

NASA Technical Memorandum 81919

NASA-TM-81919 19810008241

A USER'S GUIDE FOR DTIZE-
AN INTERACTIVE DIGITIZING AND
GRAPHICAL EDITING COMPUTER PROGRAM

Carolyn C. Thomas

FOR REFERENCE

NOT TO BE CHECKED FROM THIS ROOM

LIBRARY COPY

JANUARY 1981

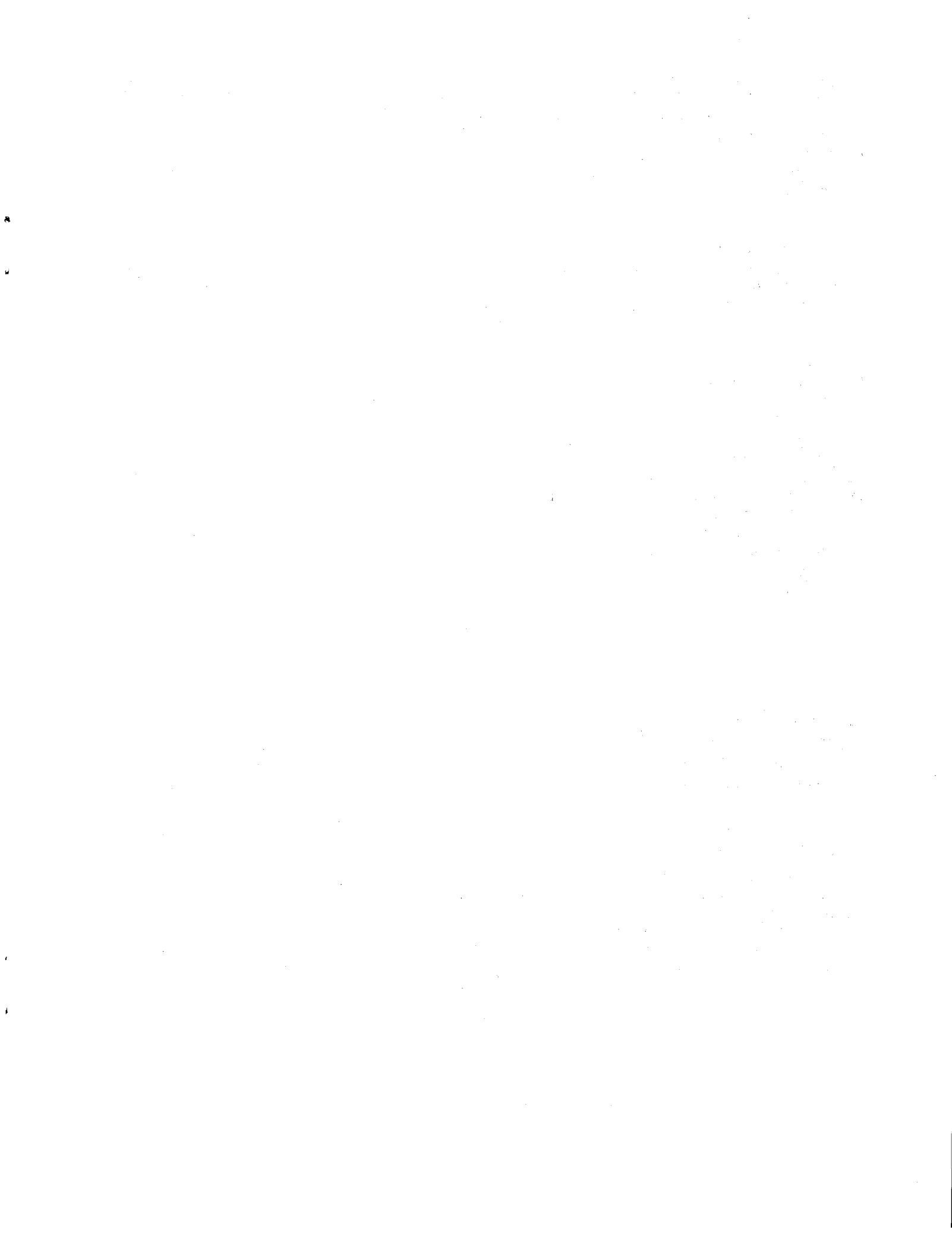
JAN 26 1981

LANGLEY RESEARCH CENTER
LIBRARY, NASA
HAMPTON, VIRGINIA

NASA

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665



A USER'S GUIDE FOR DTIZE -
AN INTERACTIVE DIGITIZING AND GRAPHICAL EDITING COMPUTER PROGRAM
Carolyn C. Thomas
Langley Research Center

SUMMARY

This report is a user's guide for DTIZE, a 2-dimensional digitizing program with graphical editing capability. DTIZE provides the capability to simultaneously create and display a picture on the display screen. Data descriptions may be permanently saved in three different formats. DTIZE creates the picture graphics in the locator mode, thus inputting one coordinate each time the terminator button is pushed. Graphic Input devices (GIN) are also used to select function commands from a command menu. These menu commands and the program's interactive prompting sequences provide a complete capability for creating, editing, and permanently recording a graphical picture file.

DTIZE is written in Fortran IV language for the Tektronix 4081 graphics system utilizing the Plot 80 Distributed Graphics Library (DGL) subroutines. The Tektronix 4953/4954 Graphics Tablet with mouse, pen, or joystick are used as Graphics Input devices to create picture graphics. DTIZE runs under the control of the Tektronix 4081 Graphic Operating System (GOS) with extended GIN features (needed for DGL graphic input routines).

INTRODUCTION

DTIZE is a 2-dimensional digitizing program with graphical editing capability which provides the capability to simultaneously create and display a picture on the Tektronix 4081 display screen. The data descriptions may be permanently saved in three different formats. The first format records the window coordinates, x and y coordinates, and a code word which describes the connectivity characteristics between data points. The connectivity information defines a move or a draw between each point and whether a solid or dashed line is to be drawn. This format is used by DTIZE for redisplaying, editing, and adding to a previously existing picture file. The second format is the Picture Data Base (PDB) format which is compatible with the Tektronix 4081 Graphics Function Manager (GFM) utility program. FM routines may be used for further picture editing and for the addition of alphanumeric text. The third format is the Plot (PLT) format and is compatible with GOS for displaying the picture.

DTIZE creates the picture graphics in the locator mode. This means that one coordinate is input each time the terminator button for the mouse, pen, or joystick is pushed. The GIN devices are also used to select function commands from a command menu shown in figure 1. These menu commands and the program's interactive prompting sequences provide a complete capability for creating, editing, and permanently recording a graphical picture file.

DTIZE is written in Fortran IV language for the Tektronix 4081 graphics system utilizing the Plot 80 Distributed Graphic Library (DGL) subroutines. The program listing is given in appendix A. The Tektronix 4081 hardware is shown in figure 2. The Tektronix 4953/4954 Graphics Tablets with mouse or pen and the joystick are used as Graphic Input devices (GIN) to create picture graphics. DTIZE runs under the control of the Tektronix 4081 Graphic Operating System (GOS) with the extended GIN features (needed for DGL graphic input routines).

N81-16760#

USE OF THE TEKTRONIX 4081 REFRESH CAPABILITY

As an operator digitizes a picture from the tablet (or inputs with the joystick), an image appears on the display monitor in refresh. Up to 25 points can be selected before this portion of the picture is automatically fixed, on the screen and written to disk. As a program option, DTIZE permits the refresh portion to be fixed at any time during a digitizing session. DTIZE has editing features which allow backup or rejection of the refresh portions of the picture and will also allow removal of one line at a time in the fixed portion of the picture.

Program Files

The operational program consists of two files, DTIZE.OBJ and DTIZE.LIB, which should be stored on the Tektronix 4081 user disk. Under GOS with DGL graphics, DTIZE is started by typing RUN DTIZE(cr). The program will prompt for user action at the bottom of the display screen.

Output File

The first prompt will ask for the name of the file to which the graphical data are to be written for storage. This file can be any name (6 char. max) with any extension (3 char. max). DAT is the default extension. The file will be written on the user disk and cannot be put in a library.

Ex:	PICT1	(Extension will be set to DAT)
	TRIAL.CCT	
	CCT/PIC	(Not a valid name)

The prompt will then ask if a file is being created or if a file previously created will be used. If creating a file, 300 blocks of space will be allotted for the output file. There is a built-in program check which will report to the monitor when 295 of these blocks have been filled. A warning message will also be displayed with a bell sound. It is the user's responsibility to open a new file with more space and copy the file to this new file. If adding to a file already created, the REDRAW command must be picked from the menu as the first command following the prompt which tells the user that he is ready to digitize.

Reference Axis

A small reference axis will be drawn in the lower left corner of the display screen. R and S are the reference axis names. The program will prompt the user to specify the x, y or z orientation axis in relation to R. An x, y, or z is then entered from the keyboard. The next prompt asks that an x, y or z axis be specified in relation to S. An x, y, or z is then entered. There is a built-in program check which will not allow R and S to be the same. The prompt then asks for a constant for the third dimension. At the present time, this constant is not used and is for future enhancement of the program to 3 dimensions. The R and S will be replaced on the small reference axes to coincide with the user's specified coordinate system.

Window Coordinates

The next prompts will ask for minimum x, minimum y, maximum x and maximum y. These define the range of coordinates along each axis in the coordinate plane. THERE ARE NO DEFAULT VALUES. The tablet and display window are set to these values. The program automatically proportions the viewport relative to the input values for the window coordinates. If the file is an existing file, there will be no prompt for the window coordinates. They will be read from the input file.

Skew Correction

There is a prompt which queries whether or not a digitizing tablet skew correction is wanted. If the choice is y, the prompt will tell the user to enter with the mouse or pen in the following corners of the desired area: lower left, lower right and upper right. When these three points have been entered, a bell will ring.

Menu Location

The menu may be placed vertically on the left or right side of the digitizing tablet. The next prompt will ask that the lower left corner and the upper right corner of the menu be entered with the mouse or pen.

Digitizing

The next prompt will be a statement that digitizing may begin. From this point, the mouse or pen may be used to input points. The program is written in locator mode, and a line is drawn between points as they are input. To pick from the menu, the mouse or pen is moved to the menu portion of the tablet and a point is input within the box for the function to be performed. Prompts are displayed at the bottom of the screen to show which function has been picked.

MENU

The menu (fig. 1) is a rectangular area made up of 15 boxes of which 13 are menu commands and 2 are blank for future use. By using the menu, the current picture can be added to and edited in several ways. The menu for the tablet is drawn on paper and placed vertically on the left or right side of the tablet. One box in the menu will allow the joystick to be chosen for editing or addition purposes. In this case, the menu appears on the right side of the display screen. A detailed explanation of each of the 13 menu commands is given in the following paragraphs.

Accept

This command causes the refresh portion of the current picture to be written into the data base and to be displayed in fixed mode on the display screen. When control is returned, the last point chosen will be the location from which the next move or draw will take place. ACCEPT is automatically picked after 25 points are chosen.

Reject

This command will cause the current refresh portion of the drawing to be cleared off the screen and return to the last point chosen before the refresh portion started.

Continue

This command will blink the last line drawn in refresh with an 'x' displayed at the last point chosen.

Backup

This command will cause the last vector drawn in the refresh portion of the picture to be removed.

Locate

This command causes a move to the next point chosen rather than a draw. The circle and spline commands return to locate mode after execution.

Remove

This command will allow removal of one vector at a time in the fixed mode portion of the picture. After the vector is removed, the picture is automatically redrawn.

Line

The default mode for vector display is a solid line. If, at some time, the mode is changed to dash, this command will return to solid line mode.

Dash

This will cause vector output to be dashed lines using the default mask stored in the computer.

Spline

Upon entrance, the refresh portion of the picture is changed to fixed mode and the points are stored on the output data file. This command will allow points to be picked along a curved line and then will fit a spline through the points using a spline routine acquired from Eglin AFB incorporated in the program. Up to 20 points on a curve can be input or as few as three. If 20 points are input, the program will automatically draw a spline in refresh, but if fewer than 20 points are picked, SPLINE must be rechosen from the menu by the user to show he is finished. The fit can be improved by picking points closer together and a peak will be reached better if the peak point and a near point on each side of the peak are chosen. The user is given the opportunity to accept or reject the spline. If accepted, the curve points are written into the data base and the curve is displayed on the screen in fixed mode. The program is returned to LOCATE mode. If rejected, an opportunity is provided to rechoose points. If the user decides not to draw a spline, the EXIT command can be selected from the menu list which permits another command to be chosen from the menu to continue the digitizing session.

Circle

The refresh portion of the picture is changed to fixed mode and the points are stored on the data file upon entering the circle subroutine. This command will allow a circle to be drawn using the DGL circle routine. A prompt asks that the center of the circle and one point on the circle be digitized. As with the spline, the user makes the choice whether to keep the circle. If accepted, the curve is written into the data base and the circle is displayed in fixed mode. If rejected, the opportunity to digitize the center of the circle and a point on the circle is given. If the user decides not to draw a circle, the EXIT command may be picked as with the SPLINE. Upon leaving this routine, the program returns to LOCATE mode.

Redraw

This command will erase the screen and redraw both the fixed and refresh portions of the picture. It is best to redraw the picture several times during a digitizing session as this will save the file. If the computer should fail for some reason, the data will not be lost.

Joy

This command will allow the joystick to be used with the same menu as the one used for the tablet. This command is replaced by TABLET on the display screen and will allow return to the tablet and mouse. The menu appears on the right-hand side of the display screen in refresh. The cursor is a box.

Exit

This command will write the refresh portion to the data base file and will then ask if a picture data base (PDB) file is desired. If a PDB file is wanted, it is written and stored under the user's file name with the PDB extension. A GFM workspace file (WSP) that has the correct window and viewport with a WSP extension will also be written. If the PDB file is used in GFM, the workspace file (WSP) file should be loaded with GFM. The user will also be asked if a plot file is wanted. If so, it will be stored under the user's file name with the PLT extension.

A SESSION WITH DTIZE

The following is a tutorial session with DTIZE. It will lead the user step by step through several commands. At the end of the tutorial, the user should experiment by trying to complete the drawing. A copy of the sample picture (fig. 3) should be placed on the graphics tablet anywhere except over the menu area.

Getting Started

To start, RUN DTIZE (cr) is entered from the keyboard. From this point, prompts will appear across the bottom of the display screen to direct the user's actions. In order to distinguish between prompts and input, the prompts will be in upper case letters and the inputs will be lower case letters.

HIT (CR) TO CONTINUE

(Depress the carriage return.)

INPUT NAME OF FILE TO SAVE DATA

shuttle (The extension DAT will be added automatically)

INPUT "C" FOR CREATING OR "P" FOR PERMANENT

c

A small axis will appear in the lower left-hand corner of the screen. The horizontal direction will be marked with an R and the vertical direction will be marked with an S. The user will then be prompted.

ENTER RELATIONSHIP OF X,Y,Z TO R

x

ENTER RELATIONSHIP OF X,Y,Z TO S

y

ENTER CONSTANT Z

(If the user had entered y and z as his R and S this prompt would have queried ENTER CONSTANT X.)

0.0

ENTER MINIMUM X

-30.0

ENTER MINIMUM Y

-17.0

ENTER MAXIMUM X

30

ENTER MAXIMUM Y

17.

DO YOU WANT SKEW CORRECTION?
TYPE "Y" FOR YES OR "N" FOR NO

y

ENTER LOWER LEFT CORNER OF PICTURE
ENTER LOWER RIGHT CORNER OF PICTURE
ENTER UPPER RIGHT CORNER OF PICTURE

(At this point, enter coordinates P1, P2, and P3 on the picture with the mouse or pen. A bell will ring when three coordinates have been entered and a prompt will ask for more information.)

ENTER LOWER LEFT CORNER OF MENU
ENTER UPPER RIGHT CORNER OF MENU

(Enter the above coordinates on the menu.)

YOU ARE NOW READY TO DIGITIZE YOUR PICTURE.

(The program now has all the necessary information to allow a picture or design to be drawn.)

Drawing The Design

The mouse should be placed on A in the example picture and the terminator pushed to enter the coordinates. Point B is entered and a refresh line will appear on the display screen. LOCATE is chosen from the menu and points C and D are entered. After picking SPLINE from the menu, the prompt YOU MUST PICK 3 POINTS BUT NO MORE THAN 20 will appear at the bottom of the display screen. With the mouse or pen, the points labeled on the example as 1,2,3. . . 10 are entered. SPLINE is again picked from the menu to indicate that all points have been entered. A refresh spline fit along with the points will appear on the screen. The prompt DO YOU WANT TO KEEP SPLINE? will appear at the bottom of the screen. A yes (y) or no (n) is entered from the keyboard. In this sample session, the user should enter a y. The user should be aware that when SPLINE was picked from the menu, the refresh portion of the picture became fixed. SPLINE is again picked from the menu and points 11, 12. . . . 30 are entered. Twenty points have been picked and the program will automatically draw a spline in refresh. A y should be the answer to the prompt DO YOU WANT TO KEEP SPLINE?. After DASH and LOCATE are selected from the menu, the points E, F, and G are entered with the mouse or pen. G should not have been input before LOCATE was again picked from the menu. To eliminate the unwanted line, the BACKUP command is picked from the menu and then LOCATE. G and H can now be input. Pick ACCEPT from the menu and the refresh portion of the picture becomes fixed. The small circle on the side of the example can be drawn by picking CIRCLE from the menu and following the prompts. The user should continue until the example is finished, and then choose EXIT in the menu. At this point the following prompts will appear.

DO YOU WANT A PDB FILE?

y

(A picture data base file will be written for you along with a workspace file. When the picture is drawn in GFM, the workspace file must be loaded as it will set the correct window and viewport.)

DO YOU WANT A PLT FILE?

y

(A plot file is written which can be displayed with the command: DISPLAY file name.)

There are now four files which will allow returning to DTIZE to edit or add to the picture, going to GFM to edit or add text, and displaying the PLT file on the screen in GOS.

APPLICATIONS

The DTIZE program creates a numerical data base of x, y points which can be formatted as input to any applications program requiring this type of data.

Potential applications include digitizing node points to pass to a finite element modeling program, digitizing contour lines from maps, and using DTIZE to create the model geometry of spacecraft or other designs.

CONCLUDING REMARKS

DTIZE, a 2-dimensional digitizing program with graphical editing capability, enables the user to simultaneously create and display a picture on the display screen. Menu commands and the program's interactive prompting sequences provide a complete capability for creating, editing, and permanently recording a graphical picture file. DTIZE is written in Fortran IV language for the Tektronix 4081 graphics system utilizing the Plot 80 Distributed Graphics Library (DGL) subroutines. The Tektronix 4953/4954 Graphics Tablet with mouse, pen, or joystick are used as Graphics Input devices to create picture graphics. DTIZE runs under the control of the Tektronix 4081 Graphic Operating System (GOS) with extended GIN features (needed for DGL graphic input routines).

This report is a user's guide for the interactive digitizing and graphical editing program. Program functions and menu commands are described and an example design is outlined and drawn. Potential applications include digitizing node points to pass to a finite element modeling program, digitizing contour lines from maps, and creating the model geometry of spacecraft or other designs.

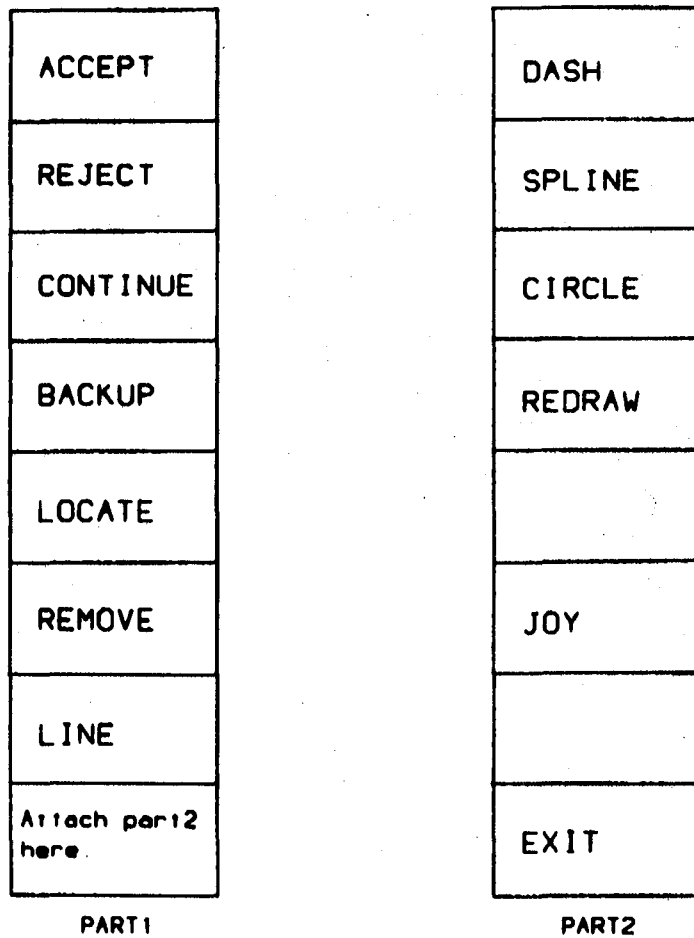
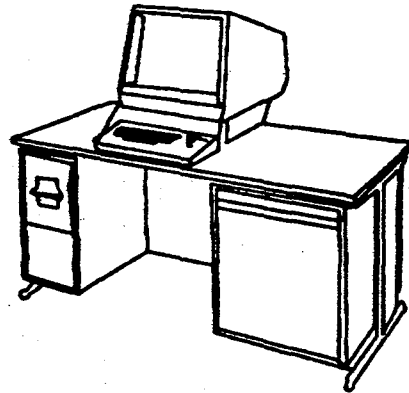
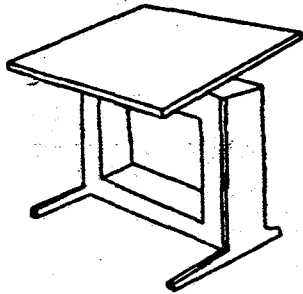


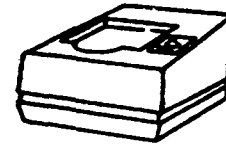
Fig. 1 - Menu For Tablet

This is a PDB file by the name of MENUUDT.PDB. It can be instanced in GFM and scaled up or down depending on the size tablet being used.

LARGE TABLET



HARD COPY



MASS STORAGE

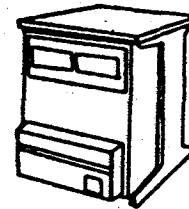


Fig. 2 - 4081 Hardware

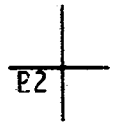
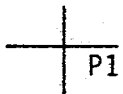
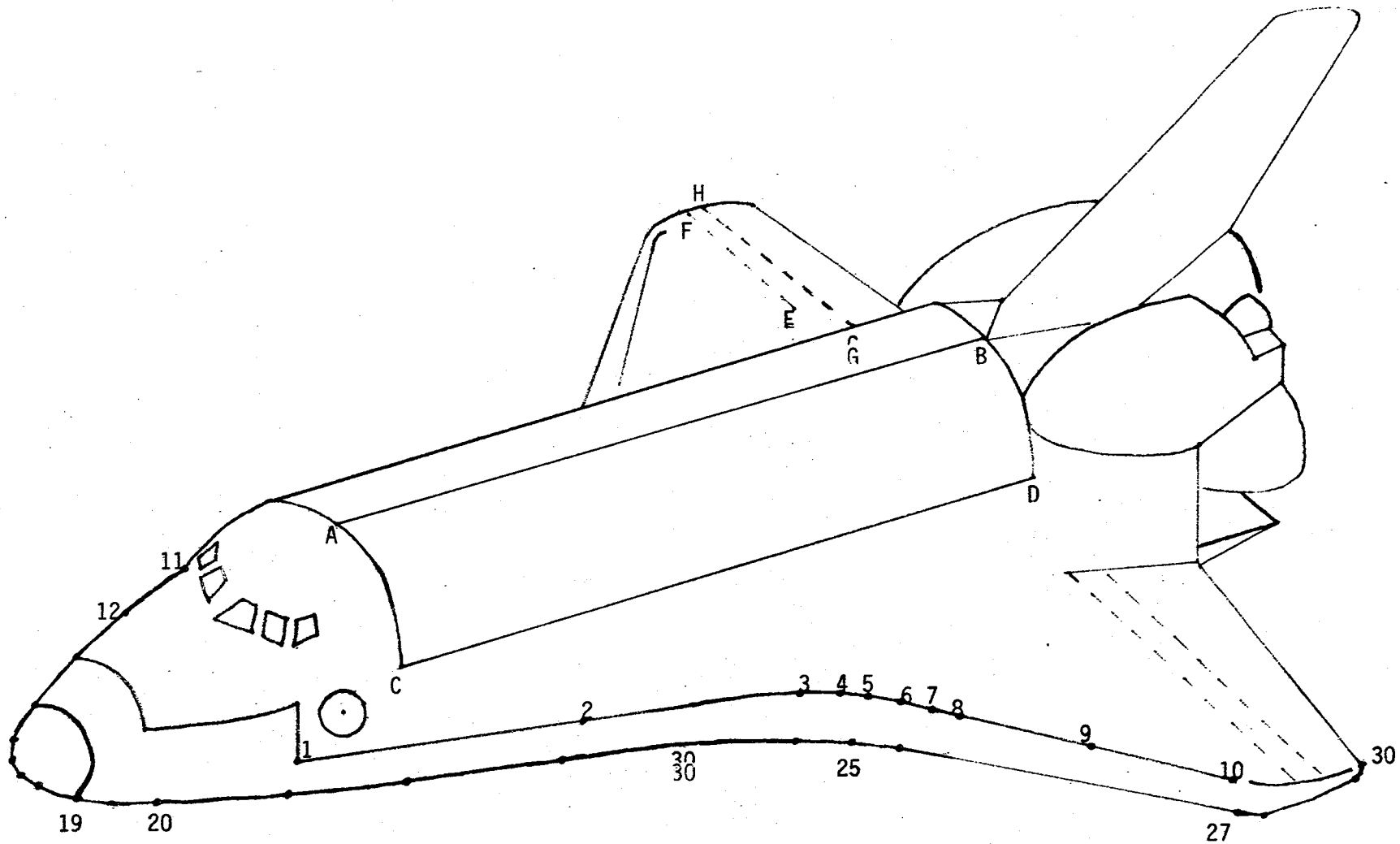
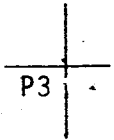
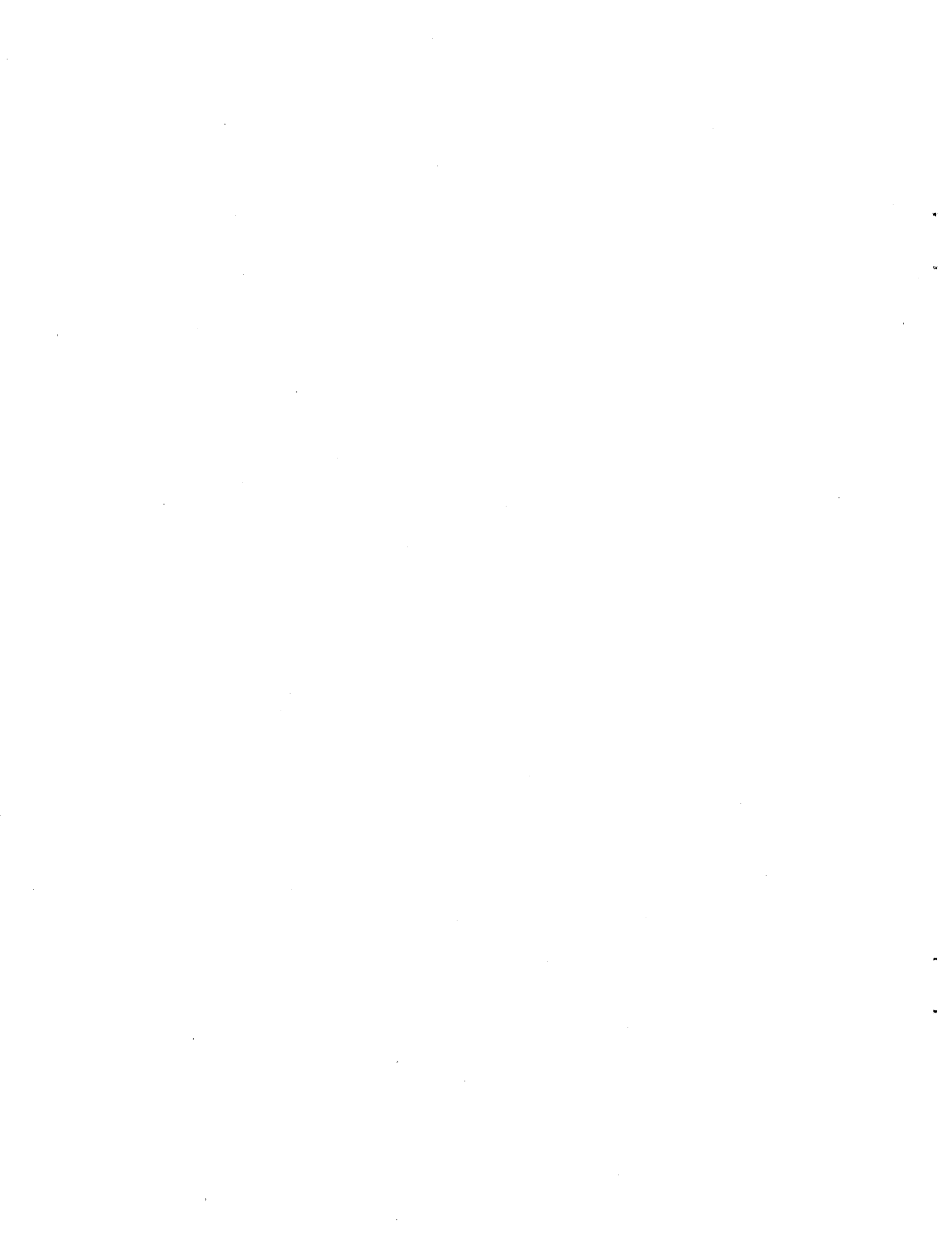
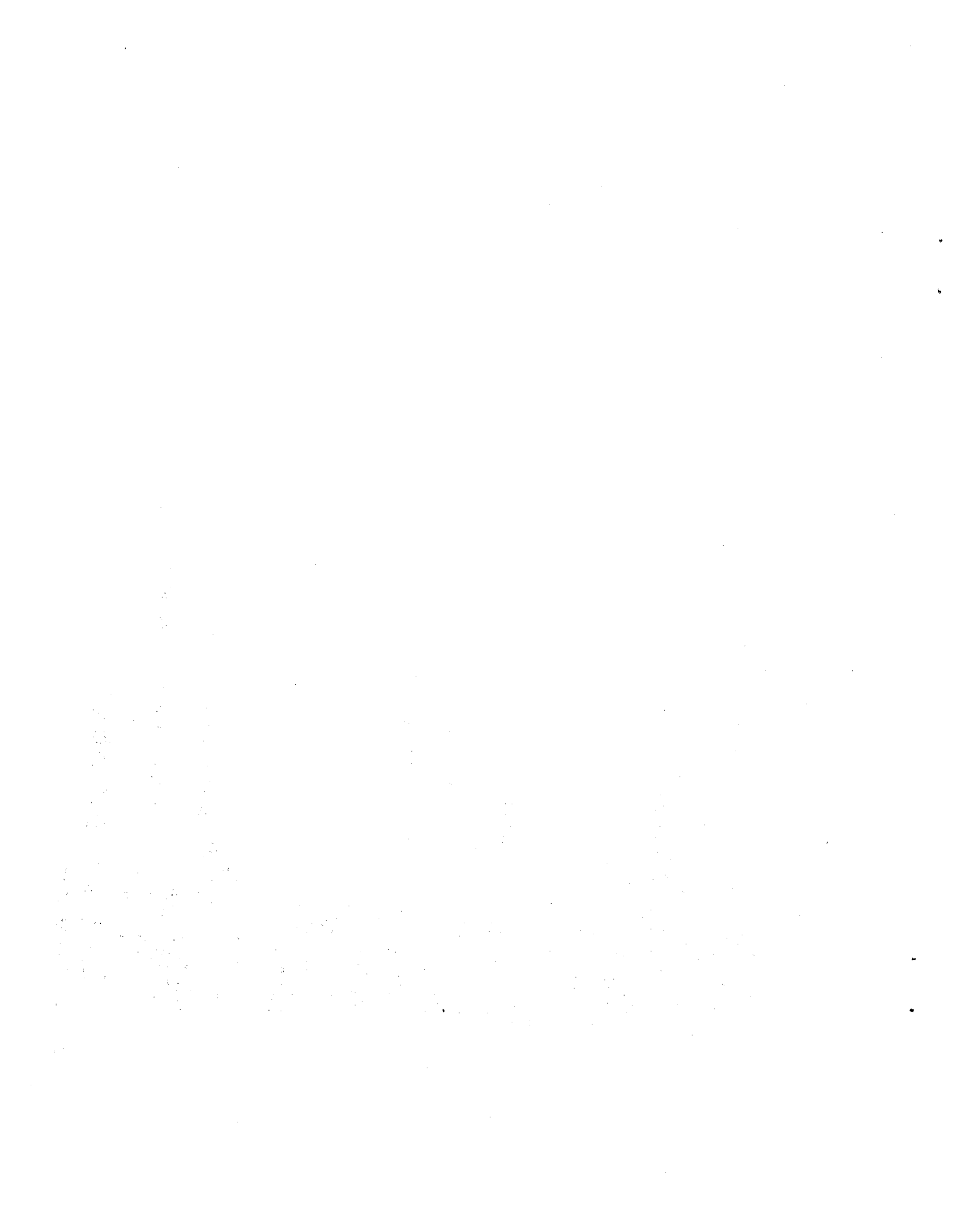


Fig. 3 - Example For Digitizing



APPENDIX A

Listing For Code 'DTIZE'




```

PROGRAM DIGIT
IMPLICIT INTEGER*2 (I-N,*)
INTEGER*1 IDD,MORD(25),IR(2),IS(2),ISKW,ILINE,NAMFIL(15),CR
INTEGER*1 FILE,C,P,Y
COMMON USKET(2,3),IR,IS,ISEG(3),IPOS(6),ISKW,CONST,IT,IPOINT(4),
&IVIEW(4),FILE
COMMON /MENU/IXY(2),IDIAC(4),NT,IDD,IM,PTXBUF(25),PTYBUF(25),
&MORD,ILINE,NAMFIL,IYLOC(16),IXMIN,IXMAX,IYMIN,IYMAX,WDIAC(4)
COMMON /PDBCOM/LUFILE,MDUM(133)
LOGICAL ISTAT(3)
EXTERNAL EDGET
DATA C/'C'/,P/'P'/,Y/'Y'/
DATA CR/'00'/,NEWJSW/'0038'/

```

C

```

CALL BFDEV (0,'DC: ')
CALL CONEW (0)
CALL BFDEV (2,'T00: ')
CALL BFDEV (5,'JOY: ')

```

C

C

```

ACCESS OVERLAY FILE
CALL BFACT (6,'DTIZE/OVER LIB ')
CALL OVLU (6)

```

C

```

CALL OVLINK ('TITLE ')

```

C

C

```

CHANGE DEFAULT EXTENSION OF FILES TO DAT
CALL BFDEX ('DAT')

```

C

C

```

INITIALIZE REFRESH EDITOR SEGMENTS AND POSTIONS
ISEG(1)-6
ISEG(2)-7
ISEG(3)-8
IPOS(1)--20000
IPOS(2)--23500
IPOS(3)--20000
IPOS(4)--22500
IPOS(5)--20000
IPOS(6)--21500
IR(2)-13
IS(2)-13

```

C

```

CALL ASKNEW (EDGET)
CALL EJSW (11,NEWJSW)
CALL EDNEW (ISEG,IPOS,2)
CALL EDPUT ('INPUT NAME OF FILE TO SAVE DATA',31)
CALL EDPUT (CR,1)
CALL ASKAN (NAMFIL,15)
CALL EDPUT (CR,1)
CALL EDPUT ('INPUT "C" FOR CREATING OR "P" FOR PERMANENT',43)
CALL EDPUT (CR,1)
CALL ASKAN (FILE,1)
CALL EDCLO
IF (FILE.EQ.C) CALL BFNEW (3,NAMFIL,300)
IF (FILE.EQ.P) CALL BFACT (3,NAMFIL,.T.)

```

C

C

```

DRAW REFERENCE AXIS

```

```
C CALL OVLINK ('REFAXS ')
C SET WINDOW,VIEWPORT,CREATE CURSOR SEGMENT, AND SKEW?
C CALL OVLINK ('SETUP ')
5 CALL OVLINK ('ITAB ')
  IF (IDO EQ 1) GO TO 10
  CALL OCISEL (LU)
  IF (LU EQ 2) GO TO 5
  CALL OVLINK ('IJOY ')
  IF (IDO EQ 1) GO TO 10
  GO TO 5
C
C CHECK TO SEE IF PDB AND/OR PLT FILES WANTED
C
10 CALL OVLINK ('PDBPLT ')
  STOP
  END
```

SUBROUTINE CKFLST (TEXT)

THIS ROUTINE CHECKS TO SEE HOW MANY BLOCKS OF DATA HAVE BEEN
WRITTEN ON THE OUTPUT FILE

```
IMPLICIT INTEGER*2 (I-N,*)  
DIMENSION ITEXT(3)  
INTEGER*4 IDO,MORD(25),IR(2),IS(2),ISKW,ILINE,NAMFIL(15),CR,FILE  
COMMON USKET (2,3),IR,IS,ISEC*29,IBOS(4),ISKW,CONST,IT,IPOINT(4),  
&VIEW(4),FILE  
COMMON /MENU/IXY(2),IDIAC(4),NT,IDO,IM,PTXBUF(25),PTYBUF(25),  
&MORD,ILINE,NAMFIL,IYLOC(16),IXMIN,IXMAX,IYMIN,IYMAX,WDIAC(4)
```

```
CALL QAFEXT(3,ITEXT)  
IF (ITEXT(1)-TEXT(2)) LT .5) GO TO 15  
RETURN
```

```
15 CALL BEEL ( , , , ) GO TO 12  
CALL EDPUT (CR,17A)  
12 CALL EDPUT ('YOU HAVE USED ',14)  
CALL EDPUT (ITEXT(2),1)  
CALL EDPUT ('BLOCKS OF THE MEMORY',25)  
CALL EDPUT (ITEXT(1),1)  
CALL EDPUT ('BLOCKS SAVED FOR ',18)  
CALL EDPUT (NAMFIL,15)  
RETURN
```

```
END
```

```
16 CALL ABVCE (20,IXA)  
17  
18  
CALL BOD (20)  
CALL AXEL ('OVERRIDE YI BY BYNATVHOGEA' 'HANSLOW' 'AV',21)  
CALL ANOAE (IXA)  
CALL OBEA (20)  
CALL D (20)  
CALL AX (20)  
CALL AXIS (21,3500) GO TO 2  
CALL ABVCE (20,IXA)  
2 IXA(5)-IXA(5)+18  
CALL BOD (20)  
CALL AXEL ('OVERRIDE IS V DICLITITIC B500KWH N5TIC DCF B00LINES',24)  
CALL ANOAE (IXA)  
CALL OBEA (20)  
CALL AIRDY (0)  
IXA(5)-51000  
IXA(11)-52000  
CALL BOD (20)  
CALL COMEA (0)  
CALL BOD (20)
```

```
DIMENSION ITEXT(3)  
INTEGER*4 ICOML  
IMPLICIT INTEGER*2 (I-N,*)  
COMMONLINE IIFE
```

SUBROUTINE TITLE
 IMPLICIT INTEGER*2 (I-N,*)
 INTEGER*1 ICONT
 DIMENSION IXY(2)

C

```

CALL BFDEV (0, 'DC ')
CALL GONEW (0)
CALL ERASE
IXY(1)--20000
IXY(2)--21000
CALL VTSCA (6)
CALL OPEN (50)
CALL VMOVE (IXY)
CALL VTEXT ('DTIZE IS A DIGITIZING PROGRAM USING DGL ROUTINES',48)
CALL POST (50)
5 IXY(2)=IXY(2)+10
CALL VPLACE (50,IXY)
IF (IXY(2) .LT. 3000) GO TO 5
CALL FIX (50)
CALL DELETE (50)
IXY(2)--21000
CALL OPEN (51)
CALL VMOVE (IXY)
CALL VTEXT ('CREATED AT NASA/LANGLEY, HAMPTON, VA.',37)
CALL POST (51)
10 IXY(2)=IXY(2)+10
CALL VPLACE (51,IXY)
IF (IXY(2) .LT. 0) GO TO 10
CALL FIX (51)
CALL DELETE (51)
IXY(2)--21000
CALL OPEN (52)
CALL VMOVE (IXY)
CALL VTEXT ('BY CAROLYN C. THOMAS      SSD/SEB',32)
CALL POST (52)
15 IXY(2)=IXY(2)+10
CALL VPLACE (52,IXY)
IF (IXY(2) .LT. -3000) GO TO 15
CALL FIX (52)
CALL DELETE (52)
IXY(2)--21000
IXY(2)--21000
CALL OPEN (54)
CALL VMOVE (IXY)
CALL VTSCA (3)
CALL VTEXT ('HIT (CR) TO CONTINUE',20)
CALL POST (54)
READ (1,1) ICONT
1 FORMAT (A1)
CALL DELETE (54)
CALL ERASE
RETURN
END

```

```

SUBROUTINE REFAXS
IMPLICIT INTEGER*2 (1-N,*)
INTEGER*2 XYMOV(10),XYAX(12)
INTEGER*1 IAXIS(5),TEXT1(35),TEXT2(35),TEXT3(27),FILE
INTEGER*1 IDO,MORD(25),IR(2),IS(2),ISKW,ILINE,NAMFIL(15),CR
COMMON USKET (2,3),IR,IS,ISEC(3),IPOS(6),ISKW,CONST,IT,IPOINT(4),
&VIEW(4),FILE
COMMON /MENU/IXY(2),IDIAG(4),NT,IDO,IM,PTXBUF(25),PTYBUF(25),
&MGRO,ILINE,NAMFIL,IYLOC(16),IXMIN,IXMAX,IYMIN,IYMAX,WDIAG
EXTERNAL EDGET
DATA XYMOV/-24000,-16000,0,300,0,-300,0,1500,1500,-1500/
DATA XYAX/-150,-150,150,0,-3000,150,150,-150,-3000,0,150,150/,
&IAXIS/'R','S','X','Y','Z'/,CR/13/
DATA TEXT1/'ENTER RELATIONSHIP OF X, Y, Z TO R',13/,
&TEXT2/'ENTER RELATIONSHIP OF X, Y, Z TO S',13/,
&TEXT3/'R AXES CANNOT EQUAL S AXES',13/

```

```

DRAW REFERENCE AXIS

```

```

IH-1
IV-2
CALL OPEN (1)
CALL VABS (.TRUE.)
CALL VMOVE (XYMOV)
CALL VABS (.FALSE.)
CALL VDRAW (XYAX)
CALL VMOVE (XYMOV(4))
CALL VDRAW (XYAX(2),3)
CALL VMOVE (XYMOV(6))
CALL VDRAW (XYAX(7))
CALL VMOVE (XYMOV(8))
CALL VDRAW (XYAX)
CALL VMOVE (XYMOV(3))
CALL VDRAW (XYAX(7),3)
CALL VMOVE (XYMOV(5))
CALL VDRAW (XYAX(2))
CALL POST (1)
2 CALL OPEN (2)
CALL VTSCA (3)
CALL VABS (.TRUE.)
CALL VMOVE (XYMOV)
CALL VABS (.FALSE.)
CALL VMOVE (XYAX(10))
CALL VTEXT (IAXIS(IV),1)
CALL VABS (.TRUE.)
CALL VMOVE (XYMOV)
CALL VABS (.FALSE.)
CALL VMOVE (XYMOV(9))
CALL VMOVE (XYAX(3))
CALL VTEXT (IAXIS(IH),1)
CALL POST (2)
IF (IH.NE.1 AND IV.NE.2) GO TO 13
9 CONTINUE
CALL EDNEW (ISEC,IPOS,2)
CALL EDPUT (TEXT1,35)

```

C
C
C

```
CALL EDGET (IR,2,LEN)
10 IF (IR(1) EQ IAXIS(3)) IH=3
   IF (IR(1) EQ IAXIS(4)) IH=4
   IF (IR(1) EQ IAXIS(5)) IH=5
   IF (IH LE 2) GO TO 9
   CALL EDPUT (TEXT2,35)
11 CALL EDGET (IS,2,LEN)
   IF (IS(1) EQ IAXIS(3)) IV=3
   IF (IS(1) EQ IAXIS(4)) IV=4
   IF (IS(1) EQ IAXIS(5)) IV=5
   IF (IV LE 2) GO TO 11
   IF (IR(1) NE IS(1)) GO TO 12
   CALL EDPUT (TEXT3,27)
   GO TO 9
12 CALL UNPOST (2)
   GO TO 2
13 CALL FIX (1)
   CALL FIX (2)
   IF (IH NE 5 AND IV NE 5) GO TO 14
   IF (IH NE 4 AND IV NE 4) GO TO 15
   IT=3
   CALL EDPUT ('ENTER CONSTANT ',15)
   CALL EDPUT (IAXIS(IT),1)
   CALL EDPUT (CR,1)
   CALL ASKR4 (CONST)
   GO TO 16
14 IT=5
   CALL EDPUT ('ENTER CONSTANT ',15)
   CALL EDPUT (IAXIS(IT),1)
   CALL EDPUT (CR,1)
   CALL ASKR4 (CONST)
   GO TO 16
15 IT=4
   CALL EDPUT ('ENTER CONSTANT ',15)
   CALL EDPUT (IAXIS(IT),1)
   CALL EDPUT (CR,1)
   CALL ASKR4 (CONST)
16 CALL EDCLO
   RETURN
   END
```

```

SUBROUTINE SETUP
IMPLICIT INTEGER*2 (I-N,#)
INTEGER*1 TEXT1(14),TEXT2(14),IR(2),IS(2),CR,Y,ISKW,ILINE,
&NAMFIL(15),IDO,MORD(25),FILE,P
INTEGER*4 LENGTH,IYMAX4
COMMON USKET (2,3),IR,IS,ISEG(3),IPOS(6),ISKW,CONST,IT,IPOINT(4),
&VIEW(4),FILE
COMMON /MENU/IXY(2),IDIAG(4),NT,IDO,IM,PTXBUF(25),PTYBUF(25),
&MORD,ILINE,NAMFIL,IYLOC(16),IXMIN,IXMAX,IYMIN,IYMAX,WDIAG(4)
DIMENSION SKEWC(6),ICROSS(6),WCOORD(2),IMENU(2)
LOGICAL STAT(3)
EXTERNAL EDGET
DATA CR/X'00'//,Y/'Y'//,ICROSS/400,400,-800,-800,800,0/,P/'P'/
DATA TEXT1/'ENTER MINIMUM '/
DATA TEXT2/'ENTER MAXIMUM '/

```

```

CREATE SEGMENT FOR DISPLAY MENU

```

```

:MENU(1)=25000
:MENU(2)=20000
CALL OPEN (99)
DO 10 MM=1,16
:MENU(2)=:MENU(2)-1500
CALL VMOVE (:MENU)
GO TO (25,24,23,22,21,20,19,18,17,16,15,10,13,10,11,10),MM
11 CALL VTEXT ('EXIT',4)
GO TO 10
13 CALL VTEXT ('TABLET',6)
GO TO 10
15 CALL VTEXT ('REDRAW',6)
GO TO 10
16 CALL VTEXT ('CIRCLE',6)
GO TO 10
17 CALL VTEXT ('SPLINE',6)
GO TO 10
18 CALL VTEXT ('DASH',4)
GO TO 10
19 CALL VTEXT ('LINE',4)
GO TO 10
20 CALL VTEXT ('REMOVE',6)
GO TO 10
21 CALL VTEXT ('LOCATE',6)
GO TO 10
22 CALL VTEXT ('BACKUP',6)
GO TO 10
23 CALL VTEXT ('CONTINUE',8)
GO TO 10
24 CALL VTEXT ('REJECT',6)
GO TO 10
25 CALL VTEXT ('ACCEPT',6)
10 :MENU(2)=:MENU(2)-500
CALL CLOSE

```

```

SET INITIAL VIEWPORT
:VIEW(1)--24575
:VIEW(2)--24575

```

C
C
C

C
C

```

IVIEW(3)-24575
IVIEW(4)-24575
CALL VIEW (IVIEW)
C
C   SET PORT VALUES EQUAL FOR TBØ AND DC
IPORT(1)--32767
IPORT(2)--24575
IPORT(3)-32767
IPORT(4)-24575
C
   IF (FILE NE P) GO TO 3Ø
   READ (3,*) (WDIAG(NN),NN-1,4)
   GO TO 31
C
C   ENTER DIAGONAL FOR WINDOW
C
3Ø CALL EDNEW (ISEG,IPOS,2)
   CALL EDPUT (TEXT1,14)
   CALL EDPUT (IR,2)
   CALL ASKR4 (WDIAG(1))
   CALL EDPUT (TEXT1,14)
   CALL EDPUT (IS,2)
   CALL ASKR4 (WDIAG(2))
   CALL EDPUT (TEXT2,14)
   CALL EDPUT (IR,2)
   CALL ASKR4 (WDIAG(3))
   CALL EDPUT (TEXT2,14)
   CALL EDPUT (IS,2)
   CALL ASKR4 (WDIAG(4))
   WRITE (3,*) (WDIAG(NN),NN-1,4)
   CALL EDCLO
31 CALL WIND (WDIAG)
   CALL VAR
   CALL QVIEW (IVIEW)
C
   CALL GILOC (2)
   CALL PORT (IPORT)
C
   CALL EDNEW (ISEG,IPOS,2)
   CALL EDPUT ('DO YOU WANT SKEW CORRECTION?',28)
   CALL EDPUT (CR,1)
   CALL EDPUT ('TYPE "Y" FOR YES OR "N" FOR NO ',32)
   CALL ASKAN (ISKW,1)
   IF (ISKW NE Y) GO TO 3
   CALL EDCLO
   CALL EDNEW (ISEG,IPOS,3)
   CALL EDPUT ('ENTER LOWER LEFT CORNER OF PICTURE ',35)
   CALL EDPUT (CR,1)
   CALL EDPUT ('ENTER LOWER RIGHT CORNER OF PICTURE ',36)
   CALL EDPUT (CR,1)
   CALL EDPUT ('ENTER UPPER RIGHT CORNER OF PICTURE ',36)
   CALL QGILOC (IXY,STAT)
   CALL P2W (IXY,SKEWC(1))
   CALL QGILOC (IXY,STAT)
   CALL P2W (IXY,SKEWC(5))
   CALL QGILOC (IXY,STAT)
   CALL P2W (IXY,SKEWC(3))

```



```

CALL UNSKEW (USKET,SKEVC,VDIAC)
3 CALL EDCLD

C
C
CREATE CROSS FOR GIN CURSOR

4 CALL OPEN (100)
  IXY(1)=0
  IXY(2)=0
5 CALL VMOVE (IXY)
6 CALL VABS ( F. )
  CALL VABS ( F. )
  IXY(1)=8000
  CALL VMOVE (IXY)
  IXY(1)--16000
  CALL VDRAW (IXY)
  IXY(1)=8000
  IXY(2)=8000
  CALL VMOVE (IXY)
  IXY(1)=0
  IXY(2)--15000
  CALL VDRAW (IXY)
  CALL CLOSE
  CALL VABS ( T )
  CALL VABS ( T )

C
C
CREATE BOX CURSOR FOR DISPLAY MENU

  IXY(1)=24000
  IXY(2)=0
  CALL OPEN (98)
  CALL VMOVE (IXY)
  CALL VABS ( F. )
  IXY(1)=7100
  IXY(2)=1000
  CALL VRECT (IXY)
  CALL CLOSE
  CALL VABS ( T. )

C
C
CREATE SEGMENT FOR X USED IN REMOVE
CALL OPEN (5)
CALL VABS ( F. )
CALL VMOVE (ICROSS(1))
CALL VDRAW (ICROSS(3))
CALL VMOVE (ICROSS(5))
CALL VDRAW (ICROSS(4))
CALL VABS ( T. )
CALL CLOSE

C
C
SET MAXIMUM AND MINIMUM FOR MENU LOCATION

CALL BELL
CALL EDNEV (ISEG,IPOS,2)
CALL EDPUT ('ENTER LOWER LEFT CORNER OF MENU',31)
CALL EDPUT (CR,1)
CALL EDPUT ('ENTER UPPER RIGHT CORNER OF MENU',32)
CALL QCILOC (IDIAC(1),STAT)
CALL QCILOC (IDIAC(3),STAT)

```

```
CALL EDCL0
IF (IDIAC(1) GT. IDIAC(3)) GO TO 7
IXMIN=IDIAC(1)
IYMIN=IDIAC(2)
IXMAX=IDIAC(3)
IYMAX=IDIAC(4)
GO TO 8
7 IXMIN=IDIAC(3)
IYMIN=IDIAC(4)
IXMAX=IDIAC(1)
IYMAX=IDIAC(2)
8 IYLOC(1)=IYMIN
IYMAX4=IYMAX
LENGTH=IYMAX4-IYMIN
IBLOCK=LENGTH/15
DO 9 J=2,15
9 IYLOC(J)=IYLOC(J-1)+IBLOCK
IYLOC(16)=IYMAX
```

C
C
C

GET RID OF SYSTEM CURSOR

```
CALL OPEN (90)
IXY(1)=32767
IXY(2)=32767
CALL VMOVE (IXY)
CALL CLOSE
CALL GICUR (90)
```

C

RETURN
END

```

SUBROUTINE ITAB
IMPLICIT INTEGER*2 (I-N,*)
INTEGER*1 IDO,MORD(25),IR(2),IS(2),ISKW,ILINE,NAMFIL(15),CR,Y
INTEGER*1 FILE
COMMON USKET (2,3),IR,IS,ISEC(3),IPOS(6),ISKW,CONST,IT,IPOINT(4),
&VIEW(4),FILE
COMMON /MENU/IXY(2),IDIAG(4),NT,IDO,IM,PTXBUF(25),PTYBUF(25),
&MORD,ILINE,NAMFIL,IYLOC(16),IXMIN,IXMAX,IYMIN,IYMAX,WDIAG(4)
DIMENSION PT(2),WCOORD(2),IVXY(2),ISXY(2),IEXT(3)
LOGICAL ISTAT(3)
EXTERNAL EDGET
DATA CR/X'0D'//,Y/'Y'//,ISXY/0,0/

```

C
C

```

ENABLE CO*OUT
ASSIGN 99 TO ICOUT
CALL ENABLE ('CO*OUT ',ICOUT)

```

C

```

6 CALL BELL
CALL EDNEW (ISEC,IPOS,1)
CALL EDPUT ('YOU ARE NOW READY TO DIGITIZE YOUR PICTURE.',43)
CALL POST (100)

```

C

```

ILINE=0
IM=0
5 CALL DELETE (20)
DO 15 M=1,25
NT=M
IBLK=0
10 CALL CICLO
4 CALL GIACT (IXY,ISTAT)
IF (IXY(1) .LT. IXMIN .OR. IXY(1) .GT. IXMAX) GO TO 7
IF (.NOT. ISTAT(3)) GO TO 4
IF (IXY(2) .LT. IYLOC(1) .OR. IXY(2) .GT. IYLOC(16)) GO TO 50
DO 11 IDO=1,15
IF (IXY(2) .GT. IYLOC(IDO).AND. IXY(2) .LT. IYLOC(IDO+1)) GO TO 12
11 CONTINUE
12 CALL MENU
IF (IDO .EQ. 1) GO TO 16
IF (IDO .EQ. 7) GO TO 5
IF (IDO .EQ. 15) GO TO 16
IF (IDO .EQ. 14) GO TO 5
IF (IDO .EQ. 13) IBLK=13
IF (IDO .EQ. 3) RETURN
GO TO 10
50 CALL EDPUT (CR,1)
CALL EDPUT ('ERROR IN MENU SELECTION TRY AGAIN!',36)
CALL BELL
GO TO 10
7 CALL P2W (IXY,WCOORD)
IF (ISKW .EQ. Y) GO TO 13
PT(1)=WCOORD(1)
PT(2)=WCOORD(2)
GO TO 14
13 CALL APPLY (USKET,WCOORD,PT)
14 CALL W2V (PT,IVXY)
ISXY(1)=IVXY(1)

```

```
ISXY(2)-IVXY(2)
CALL VPLACE (100,IVXY)
IF (.NOT. ISTAT(3)) GO TO 4
IF (IBLK .EQ. 13) CALL DELETE (4)
CALL APPEND (20)
CALL TAC (NT)
IF (ILINE .EQ. 0) CALL LINE
IF (ILINE .EQ. 1) CALL DASH
IF (IM .EQ. 0) CALL WMOVE (PT)
IF (IM .NE. 0) CALL WDRAW (PT)
CALL POST (20)
PTXBUF(NT)-PT(1)
PTYBUF(NT)-PT(2)
MORD(NT)-IM
IF (ILINE .EQ. 0) IM-1
IF (ILINE .EQ. 1) IM-2
IF (NT .EQ. M) GO TO 15
NT-NT+1
GO TO 10
15 CONTINUE
C EMPTY BUFFERS TO OUTPUT FILE
15 IF (NT .EQ. 0) GO TO 17
WRITE (3,100) (PTXBUF(K),PTYBUF(K),MORD(K),K-1,NT)
100 FORMAT (2F16.6,15)
CALL FIX (20)
CALL CKFLST (TEXT)
17 IF (!DO .EQ. 1) GO TO 20
IF (M .EQ. 25 .OR. !DO .EQ. 15) GO TO 18
GO TO 5
18 CALL OPEN (19)
CALL MOVE (PTXBUF(NT),PTYBUF(NT))
CALL POST (19)
GO TO 5
20 CALL BFCLO (3)
RETURN
C
C GO*OUT
99 CALL VPLACE (100,ISXY)
98 IF (.NOT. ISTAT(3)) GO TO 4
GO TO 10
C
END
```

```

SUBROUTINE MENU
IMPLICIT INTEGER*2 (I-N,P)
INTEGER*1 100,MORD(25),IR(2),IS(2),ISKW,ILINE,NAMFIL(15),CR,Y
INTEGER*1 FILE
COMMON USKET (2,3),IR,IS,ISEC(3),IPOS(6),ISKW,CONST,IT,IPOINT(4),
&IVIEW(4),FILE
COMMON /MENU/IXY(2),IDIAG(4),NT,IDO,IM,PTXBUF(25),PTYBUF(25),
&MORD,ILINE,NAMFIL,IYLOC(16),IXMIN,IXMAX,IYMIN,IYMAX,WDIAG(4)
DIMENSION WCOORD(2),PT(2)
LOGICAL STAT(3)
DATA CR/13/,Y/'Y'/

```

C

```

GO TO (31,32,33,32,35,36,37,38,39,40,41,42,43,44,45),IDO
31 CALL EDPUT (CR,1)
CALL EDPUT ('EXIT',4)
NT=NT-1
RETURN
32 CALL EDPUT (CR,1)
CALL EDPUT ('BLANK',5)
CALL BELL
RETURN
33 CALL EDPUT (CR,1)
CALL EDPUT ('GO TO JOY',9)
LU=5
CALL GILOC (LU)
CALL PORT (IPOINT)
CALL GICUR (9)
CALL EDCLO
RETURN
35 CALL EDPUT (CR,1)
CALL EDPUT ('REDRAW',6)
CALL ERASE
CALL BFCLO (3)
CALL BFACT (3,NAMFIL,.T.)
REWIND 3
READ (3,*) (WDIAG(NN),NN=1,4)
351 READ (3,100,END=359) X1,Y1,N1
100 FORMAT (2F16.6,IS)
IF (N1 .EQ. 10) GO TO 351
IF (N1 .EQ. 0) CALL MOVE (X1,Y1)
IF (N1 .EQ. 1) CALL LINE
IF (N1 .EQ. 2) CALL DASH
IF (N1 .EQ. 1 .OR. N1 .EQ. 2) CALL DRAW (X1,Y1)
GO TO 351
359 RETURN
36 CALL EDPUT (CR,1)
CALL EDPUT ('CIRCLE',8)
CALL OVLINK ('DRVCIR')
GO TO 41
37 CALL EDPUT (CR,1)
CALL EDPUT ('SPLINE',8)
CALL OVLINK ('DRWSPL')
GO TO 41
39 CALL EDPUT (CR,1)
CALL EDPUT ('DASH',4)
ILINE=1

```

```
RETURN
39 CALL EDPUT (CR,1)
   CALL EDPUT ('LINE',4)
   ILINE=0
   RETURN
40 CALL EDPUT (CR,1)
   CALL EDPUT ('REMOVE',6)
   CALL OVLINK ('REMOVE')
   GO TO 35
41 CALL EDPUT (CR,1)
   CALL EDPUT ('LOCATER MODE',12)
   IM=0
   RETURN
42 CALL EDPUT (CR,1)
   CALL EDPUT ('BACKUP',6)
   NT=NT-1
   CALL TRUNC (20,NT)
   CALL POST (20)
   CALL MOVE (PTXBUF(NT-1),PTYBUF(NT-1))
   RETURN
43 CALL EDPUT (CR,1)
   CALL EDPUT ('CONTINUE',8)
   CALL OPEN (4)
   CALL MOVE (PTXBUF(NT-2),PTYBUF(NT-2))
   CALL DRAW (PTXBUF(NT-1),PTYBUF(NT-1))
   CALL SYMBOL (5)
   CALL POST (4)
   CALL BINTAT (4,3)
   CALL BLINK (4,30,30)
   RETURN
44 CALL EDPUT (CR,1)
   CALL EDPUT ('REJECT',6)
   CALL DELETE (20)
   IM=0
   RETURN
45 CALL EDPUT (CR,1)
   CALL EDPUT ('ACCEPT',6)
   NT=NT-1
   RETURN
END
```



```

PT(1)-PTC(1)
PT(2)-PTC(2)
GO TO 362
364 CALL APPLY (USKET,PTC,PT)
362 CALL W2V (PT,IXY)
ISXY(1)=IXY(1)
ISXY(2)=IXY(2)
CALL VPLACE (100,IXY)
IF (NOT STAT(3)) GO TO 368
IF (IXY(1) GT. IXMIN) RETURN
CALL APPEND (35)
CALL WDRAW (PT)
CALL POST (35)
RAD=SQRT((PT(1)-CENTER(1))**2+(PT(2)-CENTER(2))**2)
C
ASSIGN 328 TO IFULL
CALL ENABLE ('*FULL ',IFULL)
C
CALL QCIR (CURVE,50,NUM,CENTER,RAD)
KK=2*NUM
CALL OPEN (36)
CALL WMOVE (CURVE)
IF (ILINE .EQ. 0) GO TO 368
IM=2
CALL DASH
GO TO 369
368 IM=1
CALL LINE
369 DO 365 KC=3, KK, 2
365 CALL WDRAW (CURVE(KC))
CALL POST (36)
CALL EDPUT (CR,1)
CALL EDPUT ('DO YOU WANT TO KEEP CIRCLE? ',29)
CALL BELL
CALL ASKAN (KCIR,1)
IF (KCIR .NE. Y) GO TO 359
CALL FIX (36)
CALL DELETE (35)
IMM=0
WRITE (3,100) CURVE(1),CURVE(2),IMM
100 FORMAT (2F16.6,15)
WRITE (3,100) (CURVE(KC),CURVE(KC+1),IM,KC-3, KK, 2)
RETURN
370 IF (STAT(3)) GO TO 371
IF (IRET .EQ. 1) GO TO 360
IF (IRET .EQ. 2) GO TO 366
371 CALL DELETE (35)
CALL DELETE (36)
RETURN
C
C
COSOUT HANDLER
367 CALL VPLACE (100,ISXY)
IF (IRET .EQ. 1) GO TO 360
GO TO 368
C
328 *FULL HANDLER
WRITE (1,*) ' NOT ENOUGH SPACE TO RETURN POINTS.'
WRITE (1,*) ' NUMBER OF POINTS REQUIRED - ',NUM

```


PAUSE
END

```

SUBROUTINE DRWSPL
  IMPLICIT INTEGER*2 (I-N,*)
  INTEGER*1 KSPL,FILE
  INTEGER*1 IDO,MORD(25),IR(2),IS(2),ISKW,ILINE,NAHFIL(15),CR,Y
  COMMON USKET (2,3),IR,IS,ISEG(3),IPOS(8),ISKW,CONST,IT,IPORT(4),
&IVIEW(4),FILE
  COMMON /MENU/IXY(2),IDIAG(4),NT,IDO,IM,PTXBUF(25),PTYBUF(25),
&MORD,ILINE,NAHFIL,IYLOC(16),IXMIN,IXMAX,IYMIN,IYMAX,WDIAG(4)
  DIMENSION WCOOR(2),PT(2),XBUF(20),YBUF(20),XSP(162),YSP(162),
&IVXY(2),ISXY(2)
  LOGICAL STAT(3)
  DATA CR/13/,Y/'Y'/

C
C   ENABLE GO*OUT HANDLER
  ASSIGN 326 TO ICOUT
  CALL ENABLE ('GO*OUT ',ICOUT)

C
C   EMPTY BUFFERS TO OUTPUT FILE
  IF (NT .EQ. 1) GO TO 319
  KK=NT-1
  WRITE (3,100) (PTXBUF(K),PTYBUF(K),MORD(K),K-1,KK)
100  FORMAT (2F16.6,15)
  CALL FIX (20)
  CALL DELETE (20)
319  CALL DELETE (20)
  CALL EDPUT (CR,1)
  CALL EDPUT ('YOU MUST PICK 3 POINTS BUT NO MORE THAN 20.',43)
  CALL BELL
  LS=1
  I20=0
  CALL OPEN (25)
  CALL PLOT
  CALL PINT (25)
  CALL CLOSE
  DO 325 L=1,20
3200 CALL CICLO
320  CALL CIAC (IXY,STAT)
  IF (IXY(1) .GT. IXMIN .AND. IXY(1) .LT. IXMAX) GO TO 322
  GO TO 321
322  IF (.NOT. STAT(3)) GO TO 320
  IF (IXY(2) .LT. IYLOC(7) .OR. IXY(2) .GT. IYLOC(8)) RETURN
  IF (L .GT. 3) GO TO 327
  CALL EDPUT (CR,1)
  CALL EDPUT ('YOU MUST INPUT AT LEAST 3 POINTS.',33)
  CALL BELL
  GO TO 320
321  CALL P2W (IXY,WCOOR)
  IF (ISKW .EQ. Y) GO TO 324
  PT(1)=WCOOR(1)
  PT(2)=WCOOR(2)
  GO TO 3250
324  CALL APPLY (USKET,WCOOR,PT)
3250 CALL H2V (PT,IVXY)
  ISXY(1)=IVXY(1)
  ISXY(2)=IVXY(2)
  CALL VPLACE (100,IVXY)

```

```

IF ( NOT .STAT(3)) GO TO 320
XBUF(LS)-PT(1)
YBUF(LS)-PT(2)
CALL APPEND (25)
CALL DRAW (XBUF(LS),YBUF(LS))
CALL POST (25)
325 LS=LS+1
I20=20
CALL EDPUT (CR,1)
CALL EDPUT ('YOU HAVE INPUT 20 POINTS WHICH IS THE MAX. ',42)
CALL BELL
327 IF (L .EQ. 20 AND I20 .EQ. 20) GO TO 323
L=L-1
323 IF (ILINE .EQ. 0) CALL LINE
CALL ECSPL (XBUF,YBUF,L,XSP,YSP)
IM=0
KK=8*L+2
CALL OPEN (26)
CALL MOVE (XSP(1),YSP(1))
IF (ILINE .EQ. 0) GO TO 13
IM=2
CALL DASH
GO TO 14
13 IM=1
CALL LINE
14 DO 329 II=2,KK
329 CALL DRAW (XSP(II),YSP(II))
CALL CLOSE
CALL POST (26)
CALL EDPUT (CR,1)
CALL EDPUT ('DO YOU WANT TO KEEP SPLINE? ',28)
CALL BELL
CALL ASKAN (KSPL,1)
CALL DELETE (25)
IF (KSPL .NE. Y) GO TO 319
CALL FIX (26)
IMOV=0
WRITE (3,100) XSP(1),YSP(1),IMOV
WRITE (3,100) (XSP(IK),YSP(IK),IM,IK-2,KK)
RETURN
C
C GO*OUT HANDLER
326 CALL VPLACE (100,ISXY)
IF (.STAT(3)) GO TO 3200
GO TO 320
C
END

```

```

SUBROUTINE EGSPL (X,Y,NP,XX,YY)
  IMPLICIT INTEGER*2 (I-N,*)
  DIMENSION X(20),Y(20),SP(20)
  DIMENSION A(20),B(20),C(20),D(20),TX(20),TY(20)
  DIMENSION XX(162),YY(162)
  SP(1)=0
  DO 115 I=2,NP
    SP(I)=SP(I-1)+SQRT((X(I)-X(I-1))**2+(Y(I)-Y(I-1))**2)
115 CONTINUE
  NIPT=8*NP+1
  DDX=SP(NP)/FLOAT(NIPT)
  V1=0
  V2=0
  *****
  C SINCE I1+I2=2 AND V1+V2=0 , INTERPOLATIONS TO THE LEFT OF X(I),Y(I)
  C AND TO RIGHT OF X(NP),Y(NP) WILL CONTINUE AT SAME SLOPES.
  C
  CALL SPLINE(SP,X,NP,A,B,C,D,TX,V1,V2)
  CALL SPLINE(SP,Y,NP,A,B,C,D,TY,V1,V2)
  T=0
  CALL SPLINT(SP,X,NP,TX,0,T,XX(1))
  CALL SPLINT(SP,Y,NP,TY,0,T,YY(1))
  DO 130 I = 1,NIPT
    T=T+DDX
  CALL SPLINT(SP,X,NP,TX,0,T,XX(I+1))
  CALL SPLINT(SP,Y,NP,TY,0,T,YY(I+1))
130 CONTINUE
  RETURN
  END

```

```

SUBROUTINE SPLINE(X,Y,N,A,B,C,D,T,V1,V2)
C
C THIS ROUTINE COMPUTES THE VECTOR OF SECOND DERIVATIVES
C IN ARRAY T WHICH ARE NEEDED TO SPLINE INTERPOLATE
C TABULAR DATA STORED IN X AND Y. ALLOWED END
C CONDITIONS INCLUDE SPECIFICATION OF Y' OR Y'' AT
C EACH END. WHEN I1 IS 1 OR 2 THEN V1 IS THE CHOSEN
C VALUE OF Y' OR Y'', RESPECTIVELY, AT X=X(I1). I2 AND
C V2 HAVE A SIMILAR MEANING FOR THE RIGHT END AT X(N).
C
C IMPLICIT INTEGER*2 (I-N,*)
C DIMENSION X(20),Y(20)
C DIMENSION A(20),B(20),C(20),D(20),T(20)
C
C FORM THE TRIDIAGONAL SYSTEM DEFINING THE SPLINE
C COEFFICIENTS
C
C NI=N-1
C DO 10 J=2,N1
C HJ=X(J)-X(J-1)
C HJ1=X(J+1)-X(J)
C HJP=HJ+HJ1
C A(J)=HJ/HJP
C B(J)=2
C C(J)=1 -A(J)
10 D(J)=6 *((Y(J+1)-Y(J))/HJ1-(Y(J)-Y(J-1))/HJ)/HJP
C
C FORM THE EQUATIONS FOR THE END CONDITIONS
C
C B(1)=1
C C(1)=0
C D(1)=V1
C A(N)=0
C B(N)=1
C D(N)=V2
C
C SOLVE FOR T(1), ..., T(N) USING GAUSS REDUCTION
C
C DO 60 K=1,N1
C R=A(K+1)/B(K)
C B(K+1)=B(K+1)-R*C(K)
60 D(K+1)=D(K+1)-R*D(K)
C T(N)=D(N)/B(N)
C DO 70 J=1,N1
C K=N-J
70 T(K)=(D(K)-C(K)*T(K+1))/B(K)
C RETURN
C END

```

```

SUBROUTINE SPLINT(X,Y,N,T,IO,XT,FT)
C
C THIS ROUTINE PRODUCES A SPLINE INTERPOLATION VALUE FT
C CORRESPONDING TO AN X-POSITION XT. WHEN IO EQUALS
C 0,1,2 OR 3, THEN FT TAKES THE VALUE OF Y, Y', Y'', OR
C THE INTEGRAL OF Y*DX BETWEEN LIMITS OF X(1) AND X
C ORIGINAL DATA POINTS ARE STORED IN X(I), Y(I), I=1,
C N. THE VECTOR T CONTAINS THE Y'' VALUES RETURNED
C FROM SUBROUTINE SPLCOF
C
C IMPLICIT INTEGER*2 (I-N,#)
C DIMENSION X(20), Y(20), T(20)
C IO=IO+1
C
C CHECK WHETHER XT IS OUTSIDE THE ORIGINAL DATA RANGE
C
C IF ( XT.LT.X(1) ) GO TO 80
C IF ( XT.GT.X(N) ) GO TO 120
C
C DETERMINE THE FIRST DATA POINT TO THE RIGHT OF XT
C
C DO 30 J=2,N
C IF ( X(J).GE.XT ) GO TO 40
30 CONTINUE
40 J1=J-1
HJ=X(J)-X(J1)
HJ6=6.*HJ
GO TO (50,60,70,72),IO
50 FT=(T(J1)*(X(J)-XT)**3 + T(J)*(XT-X(J1))**3
& +6.*(Y(J)-Y(J1))-(T(J)-T(J1))*HJ**2)*(XT-X(J1))
& +6.*HJ*Y(J1)-T(J1)*HJ**3)/HJ6
RETURN
60 FT = 1-3 *T(J1)*(X(J) - XT)**2+3.*T(J)*(XT-X(J1))**2
& + 6.*(Y(J) - Y(J1)) - (T(J) - T(J1))*HJ**2)/HJ6
RETURN
70 FT=(T(J1)*(X(J)-XT) + T(J)*(XT-X(J1)))/HJ
RETURN
72 HX=XT-X(J1)
FT= 25*T(J1)*(HJ**4-(HJ-HX)**4)+ 25*T(J)*HX**4
& + (3.*(Y(J)-Y(J1))+ 5*(T(J1)-T(J))*HJ**2)*HX**2
& + (6.*Y(J1)*HJ-T(J1)*HJ**3)*HX
FT=FT/HJ6
IF (J.EQ.2) RETURN
DO 74 K=2,J1
HK=X(K)-X(K-1)
74 FT=FT+ 5*(Y(K-1)+Y(K))*HK-(T(K-1)+T(K))*HK**3/24
RETURN
80 H2 = X(2) - X(1)
C
C DETERMINE THE SLOPE AT THE LEFT END AND INTERPOLATE
C LINEARLY
C
C SLOPE = (Y(2)-Y(1))/H2 - (2.*T(1)+T(2))*H2/6.
C GO TO (90,100,110,115), IO

```

```

90 FT=Y(1) + SLOPE* (XT-X(1))
   RETURN
100 FT=SLOPE
   RETURN
110 FT=0
   RETURN
115 HX=XT-X(1)
    FT=Y(1)*HX+ 5*SLOPE*HX**2
   RETURN

```

C
C
C

DETERMINE THE SLOPE AT THE RIGHT END AND INTERPOLATE
LINEARLY

```

120 HN=X(N) - X(N-1)
    SLOPE = (Y(N)-Y(N-1))/HN + (T(N-1)+2.*T(N))*HN/6.
    GO TO (130,100,110,140), I01
130 FT=Y(N) + SLOPE*(XT-X(N))
   RETURN
140 HX=XT-X(N)
    FT=Y(N)*HX+ 5*SLOPE*HX**2
    DO 150 J=2,N
    HJ=X(J)-X(J-1)
150 FT=FT+ 5*(Y(J-1)+Y(J))*HJ-(T(J-1)+T(J))*HJ**3/24.
   RETURN
   END

```

```

SUBROUTINE REMOVE
IMPLICIT INTEGER*2 (I-N,#)
INTEGER*1 IDO,MORD(25),IR(2),IS(2),ISKW,ILINE,NAMFIL(15),CR,Y
INTEGER*1 FILE
COMMON USKET (2,3),IR,IS,ISEG(3),IPOS(6),ISKW,CONST,IT,IPOINT(4),
&IVIEW(4),FILE
COMMON /MENU/IXY(2),IDIAG(4),NT,IDO,IM,PTXBUF(25),PTYBUF(25),
&MORD,ILINE,NAMFIL,IYLOC(16),IXMIN,IXMAX,IYMIN,IYMAX,WDIAG(4)
DIMENSION WCOORD(2),PT(2),IVXY(2),ISXY(2),IEXT(3)
LOGICAL STAT(3)
DATA CR/13/,Y/'Y'/

```

C
C
C

```

ENABLE GO*OUT HANDLER
ASSIGN 419 TO ICOUT
CALL ENABLE ('GO*OUT ',ICOUT)

```

```

CALL BFCLO (3)
CALL BFACT (3,NAMFIL,.T.)
CALL CKFLST (IEXT)
CALL BFNEW (4,'TEMP.DAT ',IEXT(1))
CALL EDPUT (CR,1)
CALL EDPUT ('PICK A POINT ON THE LINE TO BE REMOVED',38)

```

```

408 REWIND 3
READ (3,*) (WDIAC(NN),NN=1,4)
REWIND 4
WRITE (4,*) (WDIAC(NN),NN=1,4)
409 CALL GICLO
410 CALL CIACT (IXY,STAT)
IF (IXY(1) .LT. IXMIN .OR. IXY(1) .GT. IXMAX) GO TO 4104
IF (IXY(2) .GT. IYLOC(10) .AND. IXY(2) .LT. IYLOC(11) .AND.
&STAT(3)) GO TO 4151
IF (IXY(1) .GT. IXMIN .AND. IXY(1) .LT. IXMAX .AND. STAT(3)) RETURN
4104 CALL P2W (IXY,WCOORD)
IF (ISKW .EQ. Y) GO TO 4105
PT(1)=WCOORD(1)
PT(2)=WCOORD(2)
GO TO 4106
4105 CALL APPLY (USKET,WCOORD,PT)
4106 CALL W2V (PT,IVXY)
CALL VPLACE (100,IVXY)
ISXY(1)=IVXY(1)
ISXY(2)=IVXY(2)
IF (.NOT. STAT(3)) GO TO 410
CALL EDPUT (CR,1)
CALL EDPUT ('THANK YOU',9)
411 READ (3,100,END=418) X1,Y1,N1
100 FORMAT (2F16.6,I5)
412 READ (3,100,END=418) X2,Y2,N2
IF (N2 .EQ. 0) GO TO 414
IF (PT(1) .GT. X1 .AND. PT(1) .LT. X2) GO TO 413
IF (PT(1) .LT. X1 .AND. PT(1) .GT. X2) GO TO 413
IF (ABS(PT(1)-X1) .LE. .02 .OR. ABS(PT(1)-X2) .LE. .02)
&GO TO 413
GO TO 414
413 IF (PT(2) .GT. Y1 .AND. PT(2) .LT. Y2) GO TO 415
IF (PT(2) .LT. Y1 .AND. PT(2) .GT. Y2) GO TO 415

```



```

        IF (ABS(P1(2)-Y1) .LE. .02 OR ABS(P1(2)-Y2) .LE. .02)
&GO TO 415
414 WRITE (4,100) X1,Y1,N1
      X1-X2
      Y1-Y2
      N1-N2
      GO TO 412
C      COMPUTE SLOPES AND COMPARE
415 XS=(Y2-Y1)/(X2-X1)
      XSP=(PT(2)-Y1)/(PT(1)-X1)
      IF (ABS(XS-XSP) .GT. .15) GO TO 414
      CALL OPEN (4)
      CALL BINT (3)
      CALL MOVE (X1,Y1)
      CALL SYMBOL (5)
      CALL MOVE (X1,Y1)
      CALL DRAW (X2,Y2)
      CALL SYMBOL (5)
      CALL POST (4)
      CALL BLINK (4,40,20)
      CALL EDPUT (CR,1)
      CALL EDPUT ('CONFIRM LINE BY CHOOSING REMOVE AGAIN',37)
      GO TO 409
4151 CALL UNPOST (4)
      IF (N1 .NE. 0) GO TO 416
      N1-10
      416 WRITE (4,100) X1,Y1,N1
      417 N2-0
4175 WRITE (4,100) X2,Y2,N2
      READ (3,100,END=4176) X2,Y2,N2
      GO TO 4175
4176 CALL BFCLO (4)
      CALL BFDEL (3,NAMFIL)
      CALL BFREN (4,'TEMP.DAT',NAMFIL)
      RETURN
418 CALL EDPUT (CR,1)
      CALL EDPUT ('COULD NOT FIND LINE, TRY AGAIN!',31)
      GO TO 408
C
C      GO:OUT HANDLER
419 CALL VPLACE (100,ISXY)
      IF (STAT(3)) GO TO 400
      GO TO 410
      END

```

```

SUBROUTINE IJOY
IMPLICIT INTEGER*2 (I-N,#)
INTEGER*1 IDO,MORD(25),IR(2),IS(2),ISKW,ILINE,NAMFIL(15),CR,Y
INTEGER*1 FILE
COMMON USKET (2,3),IR,IS,ISEC(3),IPOS(6),ISKW,CONST,IT,IPOINT(4),
&IVIEW(4),FILE
COMMON /MENU/IXY(2),IDIAC(4),NT,IDO,IM,PTXBUF(25),PTYBUF(25),
&MORD,ILINE,NAMFIL,IYLOC(16),IXMIN,IXMAX,IYMIN,IYMAX,WDIAC(4)
DIMENSION IJLOC(16),PT(2),WCOORD(2),IVXY(2),ISXY(2)
LOGICAL STAT(3)
DATA IJLOC/-10000,-8000,-6000,-4000,-2000,0,2000,4000,6000,
&8000,10000,12000,14000,16000,18000,20000/,CR/13/,Y/'Y'/

C
C   ENABLE GO*OUT HANDLER
C   ASSIGN 99 TO ICOUT
C   CALL ENABLE ('GO*OUT ',ICOUT)

C
C   POST MENU FOR DISPLAY SCREEN

C
C   CALL POST (99)

C
C   CALL BELL
C   CALL EDNEW (ISEC,IPOS,1)
C   CALL EDPUT ('YOU ARE NOW READY TO DICITIZE YOUR PICTURE.',43)

C
ILINE=0
IM=0
5 CALL DELETE (20)
DO 70 M=1,25
NT=M
I9LK=0
34 CALL GICLO
341 IMENU=0
CALL POST (100)
35 CALL CIACT (IXY,STAT)
CALL P2W (IXY,PT)
351 IF (IXY(1) .LT. IVIEW(3) .AND. IMENU .EQ. 0) GO TO 60
IF (IMENU .NE. 0) GO TO 36
IMENU=1
CALL UNPOST (100)
CALL POST (98)
36 IXY(1)=24500
CALL VPLACE (98,IXY)
IF (.NOT. STAT(3)) GO TO 35
CALL UNPOST (98)
IF (IXY(2) .LT. IJLOC(1) .OR. IXY(2) .GT. IJLOC(16)) GO TO 50
DO 40 IDO=1,15
IF (IXY(2) .GT. IJLOC(IDO) .AND. IXY(2) .LT. IJLOC(IDO+1)) GO TO 45
40 CONTINUE
45 CALL MENU
CALL GICLO
ISXY(1)=0
ISXY(2)=0
CALL CIPOS (ISXY)
IF (IDO .EQ. 1) GO TO 72
IF (IDO .EQ. 7) GO TO 5

```

```

IF (ID0 EQ 15) GO TO 72
IF (ID0 EQ 14) GO TO 5
IF (ID0 EQ 13) IBLK-13
IF (ID0 EQ 3) RETURN
GO TO 34
50 CALL EDPUT (CR,1)
CALL EDPUT ('ERROR IN MENU SELECTION. TRY AGAIN!',36)
CALL UNPOST (98)
CALL CICLO
CALL CIPOS (ISXY)
CALL BELL
GO TO 34
60 CALL W2V (PT,IVXY)
ISXY(1)-IVXY(1)
ISXY(2)-IVXY(2)
CALL VPLACE (100,IVXY)
IF (.NOT. STAT(3)) GO TO 35
IF (IBLK EQ 13) CALL DELETE (4)
CALL APPEND (20)
CALL TAG (NT)
IF (ILINE EQ 0) CALL LINE
IF (ILINE EQ 1) CALL DASH
IF (IM EQ 0) CALL WMOVE (PT)
IF (IM NE 0) CALL WDRAW (PT)
CALL POST (20)
PTXBUF(NT)-PT(1)
PTYBUF(NT)-PT(2)
MORD(NT)-IM
IF (ILINE EQ 0) IM-1
IF (ILINE EQ 1) IM-2
IF (INT EQ M) GO TO 70
NT-NT+1
GO TO 34
70 CONTINUE
C EMPTY BUFFERS TO OUTPUT FILE
72 IF (INT EQ 0) GO TO 74
WRITE (3,100) (PTXBUF(K),PTYBUF(K),MORD(K),K-1,NT)
100 FORMAT (2F10.6,IS)
CALL FIX (20)
CALL CKFLST (IEXT)
74 IF (ID0 EQ 1) GO TO 76
IF (IM EQ 25 OR ID0 EQ 15) GO TO 75
GO TO 5
75 CALL OPEN (19)
CALL MOVE (PTXBUF(NT),PTYBUF(NT))
CALL POST (19)
GO TO 5
76 CALL BFCLO (3)
RETURN
C
C GOOUT HANDLER
99 CALL VPLACE (100,ISXY)
IF (.NOT. STAT(3)) GO TO 35
GO TO 34
C
END

```

```

SUBROUTINE KENJOY
  IMPLICIT INTEGER*2 (I-N,#)
  INTEGER*1 IDO,MORD(25),IR(2),IS(2),ISKW,ILINE,NAMFIL(15),CR,Y
  INTEGER*1 FILE
  COMMON USKET (2,3),IR,IS,ISEC(3),IPOS(6),ISKW,CONST,IT,IPORT(4),
&VIEW(4),FILE
  COMMON /MENU/IXY(2),IDIAC(4),NT,IDO,IM,PTXBUF(25),PTYBUF(25),
&MORD,ILINE,NAMFIL,IYLOC(16),IXMIN,IXMAX,IYMIN,IYMAX,WDIAC(4)
  DIMENSION WCOORD(2),PT(2)
  LOGICAL STAT(3)
  DATA CR/13/,Y/'Y'/

```

```

C
GO TO (31,32,33,32,35,36,37,38,39,40,41,42,43,44,45),IDO
31 CALL EDPUT (CR,1)
   CALL EDPUT ('EXIT',4)
   NT=NT-1
   RETURN
32 CALL EDPUT (CR,1)
   CALL EDPUT ('BLANK',5)
   CALL BELL
   RETURN
33 CALL EDPUT (CR,1)
   CALL EDPUT ('RETURN TO TABLET',16)
   LU=2
   CALL GILOC (LU)
   CALL PORT (IPORT)
   CALL GICUR (90)
   RETURN
35 CALL EDPUT (CR,1)
   CALL EDPUT ('REDRAW',6)
   CALL ERASE
   CALL BFCL0 (3)
   CALL BFACT (3,NAMFIL,I)
   REWIND 3
   READ (3,*) (WDIAC(IN),NN=1,4)
351 READ (3,100,END=359) X1,Y1,N1
100 FORMAT (2F16.6,IS)
   IF (N1 EQ. 10) GO TO 351
   IF (N1 EQ. 0) CALL MOVE (X1,Y1)
   IF (N1 EQ. 1) CALL LINE
   IF (N1 EQ. 2) CALL DASH
   IF (N1 EQ. 1 OR N1 EQ. 2) CALL DRAW (X1,Y1)
   GO TO 351
359 RETURN
36 CALL EDPUT (CR,1)
   CALL EDPUT ('CIRCLE',6)
   CALL POST (100)
   CALL OVLINK ('DRVC1J')
   GO TO 41
37 CALL EDPUT (CR,1)
   CALL EDPUT ('SPLINE',6)
   CALL POST (100)
   CALL OVLINK ('DRVSPJ')
   GO TO 41
38 CALL EDPUT (CR,1)
   CALL EDPUT ('DASH',4)

```

```

ILINE-1
RETURN
39 CALL EDPUT (CR,1)
CALL EDPUT ('LINE',4)
ILINE-0
RETURN
40 CALL EDPUT (CR,1)
CALL EDPUT ('REMOVE',6)
CALL POST (100)
CALL OVLINK ('REMOVJ',1)
GO TO 35
41 CALL EDPUT (CR,1)
CALL EDPUT ('LOCATER MODE',12)
IM-0
RETURN
42 CALL EDPUT (CR,1)
CALL EDPUT ('BACKUP',5)
NT-NT-1
CALL TRUNC (20,NT)
CALL POST (20)
CALL MOVE (PTXBUF(NT-1),PTYBUF(NT-1))
RETURN
43 CALL EDPUT (CR,1)
CALL EDPUT ('CONTINUE',8)
CALL OPEN (4)
CALL MOVE (PTXBUF(NT-2),PTYBUF(NT-2))
CALL DRAW (PTXBUF(NT-1),PTYBUF(NT-1))
CALL SYMBOL (5)
CALL POST (4)
CALL BINTAT (4,3)
CALL BLINK (4,30,30)
RETURN
44 CALL EDPUT (CR,1)
CALL EDPUT ('REJECT',6)
CALL DELETE (20)
IM-0
RETURN
45 CALL EDPUT (CR,1)
CALL EDPUT ('ACCEPT',6)
NT-NT-1
RETURN
END

```

```

SUBROUTINE DRXC1J
IMPLICIT INTEGER*2 (I-N,#)
INTEGER*1 IDO,MORD(25),IR(2),IS(2),ISKW,ILINE,NAHFIL(15),CR,Y
INTEGER*1 FILE,RCIR
COMMON USKET (2,3),IR,IS,ISEC(3),IPOS(6),ISKW,CONST,IT,IPORT(4),
&IVIEW(4),FILE
COMMON /MENU/IXY(2),IDIAC(4),NT,IDO,IM,PTXBUF(25),PTYBUF(25),
&MORD,ILINE,NAHFIL,IYLOC(16),IXMIN,IXMAX,IYMIN,IYMAX,WDIAC(4)
DIMENSION WCOORD(2),PT(2),CURVE(100),CENTER(2),PTC(2),ISXY(2)
DIMENSION IEXT(3)
LOGICAL STAT(3)
DATA CR/13/,Y/'Y'/

```

```

C
C   ENABLE CO*OUT HANDLER
C   ASSIGN 367 TO IGOUT
C   CALL ENABLE ('CO*OUT ',IGOUT)
C
359 CALL DELETE (35)
   CALL DELETE (36)
   CALL EDPUT (CR,1)
   CALL EDPUT ('DIGITIZE CENTER OF CIRCLE ',26)
   IRET=1
   ISXY(1)=-0
   ISXY(2)=0
   CALL CICLO
   CALL CIPOS(1ISXY)
360 CALL CIACT (IXY,STAT)
   CALL P2W (IXY,CENTER)
   IF (IXY(1) .LT. IVIEW(3)) GO TO 363
361 CALL UNPOST (100)
   CALL POST (98)
   IXY(1)=24000
   CALL VPLACE (98,IXY)
   IF (.NOT. STAT(3)) GO TO 360
   CALL DELETE (35)
   RETURN
363 CALL W2V (CENTER,IXY)
   ISXY(1)=IXY(1)
   ISXY(2)=IXY(2)
   CALL VPLACE (100,IXY)
   IF (.NOT. STAT(3)) GO TO 360
   CALL OPEN (35)
   CALL PLOT
   CALL PINT (25)
   CALL WDRAW (CENTER)
   CALL POST (35)
   CALL EDPUT (CR,1)
   CALL EDPUT ('DIGITIZE POINT ON CIRCLE ',24)
   IRET=2
   CALL CICLO
368 CALL CIACT (IXY,STAT)
   CALL P2W (IXY,PT)
   IF (IXY(1) .GT. IVIEW(3)) GO TO 361
362 CALL W2V (PT,IXY)
   ISXY(1)=IXY(1)
   ISXY(2)=IXY(2)

```

```
CALL VPLACE (100,IXY)
IF (NOT STAT(3)) GO TO 366
CALL APPEND (35)
CALL WDRAW (PT)
CALL POST (35)
RAD=SQRT((PT(1)-CENTER(1))**2+(PT(2)-CENTER(2))**2)
```

```
C
ASSIGN 328 TO !FULL
CALL ENABLE ('*FULL ',!FULL)
```

```
C
CALL QCIR (CURVE,50,NUM,CENTER,RAD)
KK=2*NUM
CALL OPEN (36)
CALL WMOVE (CURVE)
IF (ILINE EQ 0) GO TO 368
IM=2
CALL DASH
GO TO 369
368 IM=1
CALL LINE
369 DO 365 KC=3,KK,2
365 CALL WDRAW (CURVE(KC))
CALL POST (35)
CALL EDPUT (CR,1)
CALL EDPUT ('DO YOU WANT TO KEEP CIRCLE? ',29)
CALL BELL
CALL ASKAN (KCIR,1)
IF (KCIR NE Y) GO TO 359
CALL FIX (36)
CALL DELETE (35)
IMM=0
WRITE (3,100) CURVE(1),CURVE(2),IMM
100 FORMAT (2F16.6,IS)
WRITE (3,100) (CURVE(KC),CURVE(KC+1)),IM,KC-3,KK,2)
CALL CKFLST (TEXT)
RETURN
```

```
C
GO$OUT HANDLER
C
367 CALL VPLACE (100,ISXY)
IF (IRET EQ 1) GO TO 360
GO TO 366
```

```
C
$FULL HANDLER
328 WRITE (1,*) ' NOT ENOUGH SPACE TO RETURN POINTS. '
WRITE (1,*) ' NUMBER OF POINTS REQUIRED = ',NUM
PAUSE
END
```

```

SUBROUTINE DRWSPJ
IMPLICIT INTEGER*2 (I-N,#)
INTEGER*1 KSPL,FILE
INTEGER*1 IDO,MORD(25),IR(2),IS(2),ISKW,ILINE,NAMFIL(15),CR,Y
COMMON USKET (2,3),IR,IS,ISEG(3),IPOS(6),ISKW,CONST,IT,IPOINT(4),
&IVIEW(4),FILE
COMMON /MENU/IXY(2),IDIAC(4),NT,IDO,IM,PTXBUF(25),PTYBUF(25),
&MORD,ILINE,NAMFIL,IYLOC(16),IXMIN,IXMAX,IYMIN,IYMAX,WDIAC(4)
DIMENSION WCOORD(2),PT(2),XSP(162),YSP(162),ISXY(2),IVXY(2),
&IJLOC(16)
LOGICAL STAT(3)
DATA CR/13/,Y/'Y'/,IJLOC/-10000,-8000,-6000,-4000,-2000,0,2000,
&4000,6000,8000,10000,12000,14000,16000,18000,20000/

```

C
C
C

```

ENABLE GO*OUT HANDLER
ASSIGN 326 TO ICOUT
CALL ENABLE ('GO*OUT ',ICOUT)

```

```

EMPTY BUFFERS TO OUTPUT FILE
IF (NT EQ 1) GO TO 319
KK=NT-1
WRITE (3,100) (PTXBUF(K),PTYBUF(K),MORD(K),K=1,KK)

```

```

100 FORMAT (2F16.6,15)
CALL FIX (20)
319 CALL DELETE (26)
CALL EDPUT (CR,1)
CALL EDPUT ('YOU MUST PICK 3 POINTS BUT NO MORE THAN 20.',43)
CALL BELL

```

```

LS=1
I20=0
ISXY(1)=0
ISXY(2)=0
CALL CICLO
CALL GIPOS (ISXY)
CALL OPEN (25)
CALL PLOT
CALL PINT (25)
CALL CLOSE
DO 325 L=1,20

```

```

3200 CALL CICLO
CALL POST (100)
320 CALL GIAC (IXY,STAT)
CALL P2W (IXY,PT)
3201 IF (IXY(1) .LT. IVIEW(3)) GO TO 321
CALL UNPOST (100)
CALL POST (98)
IXY(1)=24000
CALL VPLACE (98,IXY)
IF (.NOT. STAT(3)) GO TO 320
CALL UNPOST (98)
IF (IXY(2) .LT. IJLOC(7) .OR. IXY(2) .GT. IJLOC(8)) RETURN
IF (L .GT. 3) GO TO 327
CALL EDPUT (CR,1)
CALL EDPUT ('YOU MUST INPUT AT LEAST 3 POINTS.',33)
CALL BELL
GO TO 320

```



```

321 CALL W2V (PT,IVXY)
    ISXY(1)=IVXY(1)
    ISXY(2)=IVXY(2)
    CALL VPLACE (100,IVXY)
    IF ( NOT STAT(3) ) GO TO 320
    PTXBUF(LS)=PT(1)
    PTYBUF(LS)=PT(2)
    CALL APPEND (25)
    CALL DRAW (PTXBUF(LS),PTYBUF(LS))
    CALL POST (25)
325 LS=LS+1
    I20=20
    CALL EDPUT (CR,1)
    CALL EDPUT ('YOU HAVE INPUT 20 POINTS WHICH IS THE MAX. ',42)
    CALL BELL
327 IF (L .EQ. 20 .AND. I20 .EQ. 20) GO TO 323
    L=L-1
323 IF (ILINE .EQ. 0) CALL LINE
    CALL EGSP (PTXBUF,PTYBUF,L,XSP,YSP)
    IM=0
    KK=8*L+2
    CALL OPEN (26)
    CALL MOVE (XSP(1),YSP(1))
    IF (ILINE .EQ. 0) GO TO 13
    IM=2
    CALL DASH
    GO TO 14
13 IM=1
    CALL LINE
14 DO 329 I1=2, KK
329 CALL DRAW (XSP(I1),YSP(I1))
    CALL CLOSE
    CALL POST (26)
    CALL EDPUT (CR,1)
    CALL EDPUT ('DO YOU WANT TO KEEP SPLINE? ',28)
    CALL BELL
    CALL ASK (KSPL,1)
    CALL DELETE (25)
    IF (KSPL .NE. Y) GO TO 319
    CALL FIX (26)
    IMOV=0
    WRITE (3,100) XSP(1),YSP(2),IMOV
    WRITE (3,100) (XSP(IK),YSP(IK)),IM,IK-2, KK)
    RETURN
C
C
COSOUT HANDLER
326 CALL VPLACE (100,ISXY)
    IF (STAT(3)) GO TO 3200
    GO TO 320
END

```

```

SUBROUTINE REMDVJ
IMPLICIT INTEGER*2 (I-N,*)
INTEGER*1 IDO,MORD(25),IR(2),IS(2),ISKW,ILINE,NAMFIL(15),CR,Y
INTEGER*1 FILE
COMMON USKET (2,3),IR,IS,ISEG(3),IPOS(6),ISKW,CONST,IT,IPOINT(4),
&IVIEW(4),FILE
COMMON /MENU/IXY(2),IDIAC(4),NT,IDO,IM,PTXBUF(25),PTYBUF(25),
&MORD,ILINE,NAMFIL,IYLOC(16),IXMIN,IXMAX,IYMIN,IYMAX,WDIAC(4)
DIMENSION WCOORD(2),PT(2),IVXY(2),ISXY(2),IJLOC(16)
LOGICAL STAT(3)
DATA CR/13/,Y/'Y'/,IJLOC/-1000,-800,-600,-400,-200,0,200,
&400,600,800,1000,1200,1400,1600,1800,2000/,ISXY/0,0/
C
C   ENABLE GO*OUT HANDLER
C   ASSIGN 419 TO ICOUT
C   CALL ENABLE ('GO*OUT ',ICOUT)
C
CALL BFCLD (3)
CALL BFACT (3,NAMFIL,T.)
CALL BFNEW (4,'TEMP.DAT',300)
CALL EDPUT (CR,1)
CALL EDPUT ('PICK A POINT ON THE LINE TO BE REMOVED',38)
408 REWIND 3
READ (3,*) (WDIAC(NN),NN=1,4)
REWIND 4
WRITE (4,*) (WDIAC(NN),NN=1,4)
409 CALL GICLO
CALL GIPOS (ISXY)
410 CALL GIAC7 (IXY,STAT)
CALL P2W (IXY,PT)
4100 IF (IXY(1) .LT. IVIEW(3)) GO TO 4104
CALL UNPOST (100)
CALL POST (98)
IXY(1)=2400
CALL VPLACE (98,IXY)
IF (.NOT. STAT(3)) GO TO 410
IF (IXY(2) .GT. IJLOC(10) .AND. IXY(2) .LT. IJLOC(11)) GO TO 4151
CALL UNPOST (98)
RETURN
4104 CALL W2V (PT,IVXY)
CALL VPLACE (100,IVXY)
ISXY(1)=IVXY(1)
ISXY(2)=IVXY(2)
IF (.NOT. STAT(3)) GO TO 410
CALL EDPUT (CR,1)
CALL EDPUT ('THANK YOU',9)
411 READ (3,100,END=418) X1,Y1,N1
100 FORMAT (2F16.6,15)
412 READ (3,100,END=418) X2,Y2,N2
IF (N2 .EQ. 0) GO TO 414
IF (PT(1) .GE. X1 .AND. PT(1) .LE. X2) GO TO 413
IF (PT(1) .LE. X1 .AND. PT(1) .GE. X2) GO TO 413
IF (ABS(PT(1)-X1) .LE. .02 .OR. ABS(PT(1)-X2) .LE. .02)
&GO TO 413
GO TO 414
413 IF (PT(2) .GE. Y1 .AND. PT(2) .LE. Y2) GO TO 415

```

```

IF (PT(2) LE Y1 AND PT(2) GE Y2) GO TO 415
IF (ABS(PT(2)-Y1) LE .02 OR ABS(PT(2)-Y2) LE .02)
&GO TO 415
414 WRITE (4,100) X1,Y1,N1
X1-X2
Y1-Y2
N1-N2
GO TO 412
C COMPUTE SLOPES AND COMPARE
415 IF (ABS(X2-X1) LE .05 AND ABS(PT(1)-X1) LE .05) GO TO 4150
WRITE (1,*) 'REMOVJ ',X1,X2,PT(1),Y1,Y2,PT(2)
XS=(Y2-Y1)/(X2-X1)
XSP=(PT(2)-Y1)/(PT(1)-X1)
IF (ABS(XS-XSP) GT .15) GO TO 414
4150 CALL OPEN (4)
CALL BINT (3)
CALL MOVE (X1,Y1)
CALL SYMBOL (5)
CALL MOVE (X1,Y1)
CALL DRAW (X2,Y2)
CALL SYMBOL (5)
CALL POST (4)
CALL BLINK (4,40,20)
CALL EDPUT (CR,1)
CALL EDPUT ('CONFIRM LINE BY CHOOSING REMOVE AGAIN',37)
GO TO 409
4151 CALL UNPOST (4)
IF (N1 NE 0) GO TO 416
N1=0
416 WRITE (4,100) X1,Y1,N1
417 N2=0
4175 WRITE (4,100) X2,Y2,N2
READ (3,100,END=4176) X2,Y2,N2
GO TO 4175
4176 CALL BFCLO (4)
CALL BFDEL (3,NAMFIL)
CALL BFREN (4,'TEMP.DAT',NAMFIL)
RETURN
418 CALL EDPUT (CR,1)
CALL EDPUT ('COULD NOT FIND LINE, TRY AGAIN!',31)
GO TO 408
C
C GOOUT HANDLER
419 CALL VPLACE (100,15XY)
IF (STAT(3)) GO TO 400
GO TO 410
END

```

```

SUBROUTINE PDBPLT
IMPLICIT INTEGER*2(I-N,*)
INTEGER*1 OUTFIL(15),FNAME(13),EXT(3),WSPFIL(15),
&IPDB,IPLT,IY,CR
INTEGER*1 NAMFIL(15),FILE,IDO,MORD(25),IR(2),IS(2),ISKW,ILINE
DIMENSION IATTR(8)
COMMON USKET(2,3),IR,IS,ISEG(3),IPOS(8),ISKW,CONST,IT,IPOINT(4),
&IYIEW(4),FILE
COMMON /MENU/IXY(2),IDIAC(4),NT,IDO,IM,PTXBUF(25),PTYBUF(25),
&MORD,ILINE,NAMFIL,IYLOC(16),IXMIN,IXMAX,IYMIN,IYMAX,WDIAC(4)
COMMON/PDBCOM/LUFILE,MOUM(133)
DATA IATTR/0,0,0,256,256,0/,IY/'Y',CR/'00'/
DATA FNAME/13*' '/,WSPFIL/15*' '/,OUTFIL/15*' '/
CALL PINIT
LUFILE-4
CALL EDPUT (CR,1)
CALL EDPUT ('DO YOU WANT A PDB FILE? ',24)
CALL ASKAN (IPDB,1)
CALL EDPUT (CR,1)
CALL EDPUT ('DO YOU WANT A PLT FILE? ',24)
CALL ASKAN (IPLT,1)
IF (IPDB.NE.IY.AND.IPLT.NE.IY) RETURN
CALL FSPLIT (NAMFIL,FNAME,EXT)
IF (IPDB.NE.IY) GO TO 100
CALL EDPUT (CR,1)
CALL EDPUT ('YOUR PDB FILE IS BEING WRITTEN. ',31)
CALL FMERGE (OUTFIL,FNAME,'PDB')
CALL BFNEW(4,OUTFIL)
CALL FMERGE(WSPFIL,FNAME,'WSP')
CALL BFNEW(8,WSPFIL)
CALL HEADER(IATTR)
CALL BFACT(3,NAMFIL)
READ(3,*)IXMIN,YMIN,XMAX,YMAX
XEX=XMAX-XMIN
YEX=YMAX-YMIN
XSC-1
YSC-1
IF(XEX.GT.YEX)YSC=YEX/XEX
IF(YEX.GT.XEX)XSC=XEX/YEX
SXEX=XSC*49150
SYEX=YSC*49150
IF (XMAX+XMIN.LT. 001) GO TO 10
IWX1=(XMIN/XEX)*SXEX-SXEX/2
IWX2=(XMAX/XEX)*SXEX-SXEX/2
GO TO 11
10 IWX1--SXEX/2
IWX2-SXEX/2
11 IF (YMAX+YMIN.LT. 001) GO TO 12
IWY1=(YMIN/YEX)*SYEX-SYEX/2
IWY2=(YMAX/YEX)*SYEX-SYEX/2
GO TO 13
12 IWY1--SYEX/2
IWY2-SYEX/2
13 WRITE(8,1)IWX1,IWY1,IWX2,IWY2
IWX1--IWX2
IWY1--IWY2

```

```

WRITE(6,2)IWX1,IWY1,IWX2,IWY2
1 FORMAT('WIND ',4I7)
2 FORMAT('VIEW ',4I7)
CALL BFCLO(6)
20 READ(3,*,END=50)X,Y,IB
IF(IB GT 2)GO TO 20
IX-SXEX/XEX**
JY-SYEX/YEX**
IF(IB EQ 0)CALL MOVED(IX,JY)
IF(IB EQ 1)CALL DRAND(IX,JY)
IF(IB EQ 2)CALL DASHD(IX,JY)
GO TO 20
50 CALL PCLOSE
CALL BFCLO(3)
WRITE(1,*) 'PDB FILE WRITTEN'
100 IF (IPLT NE. IY) RETURN
CALL EDPUT (CR,1)
CALL EDPUT ('YOUR PLT FILE IS BEING WRITTEN.' ,31)
CALL FMERGE (OUTFIL,FNAME,'PLT')
CALL BFNEW (15,OUTFIL)
CALL CONEW (15)
CALL BUFF (T.)
CALL BFACT (3,NAMFIL)
READ(3,*) WDIAG
CALL VIEW (VIEW)
CALL WIND (WDIAG)
101 READ(3,*,END=120) X,Y,N
IF (N EQ. 1) CALL LINE
IF (N EQ. 2) CALL DASH
IF (N EQ. 0) CALL MOVE (X,Y)
IF (N NE. 0) CALL DRAW (X,Y)
GO TO 101
120 CALL COCLO
CALL BFCLO (15)
WRITE (1,*) 'PLT FILE WRITTEN'
RETURN
END

```

C *****MODIFIED FOR DGL FILE HANDLING*****
C-----SUBROUTINE--PCLOSE-----TEKTRONIX, INC-----
C

52

CCCCCCCCCCCC
SUBROUTINE PCLOSE

AUTHOR
John F. DiNapoli

USAGE
THIS ROUTINE INSURES THAT ALL FDM'S ARE OUTPUT TO THE FILE
BEFORE IT IS CLOSED.

CALLING SEQUENCE:

CALL PCLOSE

INPUT PARAMETERS:

NONE:

IMPLICIT INTEGER*2 (I-N,#)
INTEGER*2 OUTBUF
COMMON/PDSCOM/LUFILE,MAXIND,MODE,INDEX,OUTBUF(128),LENFDM,LENOPP
IF(MODE NE 0)CALL OPAIR(-1,IX,IY)
CALL BFCLO(LUFILE)
RETURN
END

1. Report No. NASA TM-81919		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle A User's Guide for DTIZE-- An Interactive Digitizing and Graphical Editing Computer Program				5. Report Date January 1981	
				6. Performing Organization Code	
7. Author(s) Carolyn C. Thomas				8. Performing Organization Report No.	
				10. Work Unit No. 540-02-13-01	
9. Performing Organization Name and Address NASA, Langley Research Center Hampton, Virginia 23665				11. Contract or Grant No.	
				13. Type of Report and Period Covered Technical Memorandum	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546				14. Sponsoring Agency Code	
15. Supplementary Notes					
16. Abstract <p>This report is a user's guide for DTIZE, a 2-dimensional digitizing program with graphical editing capability. DTIZE provides the capability to simultaneously create and display a picture on the display screen. Data descriptions may be permanently saved in three different formats. DTIZE creates the picture graphics in the locator mode, thus inputting one coordinate each time the terminator button is pushed. Graphic Input devices (GIN) are also used to select function command menu. These menu commands and the program's interactive prompting sequences provide a complete capability for creating, editing, and permanently recording a graphical picture file.</p> <p>DTIZE is written in Fortran IV language for the Tektronix 4081 graphics system utilizing the Plot 80 Distributed Graphics Library (DGL) subroutines. The Tektronix 4953/3954 Graphics Tablet with mouse, pen, or joystick are used as Graphics Input devices to create picture graphics. DTIZE runs under the control of the Tektronix 4081 Graphic Operating System (GOS) with extended GIN features (needed for GDL graphic input routines).</p>					
17. Key Words (Suggested by Author(s)) Tektronix 4081 Digital Graphics Refresh Graphics			18. Distribution Statement Unclassified - Unlimited Subject Category: 61		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages 52	22. Price* A04

