(I-.02)
    GND(I+1)=-.029
    GND(I+2)=0.0
    GND(I+3)=X
    GND(I+4)=-.009
    GND(I+5)=0.0
    GND(I+6)=X-.05
    GND(I+7)=-.009
    GND(I+8)=.0
  CONTINUE
100 CONTINUE
SUBROUTINE JOINTS

SUBROUTINE JOINTS(IMSID,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 IMSID,CHOICE,PNUM(100),JNUM(20)
CHARACTER*7 PNAM(100),JNAM(20)
******************************************************************************
C REQUEST TYPE OF DATA INPUT
10 CALL CMAND(IMSID,10)
   CALL MENU(IMSID,CSIZE,3,CHOICE)
   IF(CHOICE.EQ.1) THEN REQUEST ADD A JOINT
      CALL ADDJ(IMSID,CSIZE,AXISR,JNAM,JNUM,JOINT,POINT,SCALE)
   ENDIF
   IF(CHOICE.EQ.2) THEN REQUEST DELETE A JOINT
      CALL DELJ(IMSID,CSIZE,AXISR,JNAM,JNUM,JOINT,SCALE)
   ENDIF
   IF(CHOICE.EQ.3) THEN REQUEST ADD A POINT
      CALL ADDP(IMSID,CSIZE,AXISR,PNAM,PNUM,POINT)
   ENDIF
   IF(CHOICE.EQ.4) THEN REQUEST DELETE A JOINT
      CALL DELP(IMSID,CSIZE,AXISR,PNAM,PNUM,POINT)
   ENDIF
   IF(CHOICE.EQ.5) THEN INQUIRE A POSITION
      CALL INQJ(IMSID,CSIZE,AXISR,PNAM,PNUM,POINT,JOINT)
   ENDIF
   IF(CHOICE.EQ.6) THEN INQUIRE AN ORIENTATION
      CALL INQO(IMSID,CSIZE,AXISR,JOINT)
   ENDIF
   IF(CHOICE.EQ.7) THEN RETURN TO JOINT, LINK, OR IC MENU
   ENDIF
RETURN
END

SUBROUTINE LINKS

SUBROUTINE LINKS(IMSID,AXISR,CSIZE,JNAM,JNUM,JOINT,PNAM,PNUM, &POINT, LNAM,LNUM,LINK,SCALE)
******************************************************************************
* ----- SUBROUTINE LINK ----- *
******************************************************************************
* THIS SUBROUTINE IS USED TO CREATE LINK DATA *
******************************************************************************
**SUBROUTINE MENU**

SUBROUTINE MENU(IHSID, CSIZE, NUM, CHOICE)

This subroutine is used to display menu items in the menu box and process the menu picks.

INTEGER IHSID, 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 DATA LENG/19/
DATA TMENU(1,1)/' TITLE TEXT, LENGTH '/
DATA TMENU(1,10)/1. NEW MODEL '/
DATA TMENU(1,12)/1. RESTART MODEL '/
DATA TMENU(1,13)/' /
DATA TMENU(1,14)/' /
DATA TMENU(1,15)/' /
DATA TMENU(1,16)/' /
DATA TMENU(1,17)/' /
DATA TMENU(1,18)/' /
DATA TMENU(2,1)/1. DEFINE '/
DATA TMENU(2,10)/1. JOINTS '/

MECHIN PROGRAM LISTING 132
DATA TMENU(2,2) /'2. DEFINE LINKS /
DATA TMENU(2,3) /'3. ENTER INITIAL CONDITIONS /
DATA TMENU(2,4) /'4. CHANGE VIEW /
DATA TMENU(2,5) /
DATA TMENU(2,6) /'5. END OF DATA /
DATA TMENU(2,15) /'INPUT /
DATA TMENU(3,1) /'1. ADD JOINT /
DATA TMENU(3,2) /'2. DELETE JOINT /
DATA TMENU(3,3) /'3. ADD POINT /
DATA TMENU(3,4) /'4. DELETE POINT /
DATA TMENU(3,5) /'5. INQUIRE POSITION /
DATA TMENU(3,16) /'7. RETURN /
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(4,3) /
DATA TMENU(4,11) /
DATA TMENU(4,12) /
DATA TMENU(4,4) /
DATA TMENU(4,7) /
DATA TMENU(4,13) /
DATA TMENU(4,8) /
DATA TMENU(4,14) /
DATA TMENU(4,9) /
DATA TMENU(4,15) /
DATA TMENU(4,16) /
DATA TMENU(5,1) /'1. ENTER POSITION /
DATA TMENU(5,10) /
DATA TMENU(5,2) /'2. RETURN /
DATA TMENU(5,3) /
DATA TMENU(5,11) /
DATA TMENU(5,4) /
DATA TMENU(5,5) /
DATA TMENU(5,12) /
DATA TMENU(5,6) /
DATA TMENU(5,13) /
DATA TMENU(5,7) /
DATA TMENU(5,14) /
DATA TMENU(5,8) /
DATA TMENU(5,15) /
DATA TMENU(5,16) /
DATA TMENU(5,9) /
DATA TMENU(5,17) /
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 /'1. ENTER '/ DATA TMENU 7,11 /' ORIENTATION '/
DATA TMENU 7,12 /'2. RETURN '/
DATA TMENU 7,13 /'2. SET COORDS '/
DATA TMENU 7,14 /'3. TOWARDS A JOINT '/ DATA TMENU 7,15 /' OR POINT '/
DATA TMENU 7,16 /'3. TOWARDS A JOINT '/ DATA TMENU 7,17 /' OR POINT '/
DATA TMENU 7,18 /'4. PARALLEL TO AN '/ DATA TMENU 7,19 /' EXISTING JOINT '/
DATA TMENU 7,20 /'5. RETURN '/
DATA TMENU 8,1 /'1. ENTER '/ DATA TMENU 8,2 /' ORIENTATION '/
DATA TMENU 8,3 /'2. RETURN '/
DATA TMENU 8,4 /'2. RETURN '/
DATA TMENU 8,5 /'2. RETURN '/
DATA TMENU 8,6 /'2. RETURN '/
DATA TMENU 8,7 /'2. RETURN '/
DATA TMENU 8,8 /'2. RETURN '/
DATA TMENU 9,1 /'1. SAME AXIS '/
DATA TMENU 9,2 /'1. SAME AXIS '/
DATA TMENU 9,3 /'2. ENTER NEW AXIS '/
DATA TMENU 9,4 /'2. ENTER NEW AXIS '/
DATA TMENU 9,5 /'2. ENTER NEW AXIS '/
DATA TMENU 9,6 /'2. ENTER NEW AXIS '/
DATA TMENU 9,7 /'2. ENTER NEW AXIS '/
DATA TMENU 9,8 /'2. ENTER NEW AXIS '/
DATA TMENU 9,9 /'2. ENTER NEW AXIS '/
DATA TMENU 10,1 /'1. SAME CENTER '/
DATA TMENU 10,2 /'1. SAME CENTER '/
DATA TMENU 10,3 /'2. ENTER CENTER '/
DATA TMENU 10,4 /'2. ENTER CENTER '/
DATA TMENU 10,5 /'2. ENTER CENTER '/
DATA TMENU 10,6 /'2. ENTER CENTER '/
DATA TMENU 10,7 /'2. ENTER CENTER '/
DATA TMENU 10,8 /'2. ENTER CENTER '/
DATA TMENU 10,9 /'2. ENTER CENTER '/
DATA TMENU 10,10 /'2. ENTER CENTER '/
DATA TMENU 11,1 /'1. ADD LINK '/
DATA TMENU 11,2 /'1. ADD LINK '/
DATA TMENU 11,3 /'2. DELETE LINK '/
DATA TMENU 11,4 /'2. DELETE LINK '/
DATA TMENU 11,5 /'3. INQUIRE LINK '/
DATA TMENU 11,6 /'3. INQUIRE LINK '/
DATA TMENU 11,7 /'4. RETURN '/
DATA TMENU 11,8 /'4. RETURN '/
DATA TMENU 12,1 /'1. TOWARDS AN MECHANICAL PROGRAM LISTING
| DATA TMENU(12,10) | 1. ZOOM IN |
| DATA TMENU(12,11) | 2. RESTORE |
| DATA TMENU(12,12) | 3. CHANGE JOINT |
| DATA TMENU(12,13) | 4. JOINT INVISIBLE |
| DATA TMENU(12,14) | 5. LINK INVISIBLE |
| DATA TMENU(12,15) | 6. RESTORE |
| DATA TMENU(12,16) | 7. RETURN |
| DATA TMENU(12,17) | 8. RESTORE |
| DATA TMENU(12,18) | 9. RETURN |
| DATA TMENU(13,1) | 10. ENTER I.C. |
| DATA TMENU(13,2) | 11. RETURN |
| DATA TMENU(13,3) | 12. DELETE I.C. |
| DATA TMENU(13,4) | 13. INQUIRE I.C. |
| DATA TMENU(13,5) | 14. RETURN |
| DATA TMENU(14,1) | 15. POSITION |
| DATA TMENU(14,2) | 16. VELOCITY |
| DATA TMENU(14,3) | 17. ACCELERATION |

MECHIN PROGRAM LISTING
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(17,5) = '/
DATA TMENU(17,6) = '/
DATA TMENU(17,7) = '/
DATA TMENU(17,8) = '/
DATA TMENU(17,9) = '/
DATA TMENU(17,10) = '/
DATA TMENU(18,1) = '/ 1. IMP /
DATA TMENU(18,2) = '/ 2. RSCR /
DATA TMENU(18,3) = '/ 3. RSSR /
DATA TMENU(18,4) = '/ 4. WRITE NO /
DATA TMENU(18,5) = '/ ANALYSIS FILE /
DATA TMENU(18,6) = '/
DATA TMENU(18,7) = '/
DATA TMENU(18,8) = '/
DATA TMENU(18,9) = '/
DATA TMENU(18,10) = '/
DATA TMENU(18,11) = '/
DATA TMENU(18,12) = '/
DATA TMENU(18,13) = '/
DATA TMENU(18,14) = '/
DATA TMENU(18,15) = '/
DATA TMENU(18,16) = '/
DATA TMENU(18,17) = '/
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/ 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 GPESTI(6)
CALL GPOPSTI(6)
CALL GPADCNI,ACLASS
CALL GPXCIIZ
CALL GPAHSC1(0)
DO 100 I=1,9
CALL GPKID(I)
CALL GPAZI(POSN,LENG,TMENU(NUM,I))
POSN(2)=POSN(2)-.08
CALL GPAZI(POSN,LENG,TMENU(NUM,I+9))
POSN(2)=POSN(2)-.18
100 CONTINUE
CALL GPCLST

******************************************************************************
* SCREEN DISPLAY
******************************************************************************
CALL GPARV(INSID,3,6,0.)
CALL GPPV(INSID,3,1,1)
CALL GPPW(INSID,PRFORM)

******************************************************************************
* REQUEST PICK INPUT
******************************************************************************

MECHIN PROGRAM LISTING

136
**SUBROUTINE NSTART**

SUBROUTINE NSTART(IHSID, FILE, CSIZE, AXISR, POINT, JOINT, LINK, IC, &PNUM, JNUM, LNUM, INUM)

* * * *
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 IHSID, PNUM(100), JNUM(20,3), LNUM(30,14), INUM(20,2), &DATA(35), LENG, ACLASS(11)
REAL AREA(6), SIZE(3), AXISR, JOINT(20,21), LINK(30,6), IC(20,4), &POINT(100,3), POSN(2), PAREA(6)
CHARACTER*7 FILE, AXIS, CLEAR, FILE, INIT
CHARACTER*19 LINE
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

INCLUSION CLASS LIST
DATA ACLASS/2/
DATA INIT/'1.0'/
DATA AXES:'/'
DATA CLEAR:'/'
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(AALL CLASS 2 DETECTABLE)
C MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
C
CALL GPPKF(IHSID,1,1,AClass,0,AClass)
C DEFINE PICK AREA
PAREA(1)=(-0.98+1/2)*CSIZE(1)
PAREA(2)=(-0.98+1/2)*CSIZE(2)
PAREA(3)=(-0.98+1/2)*CSIZE(3)
PAREA(4)=(-0.98+1/2)*CSIZE(4)
PAREA(5)=0.
PAREA(6)=CSIZE(3)
C INITIALIZE PICK PATH
C CALL GPnPK(IHSID,1,0,PPath,l,PAREA,0,DATA,1)
C PLACE PICK DEVICE IN THE EVENT MODE
C CALL GPPKMO(IHSID,1,3,2)
C Traverse ALL ACTIVE VIEWS
C
*********************************************************************
* INCLUDE A RETURN PICK
*********************************************************************
C CALL GPnPST(6)
C RETURN PICK
CALL GPADCN(6,AClass)
C CALL GPPKID(IHSID,3,0)
C CALL GPVPI(IHSID,5,0)
C
*********************************************************************
* INITIATE THE STRING DEVICE.
*********************************************************************
C CALL GPSTMO(IHSID,1,1,2)
C INITIALIZE THE STRING DEVICE
C CALL GPSTI(IHSID,1,7,Clear,1,Area,7,0,Clear)
C PLACE STRING DEVICE IN THE EVENT MODE
C CALL GPPSTO(IHSID,1,2)
C REQUEST THE FILE NAME
C
*********************************************************************
* GO TO CLOSE IF QUIT IS PICKED
*********************************************************************
C WAIT AN EVENT
3 CALL GPDAST(I000.,IHS,ICLA,IODEV)
IF(ICLA.EQ.5)THEN
   CALL GPADCN(3,0)
   CLOSE FOR RESTART
ENDIF

*********************************************************************
* PROCESS STRING EVENT. OPEN MECHIN FILE
*********************************************************************
L=7
C GET STRING
C CALL GPGST(L,LR,FILE)
IF(FILE.EQ.' ')GO TO 3
C SEE IF FILE ALREADY EXISTS
C IF(FILE.EQ.0)THEN
   CALL GPPSTO(IHSID,1,1,2)
   FILE=FILE
   FILE=Clear

MECHIN PROGRAM LISTING 138
CALL CMAND(IWSID,8)
CALL GPSTMO(IWSID,1,1,2)
CALL GPINST(IWSID,1,7,CLEAR,1,AREA,7,1,0,CLEAR)
CALL GPQST(IWSID,1,7,ISTAT,INUM,FILE1)
IF(FILE.EQ.FILEO)THEN
  CONTINUE
ELSE
  CALL GPINST(IWSID,1,7,CLEAR,1,AREA,7,1,0,CLEAR)
  CALL GPSTMO(IWSID,1,5,2)
ENDIF
GO TO 5
END
C
OPEN A NEW FILE
OPENIUNIT=15,FILE=FILE,STATUS='NEW',ERR=2l
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
INUM(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
NUM(I,J)=0
23 CONTINUE
24 CONTINUE

********************************************************************* 
* WRITE FILE NAME TO THE SCREEN                                                                                         
*********************************************************************
C
CALL FILENAME(IWSID,FILE)
********************************************************************* 
* REQUEST THE COORDINATE AXIS RANGE                                                                                     
*********************************************************************
C
500 CALL CMAND(IWSID,5)
CALL GPSTMO(IWSID,1,1,2)
C
CALL GPINST(IWSID,1,7,INIT,1,AREA,7,1,0,CLEAR)
C
REQUEST STRING FOR AXIS
CALL GPQST(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 INSID,1,0,ACLASS,1,ACLASS)
RETURN
END

SUBROUTINE ORIENT(INSID, JPOSN, POINT, JOINT, AXISR, CSIZE, ORIENT, RFLG)

*******************************************************************************
* SUBROUTINE ORIENT |----------------------------------------------------------*
* THIS SUBROUTINE IS USED TO ORIENT A VECTOR IN SPACE |---------------------*
*******************************************************************************

INTEGER INSID, ECHO, EVENT, DLEN, ECHO, ACLASS, PRFORM, CHOICE,
& PATH, RFLG, NUMV
REAL CSIZE, AREA1, AREA2, AREA3, VLOUE, VIVAL, AXISR,
& BOX, SHDI, ORIENT, MATRIX, LOVAL, HIVAL, AREA, LINE, BOX, SHDI
CHARACTER*7 CLEAR, INIT

DATA CLEAR / 'O. '/
DATA INIT / 'O. '/
DATA LX / 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 /
DATA LY / 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 /
DATA LZ / 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 /

C FLAG FOR UPDATE
C
DATA PRFORM / 2 /
C
DATA ACLASS / 1 /
DATA VLOUE / 0. /
DATA ECHO / 3 /
DATA DLEN / 0 /
DATA EVENT / 3 /
DATA ECHO / 2 /
C
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
AREA1 = .08 * CSIZE
AREA2 = .90 * CSIZE
AREA3 = .70 * CSIZE
AREA4 = .90 * CSIZE
AREA5 = .02 * CSIZE
AREA6 = .90 * CSIZE
AREA7 = .52 * CSIZE
AREA8 = .52 * CSIZE
AREA9 = .855 * CSIZE - .006
AREA10 = .855 * CSIZE
AREA11 = .855 * CSIZE
AREA12 = .855 * CSIZE
AREA13 = .52 * CSIZE
AREA14 = .52 * CSIZE
ARENA15 = .52 * CSIZE
AREA16 = .52 * CSIZE

DO 100 I = 2, 8, 2
SHDI = BOX(I) - 0.025
100 CONTINUE
DO 200 I = 1, 7, 2
SHDI = BOX(I) + 0.025
200 CONTINUE
OPEN STRUCTURE FOR THE MENU BOX (VIEW 1)

OPEN A STRUCTURE
CALL GPOPST(8)
CALL GPIS(2)
CALL GPIC(4)
CALL GPPG(1,4,2,SHD)
CALL GPIC(3)
CALL GPPG(1,4,2,BOX)
CLOSE STRUCTURE

SET SOLID INTERIOR STYLE FOR GREY BOX
CALL GPISl2l
SET COLOR FOR INTERIOR OF SHADOW BOX
CALL GPICil4l
DRAl'l GREY SHADOW BOX
CALL GPPG2(1,4,2,SHDJ
SET COLOR FOR INTERIOR OF GREY BOX
CALL GPICI13l
DRAW GREY TITLE BOX
CALL GPPG2(1,4,2,BOX)
CLOSE STRUCTURE

UPDATE THE WORKSTATION
CALL GUPWS(IHSID,PRFORM)

CHOOSE METHOD OF JOINT ORIENTATION

DIRECT ENTRY
CALL DEOIIl'ISID,AXISR,CSIZE,ORIENT,RFLG)
IF(RFLG.EQ.1)THEN REENTER POINT
ORIENT(1)=0.
ORIENT(2)=0.
ORIENT(3)=0.
RFLG=0.
GO TO 300
ENDIF

VALUATOR ENTRY
CALL VEOIIWSID,AXISR,CSIZE,ORIENT,RFLG)
IF(RFLG.EQ.1)THEN REENTER POINT
ORIENT(1)=0.
ORIENT(2)=0.
ORIENT(3)=0.
RFLG=0.
GO TO 300
ENDIF

TOWARDS AN EXISTING POINT
CALL TEPOIIHSID2 JPOSN,POINT,JOINT,AXISR,CSIZE,ORIENT,RFLG)
IF(RFLG.EQ.1)THEN REENTER POINT
ORIENT(1)=0.
ORIENT(2)=0.
ORIENT(3)=0.
RFLG=0.
GO TO 300
ENDIF

PARALLEL WITH AN EXISTING JOINT
CALL PEJOIIHSID2 JPOSN,JOINT,AXISR,CSIZE,ORIENT,RFLG)
IF(RFLG.EQ.1)THEN REENTER POINT
ORIENT(1)=0.
ORIENT(2)=0.
ORIENT(3)=0.
RFLG=0.
GO TO 300
ENDIF

RETURN WITH NO ORIENTATION
RFLG=1
ENDIF

MECHIN PROGRAM LISTING
CALL GPEST(6)
CALL GPUPHS(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),
&PI(3),P1(3),VEC(3),MAG,LOCAT(3),JPOSN(3),
&LX(2),LY(2),LZ(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,
C DATA PRFORM/2/  INCLUSION CLASS LIST(CLASS 3 -- JOINTS)
C DATA ACLASS/3,4/
DATA X '/DX=/
DATA Y '/DY=/
DATA Z '/DZ=/
DATA CLEAR '/

LX(1)=0.05
LX(2)=0.08
LY(1)=0.14
LY(2)=0.05
LZ(1)=0.05
LZ(2)=0.05
LZ(1)=0.14
LZ(2)=0.06

*****************************************************************************
* REQUEST PICK INPUT *
*****************************************************************************

C CALL GPKF(IWSID,1,2,ACLASS,0,ACLASS)  MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
C CALL GPKPMD(IWSID,1,1,2)  INITIALIZE PICK AREA
PAREA(1)=(-.98+1/2)*CSIZE(1)
PAREA(2)=(.98+1/2)*CSIZE(1)
PAREA(3)=(-.98+1/2)*CSIZE(1)
PAREA(4)=(.98+1/2)*CSIZE(1)
PAREA(5)=0
PAREA(6)=CSIZE(3)
C CALL GPINPK(IWSID,1,0,PPATH,1,PAREA,0,DATA,1)  PLACE PICK DEVICE IN THE EVENT MODE
CALL GPPKMO(IMSID,1,3,2)
CALL GPUPWS(IMSID,PRFORM)
200 CALL COMMAND(IMSID,35)

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

CALL GPOPST(6)
CALL GPADCNCl,ACLASSl CALL GPPKIDClOll CALL GPTXCIC2l CALL GPAHSCCl.Ol
RETURN

CALL GPAN2CPOSN,LENG,RETURN
CALL GPCLST CALL GPARV(INSID,3,6,0.)
CALL GPUPW(INSID,8,1,1)
CALL GPUPWS(INSID,PRFORM)

******************************************************************************
* WAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
******************************************************************************

CALL GPESTC
CHOICE=PPATH(2)
ISTRUC=PPATH(1)
GET ORIENT COORDINATES
IF(ISTRUC.EQ.18)THEN
LOCAT(1)=JOINT(CHOICE,10)
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

CALL GPESTC
CALL GPOPSTC
CALL ICON(LOCAT,AXISR)
CALL GPCLST
DISPLAY VALUES

REHIND 10
DO 350 III=1,3
WRITE(10,200)CLEAR
WRITE(10,201)
CONTINUE
REHIND 10
WRITE(10,205)LOCAT(1)
WRITE(10,205)LOCAT(2)
WRITE(10,205)LOCAT(3)
FORMAT(18,14) REHIND 10
READ(10,400)XVAL
READ(10,400)YVAL
READ(10,400)ZVAL
FORMAT(14) 

CALL GPOPST(10)
CALL GPPKIDClOll CALL GPUPWS(1,0)
CALL GPAN2(LX,20,X)
CALL GPAN2(LY,20,Y)
CALL GPAN2(LZ,16,ZVAL)
CALL GPCST

CALL COMMAND(INSID,32)
CALL MENU(INSID,CSIZE,8,CHOICE)
IF(CHOICE.EQ.1)THEN
RFLG=0
GO TO 1000
ENDIF
SUBROUTINE PICJ(J)

SUBROUTINE PICJ(WSID,JOINT,AXISR,CSIZE,END,ENDP,IAXIS,RFLG)

* ----- SUBROUTINE PICJ ----- *
* THIS SUBROUTINE WILL PROMPT FOR THE ENTRY OF JOINT*
* AT THE END OF A LINK *
* ************************************************** *

INTEGER WSID,PRFORM,PPATH(3),AClass(I),CHOICE,LENG,END,IAXIS,RFLG
REAL CSIZE(3),DATA(35),PAREA(6),L(4),JOINT(20,21),AXISR,POSN(2),
&PI(3),P(3),VEC(3),MAG,LOCAT(3),JPOSN(3),ENDP(3),
&LX(2),LY(2),LZ(2),LZ(2),LZ(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,
&
C DATA PRFORM/2/
& FLAG FOR UPDATE
C DATA ACLASS/4/
& INCLUSION CLASS LIST(CLASS 4 -- JOINTS AXIS

*** SET PICK FILTER (ALL CLASS 2 UNDETECTABLE) ***
C CALL GPPKF(WSID,1,0,AClass,2,AClass)
C MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
C CALL GPPKMO(WSID,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)=(.98+1/2)*CSIZE(2)
PAREA(5)=0
PAREA(6)=CSIZE(3)

C INITIALIZE PICK PATH
C CALL GPINP(WSID,1,0,PPATH,1,PAREA,0,DATA,1)
C PLACE PICK DEVICE IN THE EVENT MODE
C CALL GPPKMO(WSID,1,3,2)
C TRAVERSE ALL ACTIVE VIEWS
C CALL GPUPWS(WSID,PRFORM)
C GET PICK OR RETURN COMMAND
SUBROUTINE PICKJ2

SUBROUTINE PICKJ2(IWSID,JOINT,AXISR,CSIZE,END,ENDP,IAXIS,RFLG)

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* 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),11(4),JOINT(20,21),AXISR,POSN(2), &
&P(1),PI(3),VEC(3),MAG,LOCAT(3),POSM(3),ENDP(3), &
&LX(2),LY(2),LZ(2),LZ(2),LZ(2)
CHARACTER*20 X,Y,Z,CLEAR
CHARACTER*10 XVAL,YVAL,ZVAL,RETURN
DATA RETURN/'1. RETURN '/
DATA LENG/16/
DATA POSN/0.05,2.2/
DATA DATA/32,0,0,

& 2,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
C DATA PRFORM/2/ FLAG FOR UPDATE
C DATA ACLASS/5,4/ INCLUSION CLASS LIST(CLASS 4 -- JOINTS AXIS

******************************************************************************
* REQUEST PICK INPUT
******************************************************************************
C CALL GPPKF(INSID,1,2,AClass,0,AClass) MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
C CALL GPPKM(INSID,1,1,2) 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)=(-.98+1/2)*CSIZE(2)
PAREA(5)=0
PAREA(6)=CSIZE(3)
C CALL GPINPK(INSID,1,0,PPATH,1,PAREA,0,DATA,1) PLACE PICK DEVICE IN THE EVENT MODE
C CALL GPPKM(INSID,1,3,2) TRAVERSE ALL ACTIVE VIEWS
C CALL GPUPWSI(INSID,PRFORM) GET PICK OR RETURN COMMAND
******************************************************************************
* INCLUDE A RETURN PICK
******************************************************************************
C RETURN PICK
200 CALL GPOPST(6)
CALL GPADCN(1,AClass)
CALL GPPKID(101)
CALL GPTYCNI(2)
CALL GPADCH(1,0)
CALL GPANZ(POSN,LENG,RETURN)
CALL GPCLIST
CALL GPUPV(INSID,3,6,0,)
CALL GPUPWS(INSID,PRFORM)
******************************************************************************
* AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
******************************************************************************
C CALL GPAHEV(1000,INSID,ICLA,IDEV)
IF (ICLA.EQ.5) THEN
CALL GPUPTK(1,1,PPATH)
IF (PPATH(2).EQ.101) THEN
RFLG=1
END IF
ENDIF
GO TO 1000
C END
C CALL GPADCN(AClass)
C CALL GPPKID(101)
C CALL GPTYCNI(2)
C CALL GPADCH(1,0)
C CALL GPANZ(POSN,LENG,RETURN)
C CALL GPCLIST
C CALL GPUPV(INSID,3,6,0,)
C CALL GPUPWS(INSID,PRFORM)
******************************************************************************
* AWAIT AN EVENT
******************************************************************************
C CALL GPAHEV(1000,INSID,ICLA,IDEV)
IF (ICLA.EQ.5) THEN
CALL GPUPTK(1,1,PPATH)
IF (PPATH(2).EQ.101) THEN
RFLG=1
END IF
ENDIF
GO TO 1000
C END
C MAKE SURE PICK IS DEACTIVATED
1000 CALL GPPKM(INSID,1,1,2)
CALL GPEST(6)
C SET PICK FILTER(ALL CLASS 3,4 UNDETECTABLE)
C CALL GPPKF(INSID,1,0,AClass,2,AClass)
RETURN
END
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), L2(2), L3(2), LY(2), L5(2), L4(2)
CHARACTER*7 JNAM(100), LNAM(30), LNAME, JNAME
CHARACTER*20 RETURN
CHARACTER*20 X, Y, Z, CLEAR
DATA RETURN/'1. RETURN', 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
DATA DATA/32, 0, 0, 0.02, 1.1, 1.1, 1.1, &
& 1.1, 1.1, 1.1
&
& 1.1, 1.1

C DATA PRFORM/2/
C DATA ACLASS/5/
C DATA BOX/0.00, 0.545, 0.46, 0.545, 0.46, 0.73, 0.00, 0.73/

LY(1)=0.02
LY(2)=0.62
LY(1)=0.31
LY(2)=0.62
DO 10 I=2, 8, 2
SHDI=BOX(I)-0.025
10 CONTINUE
DO 20 I=1, 7, 2
SHDI=BOX(I)+0.025
20 CONTINUE

******************************************************************************
** REQUEST PICK INPUT
******************************************************************************
C CALL GPPKF(IWSID, 1, ACLASS, 0, ACLASS)
C CALL GPPMO(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 CALL GPINP(IWSID, 1, 0, PPATH, 1, PAREA, 0, DATA, 1)
C PLACE PICK DEVICE IN THE EVENT MODE
C CALL GPPOPS(IWSID, PRFORM)
C TRAVERSE ALL ACTIVE VIEWS
C CALL GPPOWS(IWSID, PRFORM)
C GET INQUIRY OR RETURN COMMAND

******************************************************************************
** INCLUDE A RETURN PICK
******************************************************************************
C CALL GPOPST(6)
C CALL GPADCN(1, ACLASS)
C CALL GPPOKDI(101)
C CALL GPTXCI(2)

MECHIN PROGRAM LISTING

147
SUBROUTINE POSIT

SUBROUTINE POSIT(IWSID,AXISR,CSIZE,LOCAT,PRFLG)

* ----- SUBROUTINE POSIT ----- *
* THIS SUBROUTINE IS USED TO POSITION A POINT IN SPACE *
*********************************************************************
INTEGER IWSID,ECHO,EVENT,DLEN,ECHO,ACLASS,PRFORM,CHOICE,
& PPATH(3),PRFLG,NNUM
REAL CSIZE(3),AREA1(6),AREA2(6),AREA3(6),VALUE1,LOVAL,HIVAL,AXISR,
& BOX1(8),SHD(8),LOCAT(3),MATRIX(1000,4),MATUSE(4,4),LINE(6),
& YBOX(15),ZBOX(15),LX(6),LY(6),LZ(6),AREA(6)
CHARACTER*7 CLEAR,INIT
DATA CLEAR/'1'/
DATA INIT/'0.'/
DATA LV/0.,0.,0.,0.,0.,0./
DATA LY/0.,0.,0.,0.,0.,0./
DATA LZ/0.,0.,0.,0.,0.,0./
C DATA PRFORM/2/  ! FLAG FOR UPDATE
C DATA ACLASS/1/  ! INCLUSION CLASS LIST

MECHIN PROGRAM LISTING  148
DATA VALUEI/O.
DATA ECHO/2.
DATA DLEN/2.
DATA EVENT/3.
DATA ECHOSH/2.

C

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)=CSIZE (2)-.006
AREA1(4)=CSIZE (2)
AREA1(5)=.00
AREA1(6)=CSIZE (3)

AREA2(1)=.52*CSIZE (1)
AREA2(2)=CSIZE (1)
AREA2(3)=CSIZE (2)-.014
AREA2(4)=CSIZE (2)
AREA2(5)=.00
AREA2(6)=CSIZE (3)

AREA3(1)=.52*CSIZE (1)
AREA3(2)=CSIZE (1)
AREA3(3)=CSIZE (2)-.022
AREA3(4)=CSIZE (2)
AREA3(5)=.00
AREA3(6)=CSIZE (3)

DO 100 I=2,8,2
    SHDI(I)=BOX(I)-0.025
100 CONTINUE
DO 200 I=1,7,2
    SHDI(I)=BOX(I)+0.025
200 CONTINUE

*********************************************************************
* OPEN STRUCTURE FOR THE VAL. BOX (VIEW 1) 
*********************************************************************
C
CALL GPOPSTC(8) SET SOLID INTERIOR STYLE FOR GREY BOX
CALL GPISI(2) SET COLOR FOR INTERIOR OF SHADOW BOX
CALL GPIC1(4) DRAG GREY SHADOW BOX
CALL GPPG2(I,4,2,SHDI) SET COLOR FOR INTERIOR OF GREY BOX
CALL GPIC1(3) DRAG GREY TITLE BOX
CALL GPPG2(1,4,2,BOX) CLOSE STRUCTURE
CALL GPCLST

*********************************************************************
* SCREEN DISPLAY 
*********************************************************************
C
CALL GPUPHS(IMSID,PRFORM)

*********************************************************************
* INITIALIZE AND ACTIVATE DIALS 1,2,3 AND STRING DEVICE 
*********************************************************************
C
PLACE VALUATORS IN THE REQUEST MODE
300 CALL GPVLOM(IMSID,1,1,ECHOSH)
CALL GPVLOM(IMSID,2,1,ECHOSH)
CALL GPVLOM(IMSID,3,1,ECHOSH)
C
CALL GPINVL1(IMSID,1,VALUE1,ECHO,AREA1,LOVAL,HIVAL,DLEN,DATA)
CALL GPINVL1(IMSID,2,VALUE1,ECHO,AREA2,LOVAL,HIVAL,DLEN,DATA)
CALL GPINVL1(IMSID,3,VALUE1,ECHO,AREA3,LOVAL,HIVAL,DLEN,DATA)
C
CALL GPVLOM(IMSID,1,EVENT,ECHOSH)
CALL GPVLOM(IMSID,2,EVENT,ECHOSH)
CALL GPVLOM(IMSID,3,EVENT,ECHOSH)
C
CALL GPPHMO(IMSID,1,10,INIT,1,AREA,12,1,0,CLEAR)
CALL GPPSHMO(IMSID,1,EVENT,ECHOSH)
UPDATE THE WORKSTATION

CALL GPUPHSI(INSID,PRFORM)

DISPLAY MENU AND GET CHOICE OF ENTER, RETURN OR VALUATOR

LOCAT1(1)=0.
LOCAT1(2)=0.
LOCAT1(3)=0.

500 CALL CMNANDI(INSID,11)
CALL ENTERI(INSID,CHOICE,CSIZE)

600 CALL GPAREV1000.,INSID,ICLASS,IDEV)
IF (ICLASS.EQ.3) THEN

PROCESS VALUATORS (NOT IN SUBROUTINE TO SPEED EXECUTION)

CALL GPSTM0(INSID,1,1,ECHOSM)
CALL GPSTVL(VALUE)

LOCAT2(1)=LOCAT1(1)
LOCAT2(2)=LOCAT1(2)
LOCAT2(3)=LOCAT1(3)

LINEX(1)=LOCAT1(1)
LINEX(2)=LOCAT1(2)
LINEX(3)=LOCAT1(3)
LINEX(4)=0.
LINEX(5)=LOCAT1(2)
LINEX(6)=LOCAT1(3)

LINEY(1)=LOCAT1(1)
LINEY(2)=LOCAT1(2)
LINEY(3)=LOCAT1(3)
LINEY(4)=0.
LINEY(5)=LOCAT1(3)
LINEY(6)=LOCAT1(3)

LINEZ(1)=LOCAT1(1)
LINEZ(2)=LOCAT1(2)
LINEZ(3)=LOCAT1(3)
LINEZ(4)=LOCAT1(1)
LINEZ(5)=LOCAT1(2)
LINEZ(6)=LOCAT1(3)

XBOX1(1)=0.00
XBOX1(2)=LOCAT1(2)+ .06*AXISR
XBOX1(3)=LOCAT1(3)+ .06*AXISR
XBOX1(4)=0.00
XBOX1(5)=LOCAT1(2)- .06*AXISR
XBOX1(6)=LOCAT1(3)- .06*AXISR

YBOX1(1)=LOCAT1(1)+ .06*AXISR
YBOX1(2)=0.00
YBOX1(3)=LOCAT1(3)+ .06*AXISR
YBOX1(4)=LOCAT1(1)- .06*AXISR
YBOX1(5)=0.00
YBOX1(6)=LOCAT1(2)- .06*AXISR

ZBOX1(1)=LOCAT1(1)+ .06*AXISR
ZBOX1(2)=LOCAT1(2)+ .06*AXISR
ZBOX1(3)=0.00
ZBOX1(4)=LOCAT1(1)- .06*AXISR

MECHIN PROGRAM LISTING 150
CALL GPTESTI(9)
CALL GPOPSTI(9)

SET POLYLINE COLOR

SET LINE TYPE DASHED

DRAW POLYLINE

CALL GPPLC1(1)
CALL GPLT(2)

SET LINE TYPE SOLID

CALL GPPLC1(1)
CALL GPPLC1(5,3,LINEY)
CALL GPPLC1(5,3,ZBOX)

SET POLYLINE COLOR

CALL GPPLC1(1)
CALL GPPLC1(5,3,LX)
CALL GPPLC1(5,3,LY)
CALL GPPLC1(5,3,LZ)
CALL GPCLST
CALL GPUPWSI(HSID,PRFORM)
GO TO 600

ENDIF

IFIICLASS.EQ.6)THEN
PROCESS STRING ENTRY

CALL GPVLMOI(HSID,1,1,ECHOSH)
CALL GPVLMOI(HSID,2,1,ECHOSH)
CALL GPVLMOI(HSID,3,1,ECHOSH)

IFRFLG.EQ.1)THEN
REENTER POINT

CALL GPVLMOI(HSID,1,3,ECHOSH)
CALL GPVLMOI(HSID,2,3,ECHOSH)
CALL GPVLMOI(HSID,3,3,ECHOSH)
CALL GPSTM0I(HSID,1.1,ECHOSH)
CALL GPSTM0I(HSID,1.2,INIT.1,AREA,12,1,0,CLEAR)

LOCAT(1)=0:
LOCAT(2)=0:
LOCAT(3)=0:
RFLG=0:
GO TO 500

ENDIF

CALL GPSTM0I(HSID,1,1,ECHOSH)
CALL GPEST(8)
CALL GPEST(9)
GO TO 1000

ENDIF

IFIICLASS.EQ.5)THEN

CALL GPGTPK(1,1,PPATH)

CHOICE=PPATH(2)

IF(CHOICE.EQ.1)THEN

DEACTIVATE VALUATORS

CALL GPVLMOI(HSID,1,1,ECHOSH)
CALL GPVLMOI(HSID,2,1,ECHOSH)
CALL GPVLMOI(HSID,3,1,ECHOSH)
RFLG=0:
CALL GPEST(6)
CALL GPEST(8)
CALL GPEST(9)
CALL GPUPWSI(HSID,2)
GO TO 1000

ENDIF

IF(CHOICE.EQ.2)THEN

DEACTIVATE VALUATORS

CALL GPVLMOI(HSID,1,1,ECHOSH)
CALL GPVLMOI(HSID,2,1,ECHOSH)
CALL GPVLMOI(HSID,3,1,ECHOSH)
RFLG=0:
CALL GPEST(6)
CALL GPEST(8)
CALL GPEST(9)
CALL GPUPWSI(HSID,2)
SUBROUTINE RDRWJT

SUBROUTINE RDRWJT(IHSID, JNAM, JNUM, JOINT, AXISR, SCALE)
********************************************************************* *
----- SUBROUTINE RDRWJT ----- *
* THIS SUBROUTINE IS USED TO REDRAW ALL JOINTS AFTER A DELETION *
* OR RESTART *
********************************************************************* *

INTEGER INUM, JNUM(20,3), JTYPE
REAL LOCAT(3), JOINT(20,21), AXISR, ORIENT(3), SCALE
CHARACTER*7 JNAME, JNAM(20)

CALL GPESTC
CALL GPESTC
CALL GPESTC
CALL GPESTC

DELETE ALL JOINTS
CALL GPUPHI(IHSID, 2)

DO 10 I=1,100

  IF(JNAM(I),.EQ., 'BBBBBBB') GO TO 1000 CHECK
  IF(JNAM(I),.EQ., 'CCCCCCC') GO TO 10

  GET JOINT NUMBER NUMJ=JNUM(I),1

  GET JOINT TYPE JTYPE=JNUM(I),2

  GET JOINT LOCATION LOCAT(I)=JOINT(I),1

  GET JOINT ORIENTATION ORIENT(I)=JOINT(I),4

  GET JOINT NAME JNAME=JNAM(I)

  DRAW JOINT CALL DRAHJ(IHSID, AXISR, JNAME, NUMJ, JTYPE, JOINT, SCALE)

10 CONTINUE

1000 RETURN
END

SUBROUTINE RDRWLT

SUBROUTINE RDRWLK(IHSID, JNAM, JNUM, JOINT, AXISR, LNAME, LNUM, LINK, SCALE)
CALL GPVP(ISID,4,1,1)
CALL GPUPHS(ISID,2)
DO 10 I=1,30
C CHECK FOR LAST JOINT
C IF(INAM(I).EQ.'BBBBBBB' GO TO 1000
C CHECK DELETED SPACES IN THE ARRAY
C IF(INAM(I).EQ.'CCCCCCC' GO TO 10
ENDIF
C NUM=LNUM(I,1)
C GET JOINT NUMBER
C DRAH JOINT
CALL DRAH(LHSID,INAM,LNUM,LINK,JOINT,JNUM,AXISR,NUML,SCALE)
10 CONTINUE
1000 RETURN END

SUBROUTINE RDRWPT

SUBROUTINE RDRWPT(ISID,PNUM,PNUM,POINT,AXISR)
******************************************************************************
* ----- SUBROUTINE RDRWPT ----- *
* THIS SUBROUTINE IS USED TO REDRAW ALL POINTS AFTER A DELETION *
* OR RESTART *
******************************************************************************
INTEGER ISID,JNUM,PNUM
REAL LOCAT(3),POINT(100,3),AXISR
CHARACTER*7 PNAME,PNUM(100)
C DELETE ALL POINTS
CALL GPVEST(ISID,12)
CALL GPUPHS(ISID,2)
DO 10 I=1,1000
C CHECK FOR LAST POINT
C IF(PNUM(I).EQ.'BBBBBBB' GO TO 1000
C CHECK DELETED SPACES IN THE ARRAY
C GET POINT NUMBER
C 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
C DRAH POINT
CALL DRAH(ISID,LOCAT,AXISR,PNAME,NUMP)
10 CONTINUE
1000 RETURN END

SUBROUTINE REINV

SUBROUTINE REINV(ISID,JNUM,JNUM,JOINT,AXISR,LNAM,LNUM,LINK, &SCALE,CSIZE)
******************************************************************************
* ----- SUBROUTINE REINV ----- *
* THIS SUBROUTINE IS USED TO MAKE LINKS AND JOINTS VISIBLE *
******************************************************************************
INTEGER ISID,JNUM,JTYPE,JEND,LNUM(30,14)
REAL LOCAT(3),JOINT(20,21),AXISR,ORIENT(3),SCALE,ENDP(3),CSIZE(3), & LINK(30,6)
CHARACTER*7 JNAME,JNUM(20),LNAM(30)
DO 20 I=1,20
  JNUM(I,13)=0
20 CONTINUE
DO 30 I=1,30
  JNUM(I,14)=0
30 CONTINUE
CALL RDRHJTI(IHSID,JNAM,JNUM,JOINT,AXISR,SCALE)
CALL RDRHLKJ(IHSID,JNAM,JNUM,JOINT,AXISR,LNAM,LNUM,LINK,SCALE)
1000 RETURN
END

SUBROUTINE RESTOR

** SUBROUTINE RESTOR(IHSID,AXISR,CSIZE) **

********************************************************************************
---this subroutine is used to restore the original-----------------------------
********************************************************************************
INTEGER IHSID,PARELL,PRFORM
REAL AXISR,AXISS(24),AXISS(24),AXISS(24),WINDOW(4),VP1(6),VP2(6),
8BOX(18),BOX(18),NEAR,FAR,POINT(3),DIST,UP(3),NORMAL(3),
&AXISS(4),POINT(3),AXISS(3),PY(3),PZ(3),HINDO(4),NEAR1,FAR1
CHARACTER*7 X,Y,Z
C C STATE THE VIEW CHARACTERISTICS FOR AXES VIEW -----------------------------------------------
C C SET THE CHARACTERISTICS CALL GPVMP3(IHSID,4,WINDOW,VP1,PARELL,POINT,DIST,NEAR,FAR)
C C ACTIVATE THE VIEW CALL GpvCH(IHSID,4,2,2,1,0,2,5,2)
C CALL GPUPHS(IHSID,PRFORM)
RETURN
END

SUBROUTINE RSHR

** SUBROUTINE RSHR(PNAM,PNUM,POINT,JNAM,JNUM,JOINT,LNAM,LNUM,LINK, **
PROGRAM RSCRA

** --- 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, 1), POINT(100, 3), AXISR, CSIZE(3), AREA(6), & IG(20, 4)
CHARACTER*7 LNAM(30), JNAM(20), PNAM(100), FILE, FILEO, FILEI, INAM(20), & NAME, LNAME1, LNAME2, FILE

DATA FILE/"RSCR"/
NP=0
NJ=0
NL=0
NI=0

DO 1 I=1, 20
  DO 1 J=1, 3
    CONN(I, J)=0
  CONTINUE
1 CONTINUE
IRET=0
AREAI1J=.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)

FILE=''

REWIND 15

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

SORT OUT LINK CONNECTIVITIES
** ********************************************************************* **

50 DO 100 J=1, 30
  IF(LNAM(J).EQ."CCCCCCC") THEN
    GO TO 90
  END IF
  IF(LNAM(J).EQ."BBBBBBB") THEN
    GO TO 105
  END IF
  LN=J
  DO 50 I=2, 13, 2
    CONN(LN, I)=JNUM(LN, I)
  CONTINUE
90 CONTINUE
100 CONTINUE
105 DO 110 I=1, 20
  CONN(I, 1)=JNUM(I, 1)
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
      END IF
      IF(CONN(J, 3).EQ.IGRD) THEN
        CONN(J, 3)=0
      END IF
    180 CONTINUE
  END IF
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
END IF

300 CONTINUE
IFICONNI,1J.EQ.4JTHEN CONNII,1J=94 III=III+1 GO TO 300 ENDIF IF(IFICONNI,1J.EQ.3JTHEN CONNII,1J=93 III=III+1 GO TO 300 ENDIF IF(IFICONNI,1J.EQ.1JTHEN CONNII,1J=91 III=III+1 GO TO 300 ENDIF

GO TO 300 ENO

C DO 301 I=1,20
WRITE(6,*)'CONN J,(CONN(I,J),J=1,3)
C 301 CONTINUE
WRITE(6,*)'III',III
IF(III.NE.4)THEN IRETT=1 RETURN
ENDIF

C *********************************************************************
* INITIALIZE THE STRING DEVICE. REQUEST NEW RESTART FILE NAME
C *********************************************************************
CALL GPMSTIIHSID,1,1,2)
C CALL GPSTIIHSID,1,7,FILE,1,AREA,7,1,0,FILE)
C CALL CMANDIIHSID,91)
C FILEN=' REQUEST STRING
CALL GPQSTIIHSID,1,7,ISTAT,NUM,FILEN)
IF(FILEN.EQ.'GO TO 5000
FILE=FILE
C SEE IF FILE ALREADY EXISTS
C IF FILE EXISTS, HAVE USER RETYPE TO USE
CALL CMANDIIHSID,8)
CALL GPQSTIIHSID,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)
C *********************************************************************
* WRITE DATA FILE
C *********************************************************************
WRITE(16,999)'RSCR ANALYSIS DATA'
999 FORMAT(A18)
DO 1000 I=1,20
IF(CONN11,1).EQ.96)THEN L1=CONN11,2) L2=CONN11,3)
ENDIF
1000 CONTINUE
DO 1010 I=1,20
IF(CONN11,1).EQ.91)THEN IF(CONN11,2).EQ.0)THEN WRITE(16,9)JOINT11,1)JOINT11,2),JOINT11,3) WRITE(16,9)JOINT11,4)JOINT11,5)JOINT11,6) CONN11,1)=81 POS=IC11,1) RINC=IC11,2) ISTEP=INUM11,2) VEL=IC11,3) ACC=IC11,4) GO TO 1011
ENDIF
ENDIF
IF(CONN11,3).EQ.0)THEN
SUBROUTINE RSCRS

**SUBROUTINE** RSCRS(PNAM, PNUM, POINT, AXISR, CSIZE, IMSID, IRET)

* * *
**THIS SUBROUTINE IS USED WRITE A RSCR SYNTHESIS FILE**
* * *
**SUBROUTINE** RSCRS( )

* * *
**DESCRIPTION**

* INTEGER PNAM, PNUM, POINT, AXISR, CSIZE, IMSID, IRET 
* REAL POINT, CSIZE, AXISR, AREA, PAREA, OR, OR3, MAT(4,4), MATSC(4,4), SCALE, MATS(4,4), ROI, PPOSN 
* CHARACTER*7 PNAM, FILE, FILEO, FILE1, FILEN 

* DATA/1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0/ 
* DATA/0.05, 2.2/ 
* DATA/0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0/ 
* DATA/32.0, 32.0/ 
* DATA/PREFIX/ 
* DATA/PRFORM/ 
* DATA/ACLS/ 
* DATA/FILE/ 

* NP=0 
* IRET=0

MECHIN PROGRAM LISTING 157
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="\nREMIND 15
********************************************************************
* PICK THE THREE BODY POSITIONS FOR SYNTHESIS
********************************************************************

C INITIALIZE PICK
C CALL GPPKG(IINSID,1,1,ACLASS,0,AClass)
C CALL GPPKMO(IINSID,1,1,2)
C DEFINE PICK AREA
PAREA(1)=(-.98+1/2)*CSIZE(1)
PAREA(2)=(-.98+1/2)*CSIZE(2)
PAREA(3)=(-.98+1/2)*CSIZE(3)
PAREA(4)=.48+1/2)*CSIZE(2)
PAREA(5)=0
PAREA(6)=CSIZE(3)
C INITIALIZE PICK PATH
C CALL GPINPK(IINSID,1,0,PPATH,1,PAREA,DATA,1)
C MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
C CALL GPPKMO(IINSID,1,1,2)
C DEFINE PICK AREA
PAREA(1)=(-.98+1/2)*CSIZE(1)
PAREA(2)=(-.98+1/2)*CSIZE(2)
PAREA(3)=(-.98+1/2)*CSIZE(3)
PAREA(4)=.48+1/2)*CSIZE(2)
PAREA(5)=0
PAREA(6)=CSIZE(3)
C INITIALIZE PICK PATH
C CALL GPINPK(IINSID,1,0,PPATH,1,PAREA,DATA,1)
C GET INQUIRY OR RETURN COMMAND
C CALL SCMMAN(IINSID,25)
C INCLUDE A RETURN PICK
C CALL GPOPST(6)
C CALL GPADCN(1,AClass)
C CALL GPKPDI(101)
C CALL GPTXCI(2)
C CALL GPASHCI(3)
C CALL GAPPN(PSON,LENG,RETURN)
C CALL GPCLST
C CALL GPARV(IINSID,3,6,0)
C CALL GPVP(IINSID,3,1,1)
C CALL GPUPHS(IINSID,2)
C CALL SCMMAN(IINSID,25)

C AWAIt A PICK EVENT TO OCCUR AND RETRIEVE IT
C CALL GPAHEV(1000,INS,ICLA,IDEV)
C IF(ICLA.EQ.9)THEN
C GET PICK
C CALL GPGTPK(1,IDEP,PPATH)
C IF NO PICK, RETURN
C IF(PPATH(2).EQ.101)THEN
IRETN=1
C CALL GPPKMO(IINSID,1,1,2)
C CALL GPPST(6)
C CALL GPUPHS(IINSID,PRFORM)
C GO TO 5000
ENDIF
C CHOICE IS SECOND ITEM OF PICK PATH
C GET DATA FOR SYN. FILE FROM POSITION PICKED
C ENDIF
C CHOICE=PPATH(2)
C CHOICE IS SECOND ITEM OF PICK PATH
C GET DATA FOR SYN. FILE FROM POSITION PICKED
C
C III=III+1
C IF(III.EQ.2)THEN
C CALL SCMMAN(IINSID,26)
C GO TO 200
C IF(III.EQ.3)THEN
C CALL SCMMAN(IINSID,34)
C ZEROS A NEXT BODY POSITION
GO TO 200
ENDIF
IF(I GT 3) THEN
GO TO 1000
ENDIF
ELSE
GO TO 200
ENDIF

C
MAKE SURE PICK IS DEACTIVATED
C
1000 CALL GPPKMD(ISID,1,1,2)
CALL GPPST(ISID,PRFORM)
C
SET PICK FILTER(ALL CLASS 4 UNDETECTABLE)
C
ENDIF

********************************************************************
* REQUEST POSITION OF THE SPHERICAL JOINT
********************************************************************
1900 CALL SCMMAN(ISID,35)
CALL SPOSIT(ISID,AXISR,CSIZE,SLOC,IRRET)
IF(IRRET.EQ.1) THEN
GO TO 1900
ENDIF
C
DRAW SPHERICAL JOINT ICON
C
ENDIF

********************************************************************
* REQUEST ORIENTATION OF THE SECOND REVOLUTE JOINT
********************************************************************
1910 CALL SCMMAN(ISID,40)
PPSON11=0.0
PPSON12=0.0
PPSON13=0.0
CALL SORIEN(ISID,PPSON,POINT,AXISR,CSIZE,RO,IRRRET)
IF(IRRRET.EQ.1) THEN
GO TO 1910
ENDIF

********************************************************************
* INITIALIZE THE STRING DEVICE, REQUEST SYNTHEIS FILE NAME
********************************************************************
C
2003 CALL GPINST(ISID,1,7,FILE,1,AREA,7,1,0,FILE)
C
2004 CALL FILEX(FILE,IRET)
C
IF(IRET.EQ.0) THEN
CALL GPSTMO(ISID,1,1,2)
FILE=FILE
FILE=*C
CALL CMDANI(ISID,8)
CALL GPINST(ISID,1,7,FILE,1,AREA,7,1,0,FILE)
CALL GPGRST(ISID,1,7,ISTAT,NM,FILE)
IF(FILE.EQ."") THEN
GO TO 5000
ENDIF
C
C OPEN A NEW FILE
OPEN(INUNIT=16,FILE=FILE,STATUS='NEW')
SUBROUTINE RSTART

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 IHSID, PNUM, JNUM, LNUM, SCLASS, IC, SCALE
REAL CSIZE, AXISR, POINT, JOINT, LINK, INUM
CHARACTER*28 FILE, CMD, QUIT, CLEAR, PNAM, JNAM, LNAM
CHARACTER*5 FLINE, CLINE
LOGICAL EXT

DATA RETURN/'RENT'/
DATA LENG/16/
DATA POSN/0.05, 2.0/
DATA DATA/2.0, 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(2)
AREA(3)=0.70*CSIZE(3)
AREA(4)=0.90*CSIZE(4)
AREA(5)=0.80*CSIZE(5)
AREA(6)=0.90*CSIZE(6)

C INITIALIZE PICK
C SET PICK FILTER(ALL CLASS 2 DETECTABLE)
CALL GPPKF(IHSID,1,1,AClass,0,AClass)

CALL GPPKMO(IHSID,1,1,2)

MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)

DEFINE PICK AREA

PAREA(1)=(-.98+.1*CSIZE(1))
PAREA(2)=(-.98+.1*CSIZE(1))
PAREA(3)=(-.98+.1*CSIZE(2))
PAREA(4)=(-.48+.1*CSIZE(2))

initial pick path

CALL GPINPKIIHSID,l,0,PPATH,l,PAREA,O,DATA,1

PLACE PICK DEVICE IN THE EVENT MODE

CALL GPPKMO(IHSID,1,2,2)

TRaverse ALL ACTIVE VIEWS

CALL GPUPHSIIHSID,2)

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

CALL GPSTMOIIHSID,1,1,2)

CALL GPINSTIIHSID,l,7,CLEAR,l,AREA,7,l,O,CLEAR

CALL GPSTMOIIHSID,l,3,2)

*********************************************************************
* QUIT IF QUIT IS PICKED
*********************************************************************

L=7

CALL GPGTST(L,LR,FILE)

CALL FILEXS(FILE,IRET)

IF(IRET.EQ.5)THEN QUIT PROGRAM

ENDIF

CALL CMMAND(IHSID,5)

WRITE THAT ACTION IS CONTINUING

CALL FILENMIIHSID,FILE)

*********************************************************************
* SET UP AXES
*********************************************************************

READ THE AXIS RANGE

MECHIN PROGRAM LISTING

161
READ(15,*)AXISR
CALL AXES(IHSID,AXISR)
READ(15,*)SCALE
CALL Scalars(IHSID,SCALE)
**********************************************************************
* SET UP POINT ARRAY
**********************************************************************
C READ(15,*)NP
READ(15,99)PNAM(I)
99 FORMAT(A7*) READ(15,*)PNUM(I)
READ(15,*)POINT(I,J),J=1,3
CONTINUE
**********************************************************************
* SET UP LINK ARRAY
**********************************************************************
C READ(15,*)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
**********************************************************************
* DRAW ALL POINTS, JOINTS, AND LINKS
**********************************************************************
CALL RRDP(1HSID,PNAM,PNUM,POINT,AXISR)
CALL RRRJ(1HSID,NAM,JNUM,JOINT,AXISR,SCALE)
CALL RRNL(1HSID,JNUM,JOINT,AXISR,LNAM,LNUM,SCALE)
CALL RICOS(IHSID,JOINT,INAM,INUM,IC,SCALE)
CALL CPSTMO(1HSID,1,1,2)
CALL GPPTOI(1HSID,1,0,ACLASS,1,ACLASS)
RETURN
END

SUBROUTINE SADDP

SUBROUTINE SADDP(IHSID,AXISR,CSIZE,PNAM,PNUM,POINT,SCALE)
**********************************************************************
* ----- SUBROUTINE SADDP ----- *
**********************************************************************
* THIS SUBROUTINE IS USED TO ADD BODY POSITION POINTS *
**********************************************************************
COMMON/PDEF/PNDEF(100)
INTEGER IHSID,NUL,PNUM(30,4),RFLG,CLASS(1), &
CHOICE
REAL AXISR,LOCAT(3),AREA(6),POINT(30,14),CSIZE(3),
& ENDP(13), ENDP(23), LNK(6), XVEC(3), YVEC(3), XI(6), X2(6),
& POINT(20, 21), SC(3), SCH(4, 4), TRM(4, 4), TRM(4, 4), TRM(4, 4), P(3),
& P(3), P(23), S(4, 4), T(4, 4), XAX(3), YAX(3), ZAX(3), NAX(3)

CHARACTER*7 PNDEF, PNAME, PNAM(30), CLEAR, CLEAR1
CHARACTER*20

DATA CLEAR /'I'
DATA CLEAR1 /'I'
DATA CLEAR2 /'I'
DATA SCM(/1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0/)
DATA TRM(/1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0/)
DATA S(/1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0/)
DATA T(/1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0/)
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)=0.02*CSIZE(1)
AREA(5)=0.90*CSIZE(1)
PNAME=''
SC(1)=SCALE
SC(2)=SCALE
SC(3)=SCALE

CALL GPSC31(SC, SCM)

********************************************************************
* SET POINT NUMBER
* SEARCH PNAM ARRAY FOR NEXT AVAILABLE SLOT (BBBBBBB)
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 NO MORE POINTS AVAILABLE
ELSE CALL SCCMAN(HSID, 46)
GET STROKE
CALL GPINST(HSID, 1, ?, CLEAR, 1, AREAl?, 1, CLEAR)
CALL GPRQST(HSID, 1, ?, ISTAT, NUM, CLEAR)
RETURN
ENDIF

********************************************************************
* GET THE POINT NAME OR DEFAULT
* REQUEST STRING
CALL SCCMAN(HSID, 47)
CALL GPSTM(HSID, 1, 1, ?, CLEAR)
CALL GPINST(HSID, 1, 1, ?, CLEAR)
CALL GPRQST(HSID, 1, 1, ?, CLEAR)
RETURN

********************************************************************
* POSITION THE POINT
CALL POSIT(HSID, AXISR, CSIZE, LOCAT, RFLG)
IF(RFLG.EQ.1) THEN
  GO TO 1000
END IF
POINT(NUMP, 1)=LOCAT(1)
POINT(NUMP, 2)=LOCAT(2)
POINT(NUMP, 3)=LOCAT(3)
CALL SDRAHPI(HSID, AXISR, PNAME, NUMP, POINT, SCALE)
DO 30 I=1,14
   POINT(NUMP,I)=0.0
30 CONTINUE
ENDIF
POINT(NUMP,4)=XAX(I)
POINT(NUMP,5)=YAX(I)
POINT(NUMP,6)=ZAX(I)
CALL SDRAHPIIHSID(IHSID,AXISR,PNAME,NUMP,POINT,SCALE)

********************************************************************
** ORIENT THE A VECTOR IN THE X Y PLANE
********************************************************************
CALL SCMMANIIHSID(36) CALL SORIENIIHSIO,LOCAT,POINT,AXISR,CSIZE,NAX,RFLG)
IF(RFLG.EQ.1)THEN
   DO 40 I=1,14
      POINT(NUMP,I)=0.0
40 CONTINUE
ENDIF
CALL VPRODI(XAX,YAX,ZAX)
CALL VPRODIZAX,XAX,YAX)
CALL VUNITIXAX,XAX)
CALL VUNITIYAX,YAX)
CALL VUNITIZAX,ZAX)
CALL SDRAHPIIHSID(IHSID,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(I)
POINT(NUMP,5)=YAX(I)
POINT(NUMP,6)=ZAX(I)
POINT(NUMP,7)=XAX(I)
POINT(NUMP,8)=YAX(I)
POINT(NUMP,9)=ZAX(I)
POINT(NUMP,10)=ZAX(I)
POINT(NUMP,11)=ZAX(I)
POINT(NUMP,12)=ZAX(I)

********************************************************************
** DRAW THE POINT
********************************************************************
1000 CALL SRDRHPIIHSID(IHSID,AXISR,PNAME,NUMP,POINT,SCALE)

RETURN
END

SUBROUTINE SASSO(IHSID,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 IHSID
REAL CSIZE(3),WIND(4),VIEW(4),WINDOC(4),VPC(4), &WINDOM(4),VPH(4),JOINT
CHARACTER*7 PNAM,JNAM,LNAM,INAM

MECHIN PROGRAM LISTING
SUBROUTINE SCALNM(IWSID,SCALE)

SUBROUTINE SCALNM(IWSID,SCALE)

******************************************************************************
* ----- SUBROUTINE SCALNM ----- *
******************************************************************************

* THIS SUBROUTINE IS USED TO JOIN SCALE ON THE SCREEN *
******************************************************************************

INTEGER IWSID,LENGT,LENGF,PRFORM
REAL POSF(2),BOX1(8),SHD1(8),POS1(2),POS2(2),AXISR

MECHIN PROGRAM LISTING
**SUBROUTINE SCHSCA**

```fortran
SUBROUTINE SCHSCA(INSID,AXISR,CSIZE,SCALE,PNUM,PNUM,POINT)
**----- SUBROUTINE SCHSCA -----**
** THIS SUBROUTINE WILL PROMPT CHANGE OF BODY POINT SCALE**
** **

```
SUBROUTINE SCHVIE

INTEGER IMSID,CHOICE,PNUM(30,4)
REAL CSIZE(3),AXISR,POINT(30,14)
CHARACTER*7 CLEAR,AXIS,INIT,INIT1,PNAM(30)
CHARACTER*20 CLEAR1,X,Y,Z
CHARACTER*12 STRING,CLEAR2

CALL SCMMANI(IMSID,69)
1 CALL GPSTMO(IMSID,1,1,2)
   CALL GPSTN(IMSID,1,12,INIT,1,AREA,12,1,0,CLEAR)
   STRING=CLEAR2
   CALL GPRQST(IMSID,1,12,ISTAT,NUM,STRING)
   REWIND 10
   WRITE(10,8)CLEAR1
   REWIND 10
   WRITE(10,12)STRING
   REWIND 10
   READ(10,*)SCALE
   IF(Scale.LE.0.JTHEN
   CALL SCMMANI(IMSID,70)
   GO TO 1
   ENDIF
   CALL SCALNM(IMSID,SCALE)
   CALL SRDRMP(IMSID,AXISR,PNAM,PNUM,POINT,SCALE)
1000 CALL GPSTMO(IMSID,1,1,2)
RETURN
END

SUBROUTINE SCHVIE(IMSID,AXISR,CSIZE,SCALE,PNAM,PNUM,POINT)

*********************************************************************
* THIS SUBROUTINE IS USED TO CHANGE THE MAIN VIEW
*********************************************************************

INTEGER IMSID,CHOICE,PNUM(30,4)
REAL CSIZE(3),AXISR,POINT(30,14)
CHARACTER*7 PNAM(30)
5 CALL SCMMANI(IMSID,10)
CALL SMENUI(IMSID,CSIZE,3,CHOICE)
SUBROUTINE SCMMAN

SUBROUTINE SCMMAN(IHSID, NUM)

******************************************************************************
* ----- SUBROUTINE SCMMAN ----- *
******************************************************************************

* THIS SUBROUTINE IS USED TO DISPLAY COMMANDS FOR SYNTHESIS *
******************************************************************************

INTEGER IHSID, ENGT, PRFORM, NUM
REAL POSL112, POSL212, POSL312, POSL412, POSL512, HINDO14, VPC141
CHARACTER*19 C(90, 5)

C COMMANDS

DATA C(1, 1)'/ PICK EITHER NEW
DATA C(1, 2)'/ 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 MECHAN
DATA C(2, 5)'/ RESTART FILE
DATA C(2, 9)'/ (UPPER CASE)

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

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

DATA C(5, 1)'/ ENTER THE RANGE
DATA C(5, 3)'/ OF THE COORDINATE
DATA C(5, 5)'/ AXES FOR THIS
DATA C(5, 8)'/ 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)'/ 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

IFICHOICE.EQ.1) THEN
CALL ZOOM(IHSID, AXISR, CSIZE)
GO TO 5
ENDIF
IFICHOICE.EQ.2) THEN
CALL RESTO(IHSID, AXISR, CSIZE)
GO TO 5
ENDIF
IFICHOICE.EQ.3) THEN
CALL SCHS(IHSID, AXISR, CSIZE, SCALE, PNAM, PNUM, POINT)
GO TO 5
ENDIF
IFICHOICE.EQ.4) THEN
CONTINUE
ENDIF
RETURN
END
DATA C(9,2)/  EXIST / DATA C(9,3)/ ENTER VALID FILE / DATA C(9,4)/ (UPPER CASE) / DATA C(10,1)/  SELECT / DATA C(10,2)/ DESIRED OPTION / DATA C(10,3)/  END OF PROGRAM / DATA C(10,4)/  FOR NOW / DATA C(10,5)/ TO QUIT / DATA C(11,1)/  INPUT LOCATION / DATA C(11,2)/ DIRECTLY / DATA C(11,3)/ (ENTER X LOCATION) / OR / DATA C(11,5)/ USE VALUATORS / DATA C(12,1)/  ENTER THE NAME OF / DATA C(12,2)/ THE POINT OR / DATA C(12,3)/ USE DEFAULT / DATA C(12,4)/  TO QUIT / DATA C(12,5)/  NUMBER OF POINTS / DATA C(13,1)/  POINT NAME / DATA C(13,2)/ ALREADY EXISTS / DATA C(13,3)/  CHOOSE DEFAULT / DATA C(14,1)/  ENTER NEW NAME OR / DATA C(14,2)/  CAN NOT BE USED / DATA C(14,3)/  RE-ENTER NAME OR / DATA C(14,4)/  CHOOSE DEFAULT / DATA C(14,5)/ NUMBER OF POINTS / DATA C(15,1)/  MAXIMUM ALLOWABLE / DATA C(15,2)/  MODIFICATION / DATA C(15,3)/  RE-ENTER DATA / DATA C(15,4)/  TO QUIT / DATA C(16,1)/  POINT IS OUTSIDE / DATA C(16,2)/ THE PREDEFINED / DATA C(16,3)/  AXIS RANGE / DATA C(16,4)/ RE-ENTER AXIS OR / DATA C(16,5)/ RE-ENTER POINT / DATA C(17,1)/  PICK POINT TO / DATA C(17,2)/ JOIN / DATA C(17,3)/ FOR POSITION / DATA C(17,4)/ INQUIRY / DATA C(17,5)/  NUMBER OF POINTS / DATA C(18,1)/  X LOCATION / DATA C(18,2)/  Y LOCATION / DATA C(18,3)/  Z LOCATION / DATA C(18,4)/  LOCATION / DATA C(18,5)/  MODIFICATION / DATA C(19,1)/  INPUT / DATA C(19,2)/  RE-ENTER DATA / DATA C(19,3)/  TO QUIT / DATA C(19,4)/  TO QUIT / DATA C(19,5)/  MODIFICATION / DATA C(20,1)/ ENTER POINT OR / DATA C(20,2)/ RETURN TO RE-ENTER / DATA C(20,3)/ POINT DATA / DATA C(20,4)/  RE-ENTER DATA / DATA C(20,5)/  TO QUIT / DATA C(21,1)/ PICK POINT TO / DATA C(21,2)/ DELETE / DATA C(21,3)/  MODIFICATION / DATA C(21,4)/  TO QUIT / DATA C(21,5)/  MODIFICATION / DATA C(22,1)/ PICK POINT OR / DATA C(22,2)/  MODIFICATION / DATA C(22,3)/  TO QUIT / DATA C(22,4)/  TO QUIT / DATA C(22,5)/  MODIFICATION / 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)/  MODIFICATION / DATA C(24,1)/ OPTION NOT YET / DATA C(24,2)/ AVAILABLE / DATA C(24,3)/  PLEASE SELECT / DATA C(24,4)/  NEW OPTION / DATA C(24,5)/  TO QUIT / 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)/ INPUT 
DATA C(27,2)/ X 
DATA C(27,3)/ DIRECTION COSINE 
DATA C(28,1)/ INPUT 
DATA C(28,2)/ Y 
DATA C(28,3)/ DIRECTION COSINE 
DATA C(29,1)/ INPUT 
DATA C(29,2)/ Z 
DATA C(29,3)/ DIRECTION COSINE 
DATA C(29,4)/ 
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(35,1)/ ENTER THE POSITION 
DATA C(35,2)/ OR THE SPHERICAL 
DATA C(35,3)/ JOINT DIRECTLY 
DATA C(35,4)/ 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 

MECHIN PROGRAM LISTING  170
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)/: /
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(52,1)/: PICK A POINT /
DATA C(52,2)/: IN THE POSITIVE /
DATA C(52,3)/: X-Z PLANE /
DATA C(52,4)/: 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)/: 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)/: PICK FIRST CORNER /
DATA C(57,2)/: OF THE /
DATA C(57,3)/: ZOOM AREA /
DATA C(57,4)/: /
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)/: 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)/: PICK A NEW POINT /
DATA C(85,1)/: ENTER A NEW /
DATA C(85,2)/: MECHANISM /
SUBROUTINE SCRNI

INTEGER IMSID, LNGTH1, STRUCT, STROKE, PRFORM, LD, ACLASS1
REAL HGHT1, POS1(2), BOXPTS(8), SHDPTS(8), MATHI(4, 4), ANG, TRAT1(3),
     MATT1(4, 4), MATT2(4, 4), MATT3(4, 4), MATB1(4, 4), TEMP1(4, 4),
     MATB2(4, 4), TRAB1(5), TRAB2(5), x(3), y(3), WIND1(4), VIEW(6),
     POIN1(5), MATRI(4, 4), POD1(2), POD2(2), POD3(2), MHTD, FOMI(2), CSIZE(3),
     APAREA(6), BX2(18), SH2(16)
CHARACTER*7 TEXT1, CLEAR
CHARACTER*30 MSG, DESC1, DESC2, DESC3

C DATA TEXT1/"MECHIN "/ OR QUIT ;/
C DATA LNGTH1/19/ COMMAND LENGTH AND LINE POSITIONS
C DATA POS1/..05,.7/ DATA POS2/..05,.6/ DATA POS3/..05,.5/ DATA POS4/..05,.4/
C DATA POS5/..05,.3/ C DATA PRFORM/2/ FLAG FOR UPDATE
C DATA WINDOC/0.00,1.00,0.00,1.00/ DATA VPC/0.01,0.2315,0.67,0.975/
*********************************************************************
* DELETE ALL PREVIOUS COMMENTS
*********************************************************************
C CALL GPATEST(5)
*********************************************************************
* OPEN STRUCTURE FOR COMMAND
*********************************************************************
C CALL GPPOST(5) OPEN A STRUCTURE
C CALL GPTXCI(2) SET TEXT COLOR TO YELLOW
C CALL GPASHC(1.0) SET ANNOTATION SIZE SCALE FACTOR
C CALL GPAN21POSL1, LNGTH1, CNUM, 1) CALL GPAN21POSL2, LNGTH1, CNUM, 2)
C CALL GPAN21POSL3, LNGTH1, CNUM, 3) CALL GPAN21POSL4, LNGTH1, CNUM, 4)
C CALL GPAN21POSL5, LNGTH1, CNUM, 5) CLOSE STRUCTURE
C CALL GPCLST
*********************************************************************
* SCREEN DISPLAY--------- VIEW INITIALIZATION GOES TO SETUP
*********************************************************************
C UPDATE THE WORKSTATION
CALL GPUPHS(IMSID, PRFORM)
RETURN
END

SUBROUTINE SCRNI1(IMSID, CSIZE)

*********************************************************************
----- SUBROUTINE SCRNI1 ----- * *
* *
* THIS SUBROUTINE IS USED TO DISPLAY SCREEN 1 --------- * *
* MECHIN TITLE SCREEN * *
*********************************************************************

INTEGER IMSID, LNGTH1, STRUCT, STROKE, PRFORM, LD, ACLASS1
REAL HGHT1, POS1(2), BOXPTS(8), SHDPTS(8), MATHI(4, 4), ANG, TRAT1(3),
     MATT1(4, 4), MATT2(4, 4), MATT3(4, 4), MATB1(4, 4), TEMP1(4, 4),
     MATB2(4, 4), TRAB1(5), TRAB2(5), x(3), y(3), WIND1(4), VIEW(6),
     POIN1(5), MATRI(4, 4), POD1(2), POD2(2), POD3(2), MHTD, FOMI(2), CSIZE(3),
     APAREA(6), BX2(18), SH2(16)
CHARACTER*7 TEXT1, CLEAR
CHARACTER*30 MSG, DESC1, DESC2, DESC3

C DATA TEXT1/"MECHIN "/ TITLE TEXT, LENGTH, HEIGHT, AND POSITION
C DATA CLEAR/"/
C DATA LNGTH1/7/ DATA HGHT1/.3/ DATA POS1/-.93,-.162/
C DATA DESC1/"A GRAPHICAL PREPROCESSOR FOR " DESCRIPTION LENGTH, HEIGHT, AND POSITION
C DATA DESC2/"SPATIAL MECHANIS/
C DATA DESC3/"ANALYSIS AND SYNTHESIS \/
DATA LD/30/
DATA HGT/D,.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 POSH/-.97,-.92/
DATA ACLASS/1/

C

IDENTITY MATRIX 1
DATA MATR1/1.,0.,0.,0.,1.,0.,0.,0.,1.,0.,0.,0.,1.,0.,0.,0.,1./
IDENTITY MATRIX 2
DATA MATR2/1.,0.,0.,0.,1.,0.,0.,0.,1.,0.,0.,0.,1.,0.,0.,0.,1./
IDENTITY MATRIX TRANS. TO
DATA MATT1/1.,0.,0.,0.,1.,0.,0.,0.,1.,0.,0.,0.,1.,0.,0.,0.,1./
IDENTITY MATRIX TRANS. BACK
DATA MATT1/1.,0.,0.,0.,1.,0.,0.,0.,1.,0.,0.,0.,1.,0.,0.,0.,1./
IDENTITY MATRIX TRANS. TO
DATA MATT2/1.,0.,0.,0.,1.,0.,0.,0.,1.,0.,0.,0.,1.,0.,0.,0.,1./
IDENTITY MATRIX TRANS. BACK
DATA MATT2/1.,0.,0.,0.,1.,0.,0.,0.,1.,0.,0.,0.,1.,0.,0.,0.,1./

TITLE STRUCTURE NUMBER
DATA STRUCT/1/
DATA STROKE/3/
DATA PRFORM/2/

STROKE PRECISION VALUE
FLAG FOR UPDATE
GreY TITLE BOX
DATA BOXPTS/-.93,-.195, .93,-.195,.93,.150,-.93,.150/
DARK GREY SHADOW BOX
DATA SHDPTS/-.88,-.245,0.98,-.245,0.98,.100,-.88,.100/
GREY PICK BOX
DATA BX/-95,-.94,-.95,-.94,-.88,-.95,-.88/
GREY PICK SHADOW
DATA SH2/-92,-.97,-.32,-.97,-.32,-.91,-.92,-.91/

ROTATION ANGLE
DATA ANG/0.02/

SHADOW TRANSLATION
DATA TRAT1/0.,0.0725,0./
DATA TRAT2/0.,0.0725,0./
DATA TRAB1/0.,-0.0725,0./
DATA TRAB2/0.,-0.0725,0./
DATA X/1.0,0./
DATA Y/0,1.0/
DATA IDEV/I/

*******************************************************************************
* OPEN STRUCTURE FOR THE TITLE
*******************************************************************************
C CALL GPOPST(1)
CALL GMLX3 MATR1,3)
CALL GIS(2)
CALL GPICI(4)
CALL GP gg2(1,4,2,SHDPTS)
CALL GPCLST

C CALL GPOPST(2)
CALL GMLX3(MATR2,3)
CALL GIS(2)
CALL GPICI(3)
CALL GP gg2(1,4,2,BOXPNTS)
CALL GPTXPR(STROKE)
CALL GPCHH(HGT1)
CALL GPTXI(2)
CALL GPACFO(1MSID,1,11)
C CALL GPTXFO(11)
CALL GPT2(POS1,LNGTHI,TEXT)
CALL GPCLST

MECHIN PROGRAM LISTING 173
CALL GPPOPST(3) OPEN A STRUCTURE FOR DESCRIPTION
CALL GPTXPR(STROKE) SET TEXT PRECISION TO STROKE
CALL GPCHH(HGHTD) SET CHARACTER HEIGHT
CALL GPTXCI(1) SET TEXT COLOR
CALL GPACFD(I,MSID,1,3) ACTIVATE TEXT FONT
CALL GPTXF0(I,3) SET TEXT FONT
CALL GPTX2(POD1,LD,DESC1) INSERT TEXT PRIMITIVE
CALL GPTX2(POD2,LD,DESC2)
CALL GPTX2(POD3,LD,DESC3)
CALL GPCLST CLOSE STRUCTURE

CALL GPPOPST(4) OPEN A STRUCTURE FOR CHOICES
CALL GPISI(2) SET SOLID INTERIOR STYLE FOR GREY BOX
CALL GPICI(4) SET COLOR FOR INTERIOR OF SHADOW BOX
CALL GGPG21(4,2,SH2) DRAM GREY SHADOW BOX
CALL GPICI(3) SET COLOR FOR INTERIOR OF GREY BOX
CALL GGPG21(4,2,BX2) DRAM GREY TITLE BOX
CALL GPADCN(1,ACLASS) INSERT CLASS NAME TO ALLOW PICKING MENU ITEM
CALL GPTXCI(2) SET TEXT COLOR
CALL GPANSC(1.0) SET ANNOTATION SIZE SCALE FACTOR
CALL GPPKID(1) SET PICK IDENTIFIER
CALL GPAN2(POS,M,LNGTH,M,MSG) DRAW GREY TITLE BOX
CALL GPCLST CLOSE STRUCTURE

*********************************************************************
* SET STRING PARAMETERS
*********************************************************************
CALL GPSTMO(IMSID,1,1,2) MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
PAREA(1)=0. PAREA(2)=CSIZE(1)
PAREA(3)=0. PAREA(4)=CSIZE(2)
PAREA(5)=0. PAREA(6)=CSIZE(3)
CALL GPINST(IMSID,1,7,CLEAR,1,PAREA,7,7,32,DATA) INITIALIZE PICK PATH
CALL GPSTMO(IMSID,1,3,2)

*********************************************************************
* SCREEN DISPLAY
*********************************************************************
CALL GPUPHS(HSID,PERM) UPDATE THE WORKSTATION

*********************************************************************
* GO ON OR CONTINUE ROTATION
*********************************************************************
CALL GPTRL5(TRA1,MATT1)
CALL GPTRL5(TRAB1,MATT1)
CALL GPTRL5(TRA2,MATT2)
CALL GPTRL5(TRAB2,MATT2)

CALL GPPDEV(I,0.0,IMSID,ICLS,IDEV) REQUEST INPUT
100 CALL GPAN(0.0,IMSID,ICLS,IDEV)

IF (ICLS.EQ.6) THEN
  CALL GPTRL5(TRA1,MATT1)
  CALL GPTRL5(TRAB1,MATT1,TEMP)
  CALL GPTRL5(TRA2,MATT2)
  CALL GPTRL5(TRAB2,MATT2)
  CALL GPPDEV(I,0.0,IMSID,ICLS,IDEV)
ENDIF

CALL GPCLST CLOSE STRUCTURE

MECHIN PROGRAM LISTING 174
CALL GPROTX(ANG,MATR2)
CALL GPCMT31(MATR2,MATR2,TEMP)
CALL GPCMT31(TMP,MATR2,MATR2)
CALL GPSTI(2)
CALL GPEP(I)
CALL GPDE
CALL GPMLX3(MATR2,3)
CALL GPCLST

C TRAVERSE ALL ACTIVE VIEWS
CALL GUPWSI(INSID,PRFOM)
ANG=ANG+0.12
GO TO 100

C

ENDIF

C

DELETE ALL STRUCTURES

200 CALL GPST(1)
CALL GPST(2)
CALL GPST(3)
CALL GPST(4)

C DEACTIVATE PICK
CALL GPSSKMO(INSID,PPRM,PRFOM)
RETURN
END

SUBROUTINE SCRN2

SUBROUTINE SCRN2(INSID,CSIZE,CHOICE)

*********************************************************************
* ----- SUBROUTINE SCRN2 ----- *
* THIS SUBROUTINE IS USED TO DISPLAY SCREEN 1 "--------*
* ANALYSIS OR SYNTHESIS CHOICE "*

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

INTEGER INSID,LENG,LENGT,STRUCT,STROKE,PRFOM,PPRM(3),ACLASS(1),
&CHOICE
REAL HGHT,POSN,BOXPTS,SHDPTS,CSIZE,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 POSN/-55,100/
DATA BOXPTS/-.93,-.195,.93,-.195,.93,.150/
DATA SHDPTS/-.88,-.245,.98,-.245,.98,.100/
POSN1=-.23
POSN2=-.010

*********************************************************************
* OPEN STRUCTURE FOR THE CHOICE BOX

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

CALL GPSTI(1)
CALL GPST(2)
CALL GPSTI(4)

C DRAW SHADOW BOX
CALL GPSSG2(1,4,2,SHDPTS)

C DRAW GREY BOX
CALL GPSSG2(1,4,2,SHDPTS)
CALL GPICI(3)
CALL GPPG2(1,4,2,BOXPTS)
CALL GPTXC2(2)
CALL GPAHSC(2.0)
CALL GPAN2(POSN,LENG,TITLE)
CALL GPCLST

CALL GPPOPST(2)
CALL GPADC2(1,AClass)
CALL GPTXC2(2)
CALL GPAHSC(1.0)
DO 100 I=1,2
CALL GPPKID(I)
CALL GPAN2(POSN,LENG,TMEN1(I))
POSN(2)=POSN(2)-.09
100 CONTINUE
CALL GPCLST

***********************************************************************
** SCREEN DISPLAY
***********************************************************************
CALL GPUPHS(IHSID,PRFORM)

***********************************************************************
** REQUEST PICK INPUT
***********************************************************************
CALL GPPKF(IHSID,1,1,AClass)
CALL GPPKMO(IHSID,1,1,2)
DEFINE PICK AREA
PAREA(1)=(BOXPTS(1)+1./2.)*MCLUE(1)
PAREA(2)=(BOXPTS(2)+1./2.)*MCLUE(1)
PAREA(3)=(BOXPTS(3)+1./2.)*MCLUE(2)
PAREA(4)=(BOXPTS(4)+1./2.)*MCLUE(2)
PAREA(5)=0.
PAREA(6)=(BOXPTS(5)+1./2.)*MCLUE(3)
CALL GPINPK(IHSID,1,0,PPATH,1,PAREA,0.0,DATA,1)
PLACE PICK DEVICE IN THE EVENT MODE
CALL GPPKMO(IHSID,1,3,2)
TRaverse all active viEws
CALL GPUPHS(IHSID,PRFORM)

***********************************************************************
** AWAIt A PICK EVENT TO OCCUR AND RETRIEVE IT
***********************************************************************
CALL GPAHEV(I1000,IHS,ICLA,IDEV)
IF(ICLA.EQ.5)THEN
GET PICK
CHOICE=PPATH(2)
DEACTIVATE PICK
CALL GPEST(1)
CALL GPEST(2)
RETURN
ELSE
IF NOT A PICK, GO TO AWAIt A EVENT
END IF

C (TEmPORARY STOP REMOVE AND FIX RETURN)
C CALL GPCHMO(IHSID,1,1,1)
C CALL GPCHC(IHSID,1,LISTS,1CHO)

MECHIN PROGRAM LISTING 176
SUBROUTINE SCRN3

*****************************************************************************
* ------ SUBROUTINE SCRN3 ------ *
*****************************************************************************

THIS SUBROUTINE IS USED TO DISPLAY SCREEN 3

ANALYSIS BACKGROUND SCREEN

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

INTEGER IMSID, LENG, LENCT, STROKE, PRFORM
REAL POSNT12J, BOX118J, BOX2(8), SHD218J, BOX3(8), SHD3(8), BOX4(8),
& BOXS1(8), BOX5(8), POSCT12
CHARACTER*35 TITLE
CHARACTER*8 CTITL

DATA CTITL/'ANALYSIS'/
DATA LENCT/8/
DATA POSCT/.882, .89/

DATA PRFORM/2/
DATA BOX1/.48, -.98, .98, -.98, .98, .48, -.48, .48/
DATA BOX2/.98, .48, .48, .98, .98, .48, -.48, .48/
DATA BOX3/.98, .37, .37, .98, -.98, .98, -.98, .98/
DATA BOX4/.48, .37, .37, .48, -.48, .48, -.48, .48/
DATA BOX51/.96, .91, -.96, .91, -.96, .91, -.96, .91/

DO 100 I=1, 7, 2
SHD2(I)=BOX2(I)+.025
SHD3(I)=BOX3(I)+.025
100 CONTINUE

DO 200 I=2, 8, 2
SHD2(I)=BOX2(I)-.025
SHD3(I)=BOX3(I)-.025
200 CONTINUE

*****************************************************************************
* OPEN STRUCTURE FOR THE DISPLAY BOX
*****************************************************************************

CALL GPOPST(1)
CALL GPIS(2)
CALL GPICI(3)
CALL GPPG2(1, 4, 2, BOX1)
CALL GPCLST

*****************************************************************************
* OPEN STRUCTURE FOR THE MENU BOX
*****************************************************************************

CALL GPOPST(2)
CALL GPIS(2)
CALL GPICI(4)
CALL GPPG2(1, 4, 2, SHD2)
CALL GPICI(3)
CALL GPPG2(1, 4, 2, BOX2)
CALL GPCLST
**OPEN STRUCTURE FOR COMMAND BOX**

```
CALL GPOPST(3)    SET SOLID INTERIOR STYLE FOR GREY BOX
CALL GPIS(2)     SET COLOR FOR INTERIOR OF SHADOW BOX
CALL GPICI(4)    DRAW GREY SHADOW BOX
CALL GPPG2(1,4,2,SHD3)    SET COLOR FOR INTERIOR OF GREY BOX
CALL GPICI(3)    DRAW GREY COMMAND BOX
CALL GPPG2(1,4,2,BOX3)    SET COLOR FOR WHITE OF COMMAND TITLE
CALL GPICI(1)    DRAW WHITE BOXES IN TITLE
CALL GPPG2(1,4,2,BOX5)    SET TEXT COLOR TO WHITE
CALL GPICI(3)    DRAW TITLE TEXT
CALL GPAN2(POSCT,LENCT,CTITL)    CLOSE STRUCTURE
CALL GPCLST
```

**OPEN STRUCTURE FOR VALUATOR BOX**

```
CALL GPOPST(4)    SET SOLID INTERIOR STYLE FOR GREY BOX
CALL GPIS(2)     SET COLOR FOR INTERIOR OF GREY BOX
CALL GPICI(3)    DRAW GREY VALUATOR BOX
CALL GPPG2(1,4,2,BOX4)    SET COLOR FOR WHITE OF COMMAND TITLE
CALL GPICI(1)    DRAW WHITE BOXES IN TITLE
CALL GPPG2(1,4,2,BOX5)    SET TEXT COLOR TO WHITE
CALL GPICI(3)    DRAW TITLE TEXT
CALL GPAN2(POSCT,LENCT,CTITL)    CLOSE STRUCTURE
CALL GPCLST
```

**SCREEN DISPLAY**

```
CALL GPUPWS(IHSID,PRFORM)    UPDATE THE WORKSTATION

RETURN
END
```

---

**SUBROUTINE SCRN4**

```
SUBROUTINE SCRN4(IHSID)
************************************************************
* ------ SUBROUTINE SCRN4 ------ *
* THIS SUBROUTINE IS USED TO DISPLAY SCREEN 4 ---------- *
* SYNTHESIS BACKGROUND SCREEN *
************************************************************

INTEGER IHSID,LENGT,LENCT,STROKE,PRFORM

REAL POSNT(2),BOX1(8),BOX2(8),SHD2(8),BOX3(8),SHD3(8),BOX4(8),
     BOX5(18),BOX51(8),POSCT(2)

CHARACTER*35 TITLE
CHARACTER*9 CTITL

C TITLE TEXT, LENGTH, HEIGHT, AND POSITION
DATA CTITL/'SYNTHESIS'/
DATA LENCT/9/
DATA POSCT/-0.850,.89/
C FLAG FOR UPDATE
DATA PRFORM/2/
C GREY MAIN SCREEN BOX
DATA BOX1/-0.48,-0.96,0.96,-0.96,0.98,-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/
GREY COMMAND BOX
DATA BOX3/-0.98,0.37,-0.525,0.57,-0.525,0.98,-0.98,0.98/
GREY POSITION BOX
DATA BOX4/-0.48,0.52,-0.02,1.0,0.52,-0.02,0.98,-0.48,0.98/
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/

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
C CALL GPOPST(1) SET SOLID INTERIOR STYLE FOR GREY BOX
C CALL GPIS(2) SET COLOR FOR INTERIOR OF GREY BOX
C CALL GPIC(3) DRAW GREY DISPLAY BOX
C CALL GPPG2(1,4,2,BOX1) CLOSE STRUCTURE
C CALL GPCLST

*********************************************************************
* OPEN STRUCTURE FOR THE MENU BOX
*********************************************************************
C OPEN A STRUCTURE
C CALL GPOPST(2) SET SOLID INTERIOR STYLE FOR GREY BOX
C CALL GPIS(2) SET COLOR FOR INTERIOR OF SHADOW BOX
C CALL GPIC(4) DRAW GREY SHADOW BOX
C CALL GPPG2(1,4,2,SHD2) SET COLOR FOR INTERIOR OF GREY BOX
C CALL GPIC(3) DRAW GREY TITLE BOX
C CALL GPPG2(1,4,2,BOX3) CLOSE STRUCTURE
C CALL GPCLST

*********************************************************************
* OPEN STRUCTURE FOR COMMAND BOX
*********************************************************************
C OPEN A STRUCTURE
C CALL GPOPST(3) SET SOLID INTERIOR STYLE FOR GREY BOX
C CALL GPIS(2) SET COLOR FOR INTERIOR OF SHADOW BOX
C CALL GPIC(4) DRAW GREY SHADOW BOX
C CALL GPPG2(1,4,2,SHD3) SET COLOR FOR INTERIOR OF GREY BOX
C CALL GPIC(3) DRAW GREY COMMAND BOX
C CALL GPPG2(1,4,2,BOX3) SET COLOR FOR WHITE OF COMMAND TITLE
C CALL GPIC(1) DRAM WHITE BOXES IN TITLE
C CALL GPPG2(1,4,2,BOX4)
C CALL GPTXC(1) SET TEXT COLOR TO WHITE
C CALL GPAHSC(1.00) SET ANNOTATION SIZE SCALE FACTOR
C CALL GPAN(1,POSCT,LENCT,CTITL) DRAW TITLE TEXT
C CALL GPCLST

*********************************************************************
* OPEN STRUCTURE FOR VALUATOR BOX
*********************************************************************
C OPEN A STRUCTURE
C CALL GPOPST(4) SET SOLID INTERIOR STYLE FOR GREY BOX
C CALL GPIS(2) SET COLOR FOR INTERIOR OF GREY BOX

MECHIN PROGRAM LISTING 179
CALL GPICI(3)
DRAW GREY VALUATOR BOX
CALL GPPG2(1,4,2,BOX4)
CLOSE STRUCTURE
CALL GPCLST

*********************************************************************
* SCREEN DISPLAY *
*********************************************************************
CALL GPUPWS(IHSID,PRFORM)

RETURN
END

SUBROUTINE SDATA

SUBROUTINE SDATA(IHSID,AXISR,CSIZE,PNAM,PNUM,POINT,SCALE)

INTEGER IHSID,CHOICE,PNUM(30,4)
CHARACTER*7 PNAM(30)
REAL AXISR,CSIZE(3),POINT(30,14),SCALE

CALL GPARVIIHSID,4,11,0.
CALL GPARVIIHSID,4,12,0.
CALL GPARVIIHSID,4,13,0.
CALL GPARVIIHSID,4,14,0.
CALL GPARVIIHSID,4,15,0.

********************************************************************
* ASSOCIATE STRUCTURES WITH VIEWS *
********************************************************************
C POSITION ICON CALL GPARVIIHSID,5,9,0.
CALL GPARVIIHSID,6,16,0.
CALL GPARVIIHSID,4,11,0.
C POINTS AND NAMES CALL GPARVIIHSID,4,12,0.
C X AXES CALL GPARVIIHSID,4,13,0.
C Y AXES CALL GPARVIIHSID,4,14,0.
C Z AXES CALL GPARVIIHSID,4,15,0.

********************************************************************
* INQUIRE TYPE OF DATA INPUT *
********************************************************************
10 CALL SCMMANIIHSID,7J CALL SMENUlrnSID,CSIZE,2,CHOICEJ
IF(CHOICE.EQ.1) THEN
BODY POSITIONS CALL SPTIIHSID,AXISR,CSIZE,PNAM,PNUM,POINT,SCALEJ
go to 10
ENDIF
IF(CHOICE.EQ.2) THEN
CHANGE VIEW CALL SCHVIEIIHSID,AXISR,CSIZE,SCALE,PNAM,PNUM,POINTJ
go to 10
ENDIF
IF(CHOICE.EQ.4) THEN
RETURN END OF DATA INPUT
ENDIF
RETURN
END

SUBROUTINE SDELP

SUBROUTINE SDELP(IHSID,CSIZE,AXISR,PNAM,PNUM,POINT,SCALE)

180
MECHIN PROGRAM LISTING
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 LEQN/16/ DATA POSN/0.05,2.2/
DATA C/C CCCC C/
DATA DATA/32,0,0,' /
DATA PRFORM/2/ DATA ACLASS /4/ INCLUSION CLASS LIST

REQUEST PICK INPUT

CALL GPPKF(IWSID,1,1,ACLASS,0,ACLASS) CALL GPPKMO(IWSID,1,1,2) DEFINE PICK AREA
PAREAl1=((-98+1/2)*CSIZE(1) PAREAl2=((-98+1/2)*CSIZE(1) PAREAl3=((-98+1/2)*CSIZE(1) PAREA21=((-48+1/2)*CSIZE(2) PAREA22=((-48+1/2)*CSIZE(2) PAREA23=((-48+1/2)*CSIZE(3)

CALL GPPNP(IWSID,1,0,PPATH,1,PAREA,0,DATA,1) PLACE PICK AREA IN THE EVENT MODE
CALL GPPKMO(IWSID,1,3,2) TRAVERSE ALL ACTIVE VIEWS
CALL GPPKF(IWSID,21) GET PICK OR RETURN COMMAND

INCLUDE A RETURN PICK

CALL GPOPST(IWSID,1) CALL GPADCN(I,ACLASS) CALL GPPKID(I,0) CALL GPTXC(I,2) CALL GPAHSCI(I,0) CALL GPAN2(POSN,LEQN,RETURN) CALL GPCLST CALL GPARY(IWSID,3,6,0.) CALL GPUP(IWSID,3,1,1) CALL GPUPM(IWSID,PRFORM)

WAIT A PICK EVENT TO OCCUR AND RETRIEVE IT

ELSE IF(IPATH.EQ.2) THEN IF NO PICK, RETURN
C CHOICE IS A PICK PATH
C DEACTIVATE POINT
C DO 300 I=1,14 POINTI(I)=0.
300 CONTINUE
C CALL SRDRMP(IWSID,AXISR,PNUM,PNUM,RETURN)
C REINTERPRET POINTS
C PRINT MECHANICAL LISTING
C GO TO 200
C ENDIF

IF NOT A PICK, GO TO WAIT AN EVENT
SUBROUTINE SDRAWP

SUBROUTINE SDRAWP(IHSID,AXISR,PNAME,NUMP,POINT,SC)

*****************************************************************************
* ----- SUBROUTINE DRAHJ ----- *
*****************************************************************************
* THIS SUBROUTINE IS USED TO DRAW A JOINT IN THE MAIN WINDOW *
*****************************************************************************

INTEGER IHSID,NUMP,CLASS(1),CLASS(1),JTYPE
REAL LOCAT(3),LOCATI(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),Z(3),NEWX(3),NEWY(3),NEWYU(3),NEWXU(3),NEWY(3),
& P(3),AXI(3),XAXI(3),ZAXI(3),X(6),Y(6),Z(6),LX(6),LY(6),SZ(6),SC(6)

CHARACTER*7 JNAME
CHARACTER*2 XN,YN,ZN
DATA XN /'X' /
DATA YN /'Y' /
DATA ZN /'Z' /
DATA CLASS(4)/
DATA CLASS(4)/
DATA MAT1(1),MAT2(1),MAT3(1),MAT4(1)
DATA MATSC(1),MATSC(1),MATSC(1),MATSC(1)
DATA MATRIX(1),MATRIX(1),MATRIX(1),MATRIX(1)
DATA P(1),P(1),P(1),P(1)
& 0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.

LOCAT(1)=POINT(NUMP,1)
LOCAT(2)=POINT(NUMP,2)
LOCAT(3)=POINT(NUMP,3)
LOCAT(1)=-.08*AXISR
LOCAT(1)=0.0
XAXI(1)=POINT(NUMP,4)
XAXI(2)=POINT(NUMP,5)
XAXI(3)=POINT(NUMP,6)
YAXI(1)=POINT(NUMP,7)
YAXI(2)=POINT(NUMP,8)
YAXI(3)=POINT(NUMP,9)
ZAXI(1)=POINT(NUMP,10)
ZAXI(2)=POINT(NUMP,11)
ZAXI(3)=POINT(NUMP,12)
X(1)=0.0
X(1)=0.0
X(1)=XAXI(1)*AXISR/10.)*2.4
X(1)=XAXI(2)*AXISR/10.)*2.4
X(1)=XAXI(3)*AXISR/10.)*2.4
Y(1)=0.0
Y(1)=0.0
Y(1)=YAXI(1)*AXISR/10.)*2.4
Y(1)=YAXI(2)*AXISR/10.)*2.4
Y(1)=YAXI(3)*AXISR/10.)*2.4
Z(1)=0.0
Z(1)=0.0
Z(1)=ZAXI(1)*AXISR/10.)*2.4
Z(1)=ZAXI(2)*AXISR/10.)*2.4
Z(1)=ZAXI(3)*AXISR/10.)*2.4

MECHIN PROGRAM LISTING

182
C CALCULATE AXIS TRANSLATION MATRIX
CALL GPTRL31LOCAT,MAT1

SCALE1=AXISR*SC SCALE2=AXISR*SC SCALE3=AXISR*SC

SC1=SC*.9 SC2=SC*.9 SC3=SC*.9


108 CALL GPSC31SC2,MSC2) CALL GPSC31SC1,MATSC1 CALL GPSC31SC3,MATRIX1

******************************************************************** * OPEN STRUCTURE AND DRAW POINT (MAKE PICKABLE) * ********************************************************************
C OPEN STRUCTURE 12
C CALL GPOPST1 CALL GPADCN1,CLASS) CALL GPPL318,3LPJ C OSE STRUCTURE CALL GPCLST
IF(X1.EQ.0.0.AN.D.X2.EQ.0.0.AN.D.X3.EQ.0.0.G0 TO 1000

C OPEN STRUCTURE 13 (X AXIS)
C CALL GPOPST1 CALL GPADCN1,CLASS) CALL GPPL318,3LPJ C OSE STRUCTURE CALL GPCLST
IF(Y1.EQ.0.0.AN.D.Y2.EQ.0.0.AN.D.Y3.EQ.0.0.GO TO 1000

C OPEN STRUCTURE 14 (Y AXIS)
CALL GPOPST(14) 
CALL GPADCN(1,CLASS) 
INSERT CLASS NAME TO ALLOW PICKING AXES (CLASS 4) 
CALL GPPKID(HNUP) 
SET PICK I.D. 
CALL GPMX3(MATI,3) 
CALL GPMX3(MATSC,1) 
SET TRANSFORMATION 
CALL GPLT(1) 
SET LINE TYPE SOLID 
CALL GPPLCI(5) 
SET LINE COLOR 
CALL GPPL3(2,3,Y1) 
ASSOCIATE Y AXIS 
CALL GPAHSCI(1) 
SET CHARACTER HEIGHT SCALE FACTOR 
CALL GPTXCII(5) 
TEXT COLOR INDEX 
CALL GPAN3(L1,Y1,YN) 
ASSOCIATE TEXT 
CALL GPCLST 
CLOSE STRUCTURE 
OPEN STRUCTURE 15 (Z AXIS) 
CALL GPOPST(15) 
INSERT CLASS NAME TO ALLOW PICKING AXES (CLASS 4) 
CALL GPADCN(1,CLASS) 
CALL GPPKID(HNUP) 
SET TRANSFORMATION 
CALL GPMX3(MATI,3) 
CALL GPMX3(MATSC,1) 
SET TRANSFORMATION 
CALL GPLT(1) 
SET LINE TYPE SOLID 
CALL GPPLCI(5) 
SET LINE COLOR 
CALL GPPL3(2,3,Z1) 
ASSOCIATE Z AXIS 
CALL GPAHSCI(1) 
SET CHARACTER HEIGHT SCALE FACTOR 
CALL GPTXCII(5) 
TEXT COLOR INDEX 
CALL GPAN3(LZ,Y1,ZN) 
ASSOCIATE TEXT 
CALL GPCLST 
CLOSE STRUCTURE 
1000 CALL GPUPHS(HMSID,2) 
RETURN 
END 

SUBROUTINE SETUP

SUBROUTINE SETUP(HMSID,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 HMSID,CSTART,CNUM,ASIZE(3),ERRIND,CHOICE 
REAL COLORS(27),CSIZE(3),UNITS,MIND(4),VIEW(4),MINDOC(4),VPC(4), 
&MINDOM(4),VPM(4),JOINT 
CHARACTER*8 ERFILE,HSTYPE,CONNID 
CHARACTER*7 PNAM,PNDEF,JNAM,JNDEF,LNAM,LNDEF,INAM,SPNAM 
C ERROR FILE
DATA ERFIL/'SYSPRINT'/
DATA HSTYPE/'5080' / WORKSTATION TYPE
DATA CONNID/'IBM5080' / CONNECTION IDENTIFIER
DATA HSTYPE/'3250' / WORKSTATION TYPE
DATA CONNID/'IBM3250' / CONNECTION IDENTIFIER
DATA HSTYPE/'GDDM' / WORKSTATION TYPE
DATA CONNID/* / CONNECTION IDENTIFIER
DATA CSTART/0/ NUMBER OF COLORS IN TABLE (BEHIND BACKGRD)
DATA CNUM/9/ COLOR TABLE
DATA COLORS/ 0., 0., 1., MAXIMUM WHITE
& 1., 1., 1., MAXIMUM YOELD
& 1., 1., 0., BACKGROUND GRY
& .40, .40, .40, DARK GREY
& .25, .25, .25, MAXIMUM BLACK
& 0., 0., 0., TURQOISE
& 0., 1., 1., RED
& 1., 0., 0., MAGENTA
& 1., 0., 1.,
DATA WIN/1.0,1.0,1.0,1.0/ WINDOW AND VIEWPORT OF TITLE SCREEN
DATA VIEW/0.0,0.0,0.0,0.0/ WINDOW AND VIEWPORT OF COMMAND BOX
DATA VPC/.01,2375,0.67,0.975/ WINDOW AND VIEWPORT OF MENU BOX
DATA VMOD/0.00,1.00,0.00,1.00/ WINDOW AND VIEWPORT OF MENU BOX
DATA VMOD/.01,2375,0.67,0.975/ WINDOW AND VIEWPORT OF MENU BOX
DATA WMOD/0.00,1.00,0.00,2.0132/ WINDOW AND VIEWPORT OF MENU BOX
DATA WMOD/.01,2375,0.67,0.975/ WORKSTATION IDENTIFIER
INSID=1 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'
MECHIN PROGRAM LISTING
OPEN AND INITIATE PHIGS

OPEN PHIGS
CALL GPOPHEFFILE,OJ
CALL GPOPHSIIHSID,CONNID,HSTYPEJ
LOAD COLOR TABLE ON THE WORKSTATION
CALL GPCRIHSID,CSTART,CNUM,COLORSJ

FILL POINT AND JOINT ARRAYS

DO 10 I=1,100
PNAMIIJ='BBBBBBB'
10 CONTINUE
DO II I=1,20
INAMIIJ='BBBBBBB'
11 CONTINUE
DO 12 I=1,30
LNAMIIJ='BBBBBBB'
12 CONTINUE
DO 13 I=1,20
SPNAMIIJ='BBBBBBB'
13 CONTINUE
DO 14 I=1,30
JOINT(I,J)=0.0
14 CONTINUE

GET DISPLAY SURFACE SIZE

INQUIRE DISPLAY SURFACE SIZE
CALL GPQADSIIHSID,ERRIND,UNITS,CSIZE,ASIZEJ

SET UP VIEWS AND ASSIGN STRUCTURES TO VIEWS

SET UP WINDOW FOR VIEW 1
CALL GPVMP2IHSID,1,HIND1,VPCJ
CALL GPVCHIIHSID,1,1,1,1,1,1/2)
ASSOCIATE THE STRUCTURE 1 WITH A VIEW 1
CALL GPARVIIHSID,1,1,0.
ASSOCIATE THE STRUCTURE 2 WITH A VIEW 1
CALL GPARVIIHSID,1,2,0.
ASSOCIATE THE STRUCTURE 3 WITH A VIEW 1
CALL GPARVIIHSID,1,3,0.
ASSOCIATE THE STRUCTURE 4 WITH A VIEW 1
CALL GPARVIIHSID,1,4,0.
ASSOCIATE THE STRUCTURE 8 WITH A VIEW 1
CALL GPARVIIHSID,1,8,0.
ASSOCIATE THE STRUCTURE 10 WITH A VIEW 1
CALL GPARVIIHSID,1,10,0.
SET VIEW PRIORITY OF VIEW1 HIGHER THAN VIEW0
CALL GPVPVHSID,1,0,1.

SET UP WINDOW FOR COMMAND BOX
CALL GPVMP2IHSID,2,HIND2,VPCJ
CALL GPVCHIIHSID,2,1,2,2,1,0,1,1,2)
ASSOCIATE THE STRUCTURE 5 WITH A VIEW 2
CALL GPARVIIHSID,2,5,0.
SET VIEW PRIORITY OF VIEW2 HIGHER THAN VIEW1
CALL GPVPVHSID,2,1,1.

SET UP WINDOW FOR MENU BOX
CALL GPVMP2IHSID,3,HIND3,VPMJ
CALL GPVCHIIHSID,3,2,2,2,1,0,1,1,2)
ASSOCIATE THE STRUCTURE 6 WITH A VIEW 3
CALL GPARVIIHSID,3,6,0.
SET VIEW PRIORITY OF VIEW3 HIGHER THAN VIEW1
CALL GPVPVHSID,3,1,1.

CALL TITLE SCREEN
CALL SCRN1IHSID,CSIZEJ

MECHIN PROGRAM LISTING

187
**SUBROUTINE SINQ**

SUBROUTINE SINQ(IHSID,CSIZE,AXISR,PNAME,PNUM,POINT)

* THIS SUBROUTINE IS USED TO INQUIRE BODY POSITIONS *

INTEGER IHSID,PRFORM,PPATH,ACLASS(1),CHOICE,PNUM(30,4),LENG
REAL CSIZE(3),DATA(35),PAREA(6),L1(4),POINT(30,14),AXISR,PSONH),
& LOCAT(3),BOX(8),SHD1(8),LX(2),LY(2),LZ(2),LXI(2),LYI(2),LZI(2)
CHARACTER*7 PNAME(30) CliARACTER*l6 RETURN
C
DATA RETURN/'1. RETURN '/ DATA LENG/16/ DATA PIX/0.05,2.2/
C
DATA X /'X = '/ DATA Y /'Y = '/ DATA Z /'Z = '/ DATA CLEAR/
C
LX1)=0.05 LX(2)=0.68
LY1)=0.05 LY(2)=0.68
LZ(I)=0.05 L2)=0.68
DATA DATA/32,o,o,o,
C
C
C
DATA PRFORM/2/ FLAG FOR UPDATE
C
DATA ACLASS/4/ INCLUSION CLASS LIST
C
DATA BOX/0.00,0.545,0.48,0.545,0.48,0.75,0.00,0.73/
DO 10 I=2,8,2
SHDI)=BOX(I)-0.025
10 CONTINUE
DO 20 I=1,7,2
SHDI)=BOX(I)+0.025
20 CONTINUE

**REQUEST PICK INPUT**

CALL GPKF(IHSID,1,1,ACLASS,0,ACLASS)
MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)

MECHN PROGRAM LISTING
CALL GPPKMO(INSID,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(INSID,1,0,PPATH,1,PAREA,0,DATA,1)
C PLACE PICK DEVICE IN THE EVENT MODE
CALL GPPKMO(INSID,1,3,2)
C TRAVERSE ALL ACTIVE VIEWS
CALL GPUSHS(INSID,PRFORM)
C GET INQUIRY OR RETURN COMMAND
CALL CMDIN(INSID,22)
*********************************************************************
* INCLUDE A RETURN PICK
*********************************************************************
C RETURN PICK
CALL GPPOST(6)
CALL GPDCMN(1,1,CLASS)
CALL GPPKID(101)
CALL GPTXC(2)
CALL GPAN(1,0)
CALL GPAN(POSN,LENG,RETURN)
CALL GPCLIST
CALL GPVAR(INSID,3,4,0.)
CALL GPVPV(INSID,3,1,1)
CALL GPUPWH(INSID,PRFORM)
C GREY VALUE BOX AND SHADOW
CALL GPPOST(8)
CALL GPISI(2)
CALL GPCII(4)
CALL GPG2(1.4,2,SHD)
CALL GPCII(3)
CALL GPG2(1.4,2,BOX)
CALL GPCLIST
CALL GPUPWH(INSID,PRFORM)
*********************************************************************
* AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
*********************************************************************
C AWAIT AN EVENT
200 CALL GPAEV(1000,INS,ICLA,IDEV)
IF(ICLA.EQ.5)THEN
C ERASE OLD POSITION ICNS
CALL GPST(9)
CALL GPST(10)
CALL GPST(11)
C GET PICK
CALL GPSTPK(1,IDEP,PPATH)
IF(PPATH(2).EQ.101)GO TO 1000
C CHOICE IS SECOND ITEM OF PICK PATH
ISTRUC = PPATH(1)
C GET LOCATION
LOCAT(1) = POINT(ISTRUC,1)
LOCAT(2) = POINT(ISTRUC,2)
LOCAT(3) = POINT(ISTRUC,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 GPPOST(11)
CALL ICON(LOCAT,AXISR)
CALL GPCLIST
CALL GPPOST(9)
CALL ICON(LOCAT,AXISR)
CALL GPCLIST
CALL GPPOST(10)
CALL GPTXC(1)
CALL GHSC(1.0)
CALL GPAN1(LX,20,X)
MECHIN PROGRAM LISTING
189
SUBROUTINE SINO(IHSID,CSIZE,AXISR,POINT)

*********************************************************************
---- SUBROUTINE SINO ----
*********************************************************************

THIS SUBROUTINE IS USED TO INQUIRE BODY POSITION ORIENTATIONS

INTEGER IHSID,PRFORM,PPATH(3),ACLASS(1),CHOICE,PNUM(30,4),LENG,
& STROC
REAL CSIZE(3),DATA(35),PAREA(6),L1(14),POINT(30,14),AXISR,POSN(2),
& BOX(8),SHD(8),LX(2),LY(2),LZ(2),LXI(2),LYI(2),LZI(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 LEN'/16'/
DATA POSN/0.05,2.2/

DATA X '/DX = '/
DATA Y '/DY = '/
DATA Z '/DZ = '/
DATA CLEAR'/

LXI(1)=0.05
LXI(2)=0.68

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

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

LXI(1)=0.14
LXI(2)=0.68

LY(1)=0.14
LY(2)=0.62

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

LXI(1)=0.14
LXI(2)=0.68

DATA DATA/32,0,0,

USE DATA

C FLAG FOR UPDATE
DATA PRFORM/2/ INCLUSION CLASS LIST
DATA ACLASS/4/ 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
* REQUEST PICK FILTER(ALLAST 2,3 DETECTABLE)
CALL GPPKF(ISD,1,1,AClass,0,AClass)
CALL GPPKMO(ISD,1,1,2)
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)
CALL GPINPK(ISD,1,0,PPath,1,PAREA,0,DATA,1)
PLACE PICK DEVICE IN THE EVENT MODE
CALL GPPKMO(ISD,1,3,2)
TRAVERSE ALL ACTIVE VIEWS
CALL GPUPH(ISD,PRFORM)
GET INQUIRY OR RETURN COMMAND
CALL SCHMAN(ISD,37)
*********************************************************************
* INCLUDE A RETURN PICK
*********************************************************************
RETURN PICK
CALL GPPOST(6)
CALL GPADCN(1,AClass)
CALL GPPKDI(101)
CALL GPTXCI(2)
CALL GPAHSCI(1,0)
CALL GPaN(POSN,LENG,RETURN)
CALL GPCLST
CALL GPAVH(ISD,3,6,0.)
CALL GPVPH(ISD,5,1,1)
CALL GPUPH(ISD,PRFORM)
CALL GPPOST(8)
CALL GPSTI(2)
CALL GPSCI(4)
CALL GPSS(1,4,2,SHD)
CALL GPSCI(3)
CALL GPSS(1,4,2,BOX)
CALL GPCLST
CALL GPUPH(ISD,PRFORM)

GREY VALUE BOX AND SHADOW
CALL GPAHST(9)
CALL GPSTI(10)
CALL GPSTI(11)
CALL GPSTI(16)
GET PICK
CALL GPGETK(1,IDEV,PPath)
IF (IF(PPath(2).EQ.101) GO TO 1000
CHOICE=PPath(2)
IF (IF(STRU.EQ.13) THEN
   ORIENT(1)=POINT(CHOICE,4)
   ORIENT(2)=POINT(CHOICE,5)
   ORIENT(3)=POINT(CHOICE,6)
ENDIF
IF (IF(STRU.EQ.14) THEN
   ORIENT(1)=POINT(CHOICE,7)
   ORIENT(2)=POINT(CHOICE,8)
   ORIENT(3)=POINT(CHOICE,9)
ENDIF
MECHIN PROGRAM LISTING 191
SUBROUTINE SMENU

* SUBROUTINE SMENU(IHSID,CSIZE,NUM,CHOICE) *
* THIS SUBROUTINE IS USED TO DISPLAY MENU ITEMS IN THE MENU BOX *
* AND PROCESS THE MENU PICKS. *
******************************************************************************

INTEGER IHSID,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)

DATA LENG/19/
DATA TMENU(1,1)/'1. NEW MODEL '/
DATA TMENU(1,2)/'2. RESTART MODEL '/
DATA TMENU(1,3)/'3. MECHIN PROGRAM LISTING '/
DATA TMENU(1,11)/ 1. ENTER BODY 
DATA TMENU(1,12)/ 2. CHANGE VIEW 
DATA TMENU(1,13)/ 3. END OF DATA INPUT 

DATA TMENU(1,14)/ 1. POSITION 
DATA TMENU(1,15)/ 2. CHANGE VIEW 
DATA TMENU(1,16)/ 

DATA TMENU(2,1)/ 1. ZOOM 
DATA TMENU(2,10)/ 2. RESTORE VIEW 
DATA TMENU(2,11)/ 3. SCALE POINTS 
DATA TMENU(2,12)/ 

DATA TMENU(2,13)/ 4. RETURN 
DATA TMENU(2,14)/ 
DATA TMENU(2,15)/ 
DATA TMENU(2,16)/ 
DATA TMENU(2,17)/ 
DATA TMENU(2,18)/ 

DATA TMENU(3,1)/ 1. ADD POINT 
DATA TMENU(3,10)/ 2. DELETE POINT 
DATA TMENU(3,11)/ 3. INQUIRE POSITION 
DATA TMENU(3,12)/ 4. INQUIRE ORIENTATION 
DATA TMENU(3,13)/ 5. RETURN 
DATA TMENU(3,14)/ 
DATA TMENU(3,15)/ 
DATA TMENU(3,16)/ 
DATA TMENU(3,17)/ 
DATA TMENU(3,18)/ 

DATA TMENU(4,1)/ 1. DIRECT ENTRY 
DATA TMENU(4,10)/ 2. VALUATOR ENTRY 
DATA TMENU(4,11)/ 3. TOWARD AN EXISTING POINT 
DATA TMENU(4,12)/ 4. RETURN 
DATA TMENU(4,13)/ 
DATA TMENU(4,14)/ 
DATA TMENU(4,15)/ 
DATA TMENU(4,16)/ 
DATA TMENU(4,17)/ 
DATA TMENU(4,18)/ 

DATA TMENU(5,1)/ 1. RSCR 
DATA TMENU(5,10)/ 
DATA TMENU(5,11)/ 
DATA TMENU(5,12)/ 
DATA TMENU(5,13)/ 
DATA TMENU(5,14)/ 
DATA TMENU(5,15)/ 
DATA TMENU(5,16)/ 
DATA TMENU(5,17)/ 
DATA TMENU(5,18)/ 

DATA TMENU(6,1)/ 1. RSCR 
DATA TMENU(6,10)/ 

MECHIN PROGRAM LISTING
DATA TMENU(6,2) /'2. RCCC /
DATA TMENU(6,11) /'2. R6CC '/ DATA
DATA TMENU(6,12) /'3. WRITE NO /
DATA TMENU(6,13) /'3. WRITE NO /
DATA TMENU(6,14) /
DATA TMENU(6,15) /
DATA TMENU(6,16) /
DATA TMENU(6,17) /
DATA TMENU(6,18) /
DATA TMENU(6,1) /

DATA DATA/32,0,0,
& DATA/32,0,0,
& DATA/32,0,0,
& DATA/32,0,0,
& FLAG FOR UPDATE
& DATA PRFORM/2/ c
& DATA ACLASS/1/
& POSHN(1)=0.05
& POSHN(2)=2.3
*********************************************************************
* OPEN STRUCTURE FOR MENU ITEMS
*********************************************************************
C CALL GPESTI(6)
C CALL GPOPSTI(6)
C CALL GPDNCN(1,ACLASS)
C CALL GPTXCI(2)
C CALL GPASHC(1,00)
DO 100 I=1,9
C CALL GPPKIDN(I)
C CALL GPXN(POSN,LENG,TMENUINUM,1)
C POSN(2)=POSN(2)-.08
C CALL GPXN(POSN,LENG,TMENUINUM,1+9)
C POSN(2)=POSN(2)-.18
100 CONTINUE
C CALL GPCLST
*********************************************************************
* SCREEN DISPLAY
*********************************************************************
C CALL GPARV(ISIDZ3,6,0)
C CALL GPVPV(ISIDZ3,1,1)
C TRAVERSE ALL ACTIVE VIEWS
C CALL GPUPW(ISIDZPRFORM)
*********************************************************************
* REQUEST PICK INPUT
*********************************************************************
C CALL GPPK(ISIDZ1,1,ACLASS,0,ACLASS)
C CALL GPPKM(ISIDZ1,1,2)
C DEFINE PICK AREA
PAREA(1)=(-.98+1/2)*CSIZE(1)
PAREA(2)=(-.52+1/2)*CSIZE(1)
PAREA(3)=(-.98+1/2)*CSIZE(1)
PAREA(4)=(.33+1/2)*CSIZE(2)
PAREA(5)=0.
PAREA(6)=CSIZE(3)
C CALL GPINP(ISIDZ1,0,0,PATH,1,PAREA,0,DATA,1)
C PLACE PICK DEVICE IN THE EVENT MODE
C CALL GPUPW(ISIDZPRFORM)
C CALL GPUPW(ISIDZPRFORM)
* AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT ******************************************
C 200 CALL GPAHEV(1000, IMS, ICLA, IDEV)
   & (ICLA.EQ.5) THEN
C   CALL GPGETP(1, 1, PPATH)
C   CHOICE = PPATH(2)
C   CALL GPSPKMO(IHS, 1, 1, 2)
C   CALL GPSPRC(6)
C   CALL GPUPMS(IHS, PFORM)
C   GO TO 1000
C ELSE
C   IF NOT A PICK, GO TO AWAIT AN EVENT
END IF
C 1000 CALL GPSPKMO(IHS, 1, 1, 2)

RETURN
END

SUBROUTINE SNSTAR

SUBROUTINE SNSTAR(IHS, FILE, CSIZE, AXIS, PNAM, PNUM, POINT)

*********************************************************************
* ----- SUBROUTINE SNSTAR ----- *
* * * * * *
* THIS SUBROUTINE WILL START A MECHIN INPUT FILE. THE USER WILL *
* BE ASKED TO INPUT THE NAME OF THE FILE AND THE RANGE OF THE *
* AXES. *
* * * * * *
*********************************************************************

INTEGER IHS, PNUM(30, 4),
& DATA(35), LEN, ACLASS(1)
REAL AREA(6), CSIZE(3), AXIS,
& POINT(30, 14), POSN(2), AREA(6)
CHARACTER*7 FILE, AXIS, CLEAR, FILEO, INIT, PNAM(30)
CHARACTER*8 FLINE
CHARACTER*50 CLEAR
DATA RETURN/'1. QUIT' /
DATA LEN/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
DATA ACLASS/2/

INCLUSION CLASS LIST

DATA INIT/'1, 0' /
DATA AXIS/' '
DATA CLEAR/' '
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

CALL GPPKF(IHSID,1,1,ACLASS,0,ACLASS)

CALL GPPKMO(IHSID,1,1,2)

MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)

CALL GPINPK(IHSID,1,0,PATH,1,PAREA,0,DATA,1)

TRaverse ALL ACTIVE VIEWS

CALL GPUPHS(IHSID,2)

* INCLUDE A RETURN PICK

RETURN PICK

RETURN PICK

REQUEST THE FILE NAME

OPEN A NEW FILE

OPEN MECHIN FILE

PROCESS STRING EVENT. OPEN MECHIN FILE

GET STRING

SEE IF FILE ALREADY EXISTS

IF FILE EXISTS, HAVE USER RETYPE TO USE

OPEN UNIT=15, FILE=FILE, STATUS='NEW', ERR=2

OPEN MECHIN PROGRAM LISTING 196
SUBROUTINE SORIEN

SUBROUTINE SORIEN(IWSID,PPOSN,POINT,AXISR,CSIZE,ORIENT,RFLG)

*****************************************************************************
* ----- SUBROUTINE SORIEN-------                                       *
*                                            *
*****************************************************************************

INTEGER IWSID,ECHOSH,EVENT,DLEN,ECHO,ACLASS,PRFORM,CHOICE,
& PPATH(3),RFLG,NUMV
REAL CSIZE(3),AREA1(6),AREA2(6),AREA3(6),VALUE1,LOVAL,HIVAL,AXISR,
& BOX(6),SHD(6),ORIENT(3),MATRIX(1000,4),MATUSE(4,4),LINE(6),
& & LINEX(6),LINOE(6),LINEZ6,JXBOX(15),PPPOSN(3),PPNT(30,14),
& & BOXI(15),ZBOX(15),LX(6),LY(6),LZ(6),OL(6),AREA(6)
CHARACTER*7 CLEAR,INIT
DATA CLEAR/' '/ DATA INIT/0.0.0.0.0.0.0./
DATA LX/0.0.0.0.0.0.0./ DATA LY/0.0.0.0.0.0.0./
DATA LZ/0.0.0.0.0.0.0.0./
DATA OL/0.0.0.0.0.0.0.0./
C DATA PRFORM/2/   Flag for update
C DATA ACLASS/1/   Inclusion class list

MECHIN PROGRAM LISTING  197
DATA DLEN/0/
DATA EVENT/3/
DATA ECHO/2/
C

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

AREAl1=.08*CSIZE1
AREAl2=.90*CSIZE1
AREAl3=.70*CSIZE2
AREAl4=.90*CSIZE2
AREAl5=.02*CSIZE3
AREAl6=.90*CSIZE3

AREAl1=0.52*CSIZE1
AREAl2=CSIZE1
AREAl3=.85*CSIZE2-.006
AREAl4=CSIZE2
AREAl5=0.
AREAl6=CSIZE3

AREAl1=.52*CSIZE1
AREAl2=CSIZE1
AREAl3=.85*CSIZE2-.014
AREAl4=CSIZE2
AREAl5=0.
AREAl6=CSIZE3

AREAl1=.52*CSIZE1
AREAl2=CSIZE1
AREAl3=.85*CSIZE2-.022
AREAl4=CSIZE2
AREAl5=0.
AREAl6=CSIZE3

DO 100 I=2,8,2
SHDI=BOX(I)-.025
100 CONTINUE

DO 200 I=1,7,2
SHDI=BOX(I)+.025
200 CONTINUE

*********************************************************************
* OPEN STRUCTURE FOR THE MENU BOX (VIEW 1)                         *
*********************************************************************
C OPEN A STRUCTURE
C CALL GPOPST8)
C CALL GGIS2)
C CALL GPICI4)
C CALL GPPG11,4,2,SHD)
C CALL GPICII3)
C CALL GPPG21,4,2,BOX)
C CALL GPCLST

*********************************************************************
* SCREEN DISPLAY                                                     *
*********************************************************************
C CALL GPUPHSINSID,PRFORM)

*********************************************************************
* CHOOSE METHOD OF JOINT ORIENTATION                                *
*********************************************************************
C CALL SMENUINSID,CSIZE,5,CHOICE)
C IF(CHOICE.EQ.1)THEN
C CALL DEOINSID,AXISR,CSIZE,ORIENT,RFLG)
C IF(RFLG.EQ.1)THEN
C ORIENT1=0.
C ORIENT2=0.
C ORIENT3=0.
C RFLG=0.
C GOTO 300
C ENDIF
C IF(CHOICE.EQ.2)THEN
C VALUATOR ENTRY

MECHIN PROGRAM LISTING 198
CALL VE0(IHSID,AXISR,CSIZE,ORIENT,RFLG)
  IF(RFLG.EQ.1) THEN
    ORIENT(1) = 0.
    ORIENT(2) = 0.
    ORIENT(3) = 0.
    GO TO 300
  ENDIF
ENDIF

IF(CHOICE.EQ.3) THEN
  CALL STEP0(IHSID,PPOSN,POINT,AXISR,CSIZE,ORIENT,RFLG)
  IF(RFLG.EQ.1) THEN
    ORIENT(1) = 0.
    ORIENT(2) = 0.
    ORIENT(3) = 0.
    GO TO 300
  ENDIF
ENDIF

IF(CHOICE.EQ.4) THEN
  RFLG = 1
ENDIF

1000 CALL GPEST(8)
CALL GPEST(6)
CALL GPUPMS(IHSID,2)
RETURN
END

SUBROUTINE SPOSIT

SUBROUTINE SPOSIT(IHSID,AXISR,CSIZE,LOCAT,RFLG)

********************************************************************* *
** VIxy " SUBROUTINE SPOSIT " **
********************************************************************* *
** THIS SUBROUTINE IS USED TO POSITION A BODY POSITION **
********************************************************************* *

INTEGER IHSID,ECHOSH,EVENT,DLEN,ECHO,ACLASS,PRFORM,CHOICE,
  & PATH(3),RFLG,NUMV
REAL CSIZE(3),AREA(6),AREA(6),AREA(6),VALUE1,LOVAL,HIVAL,AXISR,
  & BOX(15),SHO(15),LOCAT(3),MATRIX(1000,4),MATUSE(4,4),LINE(6),
  & LINE(6),LINE(6),LINE(6),LINE(6),LINE(6),
  & & BOX(15),BOX(15),LX(6),LY(6),LZ(6),AREA(6)
CHARACTER*7 CLEAR,INIT
DATA CLEAR/' '/ DATA INIT/'0,'I
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 DATA PRFORM/2/
C DATA ACLASS/1/
DATA VALUE1/0./ DATA ECHO/3/ DATA DLEN/0/ DATA EVENT/5/ DATA ECHOSH/2/
C DATA BOX/0.00,0.545,0.46,0.545,0.46,0.73,0.00,0.73/
NUMV=1
LOVAL=-AXISR
HIVAL=AXISR
AREA(1)=0.8*CSIZE(1)
AREA(2)=0.9*CSIZE(1)
AREA(3)=0.7*CSIZE(2)

MECHIN PROGRAM LISTING
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
C CALL GPOPST(8) SET SOLID INTERIOR STYLE FOR GREY BOX
C CALL GPISI(2) SET COLOR FOR INTERIOR OF SHADOW BOX
C CALL GPICI(4) DRAW GREY SHADOW BOX
C CALL GPPG2(1,4,2,SHD) SET COLOR FOR INTERIOR OF GREY BOX
C CALL GPICI(3) DRAW GREY TITLE BOX
C CALL GPPG2(1,4,2,BOX) CLOSE STRUCTURE
C CALL GPCLST

* SCREEN DISPLAY *
C UPDATE THE WORKSTATION
C CALL GPUPHS(IHSID,PRFORM)

* PLACE VALUATORS IN THE REQUEST MODE *
300 CALL GPVLMO(IHSID,1,1,ECHOSH)
   CALL GPVLMO(IHSID,2,1,ECHOSH)
   CALL GPVLMO(IHSID,3,1,ECHOSH)
C CALL GPINVL(IHSID,3,VLE, ECHO, AREA, LOVAL, HIVAL, BLEN, DATA)
   CALL GPINVL(IHSID,2,VLE, ECHO, AREA, LOVAL, HIVAL, BLEN, DATA)
   CALL GPINVL(IHSID,1,VLE, ECHO, AREA, LOVAL, HIVAL, BLEN, DATA)
C CALL GPVLMO(IHSID,1,EVENT,ECHOSH)
   CALL GPVLMO(IHSID,2,EVENT,ECHOSH)
   CALL GPVLMO(IHSID,3,EVENT,ECHOSH)
C CALL GPTMST(IHSID,1,12,INIT,1,AREA,12,1,0,CLEAR)
   CALL GPRSTMO(IHSID,1,EVENT,ECHOSH)

* SCREEN DISPLAY *
C UPDATE THE WORKSTATION
C CALL GPUPHS(IHSID,PRFORM)

* DISPLAY MENU AND GET CHOICE OF ENTER, RETURN OR VALUATOR *

LOCAT(1)=0.
LOCAT(2)=0.
LOCAT(3)=0.
500 CONTINUE
C CALL ENTER(IHSID,CHOICE,CSIZE)
CALL GPAHEV(1000, IIMSID, ICLASS, IDEV)

IFI(ICLASS.EQ.3) THEN

C PROCESS VALUATORS (NOT IN SUBROUTINE TO SPEED EXECUTION)

CALL GPSTM(IIMSID, 1, 1, ECHOSH)
CALL GPSTVL(VALUE)

IFI(IDEV.EQ.1) LOCAT(1) = VALUE
IFI(IDEV.EQ.2) LOCAT(2) = VALUE
IFI(IDEV.EQ.3) LOCAT(3) = VALUE

*********************************************************************
* GET POSITION ICON
*********************************************************************

LX14 = LOCAT(1)
LY15 = LOCAT(2)
LZ16 = LOCAT(3)

LINEX11 = LOCAT(1)
LINEX12 = LOCAT(2)
LINEX13 = LOCAT(3)
LINEX14 = LOCAT(4)
LINEX15 = LOCAT(2)
LINEX16 = LOCAT(3)

LINEY11 = LOCAT(1)
LINEY12 = LOCAT(2)
LINEY13 = LOCAT(3)
LINEY14 = LOCAT(1)
LINEY15 = 0.00
LINEY16 = LOCAT(3)

LINEZ11 = LOCAT(1)
LINEZ12 = LOCAT(2)
LINEZ13 = LOCAT(3)
LINEZ14 = LOCAT(1)
LINEZ15 = LOCAT(2)
LINEZ16 = LOCAT(3)

XBOX11 = 0.00
XBOX12 = LOCAT(2) + 0.06*AXISR
XBOX13 = LOCAT(3) + 0.06*AXISR
XBOX14 = 0.00
XBOX15 = LOCAT(2) - 0.06*AXISR
XBOX16 = LOCAT(3) - 0.06*AXISR

YBOX11 = LOCAT(1) + 0.06*AXISR
YBOX12 = 0.00
YBOX13 = LOCAT(3) + 0.06*AXISR
YBOX14 = LOCAT(1) - 0.06*AXISR
YBOX15 = 0.00
YBOX16 = LOCAT(3) - 0.06*AXISR

ZBOX11 = LOCAT(1) + 0.06*AXISR
ZBOX12 = LOCAT(2) + 0.06*AXISR
ZBOX13 = LOCAT(3) + 0.06*AXISR
ZBOX14 = LOCAT(1) - 0.06*AXISR
ZBOX15 = 0.00
ZBOX16 = LOCAT(3) - 0.06*AXISR

CALL GPESTI(9)
CALL GOPSTI(9)

C SET POLYLINE COLOR

MECHIN PROGRAM LISTING
CALL GPPLCI(1) SET LINE TYPE DASHED
CALL GPLT(2) DRAW POLYLINE
CALL GPPL(2,3,LINEX) CALL GPPL(2,3,LINEX)
CALL GPPL(2,3,LINEZ)
CALL GPLT(1) CALL GPPL(5,3,XBOX) CALL GPPL(5,3,XBOX)
CALL GPPL(8,3,ZBOX)
CALL GPPLCI(5) SET POLYLINE COLOR
CALL GPPL(2,3,LX) CALL GPPL(2,3,LY) CALL GPPL(2,3,LZ)
CALL GPCLST CALL GPUPHSIHSID,PRFORM GO TO 600
ENDIF
IFIICLASS.EQ.6 THEN PROCESS STRING ENTRY
CALL GPVMO(IHSID,1,1,ECHOSIM) CALL GPVMO(IHSID,2,1,ECHOSIM)
CALL GPVMO(IHSID,3,1,ECHOSIM) CALL GPVMO(IHSID,4,1,ECHOSIM)
RFLG=0 CALL DEP(IHSID,AXISR,CSIZE,LOCAT,RFLG)
IFIIFLAG.EQ.1 THEN REENTER POINT
CALL GPVMO(IHSID,1,1,ECHOSIM) CALL GPVMO(IHSID,2,1,ECHOSIM)
CALL GPVMO(IHSID,3,1,ECHOSIM) CALL GPVMO(IHSID,4,1,ECHOSIM)
CALL GPINST(IHSID,1,12,INIT,1,AREA,12,1,0,CLEAR)
CALL GPSTMO(IHSID,1,1,5,ECHOSIM)
LOCAT(1)=0. LOCAT(2)=0. LOCAT(3)=0.
RFLG=0 GO TO 500
ENDIF
CALL GPSTMO(IHSID,1,1,ECHOSIM) CALL GPSTMO(9)
CALL GPSTMO(IHSID,1,1,ECHOSIM) CALL GPSTMO(9)
GO TO 1000
ENDIF
IFIIFCLASS.EQ.5 THEN CALL GPSTPK(1,1,PPATH)
CHOICE=PPATH(2)
IFIIFCHOICE.EQ.1 THEN DEACTIVATE VALUATORS
CALL GPVMO(IHSID,1,1,ECHOSIM) CALL GPVMO(IHSID,2,1,ECHOSIM)
CALL GPVMO(IHSID,3,1,ECHOSIM) CALL GPVMO(IHSID,4,1,ECHOSIM)
CALL GPSTMO(IHSID,1,1,ECHOSIM)
RFLG=0 CALL GPSTESI(6) CALL GPSTESI(8)
CALL GPSTESI(9) CALL GPUPHSIHSID,PRFORM
GO TO 1000
ENDIF
IFIIFCHOICE.EQ.2 THEN DEACTIVATE VALUATORS
CALL GPVMO(IHSID,1,1,ECHOSIM) CALL GPVMO(IHSID,2,1,ECHOSIM)
CALL GPVMO(IHSID,3,1,ECHOSIM) CALL GPVMO(IHSID,4,1,ECHOSIM)
CALL GPSTMO(IHSID,1,1,ECHOSIM)
RFLG=1 CALL GPSTESI(6) CALL GPSTESI(8)
CALL GPSTESI(9) CALL GPUPHSIHSID,PRFORM
GO TO 1000
ENDIF
ENDIF
1000 RETURN
END

SUBROUTINE SPT

SUBROUTINE SPT(IHSID,AXISR,CSIZE,PNAME,PNUM,POINT,SCALE) 
*****************************************************************************
--- SUBROUTINE SPT ----

THESE SUBROUTINES IS USED TO CREATE BODY POSITION DATA

REAL AXISR, CSIZE(3), POINT(30, 14), SCALE
INTEGER IMSID, CHOICE, PNUM(30, 4)

C REQUEST TYPE OF DATA INPUT
10 CALL CMMANDI(IMSID, 10)
CALL SMENUI(IMSID, CSIZE, 4, CHOICE)
 IF (CHOICE.EQ.1) THEN
   REQUEST ADD A POINT
   CALL SADDPI(IMSID, AXISR, CSIZE, PNAM, PNUM, POINT, SCALE)
   GO TO 10
 ENDIF
 IF (CHOICE.EQ.2) THEN
   REQUEST DELETE A POINT
   CALL SDELP(IMSID, AXISR, CSIZE, PNAM, PNUM, POINT, SCALE)
   GO TO 10
 ENDIF
 IF (CHOICE.EQ.3) THEN
   INQUIRE A POSITION
   CALL SINQI(IMSID, CSIZE, AXISR, PNAM, PNUM, POINT)
   GO TO 10
 ENDIF
 IF (CHOICE.EQ.4) THEN
   INQUIRE AN ORIENTATION
   CALL SINQOCI(IMSID, CSIZE, AXISR, POINT)
   GO TO 10
 ENDIF
 C RETURN
ENDIF
RETURN
END

--- SUBROUTINE SRDRWP ----

THESE SUBROUTINES IS USED TO REDRAW ALL POINTS BODY POSITIONS
AFTER A DELETION OR RESTART

INTEGER IMSID, NUMP, PNUM(30, 4)
REAL LOCAT(3), POINT(30, 14), AXISR, SCALE
CHARACTER*7 PNAM(30)

C DELETE ALL POINTS
CALL GPESTI(12)
CALL GPESTI(13)
CALL GPESTI(14)
CALL GPESTI(15)
CALL GPUPMS(IMSID, 2)

DO 10 I=1, 30
C CHECK FOR LAST POINT
C IF (PNAM(I).EQ. 'BBBBBBB') GO TO 1000
C CHECK DELETED SPACES IN THE ARRAY
C 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
PNAM=PNAM(I)
C DRAW POINT
CALL SRDRWP(IMSID, AXISR, PNAM, NUMP, POINT, SCALE)

MECHIN PROGRAM LISTING
**SUBROUTINE SRSTAR**

SUBROUTINE SRSTAR(IMSID, 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 IMSID, PNUM(30, 4),
REAL AREA(3), CSIZE(3), AXISR, POINT(30, 14),
IC(20, 4), POSN(2), PAREA(6),
CHARACTER*7 FILE, CMD, QUIT, CLEAR, PNAM(30),
CHARACTER*16 FLINE, CLINE

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

DATA DATA/32,0,0,
& 1,1,1,
& 1,1,1,1,
& 1,1,1,1,1,
& 1,1,1,1,1,
& 1,1,1,1,
& 1

DATA ACLASS/2/

DATA CLEAR/''/
DATA CMD/''/
DATA 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
******************************************************************************

CALL GPKF(IMSID,1,1,AClass,0,AClass)
CALL GPPKMD(IMSID,1,1,2)

DEFINPE PICK AREA
PAREA(1)=(-.98+1/2)*CSIZE(1)
PAREA(2)=(-.98+1/2)*CSIZE(2)
PAREA(3)=(-.98+1/2)*CSIZE(2)
PAREA(4)=(.48+1/2)*CSIZE(2)
PAREA(5)=0.
PAREA(6)=CSIZE(3)

CALL GPINP(IMSID,1,0,PPathl,PAREA,0,DATA,1)
CALL GPPKMD(IMSID,1,3,2)

CALL GPUPHS(IMSID,2)

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

CALL GPOPST(6)
CALL GPADCN(1,AClass)
CALL GPPKID(101)

MECHIN PROGRAM LISTING

204
CALL GPTXCI(2)
CALL GPAHSC(1,0)
CALL GPAN3(POSN,LENG,RETURN)
CALL GPCLS1
CALL GPARV(IHSID,3,6,0.)
CALL GPVPI(IHSID,3,1,1)
CALL GPUPMS(IHSID,2)

*********************************************************************
* INITIALIZE THE STRING DEVICE
*********************************************************************
CALL COMMAND(IHSID,2) REQUEST THE FILE NAME
5 CALL GPSTM0(IHSID,1,1,2)
CALL GPINST(IHSID,1,7,CLEAR,1,AREA,7,1,0,CLEAR)
CALL GPSTM0(IHSID,1,3,2)

*********************************************************************
* QUIT IF QUIT IS PICKED
*********************************************************************
C 6 CALL GPAHEV(1000.,IHS,ICLA,IDEV)
IF(ICLA.EQ.5)THEN QUIT PHIGS
CALL GDAST
CALL GPCLPH STOP
ENDIF

*********************************************************************
* PROCESS STRING EVENT
*********************************************************************
C L=7 GET STRING
CALL GPGTST(L,LR,FILE)
C FIND OUT IF FILE EXISTS
CALL FILEXS(FILE,IRET)
IF(IRET.EQ.1)THEN CALL COMMAND(IHSID,9) FILE=CLEAR
GO TO 6 ENDIF
C OPEN UNIT 15, FILETYPE 'FILE'
OPEN(UNIT=15, FILE=FILE, ERR=6) REMIND 15
READ(15, 10, ERR=6, END=6)FLINE
10 FORMAT(A29) IF(FLINE.NE.CLINE)THEN CALL COMMAND(IHSID,9)
FILE=CLEAR
GO TO 6 ENDIF
CALL GPPKMO(IHSID,1,1,2)
CALL GPSTMO(IHSID,1,1,2)
C CALL COMMAND(IHSID,3) WRITE THAT ACTION IS CONTINUING
C CALL FILENM(IHSID,FILE) WRITE FILENAME TO THE SCREEN

*********************************************************************
* SET UP AXES
*********************************************************************
C READ THE AXIS RANGE
READ(15,*)AXISR CALL AXES(IHSID,AXISR)
C READ JOINT SCALE FACTOR
READ(15,*)SCALE CALL SCALNM(IHSID,SCALE)

*********************************************************************
* SET UP POINT ARRAY
*********************************************************************
C READ THE NUMBER OF POINTS
DO 100 I=1,NP READ THE POINT DATA
100 CONTINUE

*********************************************************************
* DRAW ALL POINTS
*********************************************************************
CALL SRDRPH(IHSID,AXISR,PNAM,PNUM,POINT,SCALE)

MECHIN PROGRAM LISTING
SUBROUTINE STEPO

SUBROUTINE STEPO(IHSID, PPOSN, POINT, AXISR, CSIZE, LOCAT, RFLG)
*********************************************************************
*** ----- SUBROUTINE STEPO ----- ***
*** THIS SUBROUTINE WILL PROMPT FOR THE ENTRY OF ***
*** ORIENTATION COORDINATES TOWARD AN EXISTING POINT ***
*********************************************************************

INTEGER IHSID, PRFORM, PPATH(3), ACLASS(1), CHOICE, PNUM(30, 4), LENG,
REAL CSIZE(3), DATA(35), PAREA(6), L1(4), AXISR, PPOSN(2),
& X(3), Y(3), Z(3), VEC(3), MAG, LOCAT(3), PPOSN(3), POINT(30, 14),
& AXI(2), LX(2), LY(2), LX1(2), LZ(2), LZ1(2)
CHARACTER*7 PNAM(30)
CHARACTER*20 X, Y, CLEAR
CHARACTER*16 XVAL, YVAL, ZVAL, RETURN
DATA RETURN/1. RETURN /
DATA LENG/16/
DATA POSN/0.05, 2.2/

C DATA X/ ' ' /DX= '/
C DATA Y/ ' ' /DY= '/
C DATA Z/ ' ' /DZ= '/
C DATA CLEAR '/

C DATA X(1)=0.05 C DATA Y(1)=0.14
C DATA X(2)=0.68 C DATA Y(2)=0.62
C DATA X(3)=0.05 C DATA Y(3)=0.14
C DATA Z(1)=0.05 C DATA Z(2)=0.56
C DATA Z(1)=0.14 C DATA Z(1)=0.56

*********************************************************************
*** REQUEST PICK INPUT ***
*********************************************************************
C CALL GPSTMO(IHSID,1,1,2)
C CALL GPPKF(IHSID,1,0, ACLASS,1, ACLASS)
RETURN
END
CALL GPUPHS(ISID,PRFORM)
GET PICK OR RETURN COMMAND

200 CALL SCMANN(ISID,34)

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

C RETURN PICK
CALL GPUPST(6)
CALL GPADCN(I,ACLASS)
CALL GPPKID(I)
CALL GPTXCI(I)
CALL GPAHSCI(I,0)
CALL GPAN2I(POSN,LEN,RETURN)
CALL GPCLST
CALL GPARV(ISID,3,6,0.)
CALL GPVP(ISID,3,1,1)
CALL GPUPHS(ISID,PRFORM)

******************************************************************************
* AWAiit A PICK EVENT TO OCCUR AND RETRIEVE IT
******************************************************************************
C AWAiit AN EVENT
210 CALL GPHEV(I1000.,INS,ICLA,IDEV)
IF (ICLA.EQ.5) THEN
GET PICK
CALL GPSTPK(3, I, PPATH)
IF (PPATH(2), EQ. 101) THEN
RETURN PICKED
ENDIF

C CHOICE IS SECOND ITEM OF PICK PATH
C CHOICE=PPATH(2)
C GET POINT COORDINATES
P(1)=POINT(CHOICE,1)
P(2)=POINT(CHOICE,2)
P(3)=POINT(CHOICE,3)
P(1)=POSN(1)
P(2)=POSN(2)
P(3)=POSN(3)

VEC(1)=P(1)-P(1)
VEC(2)=P(2)-P(2)
VEC(3)=P(3)-P(3)

IF (VECl1J.EQ.0.0.AND. VEC12J.EQ.0.0.AND. VEC13J.EQ.0.0) THEN
POINT AND ORIGIN ARE THE SAME -- RE-ENTER
CALL COMMAND(ISID,64)
GO TO 210
ENDIF
MAG=(VECl1J**2)+(VECl2J**2)+(VECl3J**2)**.5
LOCAT1)=VEC11J/MAG
LOCAT2)=VEC12J/MAG
LOCAT3)=VEC13J/MAG

C DRaW ORIENTATION ICON
CALL GPSTI(16)
CALL GPOPSTI (16)
CALL GPTXI(1)
CALL GPAHSCI(1,0)
CALL GPAN2I(LX,20,X)
CALL GPAN2I(LY,20,Y)
CALL GPAN2I(LZ,20,Z)
CALL GPAN2I(LZ,20,2)

THE MECHIN PROGRAM LISTING

207
CALL GPANZ(LZ1,16,ZVAL)
CALL GPCLST

C ENTER ORIENTATION OR RETURN
CALL CMNAND(IWSID,3)
CALL MENU(IWSID,CSIZE,8,CHOICE)
IF(CHOICE.EQ.1)THEN
    RFLG=0
    GO TO 1000
ENDIF
IF(CHOICE.EQ.2)THEN
    RFLG=1
ENDIF
ELSE
    LOCAT1=0.
    LOCAT2=0.
    LOCAT3=0.
    GO TO 1000
ENDIF
GO TO 200

C IF NOT A PICK, GO TO AWAIT AN EVENT

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,弗L1E8CHARACTER*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=' 
REMIND 15
*********************************************************************
* INITIALIZE THE STRING DEVICE. REQUEST NEW RESTART FILE NAME 
*********************************************************************
C INITIALIZE STRING MODE
C CALL GPSTMO(IWSID,1,1,2)
C 1 CALL GPINST(IWSID,1,7,FILE,1,AREA,7,1,0,FILE)
C CALL CMNAND(IWSID,60)
C REQUEST FILE NAME
C 2 CALL GPRQST(IWSID,1,7,1STAT,NP,FILE1)
IF(FILE1.EQ.' ')THEN
    GO TO 2
ENDIF
IF(FILE1.EQ.FILE)THEN
    GO TO 9
ENDIF

MECHIN PROGRAM LISTING 208
FILE=FILE1
SEE IF FILE ALREADY EXISTS
3 CALL FILEX(FILE,IRET)
   IF(IRET.EQ.0)THEN
      CALL GPSTMO(IHSID,1,1,2)
      FILE=FILE
      CALL CHMND(IHSID,8)
      CALL GPINST(IHSID,1,FILE,1,AREA,1,0,FILE)
      CALL GPROST(IHSID,1,ISTAT,NUM,FILE)
      IF(FILE.EQ.'')THEN
         GO TO 2
      ENDIF
      IF(FILE.NE.FILEO)THEN
         ENDIF
      ENDIF
      GO TO 3
   ENDIF
CLOSE (UNIT=1)
CALL GPSTMT(IHSID,1,1,2)
OPEN(UNIT=15,FILE=FILE,STATUS='NEW',ERR=2)
*** WRITE FIRST LINE OF THE RESTART FILE AND AXIS RANGE
*** WRITE POINT DATA TO THE RESTART FILE
SUBROUTINE SYFIN(IHSID,CSIZE)
*********************************************************************
* ----- SUBROUTINE SYFIN ----- *
* THIS IS THE SYNTHESIS PORTION OF MECHIN. THIS ROUTINE WILL
* ALLOW THE INPUT OF SPATIAL MECHANISM PARAMETERS AND MILL
* WRITE AN INPUT FILE FOR AN SYNTHESIS PACKAGE BASED ON THESE
* PARAMETERS.
* COMMON/SYN/PNAM(30)
* INTEGER IHSID,CHOICE,PNUM(30,4)
* REAL CSIZE(5),AXISR,POINT(30,14),SCALE
* CHARACTER*7 PNMH,FILE
*********************************************************************
*** CALL BACKGROUND SCREEN
*** CALL FILE WITH SYNTHESIS BACKGROUND

MECHIN PROGRAM LISTING
209
SUBROUTINE TEJOX

SUBROUTINE TEJOX(IHSID, NUMJ, IAIXS, 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 IHSID, PRFORM, PPATH(3), ACLASS(2), CHOICE, LENG, IAXIS, RFLG
REAL CSIZE(3), DATA(35), PAREA(6), L(6), 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.
DATA LENG/16/
DATA DATA/32,0,0,
& 2,2,1,1,
& 2,2,1,1,
& 2,2,1,1,
& 2,2,1,1,
& 2,2,1,1,
& 2,2,1,1,
& C DATA PRFORM/2/ FLAG FOR UPDATE
& C DATA ACLASS/3,4/ INCLUSION CLASS LIST (CLASS 3, 4 JOINT AXES)

*********************************************************************
* REQUEST PICK INPUT *
*********************************************************************
C CALL GPPKF(IHSID, 1, 2, ACLASS, 0, ACLASS)
M MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
C CALL GPPMKO(IHSID, 1, 1, 2)
D DEFINE PICK AREA
PAREA(1) = ((- 98*1)/2)*CSIZE(1)
PAREA(2) = ((.98*1)/2)*CSIZE(1)
PAREA(3) = (1 - .981) / 2) * CSIZE(2)  
PAREA(4) = (1 + .81) / 2) * CSIZE(2)  
PAREA(5) = 0  
PAREA(6) = CSIZE(3)  

CALL GPPKM(INSID,1,1,2)  
INITIALIZE PICK PATH  
CALL GPPMKO(INSID,1,3,2)  
PLACE PICK DEVICE IN THE EVENT MODE  
CALL GPUPHS(INSID,PRFORM)  
TRaverse ALL ACTIVE VIEWS  
GET PICK OR RETURN COMMAND  
**************************************************************************  
INCLUDE A RETURN PICK  
**************************************************************************  

CALL GPOPST(16)  
CALL GPADCN(1,ACLASS)  
RETURN PICK  
CALL GPPKID(0)  
CALL GPTXC(2)  
CALL GPAHSC(0)  
CALL GPPKMO(INSID,3,6,0)  
CALL GPUPH(INSID,1,1)  
**************************************************************************  

AHAIT EVENT TO OCCUR AND RETRIEVE IT  
**************************************************************************  

C AHAIT AN EVENT  
300 CALL GPAHEV(1000, IHS, ICLA, IDEV)  
IF (ICLA.EQ.5) THEN  
CALL GPPTPK(1,1,PPATH)  
CALL GPADCN(1,ACLASS)  
GET PICK  
IF (PPATH(2).EQ.101) THEN  
RFLG=1  
GO TO 1000  
ENDIF  
END = PPATH(1)  
ISTRUCT = 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)  
ENDIF  
IF (IAXIS.EQ.11) THEN  
ZVEC(1) = JOINT(NUMJ,4)  
ZVEC(2) = JOINT(NUMJ,5)  
ZVEC(3) = JOINT(NUMJ,6)  
ENDIF  
IF (IAXIS.EQ.21) 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  
X POINT ON THE Z VECTOR  
CALL COMMAND(INSID,54)  
GO TO 300  
ENDIF  
CALL VPROD(YVEC,ZVEC,XVEC)  
XMAG = (YVEC(1)**2) + (YVEC(2)**2) + (YVEC(3)**2)**.5  
YMAG = (YVEC(1)**2) + (YVEC(2)**2) + (YVEC(3)**2)**.5  
XVEC(1) = YVEC(1) / XMAG  
XVEC(2) = YVEC(2) / XMAG  
XVEC(3) = YVEC(3) / XMAG  
YVEC(1) = YVEC(1) / YMAG  
YVEC(2) = YVEC(2) / YMAG  
YVEC(3) = YVEC(3) / YMAG  
ENDIF  
MAKE SURE PICK IS DEACTIVATED  
1000 CALL GPPKM(INSID,1,1,2)  
CALL GPEST(6)  
SET PICK FILTER(ALL CLASS 2 UNDETECTABLE)  
CALL GPKF(INSID,1,0,ACLASS,2,ACLASS)  
RETURN  
END  

MECHIN PROGRAM LISTING
SUBROUTINE TEPO

integer IHSID,PRFORM,PPATH(3),ACLASS(3),CHOICE,PNUM(100),LENG,
& RFLG
real CSIZE(3),PAREA(6),LX1(4),LY1(4),LZ1(4),JOINT(20,21),AXISR,POSN(2),
& P(3),PI(3),VEC(3),MAG,LOCAT(3),JPOSN(3),POINT(100,3),
& LX(2),LY(2),LZ(2)
character*7 PNAM(100)
character*20 XVAL,YVAL
character*16 XVAL,YVAL,val,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,
data x /'dx= '
data y /'dy= '
data z /'dz= '
data clear /' '
data x(1)=0.05
data x(2)=0.68
x(1)=0.14
x(2)=0.68
ly(1)=0.05
ly(2)=0.62
ly(1)=0.14
ly(2)=0.62
lz(1)=0.05
lz(2)=0.56
lz(1)=0.14
lz(2)=0.56

********************************************************************* *
REQUEST PICK INPUT
*********************************************************************

call gppkfi(ihsid,1,3,aclass,0,aclass)
call gppkm(ihsid,1,2)
call gppfnk(ihsid,1,3,1,0,aclass,0,aclass)
call gppkm(ihsid,1,3,2)
call gppupk(ihsid,prform)
call cmmand(ihsid,34)
call include a return pick
CALL GPADCN(1,ACLASS)
CALL GPPKID(101)
CALL GPTMC(1.0)
CALL GPADSC(1.0)
CALL GPAN2(POSN,LEN,RETURN)
CALL GPCLST
CALL GPARN(INSID,3,6,0)
CALL GPPV(INSID,3,1,1)
CALL GPPNHS(INSID,PRFORM)

**************************************************************XXXXXX*
* AWAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
**************************************************************XXXXXX*

CALL GPAHEV(1000,INS,ICLA,IDEV)

IF(ICLA.EQ.5) THEN
  GET PICK
  CALL GPGTPK(3,I,PPATH)
  IF RETURN PICKED
  IF(PPATH(2).EQ.101) THEN
    RFLG=1
    GO TO 1000
  ENDIF

  CALL GPEST(6)
  CHOICE IS SECOND ITEM OF PICK PATH
  CHOICE=PPATH(2)

  GET POINT COORDINATES
  IS=PPATH(1)
  IF(IS.EQ.12) THEN
    GET POINT COORDINATES
    P1(1)=POINT(CHOICE,1)
    P1(2)=POINT(CHOICE,2)
    P1(3)=POINT(CHOICE,3)
  ENDIF

  GET POINT COORDINATES
  IF(IS.EQ.13.OR.IS.EQ.14.OR.IS.EQ.17.OR.IS.EQ.18) THEN
    GET POINT COORDINATES
    P1(1)=JOINT(CHOICE,1)
    P1(2)=JOINT(CHOICE,2)
    P1(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(VE(1).EQ.0.0.AND.VE(2).EQ.0.0.AND.VE(3).EQ.0.0) THEN
    POINT AND ORIGIN ARE THE SAME -- RE-ENTER
    CALL COMMAND(INSID,84)
    GO TO 210
  ENDIF

  MAG=(VE(1)**2)+(VE(2)**2)+(VE(3)**2)**.5
  LOCAT(1)=VE(1)/MAG
  LOCAT(2)=VE(2)/MAG
  LOCAT(3)=VE(3)/MAG

  DRAW ORIENTATION ICON
  CALL GPEST(16)
  CALL GPOPST(16)
  CALL ICON(ILOCAT,AXISR)
  CALL GPCLST

  DISPLAY VALUES
  CALL GPOPST(16)
  CALL GPTMC(1.0)
  CALL GPADSC(1.0)
  CALL GPAN2(POSN,LEN,RETURN)

  CALL GPARN(INSID,3,6,0)
  CALL GPPV(INSID,3,1,1)
  CALL GPPNHS(INSID,PRFORM)

  CALL GPADCN(1,ACLASS)
  CALL GPPKID(101)
  CALL GPTMC(1.0)
  CALL GPADSC(1.0)
  CALL GPAN2(POSN,LEN,RETURN)
  CALL GPCLST
  CALL GPARN(INSID,3,6,0)
  CALL GPPV(INSID,3,1,1)
  CALL GPPNHS(INSID,PRFORM)
CALL GPAN2(LZ1,16,ZVAL)
CALL GPCLST

C ENTER ORIENTATION OR RETURN
CALL CHMDP(IWSID,2)
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
  IF NOT A PICK, GO TO AWAIT AN EVENT
ENDIF

C MAKE SURE PICK IS DEACTIVATED
1000 CALL GPPKMD(IWSID,1,1,2)
CALL GPESTL61
CALL GPESTL10
CALL GPESTL16)
C SET PICK FILTER(ALL CLASS 2 UNDETECTABLE)
CALL GPPKF(IWSID,1,0,ACLASS,3,ACLASS)
RETURN

SUBROUTINE TEPOX

SUBROUTINE TEPOX(IHSID,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 IHSID,PRFORM,PPATH(3),ACLASS111,CHOICE,LENG,IAXIS,
&RFLG
REAL CSIZE131,DATA135,PI1(6),P1(10),POINT1100L,VECl3J,YVECl3l,
&MAG,LOCAT131,XVECl3J,YVECl3l,ZVECl3J)
CHARACTER*20 CLEAR CHARACTER16 RETURN
DATA RETURN/'1. RETURN'
DATA LENG/16/
DATA PGN0.05,2.2/
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;
C DATA PRFORM/2/ FLAG FOR UPDATE
C DATA ACLASS/2/ INCLUSION CLASS LIST(CLASS 4 -- JOINTS AXIS

***********************************************************************
* REQUEST PICK INPUT
***********************************************************************
C CALL GPPKF(IWSID,1,1,ACLASS,3,ACLASS)
C SET PICK FILTER(ALL CLASS 3,4 DETECTABLE)
C CALL GPPKMD(IWSID,1,1,2)
C MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
C DEFINE PICK AREA
PAREAl11=(1.98+1)/2)*CSIZE111

MECHIN PROGRAM LISTING 214
PAREAl2 = ((.98 + 1/2) * CSIZE(1))
PAREAl3 = ((-.98 + 1/2) * CSIZE(1))
PAREAl4 = ((.48 + 1/2) * CSIZE(2))
PAREAl5 = 0
PAREAl6 = CSIZE(3)

CALL GPINPK(IMSID, 1, 0, PPATH, 1, PAREA, 0, DATA, 1)

PLACE PICK DEVICE IN THE EVENT MODE

CALL GPUPHS(IMSID, PRFORM)

GET PICK OR RETURN COMMAND

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

200 CALL GPOPST(6)

CALL GPADCN(ACLASS)

CALL GPPKID(101)

CALL GPCLST

CALL GPARKHSID,3,6,0.

CALL GPVPHS(IMSID, 1, 1)

CALL GPUPHS(IMSID, PRFORM)

RETURN PICK

*********************************************************************
*
* WAIT A PICK EVENT TO OCCUR AND RETRIEVE IT
*********************************************************************

300 CALL GPAHEV(1000, JMS, ICLA, IDEV)

IF (ICLA.EQ.5) THEN

GET PICK

IF (PPATH(2).EQ.101) THEN

RFLAG=1

GO TO 1000

ENDIF

CHOICE IS SECOND ITEM OF PICK PATH

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

X POINT ON Z VECTOR

CALL COMMAND(IMSID, 35)

GO TO 300

ENDIF

CALL VPROD(YVEC, ZVEC, XVEC)

XMAG=((YVEC(1)**2)+(YVEC(2)**2)+(YVEC(3)**2))**.5

YMAG=((YVEC(1)**2)+(YVEC(2)**2)+(YVEC(3)**2))**.5

XVEC(1)=YVEC(1)/XMAG

XVEC(2)=YVEC(2)/XMAG

XVEC(3)=YVEC(3)/XMAG

YVEC(1)=YVEC(1)/YMAG

YVEC(2)=YVEC(2)/YMAG

YVEC(3)=YVEC(3)/YMAG

ENDIF

MAKE SURE PICK IS DEACTIVATED

1000 CALL GPPKMO(IMSID, 1, 1, 2)

CALL GPEST(1)

SET PICK FILTER(ALL CLASS 2 UNDETECTABLE)

CALL GPPKF(IMSID, 1, 0, ACLASS, 1, ACLASS)

RETURN
SUBROUTINE VEO

SUBROUTINE VEO(IHSID,AXISR,CSIZE,ORIENT,RFLG)

*********************************************************************
* ----- SUBROUTINE VEO ----- *

* THIS SUBROUTINE WILL PROMPT FOR THE VALUATOR ENTRY OF *
* THE JOINT ORIENTATION *

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

INTEGER IHSID,ECHOSH,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),
& LNE(4),LNEY(6),LNEX(6),XBOX(15),&
& YBOX(15),ZBOX(15),LX(6),LY(6),LZ(6),OL(6),AREAl(6)
CHARACTER*7 CLEAR,INIT
DATA CLEAR/'0.0.0.0.0.0.0.0./
DATA INIT/'0.0.0.0.0.0.0.0./
DATA LX/O.,o.,o.,o.,o.,o.,o./
DATA LY/O.,o.,o.,o.,o.,o.,o./
DATA LZ/O.,o.,o.,o.,o.,o.,o./
DATA PRFORM/2/
DATA ACLASS/l/
DATA VLUEI/O./
DATA ECHOSH/2/

C DATA PRFORM/2/          FLAG FOR UPDATE

C DATA ACLASS/1/         INCLUSION CLASS LIST
C DATA VLUEI/0./
C DATA ECHO/3/
C DATA DLEN/0./
C DATA EVENT/3/
C DATA ECHOSH/2/

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

AREA1(1)=.08*CSIZE(1)
AREA1(2)=.90*CSIZE(1)
AREA1(3)=.70*CSIZE(1)
AREA1(4)=.90*CSIZE(2)
AREA1(5)=.02*CSIZE(1)
AREA1(6)=.90*CSIZE(3)
AREA2(1)=.52*CSIZE(1)
AREA2(2)=CSIZE(1)
AREA2(3)=.855*CSIZE(2)-.006
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)-.014
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(IHSID,1,1,ECHOSH)
CALL GPVLMO(IHSID,2,1,ECHOSH)
CALL GPVLMO(IHSID,3,1,ECHOSH)

C INITIALIZE DIALS FOR X, Y, Z INPUT
CALL GPNVL(IHSID,1,VLUE1,ECHO,AREAl,LOVAL,HIVAL,DLEN,DATA)
CALL GPNVL(IHSID,2,VLUE1,ECHO,AREAl,LOVAL,HIVAL,DLEN,DATA)
CALL GPNVL(IHSID,3,VLUE1,ECHO,AREAl,LOVAL,HIVAL,DLEN,DATA)

C ACTIVATE DIALS FOR X, Y, Z INPUT
CALL GPVLHO(IHSID,1,EVENT,ECHOSH)
CALL GPVLHO(IHSID,2,EVENT,ECHOSH)
CALL GPVLHO(IHSID,3,EVENT,ECHOSH)
UPDATE THE WORKSTATION
CALL GPUPWS(IHSID,PRFORM)
CALL GPARM(IHSID,16)

DISPLAY MENU AND GET CHOICE OF ENTER, RETURN OR VALUATOR

CALL CMDAND(IHSID,33)
CALL ENTERO(IHSID,CHOICE,CSIZE)

CALL GPAREV(1000,IHSID,ICLASS,IDEV)

PROCESS VALUATORS (NOT IN SUBROUTINE TO SPEED EXECUTION)

CALL GPGTVL(IVALUE)

GET ORIENT. ICON

CALL GPESTI
CALL GPOPSTI
CALL GPPLCI
CALL GPLT
SET POLYLINE COLOR
SET LINE SOLID
SET POLYLINE COLOR

CALL GPVLM(O(IHSID,1,1),RFLG)
CALL GPVLM(O(IHSID,2,1),RFLG)
CALL GPVLM(O(IHSID,3,1),RFLG)

DEACTIVATE VALUATORS

ENDIF

DEACTIVATE VALUATORS

ENDIF

DEACTIVATE VALUATORS

GO TO 600
GO TO 600
GO TO 1000
GO TO 1000
SUBROUTINE VUNIT

SUBROUTINE VUNIT(V, U)

******************************************************************************
----- SUBROUTINE VUNIT -----
******************************************************************************

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)**0.5
DO 20 I = 1, 3
  U(I) = V(I)/MAG
20 CONTINUE
RETURN
END

SUBROUTINE WRANAL

SUBROUTINE WRANAL(PNAM, PNAM, POINT, JNAM, JNUM, JOIN, LNAM, LNUM, LINK, & CSIZE, INAM, INUM, IC, IHSID)

******************************************************************************
----- SUBROUTINE WRANAL -----
******************************************************************************

INTEGER PNAM, PNAM, POINT, JNAM, JNUM, JOIN, LNAM, LNUM, LINK, & CSIZE, INAM, INUM, IC, IHSID
REAL LINK, JOIN, POINT, AXISR, CSIZE, AREA, IC, IHSID
CHARACTER*7 LNAM, JNAM, FILE

* FIND OUT TYPE OF ANALYSIS FILE TO BE WRITTEN *
CALL CMDNAM(IHSID, 90)
10 CALL MENU(IHSID, CSIZE, 18, ICH)

C IF(ICH.EQ.1) THEN
  IMP CALL IMP(PLAN, PNAM, POINT, JNAM, JNUM, JOIN, LNAM, LNUM, LINK, & CSIZE, INAM, INUM, IC, IHSID)
C ENDIF

C IF(ICH.EQ.2) THEN
  RSCR CALL RSCR(PLAN, PNAM, POINT, JNAM, JNUM, JOIN, LNAM, LNUM, LINK, & CSIZE, INAM, INUM, IC, IHSID, IRETT)
  IF(IRETT.EQ.1) THEN
    CALL CMDNAM(IHSID, 90)
    GO TO 10
  ENDIF
C ENDIF

C IF(ICH.EQ.3) THEN
  RSR (NOT YET AVAILABLE)
  GO TO 10
C ENDIF

MECHIN PROGRAM LISTING
SUBROUTINE WRREST

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

***** SUBROUTINE WRREST *****

*** THIS SUBROUTINE IS USED WRITE THE ANALYSIS RESTART FILE ***

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

INTEGER PNUM(100), JNUM(10), INUM(100), 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)
FILEO = ' 

REWIND 15

INITIALIZE STRING MODE
CALL GIPSMO(IMSID, 1, 1, 2)
CALL GIPST(IMSID, 1, 7, FILE1), AREA(7, 1, 0, FILE)
REQUEST FILE NAME
CALL CMDMSO(IMSID, 60)
REQUEST STRING
CALL GPGRST(IMSID, 1, 7, ISTAT, NUM, FILE1)
IF(FILE1.EQ. ' ' )THEN
   GO TO 2
ENDIF
IF(FILE1.EQ.FILE)THEN
   GO TO 9
ENDIF

FILE = FILE1

SEE IF FILE ALREADY EXISTS
CALL FILEX(FILE, IRET)

IF(IRET.EQ.0) THEN
   CALL GPSTMO(IMSID, 1, 1, 2)
   FILEO = FILE
   FILE = FILEO
   CALL CMDMSO(IMSID, 0)
   CALL GPSTN(IMSID, 1, 7, FILE, 1, AREA, 7, 1, 0, FILE)
   CALL GPGRST(IMSID, 1, 7, ISTAT, NUM, FILE)
   IF(FILE1.EQ. ' ' ) THEN
      GO TO 2
   ENDIF
IF(FILE1.NE.FILEO) THEN
   GO TO 3
ENDIF

ENDIF

CLOSE(UNIT=15)
OPEN (UNIT=15, FILE=FILE, STATUS='NEW', ERR=2)

MECHIN PROGRAM LISTING
* WRITE FIRST LINE OF THE RESTART FILE AND AXIS RANGE

9 WRITE(15, 10) IFLINE
10 FORMAT(A20)
   WRITE(15, *) AXISR
   WRITE(15, *) SCALE

* WRITE POINT DATA TO THE RESTART FILE

DO 80 I = 1, 100
   IF (PNAMI(I).EQ. 'BBBBBBB') THEN
      GO TO 90
   ENDIF
   NP = NP + 1
80 CONTINUE

90 WRITE(15, *) NP

DO 100 I = 1, 100
   IF (PNAMI(I).EQ. 'BBBBBBB') THEN
      GO TO 110
   ELSE
      WRITE(15,*) PNUM(I), PPOINT(J, J), J = 1, 3
   ENDIF
100 CONTINUE

* WRITE JOINT DATA TO THE RESTART FILE

110 DO 180 I = 1, 20
   IF (JNAMI(I).EQ. 'BBBBBBB') THEN
      GO TO 190
   ENDIF
   NJ = NJ + 1
180 CONTINUE

190 WRITE(15, *) NJ

DO 200 I = 1, 20
   IF (JNAMI(I).EQ. 'BBBBBBB') THEN
      GO TO 210
   ELSE
      WRITE(15, 191) JNAM(I), 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 290 I = 1, 30
   IF (LNAMI(I).EQ. 'BBBBBBB') THEN
      GO TO 290
   ENDIF
   NL = NL + 1
280 CONTINUE

290 WRITE(15, *) NL

DO 300 I = 1, 20
   IF (LNAMI(I).EQ. 'BBBBBBB') THEN
      GO TO 310
   ELSE
      WRITE(15, 291) LNAM(I), (NUM(I, J), J = 1, 16)
      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) LNAM(I),
   WRITE(15, *) (NUM(I, I), NUM(I, 2)
391 CONTINUE
SUBROUTINE WRSYN

SUBROUTINE WRSYN(PNAM,PNUM,POINT,AXISR,CSIZE,IMSID)
*****************************************************************************
** ----- SUBROUTINE WRSYN *****
**
** THIS SUBROUTINE IS USED WRITE THE SYNTHESIS DATA FILE
**
*****************************************************************************
INTEGER PNUM(30,4),NP
REAL POINT(30,14),AXISR,CSIZE(3),AREA
CHARACTER*7 PNAM(30),FILE,FILEO,FILE1,FILEN
*****************************************************************************
* FIND OUT TYPE OF SYNTHESIS FILE TO BE WRITTEN
*****************************************************************************
CALL SCMMANIIHSID,231 10 CALL SMENUIIHSID,CSIZE,6,IICH)
IFIICH.EQ.1 THEN
CALL RSCRSI(PNAM,PNUM,POINT,AXISR,CSIZE,IMSID,IRET)
IFI(IRET,EQ.1) THEN
GO TO 10
ENDIF
IFIICH.EQ.2 THEN
IF( IICH.EQ.3 THEN
RETURN
CONTINUE
ENDIF
RETURN
END

SUBROUTINE ZOOM

SUBROUTINE ZOOM(IMSID,AXISR,CSIZE)
*****************************************************************************
** ----- SUBROUTINE ZOOM *****
**
** THIS SUBROUTINE IS USED TO ZOOM IN ON AN AREA ON THE SCREEN
**
*****************************************************************************
INTEGER IMSID,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),POSZ(3),CSIZE(3),M1(4)
C
DATA PRFORM/2,
DATA DIst/0.5/
DATA PARELL/1/ DATA POINT/0.5/1.5/5.0/
DATA NORMAL/1.0/0./
DATA UP/0.0,0.1/ DATA ILOC/0.,0.,0./ DATA BOX1/-48,-48,-98,-98,-98,-98,-48,-48/
DATA DATA/1,75*0/
NEAR=AXISR+1.64*AXISR
FAR=AXISR-1.64*AXISR
VP1(1)=IBOX1(1)+1.1/2.
VP1(2)=IBOX1(1)+1.1/2.
VP1(3)=IBOX1(4)+1.1/2.
VP1(4)=BOX1(6)*1.1/2.
VP1(5)=BOX1(1)*1.1/2.
VP1(6)=BOX1(1)*1.1/2.

*********************************************************************
* ACTIVATE THE LOCATOR TO GET NEW VIEW PORT
*********************************************************************
C
MAKE SURE PICK IS DEACTIVATED (REQUEST MODE)
C
CALL GPPOMO(IMSID,1,1,2)
CALL GPLCHO(IMSID,1,1,2)
C
DEFINE PICK AREA
PAREA(1)=VP1(1)*CSIZE(1)
PAREA(2)=VP1(2)*CSIZE(1)
PAREA(3)=VP1(3)*CSIZE(1)
PAREA(4)=VP1(4)*CSIZE(1)
PAREA(5)=VP1(5)*CSIZE(1)
PAREA(6)=VP1(6)*CSIZE(1)
C
INITIALIZE LOCATOR
CALL GPMNLC(IMSID,1,4,(LOC,1,PAREA,0,DATA)
C
TRaverse ALL ACTIVE VIEWS
CALL GPUPHS(IMSID,PRFORM)
CALL CMDND(IMSID,57)
C
CALL GPRQLC(IMSID,1,ISTAT,IV,POS1)
C
INITIALIZE LOCATOR FOR BOX
CALL GPMNLC(IMSID,1,4,POS1,5,PAREA,72,DATA)
C
CALL CMDND(IMSID,58)
C
POS1(1)=POS1(1)-(.6550*AXISR)
POS1(2)=POS1(1)-(.4912*AXISR)
POS2(1)=POS2(1)-(.6550*AXISR)
POS2(2)=POS2(1)-(.4912*AXISR)
XRANG=ABS(POS1(1)-POS2(1))
YRANG=ABS(POS1(2)-POS2(2))
IF XRANG.GE.YRANG THEN
   WINDOW=1=(POS1(1)+POS1(1))/2.-(XRANG/2.)
   WINDOW=2=(POS1(1)+POS1(1))/2.+(YRANG/2.)
   WINDOW=3=(POS1(2)+POS1(2))/2.-(YRANG/2.)
   WINDOW=4=(POS1(2)+POS1(2))/2.+(YRANG/2.)
   POINT(1)=(POS1(1)+POS1(1))/2.+(1.5)
   POINT(2)=(POS1(2)+POS1(2))/2.+(1.5)
ENDIF
IF(YRANG.LT.XRANG) THEN
   WINDOW=1=(POS1(1)+POS1(1))/2.-(YRANG/2.)
   WINDOW=2=(POS1(1)+POS1(1))/2.+(XRANG/2.)
   WINDOW=3=(POS1(2)+POS1(2))/2.-(XRANG/2.)
   WINDOW=4=(POS1(2)+POS1(2))/2.+(XRANG/2.)
   POINT(1)=(POS1(1)+POS1(1))/2.+(1.5)
   POINT(2)=(POS1(2)+POS1(2))/2.+(1.5)
ENDIF

*********************************************************************
* SET THE VIEW CHARACTERISTICS FOR AXES VIEW
*********************************************************************
C
SET THE CHARACTERISTICS
CALL GPVMPS(IMSID,4,MINDOM,VP1,PAEALL,POINTP,DIST,NEAR,FAR)
C
ACTIVATE THE VIEW
CALL GPVCH(IMSID,4,2,2,1,0,2,5,2)
C
RETURN
END
The vita has been removed from the scanned document