

AD-A102 795

NAVAL RESEARCH LAB WASHINGTON DC
GENERAL PURPOSE HIGH-RESOLUTION PLOTTING PACKAGE FOR TEKTRONIX --ETC(U)
AUG 81 C M RUSSIERO, R W ANDERSON
NRL-MR-4533

F/G 9/2

NL

UNCLASSIFIED

1 OF 1
AD
P000706

END
DATE
FILMED
9-81
DTIC

AD A102736

⑥ General Purpose High Resolution Storage
Fusing for Television 602 Filter and
Image CRT Terminals.

⑩ 100-100-100-100-100-100

Chemical Development Department
Naval Research Laboratory
P.O. Box 117
Orlando, FL 32816

⑫ 34

⑭ NRL-MR-4533

⑪ 37 AUG 1961

⑮ RR010F

⑯ RR010842



RECEIVED
S-1000

NAVAL RESEARCH LABORATORY
Washington, D.C.

Approved for public release; distribution unlimited

251 950

81 8 11 054

int

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NRL Memorandum Report 4533	2. GOVT ACCESSION NO. -1D-A102 795	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) GENERAL PURPOSE HIGH-RESOLUTION PLOTTING PACKAGE FOR TEKTRONIX 4662 PLOTTER AND COMPATIBLE CRT TERMINALS	5. TYPE OF REPORT & PERIOD COVERED Interim report on a continuing NRL problem.	
7. AUTHOR(s) C. M. Ruggiero and R. W. Anderson	6. PERFORMING ORG. REPORT NUMBER	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Underwater Sound Reference Detachment Naval Research Laboratory P.O. Box 8337, Orlando, FL 32856	8. CONTRACT OR GRANT NUMBER(s)	
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Direct Funding Systems Research & Technology Directorate Washington, DC 20375	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61153N-11; RR011-08-42; 0585-00	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	12. REPORT DATE August 17, 1981	
	13. NUMBER OF PAGES 33	
	15. SECURITY CLASS. (of this report) UNCLASSIFIED	
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Cathode ray tube (CRT) Tektronix 4662 High resolution Interactive plotting FORTRAN IV PLUS		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) An interactive plotting package to accommodate XY data in any format is presented. Automatic scaling, choice of line types, labels, and multiple curves on a single plot are some of the options provided to make this package versatile and easy to use.		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

CONTENTS

	<u>Page</u>
1. IDENTIFICATION	1
2. PURPOSE.	1
3. USAGE.	1
4. TEST METHOD AND RESULTS.	4
APPENDICES	
A. SOURCE LANGUAGE LISTING.	7
B. SUBROUTINES.	15
XYPLOT (On-Line Data Entry Module)	15
SCALE (Data Scale Module).	16
AXIS (Label and Axis Output Module).	16
PLREAD (Data Input Module)	18
PON, POFF (A4662 Plotter On/Off Module).	22
TKDASH (Line Plotter Module)	22
MINMAX (Data Boundary Module).	25
TPLOT (Point Plotter Module)	25
TERM (Plotter Termination Module).	26
A4662 (Label Size Adjustment Module)	27
C. COMPILATION AND TASKBUILDING	28
D. UEDIT (Unformatted File Text Editor)	30

GENERAL PURPOSE HIGH-RESOLUTION PLOTTING PACKAGE
FOR
TEKTRONIX 4662 PLOTTER AND COMPATIBLE CRT TERMINALS

1. IDENTIFICATION

1.1. Identification Name: PLOTTER

1.2. Subroutines: XYPLOT, SCALE, AXIS, PLREAD, PON, POFF, TKDASH, MINMAX,
TPLOT, TERM, A4662

1.3. Programming Language:

Language: FORTRAN 4-Plus, Version 3.0
Routine Type: Source Program and Subroutines
Operating System: RSX-11M, Version 3.2

1.4. Computer: Digital Equipment Corp. PDP-11/45

1.5. Program Availability:

Submittal: Program Descriptions and Listings
On File: USRD Source Library

2. PURPOSE

2.1. Description of the Routine: The program PLOTTER generates a 12-bit resolution plot of abscissa and ordinate values. This program can adapt to formatted or unformatted data contained in one or two files. PLOTTER also has the capability to allow the user to manually enter points for an XY plot. The program contains the following options:

- Automatic Scaling
- Point Plot
- Choice of Line (solid, dashed, dot-dash, small dashes)
- Create New Data File
- Labels
- Multiple Curves on One Graph

2.2. Program Background: The PLOTTER program was designed to provide USRD with an interactive graphics package that is both versatile and easy to use.

3. USAGE

3.1. Preparation of Terminal and Plotter: The preparation needed before running the program PLOTTER depends upon the instrument used; either the Tektronix CRT or the Tektronix 4662 plotter may be used for plotting. If a Tektronix CRT is used, one needs only to be logged on the computer and to type RUN PLOTTER.

Manuscript submitted June 3, 1981.

Since the program is self-explanatory (see Sec. 4.1.) there is no need for further instructions. However, if the Tektronix 4662 flat-bed plotter (Fig. 1) is used, the plotter must be prepared before running the program. This involves a few simple steps.

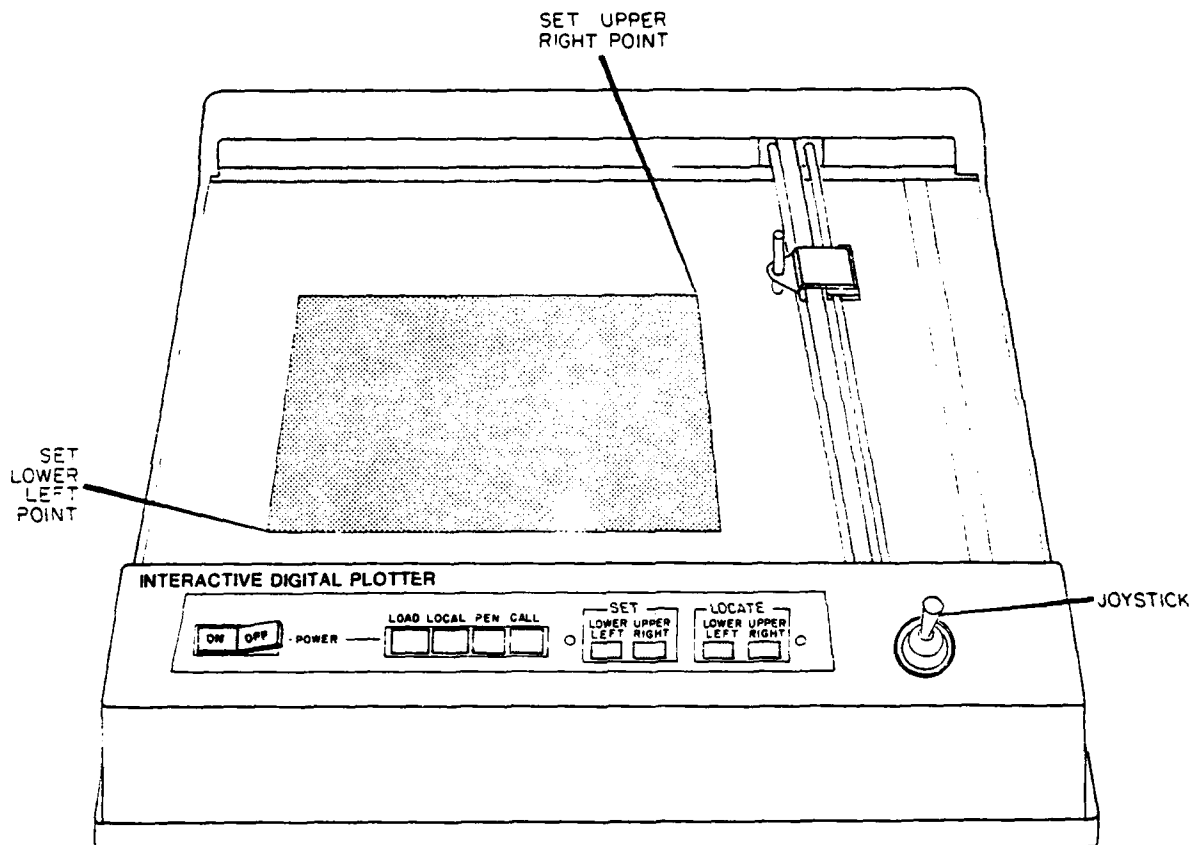


Fig. 1 - Flat-bed plotter (Tektronix 4662)

- Rock the POWER switch to the right to turn the plotter on. The POWER indicator will light and the pen will move down along the right boundary until it reaches the lower right corner where it will stop.
- Depress the LOAD key to its locked position. This will lift the pen and move it to the LOAD position (upper right corner of platen), and the electrostatic paper hold-down will be disengaged.
- Position a new piece of paper on the platen so the bottom edge of the paper lays evenly along the paper guide.

- Press the LOAD key to release it from its locked position; this will activate the electrostatic paper hold-down. If "bubbles" appear under the paper, smooth them out with your hand.
- Choose a scale so that the graph fits the page. Use the joystick to position the pen to the lower left corner of the paper.
- Press the SET LOWER LEFT key and hold it down until the plotter bell rings.
- Using the joystick, move the pen to the upper right corner of the paper.
- Press the SET UPPER RIGHT key and hold it down until the plotter bell rings.

After logging-in on the terminal, you may now type RUN PLOTTER <CR> (<CR> represents depressing the RETURN key). The graphs produced on the CRT and on the flat-bed plotter (Tektronix 4662) are identical except there is 12-bit resolution on the Tektronix 4662 and 10-bit resolution on the CRT plots. To ensure a 12-bit resolution, the rear-panel switches of the 4662 must be properly set. These parameter settings are vital to proper system operation. Figure 2 shows the switch positions.

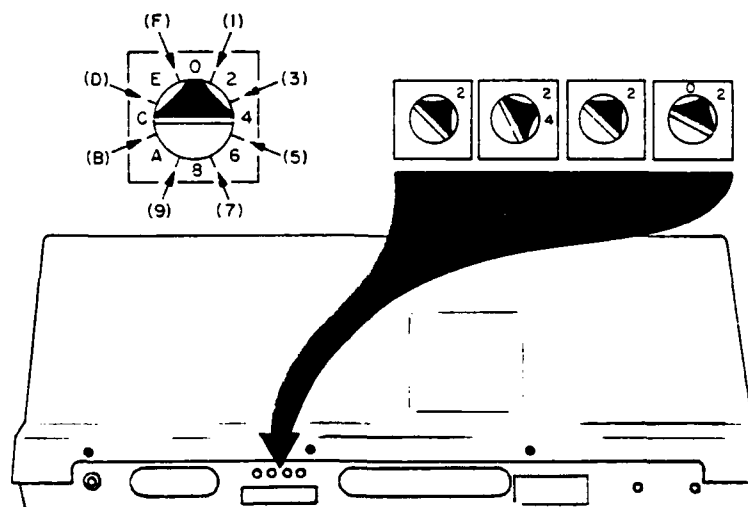


Fig. 2 - There are four hexadecimal switches labeled A, B, C, and D in the rear panel of the 4662. The switches should be set at the 2321 position as shown.

3.2. Input: The user must supply the name of the single file containing the XY coordinates or the names of the files containing the X data and the Y data. The data files can be formatted or unformatted. As an option, the XY data can be entered interactively. This option will store the data in two unformatted files: DATA.X.DAT. and DATA.Y.DAT.

3.3. Output: A typical output is shown in Section 4. The output is primarily the CRT plot or 4662 plot with appropriate labels.

3.4. Limitations: The data arrays have been dimensioned at 1000. To accommodate more than 1000 points the source-program statement must be changed.

3.5. Format: PLOTTER can accommodate data file(s) containing one X or Y value, or one XY coordinate per line in any format.

4. TEST METHOD AND RESULTS

To test PLOTTER, four data files were created in different formats. The interaction found in Section 4.1. is an example of the type of responses needed for this routine. Sections 4.2. and 4.3. give examples of the graphs produced by PLOTTER on the Tektronix 4662.

4.1. Running PLOTTER: This is an example of the interaction involved in running the program.

```
RUN PLOTTER <CR>
DO YOU WANT TO CREATE NEW DATA FILES? Y/N: N <CR>
ENTER NUMBER OF POINTS TO BE PLOTTED: 200 <CR>
COORDINATES CONTAINED IN 1 OR 2 FILES? 1 <CR>
ARE FILES (1)UNFORMATTED OR (2)FORMATTED: 2 <CR>
ENTER FORMAT TO USE FOR READING: (IE. (2F10.0) ) (F4.1,E7.1) <CR>
FILENAME FOR X,Y DATA: CAL10.DAT <CR>
DO YOU WANT AUTOMATIC SCALING? Y/N N <CR>
XMIN = 0.20000000E+01
XMAX = 0.70400002E+02
YMIN = 0.49976001E+01
YMAX = 0.50847101E+01
ENTER THE XMIN,XMAX,YMIN,YMAX YOU WISH TO
USE SEPARATED BY COMMAS: 2.0,30.0,4.97,5.10 <CR>
ENTER THE # OF X TIC MARKS,# OF Y TIC MARKS: 2,5 <CR>
DO YOU WANT POINT PLOT? Y/N N <CR>
DO YOU WANT LABELS? IF USING CRT, TYPE N: Y <CR>
TYPE IN LABEL FOR PLOT: CALIBRATION AT 10°C <CR>
TYPE IN LABEL FOR X AXIS: PRESSURE <CR>
TYPE IN LABEL FOR Y AXIS: VOLUME <CR>
DO YOU WANT ANOTHER CURVE ON THIS AXIS? Y/N N <CR>
YOU HAVE A CHOICE AS TO WHICH TYPE LINE TO USE.
SOLID LINE - TYPE 1 DASHED LINE - TYPE 2
DASH-DOT LINE - TYPE 3 SHORT DASHES - TYPE 4 1 <CR>
DO YOU WANT TO CREATE NEW DATA FILES? Y/N: N <CR>
ENTER NUMBER OF POINTS TO BE PLOTTED: 200 <CR>
COORDINATES CONTAINED IN 1 OR 2 FILES? 1 <CR>
ARE FILES (1)UNFORMATTED OR (2)FORMATTED? 2 <CR>
ENTER FORMAT TO USE FOR READING: (IE. (2F10.0) ) (2F10.0) <CR>
FILENAME FOR X,Y DATA: PCAL10.DAT <CR>
DO YOU WANT ANOTHER CURVE ON THIS AXIS? Y/N N <CR>
TO GRAPH ANOTHER CURVE, TYPE RUN PLOTTER
>
```

NOTE: All underlined portions are user-supplied.

4.2. PLOTTER Output Example: Figure 3 is a typical plot of four experimental data sets and a rational-fraction curve fit to the data.

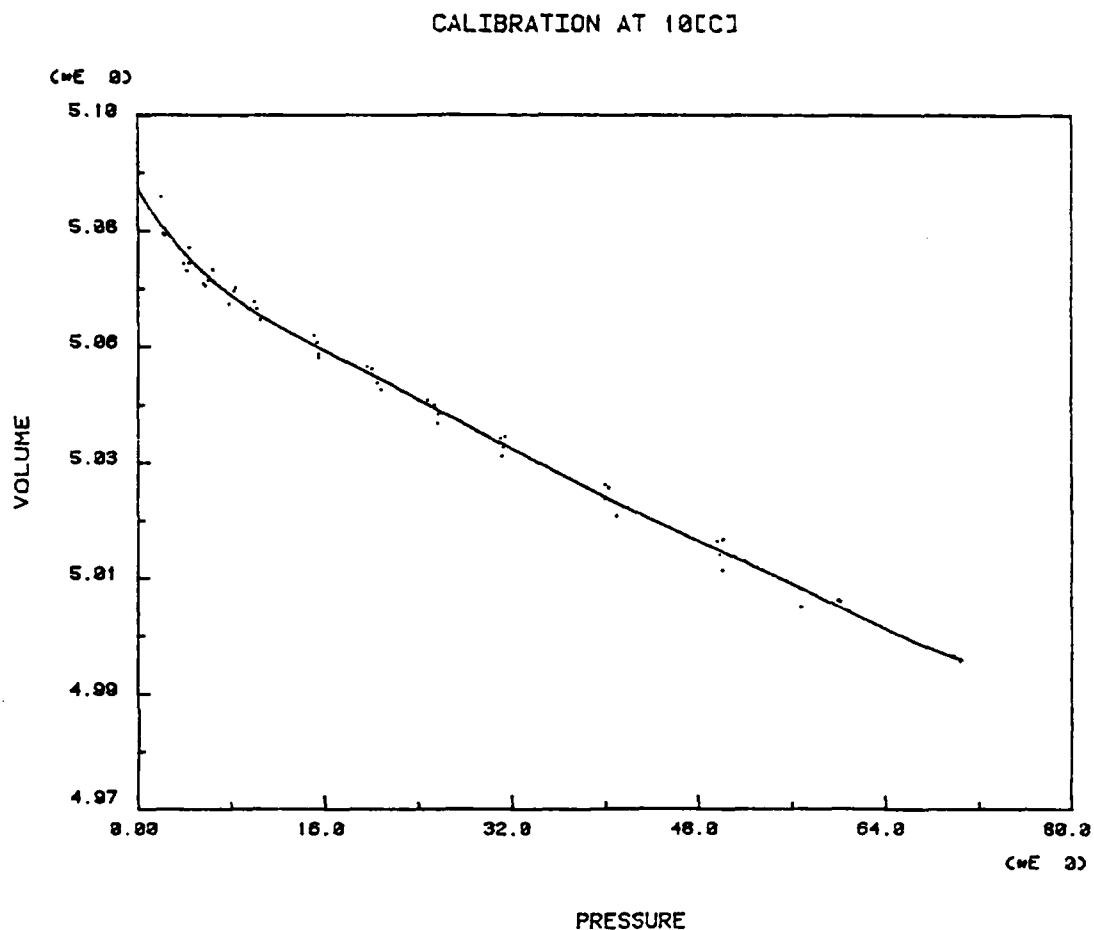


Fig. 3

4.3. PLOTTER Output Example: Figure 4 is another example of the output from the Tektronix 4662 plotter. This plot illustrates the various line types that can be obtained.

PYCNOMETER DATA

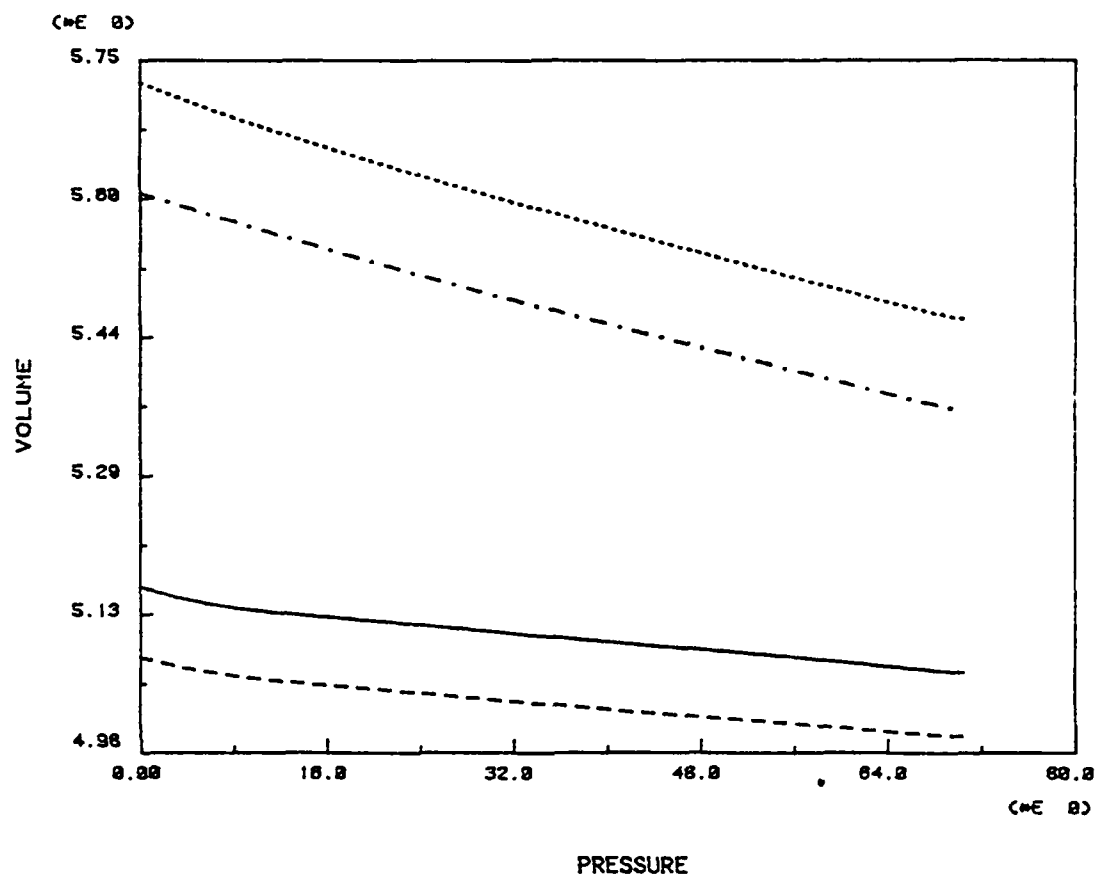


Fig. 4

Acknowledgments

The authors wish to thank R.E. Scott, L. Dwight Luker, and J.D. George for the various subroutines they have provided us. Also, consultations with Anthony J. Rudgers and L. Dwight Luker were very much appreciated.

```

C      : .....
C      : PLOTTER PROGRAM
C      : SEPTEMBER, 1980
C      : PROGRAMMED BY TINA RUGGIERO AND BOB ANDERSON
C      : .....

```

```
C : .....  
C :  
C : THIS PROGRAM IS DESIGNED FOR THE TEKTRONIX-4662 FLAT BED PLOTTER  
C : IN WHERE X AND Y DATA CAN BE PLOTTED IN THE FOLLOWING FORMS:  
C :   1)X AND Y DATA IN SEPARATE FILES-UNFORMATTED  
C :   2)X AND Y DATA IN SEPARATE FILES-FORMATTED  
C :   3)X AND Y DATA IN SAME FILE-UNFORMATTED  
C :   4)X AND Y DATA IN SAME FILE-FORMATTED OR  
C :   5)POINTS CAN BE INPUTED MANUALLY.  
C :  
C : .....
```

```

C .....
C :
C : SUBROUTINES USED ARE: TPLLOT, XYPLOT, TERM, AXIS, MINMAX, TKDASH,
C : A4662, PON, POFF, SCALE, PLREAD
C :
C : .....

```

```

C      : .....
C      :
C      :      LIMITATIONS:
C      :      DATA ARRAYS ARE DIMENTIONED AT 1000 POINTS
C      :      DATA MUST BE REAL
C      :
C      : .....

```

```

C      : .....
C      :
C      :     NOTE:
C      :     THERE IS A PROGRAM CALLED 'UEDIT' WHICH CAN BE USED FOR
C      :     EDITING UNFORMATTED SINGLE ARRAYS.
C      :
C      : .....

```

```

C      :
C      :
C      :      BEFORE RUNNING PROGRAM, TURN PLOTTER ON, LOAD PAPER, AND SET THE
C      :      LOWER LEFT AND UPPER RIGHT POINTS
C      :

```

```

BYTE DATA(32), DATAY(32), COOR(4), ILFA(70), ITXT(148)
REAL X(2050), Y(2050), XMIN, XMAX, YMIN, YMAX, XMIN1, XMAX1, YMIN1, YMAX1,
1  XRANG, YRANG, DASH(8)
INTEGER NPOINT, NPONT, LENX, LENY, NANS,
1  ITYPE, XVAL(2050), YVAL(2050),
1  ITYMAX, ITXMIN, ITYMIN

```

```

CALL ERRSET(29,...FALSE....FALSE...,)
CALL ERRSET(30,...FALSE....FALSE...,)

```

```

C      :
C      :
C      :      ICENT IS HALF THE NUMBER OF CHARACTERS USED IN THE LABEL
C      :

```

```

ICENT=42
NDASH=1

```

```

C      :
C      :
C      :      POFF TURNS THE PLOTTER OFF
C      :

```

```

10      CALL POFF(5)
      WRITE(5,11)

```

```

C      :
C      :
C      :      IF YOU HAVE YOUR DATA ON A FILE YOU DON'T WANT A
C      :      NEW DATA FILE.
C      :

```

```

11      FORMAT(/,'DO YOU WANT TO CREATE NEW DATA FILES? Y/N: ')
      READ (5,12)NANS
12      FORMAT(A1)
      NANS='N'
      IF (NANS .EQ. 'N') GO TO 16
      IF (NANS .EQ. 'Y') GO TO 15
      GO TO 10

```

```

C      :
C      :
C      :      XYPLOT IS WHERE NEW DATA FILE IS CREATED IF POINTS
C      :      ARE ENTERED MANUALLY
C      :

```

```

15  CALL XYPLOT(X,Y,NPONT)
    GO TO 20
16  WRITE(5,17)
    READ(5,18)NPONT
17  FORMAT(/,'ENTER NUMBER OF POINTS TO BE PLOTTED: ')

```

```

C  : .....
C  :
C  :     IF YOU DON'T KNOW THE NUMBER OF POINTS IN THE FILE,
C  :     MAKE SURE YOU ENTER A NUMBER THAT IS POSITIVELY
C  :     LARGER THAN THE NUMBER OF POINTS YOU HAVE.
C  :
C  : .....

```

```

18  FORMAT(I4)

```

```

C  : .....
C  :
C  :     PLREAD DETERMINES THE NUMBER OF FILES AND THE FORMAT THAT YOUR
C  :     DATA IS IN
C  :
C  : .....

```

```

CALL PLREAD(X,Y,NPOINT)

```

```

20  NPONT=NPONT
    IF(ANSW .EQ. 'Y')CALL PON(5)
    IF(ANSW .EQ. 'Y')GO TO 60

```

```

C  : .....
C  :
C  :     MINMAX DETERMINES THE MINIMUM AND MAXIMUM VALUE OF AN ARRAY
C  :
C  : .....

```

```

CALL MINMAX(X,XMIN,XMAX,NPONT)
CALL MINMAX(Y,YMIN,YMAX,NPONT)
WRITE(5,21)

```

```

C  : .....
C  :
C  :     AUTOMATIC SCALING WILL SCALE YOUR DATA AND WILL
C  :     CALL THE SUBROUTINE AXIS.
C  :
C  : .....

```

```

21  FORMAT(/,'DO YOU WANT AUTOMATIC SCALING? Y/N  ')
    READ(5,23)AUTO
23  FORMAT(A1)
    IF(AUTO.NE. 'Y')GO TO 30

```

```

C  .....
C  :
C  :   SCALE DRAWS AND LABELS AXIS
C  :
C  :
C  :
C  :
C  .....

```

```

    CALL SCALE(XMIN-(XMIN*5.0E-05),XMAX+(XMAX*5.0E-05),XMIN1,XMAX1,NTICX)
    CALL SCALE (YMIN-(YMIN*5.0E-05),YMAX+(YMAX*5.0E-05),YMIN1,YMAX1,NTICY)
    GO TO 42
C  THE X AND Y MIN AND MAX WILL BE PRINTED.

```

```

30  WRITE(5,31)XMIN,XMAX,YMIN,YMAX
31  FORMAT(' XMIN=',E15.8,/, ' XMAX=',E15.8,/, ' YMIN=',E15.8,/,
    1  ' YMAX=',E15.8)
    WRITE(5,32)
    WRITE(5,33)
32  FORMAT(/,' ENTER THE XMIN,XMAX,YMIN,YMAX YOU WISH TO')
33  FORMAT('S USE SEPARATED BY COMMAS:  ')
    READ(5,34)XMIN1,XMAX1,YMIN1,YMAX1
34  FORMAT(4E15.0)
    WRITE(5,40)

```

```

C  .....
C  :
C  :   TIC MARKS ARE THE DIVISIONS ON THE AXIS
C  :
C  :
C  :
C  :
C  .....

```

```

40  FORMAT(/,'ENTER THE # OF X TIC MARKS,# OF Y TIC MARKS:  ')
    READ (5,41)NTICX,NTICY
41  FORMAT(2I2)
42  CONTINUE

```

```

C  .....
C  :
C  :   THESE VALUES ARE THE MAX. & MIN OF THE TEKTRONICS PLOTTER
C  :
C  :
C  :
C  :
C  .....

```

```

TXMIN=600
TXMAX=3999
TYMIN=550
TYMAX=2731
ITXMIN=TXMIN
ITXMAX=TXMAX
ITYMIN=TYMIN
ITYMAX=TYMAX
GO TO 72

```

```

C      : .....
C      :
C      : THE SUBROUTINE SCALE, AUTOMATICALLY SCALES THE DATA AND CALLS
C      : SUBROUTINE AXIS
C      : .....
50 CALL PON(5)
   CALL TERM(0,0)

51 CALL AXIS(XMIN1,XMAX1,NTICX,TXMIN,TXMAX,YMIN1,YMAX1,NTICY,TYMIN,
1 TYMAX,0)

C      : .....
C      :
C      : A4662 CHANGES THE SIZES OF THE CHARACTERS IN LABEL
C      : .....

   CALL A4662(5,'A',56,80,0,0)

C      SCALE THE POINTS

XRANG=XMAX1-XMIN1
TXRANG=TXMAX-TXMIN
YRANG=YMAX1-YMIN1
TYRANG=TYMAX-TYMIN

C      : .....
C      :
C      : THIS SECTION DETERMINES THE TYPE LINE DESIRED.  WHEN NDASH=1
C      : -SOLID LINE,NDASH=2-DASHED LINE,NDASH=3-DASH-DOT,
C      : AND NDASH=4 IS SMALL DASHES.
C      : .....

60 CONTINUE
   IF (NDASH .EQ. 1) DASH(1)=2000.0
   IF (NDASH .EQ. 2) DASH(1)=40.0
   IF (NDASH .EQ. 2) DASH(2)=40.0
   IF (NDASH .EQ. 3) DASH(1)=50.0
   IF (NDASH .EQ. 3) DASH(2)=50.0
   IF (NDASH .EQ. 3) DASH(3)=4.0
   IF (NDASH .EQ. 3) DASH(4)=50.0
   IF (NDASH .EQ. 4) DASH(1)=10.0
   IF (NDASH .EQ. 4) DASH(2)=25.0
   IF (NDASH .EQ. 4) DASH(3)=10.0
   IF (NDASH .EQ. 4) DASH(4)=25.0
   IF (NDASH .EQ. 4) DASH(5)=10.0
   IF (NDASH .EQ. 4) DASH(6)=25.0
   IF (NDASH .EQ. 4) NDASH=6
   IF (NDASH .EQ. 3) NDASH=4
   MO=1

```

```

DO 62 I=1,NPONT
IF(Y(I).LT.YMIN1 .OR. X(I) .LT. XMIN1 .OR. X(I) .GT. XMAX1)GOTO 62
XVAL(MO)=INT(((X(I)-XMIN1)*(TXRANG))/(XRANG))+TXMIN)
YVAL(MO)=INT(((Y(I)-YMIN1)*(TYRANG))/(YRANG))+TYMIN)
61  FORMAT(2E15.8,2I4)

C .....
C :
C :      TPLOT POSITIONS PEN AND WHEN ITYPE<0, A POINT IS DRAWN
C :
C : .....

IF(NAN .EQ. 'Y') CALL TPLOT (XVAL(MO),YVAL(MO),-1)

C .....
C :
C :      TKDASH PLOTS THE DESIRED TYPE LINE
C :
C : .....

IF (NAN .NE. 'Y') CALL TKDASH(XVAL(MO),YVAL(MO),MO,DASH,NDASH)
MO=MO+1
62  CONTINUE
CALL TPLOT(XVAL(MO),YVAL(MO),0)

C .....
C :
C :      TERM(1,0) DUMPS THE BUFFER
C :
C : .....

CALL TERM(1,0)
CLOSE(UNIT=1)
CLOSE(UNIT=2)
63  CONTINUE
CALL POFF(5)
WRITE(5,70)

C .....
C :
C :      THIS ALLOWS FOR UP TO 4 CURVES ON 1 GRAPH.
C :
C : .....

70  FORMAT(/,'DO YOU WANT ANOTHER CURVE ON THIS AXIS? Y/N  ')
READ(5,71)ANSW
71  FORMAT(A1)
IF(ANSW .EQ. 'Y') GO TO 120
GO TO 140
72  CONTINUE
ITYPE=0
WRITE(5,80)

```

```

C      : .....
C      :
C      :     IF YOU DON'T WANT A LINE PLOT YOU WILL GET A
C      :     POINT PLOT.
C      :
C      : .....

```

```

80      FORMAT(/,'$DO YOU WANT POINT PLOT? Y/N  ')
      READ(5,81)NAN
81      FORMAT(A1)
      IF (NAN .EQ. 'Y') ITYPE=-1
      WRITE(5,82)

```

```

C      : .....
C      :
C      :     THERE ARE THREE LABELS.  ONE LABEL IS THE TITLE LABEL
C      :     AND THEN THERE IS A LABEL FOR THE X AND Y AXIS.
C      :
C      : .....

```

```

82      FORMAT(/,'$DO YOU WANT LABELS? IF USING CRT, TYPE N:  ')
      READ(5,83)ILAB
83      FORMAT(A2)
      IF(ILAB .EQ. 'N') GO TO 50
      WRITE(6,84)
84      FORMAT(/,'$TYPE IN LABEL FOR PLOT:  ')

```

```

C      : .....
C      :
C      :     ICNT IS THE LENGTH OF WORD, ILFA IS THE WORD
C      :
C      : .....

```

```

90      READ(5,91)ICNT,ILFA
91      FORMAT(Q,70A1)
      IF(ICNT.EQ.0)GO TO 93
      DO 92 I=1,ICNT
      ITXT(I)=ILFA(I)
92      CONTINUE
      NT=ICNT
      IF(A .EQ.1)GO TO 111
      IF(B .EQ.1)GO TO 113

```

```

C      : .....
C      :
C      :     PON TURNS THE PLOTTER ON
C      :
C      : .....

```

```

93      CALL PON(5)

```

```

C .....
C : TERM(0,0) CLEARS THE SCREEN
C :
C :
C .....

```

```

CALL TERM(0,0)
CALL TPLOT(0,2400,0)
IF(NT.GT. 1)WRITE (6,100)(ITXT(IX),IX=1,NT)
100 FORMAT(' ',T(ICENT-(ICNT/2)),<NT>A1)
CALL POFF(5)
WRITE(5,110)
110 FORMAT(/,'STYPE IN LABEL FOR X AXIS: ')
A=1
GO TO 90
111 CALL TPLOT(0,250,0)
ICENT=50
CALL PON(5)
CALL A4662(5,'A',50,75,0,0)
CALL TERM(2,0)
WRITE(6,100)(ITXT(IX),IX=1,NT)
CALL POFF(5)
WRITE(5,112)
112 FORMAT(/,'STYPE IN LABEL FOR Y AXIS: ')
B=1
A=0
GO TO 90
113 CALL TPLOT(125,0,0)
ICENT=35
CALL PON(5)
CALL A4662(5,'A',50,75,90,0)

CALL TERM(2,0)
WRITE(6,100)(ITXT(IX),IX=1,NT)
CALL A4662(5,'A',44,60,0,0)

GO TO 51
120 WRITE(5,121)
121 FORMAT(/,' YOU HAVE A CHOICE AS TO WHICH TYPE LINE TO USE. ')
WRITE(5,122)
122 FORMAT(/,' SOLID LINE - TYPE 1 DASHED LINE - TYPE 2')
WRITE(5,123)
123 FORMAT('S DASH-DOT LINE - TYPE 3 SHORT DASHES - TYPE 4 ')
READ (5,124)NDASH
124 FORMAT(I1)
GO TO 10
125 WRITE(5,130)
130 FORMAT(' ENTERED WRONG DATA FILE')
CLOSE(UNIT=5)
CALL PLREAD(X,Y,NPOINT)
131 WRITE(5,130)
CALL PLREAD(X,Y,NPOINT)
140 WRITE(5,141)
141 FORMAT(/,' TO GRAPH ANOTHER CURVE, TYPE RUN PLOTTER')
END

```

APPENDIX B

SUBROUTINES

The following subroutines are those required for the program PLOTTER:

Subroutine: XYPLOT

```

C      : .....
C      :
C      : THIS SUBROUTINE CREATES TWO UNFORMATTED REAL ARRAYS
C      : PROGRAMMED BY TINA RUGGIERO
C      : SEPTEMBER, 1980
C      : .....

SUBROUTINE XYPLOT(X,Y,NPONT)

C      : .....
C      :
C      : THE ARGUMENTS X,Y,NPONT ARE RESPECTIVELY, THE ARRAY CONTAINING
C      : ABSISSAS, THE ARRAY CONTAINING ORDINATES, AND THE NUMBER OF
C      : POINTS TO BE PLOTTED.
C      : .....

REALX(1),Y(1)
INTEGER I,SUB,ANS,NPONT

WRITE(5,10)
10  FORMAT(//,' ENTER ORDERED PAIR SEPARATED BY A COMMA')

I=0
WRITE(5,20)

20  FORMAT('/ WHEN FINISHED ENTERING DATA,TYPE "CTRL Z"')
OPEN(UNIT=1,NAME='DATA.X.DAT',TYPE='NEW',FORM='UNFORMATTED')
OPEN(UNIT=2,NAME='DATA.Y.DAT',TYPE='NEW',FORM='UNFORMATTED')
15  I=I+1
WRITE(5,22)
22  FORMAT('/SDATA? ')
READ(5,30,END=49)X(I),Y(I)
30  FORMAT(2E15.8)
GO TO 15
49  CLOSE(UNIT=5)
WRITE(5,50)
50  FORMAT(/,'DO YOU WANT TO CHANGE ANY VALUES (Y/N)? ')
READ(5,60)ANS
60  FORMAT(A1)
IF (ANS.EQ.'N')GO TO 40
WRITE(5,70)
70  FORMAT('ENTER SUBSCRIPT NUMBER OF CORRECTION ')

READ(5,30)SUB
80  FORMAT(I2)
WRITE(5,90)
90  FORMAT('RE-ENTER BOTH THE X AND Y VALUES ')
READ(5,30)X(SUB),Y(SUB)
GO TO 49
40  NPONT=I-1
DO 91 I=1,NPONT

```

(continued)

Subroutine: XYPLOT (continued)

```

      WRITE(1)X(I)
      WRITE(2)Y(I)
91    CONTINUE
      CLOSE(UNIT=1)
      CLOSE(UNIT=2)
      WRITE(5,800)
800   FORMAT(/,' REMEMBER YOU HAVE JUST CREATED TWO UNFORMATTED ARRAYS')
      WRITE (5,801)
801   FORMAT('          WHICH ARE CALLED DATAX.DAT AND DATAY.DAT')
      WRITE(5,100)
100   FORMAT(/,' DATA FILE CLOSED')
      RETURN
      END

```

Subroutine: SCALE

Entry: AXIS

```

      SUBROUTINE SCALE(DMN,DMX,SMN,SMX,NDIV)
C    "SCALE" DETERMINES AN APPROPRIATE SCALE WITH NEAT LABELS
C    AT MULTIPLES OF 1.2, OR 5.
C    REQUIRES:
C      DMN=MIN DATA VALUE
C      DMX=MAX DATA VALUE
C    RETURNS:
C      SMN=MIN SCALE VALUE
C      SMX=MAX SCALE VALUE
C      NDIV=NUMBER OF SCALE DIVISIONS
C    L. D. LUKER      8/15/80
C    REVISED BY TINA RUGGIERO
      TEKINT(Z)=ANINT(Z-.499999)
      NFMT(VA)=MINI(3.-TEKINT(1.+ALOG10(ABS(VA))),2.)
D      TYPE *, 'SCALE ', ' DMN=',DMN,' DMX=',DMX
      R=(DMX-DMN)/6.
      S=10.**((TEKINT(ALOG10(R)))
      T=R/S
      IF (T.GT.2) GOTO 110
      IF (T.EQ.1) GOTO 130
      S=S*2
      GOTO 130
110   IF (T.GT.5) GOTO 120
      S=5*S
      GOTO 130
120   S=10*S
130   SMN=TEKINT(DMN/S)
      SMN=S*(SMN+2)
140   IF ((SMN-DMN)/(DMX-DMN).LE.1E-5) GOTO 150
      SMN=SMN-S
      GOTO 140
150   SMX=TEKINT(DMX/S)
      SMX=S*(SMX-2)
160   IF ((DMX-SMX)/(DMX-DMN).LE.1E-5) GOTO 170
      SMX=SMX+S
      GOTO 160
170   NDIV=NINT((SMX-SMN)/S)
D      TYPE *, 'SCALE ', ' SMN=',SMN,' SMX=',SMX,' NDIV=',NDIV
      RETURN

```

(continued)

Subroutine: SCALE (continued)
Entry: AXIS (continued)

```

C      ENTRY AXIS(XMN,XXM,NXDIV,TXMN,TXXM,YMN,YMY,NYDIV,TYMN,TYMX,IZL)
C      DRAWS AND LABELS THE X & Y AXES
C      REQUIRES:
C      XMN=MIN X VAL
C      XXM=MAX X VAL
C      NXDIV=NUMBER OF X SCALE DIVISIONS
C      TXMN=TERMINAL X VAL CORRESPONDING TO MIN X VAL
C      TXXM=TERMINAL X VAL CORRESPONDING TO MAX X VAL
C      ALSO ALL THE ABOVE FOR Y
C      IZL=0 DON'T DRAW ZERO LINE , IZL=1 DRAW ZERO LINE
D      TYPE *, 'AXIS=',XMN,XXM,NXDIV,TXMN,TXXM,YMN,YMY,NYDIV,TYMN,TYMX
      T1=AMAX1(ABS(XMN),ABS(XXM))
      T1=10.**((3*TEKINT((TEKINT(ALOG10(T1+ABS((XXM-XMN)/1E4)))))/3.))
      X1=XMN/T1
      XDIV=(XXM-XMN)/(2*NXDIV)
      XDX=(TXXM-TXMN)/(2*NXDIV)
      IX=TXMN
      IY=TYMN
      CALL TPLLOT(IX-144,IY-20,0)
      N=2
      IF (ABS(X1).GT..001) N=NFMT(X1)
      CALL TERM(2,0)
      WRITE(6,210)X1
      DO 180 I=1,NXDIV*2
      X1=(XMN+I*XDIV)/T1
      IF (ABS(X1).LT..1) X1=0.
      IX=TXMN+(I-1)*XDX
      CALL TPLLOT(IX,IY,0)
      IX=TXMN+I*XDX
      CALL TPLLOT(IX,IY,1)
      CALL TPLLOT(IX,IY+20,1)
      IF (I/2..NE.AINT(I/2.)) GOTO 180
      CALL TPLLOT(IX,IY+40,1)
      CALL TPLLOT(IX-144,IY-20,0)
      N=2
      IF (ABS(X1).GT..001) N=NFMT(X1)
      CALL TERM(2,0)
      WRITE(6,210)X1
180  CONTINUE
      CALL TPLLOT(IX-240,IY-120,0)
      CALL TERM(2,0)
      IT1=ANINT(ALOG10(T1))
      WRITE(6,220)IT1
C
      T1=AMAX1(ABS(YMN),ABS(YMY))
      T1=10.**((3*TEKINT((TEKINT(ALOG10(T1+ABS((YMY-YMN)/1E4)))))/3.))
      Y1=YMN/T1
      YDIV=(YMY-YMN)/(2*NYDIV)
      YDX=(TYMX-TYMN)/(2*NYDIV)
      IY=TYMN
      IX=TXMN
      CALL TPLLOT(IX-288,IY+80,0)
      N=2
      IF (ABS(Y1).GT..001) N=NFMT(Y1)
      CALL TERM(2,0)
      WRITE(6,210)Y1
      DO 190 I=1,NYDIV*2
      Y1=(YMN+I*YDIV)/T1
      IF (ABS(Y1).LT..1) Y1=0.
      IY=TYMN+(I-1)*YDX
      CALL TPLLOT(IX,IY,0)
      IY=TYMN+I*YDX

```

(continued)

Subroutine: SCALE (continued)

Entry: AXIS (continued)

```
      CALL TPLLOT(IX,IV,1)
      CALL TPLLOT(IX+20,IV,1)
      IF (I/2..NE.AINT(I/2.)) GOTO 190
      CALL TPLLOT(IX+40,IV,1)
      CALL TPLLOT(IX-280,IV+56,0)
      N=2
      IF (ABS(Y1).GT..001) N=NFMT(Y1)
      CALL TERM(2,0)
      WRITE(6,210)Y1
190   CONTINUE
      CALL TPLLOT(IX-316,IV+160,0)
      CALL TERM(2,0)
      IT1=ANINT(ALOG10(T1))
      WRITE(6,220)IT1
      ITXMN=TXMN
      ITXMX=TXMX
      ITYMN=TYMN
      ITYMX=TYMX
      CALL TPLLOT(ITXMN,ITYMX,0)
      CALL TPLLOT(ITXMX,ITYMX,1)
      CALL TPLLOT(ITXMX,ITYMN,1)
      IF (IZL.EQ.0) GOTO 200
      IF (YMN*YMX.GT.0) GOTO 200
      IYZERO=(-YMN)*(TYMX-TYMN)/(YMX-YMN)+TYMN
      CALL TPLLOT(ITXMN,IYZERO,0)
      CALL TPLLOT(ITXMX,IYZERO,1)
200   RETURN
C   FORMAT STATEMENTS
210   FORMAT(' ',F5.4,N)
220   FORMAT(' ',I3,'')
      END
```

Subroutine: PLREAD

```
C   : .....
C   :
C   : THIS SUBROUTINE IS DESIGNED TO READ X AND Y COORDINATES FROM EITHER
C   : FORMATTED OR UNFORMATTED FILES CONTAINING X, Y, OR X AND Y DATA.
C   : AS IT READS THE NUMBER OF POINTS SPECIFIED (NPTS) IT ALSO DETERMINES
C   : THE MINIMUM AND MAXIMUM VALUES OF EITHER OR BOTH COORDINATES. IT ALSO
C   : MODIFIES NPTS IF AN ATTEMPT IS MADE TO READ MORE VALUES (OR SETS OF
C   : VALUES) THAN ARE CONTAINED IN THE SPECIFIED FILE.
C   : .....
C   :
C   : WRITTEN BY BOB ANDERSON AND TINA RUGGIERO
```

SUBROUTINE PLREAD(X,Y,NPTS)

(continued)

Subroutine: PLREAD (continued)

```
C      : .....
C      : THE FILE NAME(S) ARE SAVED ALONG WITH THE NUMBER OF FILES AND THE
C      : FILE TYPE(S). FILE TYPE = 2 * NUMBER OF FILES + 1 FOR UNFORMATTED, OR
C      : 2 FOR FORMATTED.
C      : .....
```

COMMON FILEX,FILEY,IFTYP

```
C      : .....
C      : ARRAYS TO SAVE UP TO NPTS COORDINATE VALUES.
C      : .....
```

DIMENSION X(1),V(1)

```
C      : .....
C      : ARRAYS IN COMMON TO SAVE THE FILE NAME(S). FILEX IS USED FOR ONLY ONE
C      : FILE.
C      : .....
```

BYTE FILEX(32),FILEY(32),FMT(32)

```
C      : .....
C      : DETERMINE IFTYP BY FIRST FINDING OUT HOW MANY FILES ARE TO BE USED.
C      : .....
```

```
50      CLOSE(UNIT=5)
        WRITE(5,1000)
1000     FORMAT('SCCOORDINATES CONTAINED IN 1 OR 2 FILES? ')
        READ(5,1010,END=50)NFILES
1010     FORMAT(I4)
        IF(NFILES.LT.1.OR.NFILES.GT.2) GOTO 50
```

```
C      : .....
C      : AND THEN FIND OUT IF IT OR THEY ARE FORMATTED OR UNFORMATTED.
C      : .....
```

(continued)

Subroutine: PLREAD (continued)

```

100  CLOSE(UNIT=5)
      WRITE(5,1020)
1020  FORMAT(/,'SARE FILES (1)UNFORMATTED OR (2)FORMATTED: ')
      READ(5,1010,END=100) ITYPE
      IFTYP=NFILES*2+ITYPE

```

```

C      .....
C      :
C      : NOW READ THE DATA UP TO NPTS. MODIFY NPTS IF < PARAMETER PASSED.
C      :
C      : .....

```

```

105  IF(IFTYP.EQ.3) GOTO 200
      IF(IFTYP.EQ.4) GOTO 300
      IF(IFTYP.EQ.5) GOTO 400
      IF(IFTYP.EQ.6) GOTO 500
      GOTO 50

```

```

C      .....
C      :
C      : BOTH COORDINATES ARE CONTAINED IN ONE FILE AND IT IS UNFORMATTED.
C      :
C      : .....

```

```

200  WRITE(5,1050)
1050  FORMAT(/,'$FILENAME FOR X,Y DATA: ')
      READ(5,1060,END=50)LEN,FILEX
C      NOTICE LAST CHANCE TO START OVER! (^Z)
1060  FORMAT(Q,32A1)
      FILEX(LEN+1)=0
      OPEN(UNIT=1,NAME=FILEX,FORM='UNFORMATTED',TYPE='OLD',ERR=800)
C      NOTICE THERE MUST BE AT LEAST ONE POINT.
      DO 220 I=1,NPTS
          READ(1,END=240)X(I),Y(I)
          J=I
220  CONTINUE
C      MODIFY NPTS TO REFLECT THE ACTUAL NUMBER OF POINTS READ.
240  NPTS=J
      CLOSE(UNIT=1)
      GOTO 900

```

```

C      .....
C      :
C      : BOTH COORDINATES ARE CONTAINED IN ONE FILE AND IT IS FORMATTED.
C      :
C      : .....

```

(continued)

Subroutine: PLREAD (continued)

```

300      CLOSE(UNIT=1)
        WRITE(5,3000)
3000     FORMAT(/'SENER FORMAT TO USE FOR READING: (IE.  (2F10.0)  ) ')
        READ(5,3010)FMT
3010     FORMAT(32A1)
        WRITE(5,1050)
        READ(5,1060,END=50)LEN,FILEX
        FILEX(LEN + 1)=0
        OPEN(UNIT=1,NAME=FILEX,TYPE='OLD',ERR=800)
        DO 330 I=1,NPTS
          READ(1,FMT,ERR=300,END=240)X(I),Y(I)
          J=I
330      CONTINUE
        GO TO 240

C      : .....:
C      :      :
C      :      COORDINATES ARE CONTAINED IN TWO FILES AND ARE UNFORMATTED      :
C      :      :
C      : .....:

400      WRITE(5,160)
160      FORMAT(/'SENER FILENAME FOR X VALUES:  ')
        READ (5,1060,END=900)LENX,FILEX
        WRITE(5,360)
360      FORMAT(/'SENER FILENAME FOR Y VALUES:  ')
        READ(5,1060,END=900)LENY,FILEY
        FILEX(LENX+1)=0
        FILEY(LENY+1)=0
        OPEN(UNIT=1,NAME=FILEX,TYPE='OLD',ERR=800,FORM='UNFORMATTED')
        OPEN(UNIT=2,NAME=FILEY,TYPE='OLD',ERR=800,FORM='UNFORMATTED')
        DO 460 I=1,NPTS
          READ(1,END=340)X(I)
          READ(2,END=340)Y(I)
          J=I
460      CONTINUE
340      NPTS=J
        CLOSE(UNIT=1)
        CLOSE(UNIT=2)
        GO TO 900

C      : .....:
C      :      :
C      :      COORDINATES ARE IN TWO FILES AND ARE FORMATTED      :
C      :      :
C      : .....:

500      CLOSE(UNIT=1)
        WRITE(5,3000)
        READ(5,3010)FMT
        WRITE(5,160)
260      READ(5,260,END=900)LENX,FILEX
        FORMAT(Q,32A1)
        WRITE(5,360)
        READ(5,260,END=900)LENY,FILEY
        FILEX(LENX+1)=0
        FILEY(LENY+1)=0
        OPEN(UNIT=1,NAME=FILEX,ERR=800,TYPE='OLD')
        OPEN(UNIT=2,NAME=FILEY,ERR=800,TYPE='OLD')
        DO 560 I=1,NPTS
          READ(1,FMT,END=340)X(I)
          READ(1,FMT,END=340)Y(I)
          J=I

```

(continued)

Subroutine: PLEAD (continued)

```
568 CONTINUE
      GO TO 348
888 WRITE(5,881)
881  FORMAT(//,' YOU ENTERED WRONG DATA FILE(S)! ')
      GO TO 185
988 RETURN
      END
```

Subroutine: PON

Entry: POFF

```
C      SUBROUTINE PON (LUN)
C      SUBROUTINE TO TURN ON OR OFF THE TEKTRONIX 4662 PLOTTER.
C      WRITTEN BY RICK SCOTT
      BYTE PLON(3),PLOFF(3)
      INTEGER IPRM(6)
      DATA PLON/27,65,69/PLOFF/27,65,78/

      CALL GETADR (IPRM,PLON)
      IPRM(2)=3
      CALL QIO ("418,LUN,24.,,IPRM, )
      CALL WAITFR (24)
      RETURN

      ENTRY POFF
      CALL GETADR (IPRM,PLOFF)
      IPRM(2)=3
      CALL QIO ("418,LUN,24.,,IPRM, )
      CALL WAITFR (24)
      RETURN

      END
```

Subroutine: TKDASH

```
C      TKDASH.FTN      J.D.GEORGE OCTOBER 1975
C
C      THE PURPOSE OF SUBROUTINE TKDASH IS TO PLOT A CURVE AS A SERIES OF
C      ALTERNATING BRITE AND DARK LINE SEGMENTS OF ARC LENGTHS SPECIFIED
C      BY THE USER
C
C      SUBROUTINE TKDASH(IX,IY,N,DASH,NDASH)
C
C      IX,IY ARE SCREEN OR PAPER COORDINATES
C
C      N IS THE NUMBER OR INDEX OF THE POINT IX,IY
C      N=1, IS TREATED SEPERATELY, N.GE.1
```

(continued)

Subroutine: TKDASH (continued)

```

C      DASH      IS AN ARRAY OF SCREEN COORDINATE ARC LENGTHS
C                FOR ALTERNATELY BRITE AND DARK LINE SEGMENTS
C                ODD INDICES ARE BRITE SEGMENTS
C                EVEN INDICES ARE DARK SEGMENTS
C                I.E.      1      1      2      3      4
C                        DASH(1) 10    10    2    10
C
C                        BRITE    DARK    BRITE    DARK
C
C      NDASH      IS THE LENGTH OF THE DASH ARRAY
C                4 SHOULD PROVIDE A WIDE RANGE OF SYMBOLS
C                TO FORCE ALTERNATE BRITE-DK LINE SEGMENTS
C                NDASH IS EVEN
C                FOR SOLID LINE USE NDASH = 1 & DASH(1)=LARGE#
C
C      *****
C      SUBROUTINES REQUIRED:TPLOT
C      *****
C
C      SUBROUTINE TKDASH(IX,IY,N,DASH,NDASH)
C      COMMON /LUN /LUN
C      DIMENSION DASH(NDASH)
C      DATA ZERO/0.0/
C      LUN=5
C      IF(N.GT.1)GOTO 100
C
C      THE FIRST POINT INITIALIZES THINGS
C
C      XLAST=IX
C      YLAST=IY
C      LASTDK=1
C      IDASH=1
C      OLDARC=ZERO
C      CALL TPLOT (IX,IY,0)
C      RETURN
C
C      ENTRY FOR N.GT.1
C
C 100      CONTINUE
C          X=IX
C          Y=IY
C
C      THE CODE BELOW IS REPEATED UNTIL HAVE PLOTTED SEGMENTS TO
C      POINT IX,IY
C
C 200      CONTINUE
C
C          DX=X-XLAST
C          DY=Y-YLAST
C          ARC=SQRT(DX*DX+DY*DY)
C          IF(ARC.EQ.ZERO)GOTO 300
C
C      THE PATH DEPENDS ON WHETHER ARC EXTENDS BEYOND THE NEXT
C      LINE SEGMENT SPECIFIED IN DASH(IDASH)
C
C          IF((OLDARC+ARC).GE.(DASH(IDASH)))GOTO 300
C
C      THE ARC TERMINATES WITHIN THE CURRENT LINE SEGMENT

```

(continued)

Subroutine: TKDASH (continued)

```
      XINC=DX
      YINC=DY
      OLDARC=OLDARC+ARC
      GOTO 400

C
C
C THE ARC TERMINATES AT OR BEYOND THE CURRENT LINE SEGMENT
C
C
300  CONTINUE
      XINC=DX*(DASH(IDASH)-OLDARC)/ARC
      YINC=DY*(DASH(IDASH)-OLDARC)/ARC
      OLDARC=ZERO

C
400  CONTINUE
      X0=XLAST+XINC
      Y0=YLAST+YINC

C
C IF IDASH IS EVEN PLOT DARK VECTOR
C IF IDASH IS ODD PLOT BRITE VECTOR
C
C MODIFY TO NOTE THE TRANSITION FROM LITE TO DARK
C MOVE TO EDGE WITH DK VECTOR, THEN PUT DOWN DK VECTOR AT EDGE
C
C FOR DK VECTOR SKIP PLOTTING UNTIL SENSE LITE-TO-DK TRANSITION
C
      IDARK=MOD(IDASH,2)
      IX0=X0
      IY0=Y0
      IF(IDARK.EQ.0)GOTO 410
      IF(LASTDK.EQ.IDARK)GOTO 405
      IXLAST=XLAST
      IYLAST=YLAST
      CALL TPLT (IXLAST,IYLAST,0)
      CALL TPLT (IXLAST,IYLAST,1)
405  CONTINUE
      CALL TPLT (IX0,IY0,IDARK)
410  LASTDK=IDARK

C
C SETUP FOR NEXT POINT
C
      XLAST=X0
      YLAST=Y0

C
C REPEAT PLOTTING UNTIL ARC TERMINATES WITHIN A
C SEGMENT OF DASH
C
C I.E. OLDARC.NE.ZERO
C
      IF(OLDARC.NE.ZERO)GOTO 1000
      IDASH=MOD(IDASH,NDASH)+1
      GOTO 200

C
C EXIT
C
1000  CONTINUE
      RETURN
      END
```

Subroutine: MINMAX

```

C      : .....
C      :      THIS SUBROUTINE DETERMINES THE MINIMUM AND MAXIMUM VALUE OF ARRAY
C      :      "A" WHERE N IS THE NUMBER OF POINTS.
C      : .....
C      WRITTEN BY TINA RUGGIERO
C      SUBROUTINE MINMAX(A,MIN,MAX,N)
C      REAL A(N),MAX,MIN
C      INTEGER I,N
C      MIN=A(1)
C      MAX=A(1)
C      DO 38 I=2,N
C      IF (A(I).LT.MIN)MIN=A(I)
C      IF (A(I).GT.MAX)MAX=A(I)
38    CONTINUE
C      RETURN
C      END

```

Subroutine: TPLOT

```

C      SUBROUTINE TPLOT (IX,IY,M)
C      WRITTEN BY RICK SCOTT
C      REVISED BY BOB ANDERSON
C      SUBROUTINE TO PLOT ON THE TEKTRONIX 4010 AND 613 DISPLAY
C      TERMINALS (AS CHOSEN IN "TERM" SUBROUTINE).
C      FORTRAN-IV BUFFERED VERSION.

C      VALUES TO PLOT: IX,IY
C      MODES TO PLOT: M>0 (BRIGHT), M=0 (DARK), M<0 (POINT)
C      LUN 6: THIS SUBROUTINE USES QIOB.
C      REMEMBER TO PURGE THE BUFFER WHEN DONE (WITH TERM).

C      BYTE IOUT(8)
C      I=0
C      IOUT(1)=000
C      IF (M.GT.0) GOTO 11

C      INITIAL PLOT, DARK PLOT, POINT PLOT--
10    I=I+1
C      IOUT(I)="35

C      ALL MODES--SEPARATE COORDINATES INTO HIGH- AND LOW-ORDER BYTES
11    I=I+1
C      IOUT(I)=IY/128+32
C      I=I+1
C      IOUT(I)=(IY-1*(IY/4))+4
C      IOUT(I)=IOUT(I)+(IX-4*(IX/4))+96
C      I=I+1
C      IOUT(I)=IY-128*(IY/128)
C      IOUT(I)=96+(IOUT(I)/4)
C      I=I+1
C      IOUT(I)=IX/128+32
C      I=I+1
C      IOUT(I)=IX-128*(IX/128)
C      IOUT(I)=64+(IOUT(I)/4)

```

Subroutine: TPLOTT (continued)

```

C      IF (M.GE.8) GOTO 28
C      REINFORCE FOR POINT PLOT
12     I=I+1
        IOUT(I)=IOUT(I-1)

C      EXECUTE QIO AND RETURN
28     CALL QIOB ("518,6,24,8,IOUT,I,ISW)
        CALL WAITFR (24)

        RETURN
        END

```

Subroutine: TERM

```

C      SUBROUTINE TERM(K,L)
C      WRITTEN BY RICK SCOTT
C      THIS SUBROUTINE WILL MANIPULATE THE TERMINAL--
C      K=8, L=8 ERASE SCREEN
C      K=1, L=8 COPY SCREEN
C      K=2, L=8 RETURN TO ALPHA MODE
C      K=3, L=8 PURGE THE QIO BUFFER
C      K=8, L=C IMPLEMENT MULTIPLEXER
C      WHERE B IS BOARD SELECT NUMBER 8-3
C      WHERE C IS CONTROL NUMBER TERMINAL(1), A(2), B(3), C(4)
C      (COMBINATIONS OF TERMINALS ARE ALLOWED)

C      THIS PARTICULAR VERSION IS FOR USE WITH BUFFERED
C      PLOTTING, AND EVERY CALL TO TERM WILL PURGE THE BUFFER.

        BYTE IOUT(3)
        I=2
        IOUT(1)="33      IESCAPE

        IF (L.NE.8) GOTO 18
        KX=K+1
        GOTO (2,3,5,48),KX
2       IOUT(2)="14      ICLEAR THE SCREEN
        GOTO 38
3       IOUT(2)="27      ICOPY THE SCREEN
        GOTO 38
5       IOUT(1)="37      IRETURN TO ALPHA MODE
        I=1
        GOTO 38

C      PREPARE ASCII CHARACTERS FOR MUX BOARD AND CONTROL NUMBERS
18     IOUT(2)=K+"68
        IOUT(3)=2** (L-1) + "68
        I=3

C      OUTPUT THE CONTROL SEQUENCE
38     CALL QIOB ("618,6,24,8,IOUT,I,ISW)
C      AND PURGE THE BUFFER
48     CALL QIOB ("618,6,24,8,IDAT,8,ISW)

```

(continued)

Subroutine: TERM (continued)

```

C      PAUSE A MOMENT IF SCREEN IS BEING CLEARED
      IF ((K+L).NE.0) RETURN
      CALL WAIT (1,2,M)
      RETURN
      END

```

Subroutine: A4662

```

C      SUBROUTINE A4662 (LUN,DEV,KX,KY,KA,KF)
C      WRITTEN BY RICK SCOTT
C
C      SUBROUTINE TO SET UP THE ALPHABET OF PLOTTER DEV ON LINE LUN--
C      SIZE OF CHARACTERS (X AND Y), ANGLE, AND FONT.
C
C      ENTRY A4662R WILL RESET DEFAULT VALUES, WHICH ARE--
C      SIZE (56X88), ANGLE (0), FONT (0).
C
      INTEGER IPRM(6)
      BYTE STRING(23),IRST(3),DEV
C
      DATA STRING/'33','A','T',0,'33','A','I',3*0,' ',3*0,4,
1"33','A','J',4*0,4/IRST/'33','A','V'/
C
      STRING(2)=DEV
      STRING(4)=KF+"60
      STRING(6)=DEV
      ENCODE (3,100,STRING(8)) KX
      ENCODE (3,100,STRING(12)) KY
100  FORMAT (I3)
      STRING(17)=DEV
      ENCODE (4,101,STRING(19)) KA
101  FORMAT (I4)
C
      CALL GETADR (IPRM,STRING)
      IPRM(2)=23
C
200  CALL QIO ("410,LUN,24...,IPRM.)
      CALL WAITFR (24)
      RETURN
C
      ENTRY A4662R (LUN,DEV)
C
      IRST(2)=DEV
      CALL GETADR (IPRM,IRST)
      IPRM(2)=3
      GOTO 200
C
      END

```

APPENDIX C

COMPILATION AND TASKBUILDING

The Digital Equipment Corporation RSX-11M operating system provides an indirect command file processor that will pass commands to a system utility from a file (indirectly) rather than interactively (directly) from your terminal. This facility allows the user to create a file containing commands only once, minimizing effort in rebuilding subsequent tasks and helping to eliminate typographical or syntax errors.

The indirect-command-file processor allows multiple levels of files, which expands the flexibility of its use. In the examples provided, note the second level of indirectness used in PLOTTER.CMD. The system-utility task name can be included in the indirect command file, as in Example 1, or can be external to the command file as in Examples 2 and 3. The processor completely executes one command before it goes on to the next command. The current command is also displayed on the user's terminal allowing the user to monitor progress.

Example 1 is an indirect command file used to completely manage the rebuilding of the task PLOTTER. The semicolon is used to denote comment and is therefore ignored. Lines 2 through 6 in Example 1 contain commands to the system utility PIP (Peripheral Interchange Program) to delete or purge the user's area of unnecessary or unwanted files. Line 7 is a call to the system FORTRAN Four Plus compiler (F4P), passing it a second level of indirect commands containing F4P commands. Finally, line 8 in Example 1 instructs the System Task Building (TKB) to receive its commands from the file PLOTTER.TKB.

```
;PLOTTER.CMD
PIP PLOTTER.TSK;*/DE
PIP *.FTN/PU:3
PIP *.F4P/PU
PIP *.TKB/PU
PIP *.OBJ;*/DE
F4P @PLOTTER.F4P
TKB @PLOTTER.TKB
```

Example 1

Example 2 is an indirect command file containing commands for F4P to accomplish recompilation of all the FORTRAN source files used in the task PLOTTER.

```

;RE-COMPILE SOURCES
PLOTTER=PLOTTER
TERM=TERM
TPLOT=TPLOT
SCALE=SCALE
MINMAX=MINMAX
QIOB=QIOB
PON=PON
XYPLOT=XYPLOT
A4662=A4662
PLREAD=PLREAD
TKDASH=TKDASH

```

Example 2

The default extensions are .FTN for the source files (right of equal sign) and .OBJ for the created object files (left). Example 3 is an indirect command file containing commands for TKB to accomplish linking of all the necessary objects and allocation of space for a new task image. The default extension for the input file(s) is .OBJ and the output extension is .TSK.

```

;PLOTTER.TKB
PLOTTER=PLOTTER,TERM,TPLOT,SCALE,MINMAX,QIOB,PON,XYPLOT,A4662,PLREAD,TKDASH
/
ASG=TI:5:6,SY:1:2
ACTFIL=3

```

Example 3

To execute the command file PLOTTER.CMD, the user types '@PLOTTER'.

APPENDIX D

UEDIT

The program UEDIT allows the user to edit an unformatted file that has been created. Data may be changed or new data may be added.

```

PROGRAM UEDIT
C
C   BYTE FILE(32),ANS(4)
C
C   CALL ERRSET(29,...FALSE,...FALSE...)
C   CALL ERRSET(39,...FALSE,...FALSE...)
C---  GET THE FILENAME ---
C
50    WRITE (5,1000)
1000  FORMAT (/ 'ENTER FILENAME:')
READ (5,1010,END=800) LF,FILE
1010  FORMAT (Q,32A1)
FILE(LF+1)=0
C
OPEN (UNIT=1,NAME=FILE,TYPE='OLD',FORM='UNFORMATTED',
1     ERR=700)
C
C---  CHECK FOR VERSION # ---
C
90    DO 100 I=1,LF
100   IF (FILE(I).EQ.';') FILE(I)=0
CONTINUE
C
C---  OPEN THE TEMPORARY WORK FILE ---
C
OPEN (UNIT=2,NAME='TEMP.DAT',ACCESS='DIRECT',
1     FORM='UNFORMATTED',RECORDSIZE=1,TYPE='NEW')
C
C---  XFER DATA TO TEMP WORK FILE ---
C
NREC=0
READ (1,END=250) VALUE
NREC=NREC+1
WRITE (2,NREC) VALUE
GOTO 200
C
250   CLOSE (UNIT=1)
C
C---  READY TO EDIT DATA FILE ---
CALL EDIT (NREC)
C
C---  FINISHED WITH EDIT ---
C
OPEN (UNIT=1,NAME=FILE,FORM='UNFORMATTED',TYPE='NEW')
C
C---  RE-WRITE DATA ---
C
DO 400 I=1,NREC
READ (2,I) VALUE
WRITE (1) VALUE
CONTINUE
400   GOTO 800

```

(continued)

```

C
C--- OPEN FAILURE ON INPUT FILE ---
C
700 WRITE (5,1050)
1050 FORMAT ('$CREATE NEW FILE?')
      READ (5,1055,END=800) ANS
1055 FORMAT (4A1)
      IF (ANS(1).NE.'Y') GOTO 50
      OPEN (UNIT=1,NAME=FILE,FORM='UNFORMATTED',TYPE='NEW')
      GOTO 90
C
C--- GO BYE BYE
C
800 WRITE (5,1060)
1060 FORMAT ('$ EDITTING SESSION COMPLETE.')
      CLOSE (UNIT=1)
      CLOSE (UNIT=2,DISPOSE='DELETE')
      CALL EXIT
      END

```

