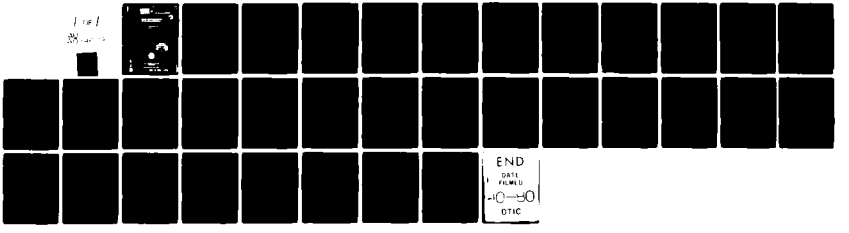
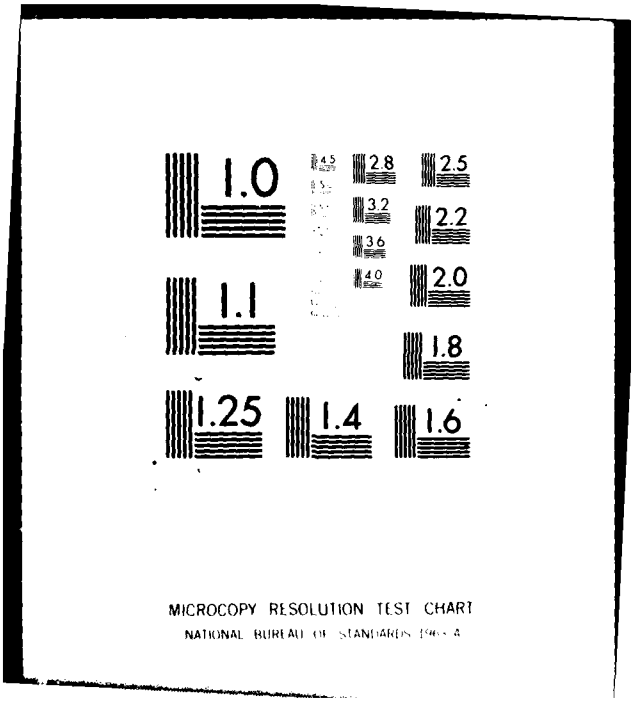
AD-A089 673 NAVAL RESEARCH LAB WASHINGTON DC F/G 9/2
A SCALING AND PLOTTING ROUTINE FOR TWO DIMENSIONAL DATA.(U)
SEP 80 T R O'NEAL
UNCLASSIFIED NRL-MR-4351 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

AD A08

(9) Memorandum Report

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(14) REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
(1) NRL-MR-NRL Memorandum Report 4351	AD-A089 673	

4. TITLE (and Subtitle)
(6) A SCALING AND PLOTTING ROUTINE FOR TWO DIMENSIONAL DATA

5. TYPE OF REPORT & PERIOD COVERED
Interim report on a continuing NRL problem.

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
(10) Terry R. O'Neal

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Naval Research Laboratory
Washington, D.C. 20375

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
61153N; RR-0024-03-41;
61-0027-0-0

11. CONTROLLING OFFICE NAME AND ADDRESS
(11) 30

12. REPORT DATE
September 1980

13. NUMBER OF PAGES
33

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)
(12) 34

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited.
(15) RR0024103 (17) RR002403

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)
Hewlett-Packard HP 1000 Computer
Graphics
Scaling
Plotting

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
This report provides a general plotting routine to express x, y type tabular data in a 2 dimensional graphical form. The program contains a scaling routine to maximize plot coverage and calculate axis length and tick marks. The routines are written in Fortran IV computer language and designed to run on a Hewlett-Packard HP 1000 computer system with Graphics 1000 plotting software and a graphics terminal or plotter.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

251950

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

CONTENTS

INTRODUCTION	1
FEATURES OF PROGRAM	2
COMPUTER AND SYSTEM CONFIGURATION	3
USER CHANGES	3
PROGRAM BACKGROUND	3
CAUTIONS	4
INPUT	4
LOADING PROCEDURE ON HP 1000	5
PROGRAM SOURCE AVAILABILITY	5
ERROR MESSAGES	5
TESTING OF PROGRAM	6
FLOWCHARTS	6
EXAMPLES	6
EXECUTION OF PROGRAM	6
SUBROUTINES	7
PLOTTER SET-UP	8
PLOTTING ACCURACY	8
MNEMONIC LIST	9
FORMULAS	11
FLOW CHARTS	12
EXAMPLES	15
REFERENCES	20
ACKNOWLEDGMENTS	21
APPENDIX A — Sample Dialogue of Program Execution	22
APPENDIX B — Program Listings	23

A SCALING AND PLOTTING ROUTINE FOR TWO DIMENSIONAL DATA

Introduction

The recent advent of more reliable, more accurate and faster plotting and graphics output devices and recent attempts by computer manufacturers to standardize graphics plotting software (Core System[1] graphics standard, HP Graphics-1000, etc.) have required new application software development of general scientific data-plotting routines. Our specific needs in chemical experimentation require the capability for plotting of a wide variety of X vs. Y data types and magnitudes, with and without "error bar"-type error limits on each data point. In addition, it is often necessary to do linear least squares calculations on these same data and plot a least squares regression line on the same plot for visual indication of linearity, scatter and goodness-of-fit. Finally, the capability to provide journal-ready plots to eliminate the need for the user to make decisions about scaling is highly desirable.

The program GPLOT satisfies these basic requirements and contains such additional capabilities as multiple data sets on a single graph, multi-colored plots and variable origin starting location. The main routine does the plotting; two subroutines are used to scale the data and calculate a least squares regression. All three routines are written in Fortran IV[2]. This software was developed to provide maximum flexibility with a minimum of effort by the user. It is designed to run on a Hewlett-Packard 1000 computer system under an RTE IV operating system and makes use of the device-independent features of Hewlett-Packard's Graphics-1000 Software as well as some H.P. extensions to standard Fortran IV.

This document is designed to serve the purposes of a users' guide and an operations manual, and to provide sufficient documentation for program maintenance.

Manuscript submitted July 31, 1980.

Features of Program

1. Automatic scaling.
2. Least squares line with slope, Y-intercept and standard deviations of each.
3. Error bars.
4. Multiple plots with different symbols on same graph.
5. Data source can be from disc file, cartridge tape or typed in from the keyboard of a terminal.
6. Output can be on graphics terminal or plotter.
7. Plots can be line, symbols, or symbols connected by line.
8. Axis labeling and title are entered from terminal.
9. Origin can be determined automatically or forced to start at (0,0).
10. Tick marks are labeled.
11. Multiple color plots with automatic pen changing.
12. A manual scaling option can plot multiple data sets, which have different maximum and minimum values, on a single set of axes.
13. The routine can handle up to 100 points per data set.

Computer and System Configuration

This program could be modified to run on many different computers with different plot packages; however, this routine was written for the Hewlett-Packard HP 1000 family of computers using Graphics-1000 (HP 92840A graphics plotting software). The operating system is RTE IVB with updated software revision code 2001. A graphics device, HP 2648A graphics terminal and/or HP 9872B plotter, is needed to do the plotting. Older versions of the RTE operating system and earlier plotters can be used so long as graphics 1000 limitations are satisfied. EMA and spooling features were not utilized. This routine requires a 24K-word partition to run.

User Changes

This program was tested on a system probably configured differently from that of the user. It is the responsibility of the user to make changes to logical unit assignments to implement this program on his system. Changes will need to be made to lines 21 through 26, 31, 226 and 294 of program GPLOT. LUG is the logical unit number of the graphics device. ID is the identification number assigned by the device link table to a graphics logical device. Line 44 of program GPLOT does not permit a logical unit number greater than 30. This limit was set to avoid input errors and may need to be changed by the user.

Program Background

The Hewlett-Packard plot package (HP 92840A graphics plotting software) does not contain a scaling routine. A major reason for development of this routine was the absence of useable existing scale routines. Most CalComp-type routines provide very limited plotting capabilities. The HP CalComp scale routine requires placement of a tick mark at one inch intervals on each axis; this requirement is very restrictive in maximizing plot size. A more generally applicable routine was required, which would provide a minimum of unused plotting area for a wide variety of data types.

Cautions

The routine GPLOT and the Graphics-1000 routines which it calls are not omniscient. It is possible to enter responses or data which will produce unpredictable results. It is also possible to use portions of the program or to modify it in such a way as to produce other-than-desirable results.

The subroutine SCALE transforms the array X into exponential notation, storing the exponent in IEXP. This process changes the array X by some factor of 10. If the user calls this subroutine from any other program he should be aware that any value passed back to the calling routine could be changed. For this reason X and IEXP must be used together.

Certain data characteristics, such as failing to separate the data on an input line with commas or spaces, will not be detected as an error by this program.

Some errors in input data can generate error conditions in Hewlett-Packard library routines. As written, the program allows these to be printed on the standard list device (logical unit 6). To avoid this, the user must supply his own error routine as described in ERØ.E[5].

Input

Data can be input to the program from a disc file, cartridge or paper tape or typed in from the keyboard of a terminal. Disc files must be type 3 or 4[6]. The program can handle an array up to 100 data points. The format is X, Y, DELX, DELY using free field input.[2] The X and Y pair is the position of the point along the X and Y axis respectively. The optional pair DELX and DELY is the standard deviation in X and Y. These values are used for drawing error bars and will be doubled and scaled to plotter units to provide the horizontal or vertical separation of error bars. All numeric input data must be in the range of 10^{-38} to 10^{38}.

Loading Procedure on HP 1000

The loader must be loaded as a large background program (type 4) and will require a size of 24 pages to accommodate the Graphics-1000 routines. After compiling &GPLOT, &SCALE and &LSREG, execute the loader interactively as follows:

```
RU,LOADR,,,,LB
RE,%GPLOT
RE,%SCALE
RE,%LSREG
RE,%DLTBL
SE,%GPS
END
```

where %GPS is the Graphics-1000 library file created when Graphics-1000 was loaded.[3] Alternatively, the source version available from the author contains a loader command file which can be used. The loaded program will occupy approximately 23K words of memory. It would be possible to decrease the main program space somewhat by using EMA[4] for array space. However, the bulk of this 23K words is required for Graphics-1000 routines.

Program Source Availability

A program source is available from the author on a user-supplied Hewlett-Packard 264X-type cartridge tape. This tape contains 5 files. The first file is a description of what is on the tape. The second file is a command file which may be used by the loader to load the programs. The third file is the source for program GPLOT. The fourth file is the source for subroutine SCALE. The fifth file is the source for subroutine LSREG.

Error Messages

Program GPLOT checks for errors that could occur when reading a disc resident file. Messages to the user are sent with the name of the file if a problem is encountered. The program has several built in checks to catch typing errors by the user.

Testing of Program

The author has used a wide variety of data types, multiple files, and several input sources to debug this program. In addition, the program has been used extensively by six individuals with different applications and occasionally by about twenty others.

Flowcharts

Figures 1 and 2 show the logic flow in the main program GPLOT and in the subroutine SCALE. The calculations performed by subroutine LSREG appear in the Formulas section of this report. A complete program listing appears in the Appendix.

Examples

Figures 3, 4, and 5 are examples of different types of plots that GPLOT can produce. Figure 3 is a symbol-only plot with error bars and a least squares line. Figure 4 is a line plot of 51 data points. Figure 5 is a plot of 3 data sets using different symbols for each data set. These three figures demonstrate the minimization of unused plot area resulting from the algorithm in SCALE.

Execution of Program

The execution of GPLOT can be performed by supplying terminal and graphics device Logical Unit Numbers (LU and LUG) via the run string parameters. These parameters (globals) are retrieved by a call to the Hewlett-Packard routine RMPAR in GPLOT. If these parameters are not supplied, the program retrieves LU and prompts the user for the graphics device LUG. Device selection is to be made between the plotter, for a hard copy, or the graphics terminal. The data source is from a disc file, cartridge or paper tape or entered via the keyboard of a terminal. If the source is a disc file the name of the file is requested. If the source is other than the disc a logical unit number is requested. One

of three types of plots can be selected; a straight line connecting each point, a symbol at each point or a symbol at each point with connecting lines. Labels are entered from the users terminal, any ASCII character is permitted (capital and lower case letters, numbers and symbols). The X and Y axis labels cannot exceed 30 characters and the title of the plot cannot exceed 40 characters in length; characters beyond these limits are ignored. The starting position of the origin can be forced to start at X = 0 and Y = 0 or the user can let the scale routine determine an origin that will maximize the size of the plot *vs.* the size of the axes. Scaling is performed automatically by the SCALE subroutine, however a manual override is provided. The manual scaling mode is used to increase the limits between the maximum and minimum values of an axis. This feature is necessary when plotting multiple plots on one graph when the maximum and minimum values of all of the data sets are not within one data set. When plotting multiple data sets the first set plotted must have the smallest and largest values of all of the data sets. If this condition cannot be met the user must specify manual scaling and enter minimum and maximum values of the entire set of data. A least squares line can be drawn on the plot with slope, y-intercept and respective standard deviations printed on the users terminal or printer. Error bars can be drawn around each point provided requirements in the input section of this paper have been met. Error bars that are small enough to distort the symbol printed at a data point are suppressed and a message is printed on the users terminal of this action. Six or less plots can be made using the same set of axes provided all data sets fall within the limits of the first data set plotted. The user has the option to make pen color changes when doing multiple plots on the same axis.

Subroutines

The subroutine SCALE uses a table look-up method, based on the difference between maximum and minimum values, to determine axis scaling and number of tick marks to be placed on each axis. The SCALE routine uses an algorithm that shifts decimal points to increase numbers that are less than one and

decrease numbers that are greater than 1000. This method can handle a difference of any order of magnitude and the data will be scaled to cover a minimum of 50% of each axis.

The subroutine LSREG does a least squares linear regression calculation including the standard deviation of the slope and y intercept.

Plotter Setup

The HP 9872B plotter is used when hardcopy results of GPLOT are required. The plot generated by GPLOT is designed to fit on standard 8½ X 11 inch paper with adequate margins for publication. In order to center the plot the paper must be placed ½ inch from the left side and 1 inch from the bottom lower left corner of the plotter bed. This displacement is necessary because the HP 9872B plotter with advance option OFF places the lower left corner of the plotting window at (520, 380) instead of (0,0). For multicolored plots pen placement is as follows; pen 1 is black, 2 is red, 3 is green, and 4 is blue. If the plot is to be one color that color pen must be in pen holder 1.

Plotting Accuracy

The HP 9872B plotter is divided into plotter units where one unit = 0.025 *mm*. This is to say that the overall resolution of the plot is one part in N where N is the number of plotter units occupied in the X or Y direction. The X axis is 7.7 inches (7,823 plotter units) and the Y axis is 5.25 inches (5,334 plotter units). The plotter resolution is one part in 7,823 in the X direction and one part in 5,334 in the Y direction. In the worst case of plot coverage *vs.* axis length (50%) the plot resolution would be one half of the plotter resolution.

Mnemonic List

BARX - Height of X error bar tick mark
BARY - Height of Y error bar tick mark
DELX - Experimental error in X (standard deviation)
DELY - Experimental error in Y (standard deviation)
DIF - Difference between max and min
HH - Half height in character cells
HW - Half width in character cells
IAX - Label of X axis (30 characters max)
IAY - Label of Y axis (30 characters max)
IBAR - Type of error bars
ID - Identification number
IDONE - Check for termination or multiple plot
IEXP - Exponent of base 10 in X data scale
IFMT - Source of data
IHED - Title of plot (40 characters max)
ILINE - Check for least squares line
IPEN - Pen number of plotter
ISCAL - Set to zero for automatic scaling, 1 for manual
IZERO - Set to zero to force origin to start at (0,0)
JCHAR - Character to be plotted at each data point
JEXP - Exponent of base 10 in Y data scale
JJ - Type of plot
KK - Number of plots on same axis
LU - Logical unit
LUG - Logical unit of graphics device
LUT - Logical unit of cartridge tape or keyboard
NAME - Name of data file
NOBAR - Check to see if error bar was too small to plot
NP - Number of points
S1 - Standard deviation of slope
S2 - Standard deviation of Y intercept
SLOPE - Slope of least squares line
SXTIC - Interval between X tick marks
SYTIC - Interval between Y tick marks

X - Displacement along X axis

XBAR1 - Distance to right of character of X error bar

XBAR2 - Distance to left of character of X error bar

XEND - X value at end of least squares line

XMAX - Maximum value of X

XMIN - Minimum value of X

XST - X value at start of least squares line

XTIC - Number of tick marks on X axis

Y - Displacement along Y axis

YBAR1 - Distance above character of Y error bar

YBAR2 - Distance below character of Y error bar

YEND - Y value at end of least squares line

YINT - Y intercept of least squares line

YMAX - Maximum value of Y

YMIN - Minimum value of Y

YST - Y value at start of least squares line

YTIC - Number of tick marks on Y axis

Formulas

The following formulas were used in subroutine LSREG:

$$\text{Slope} = \frac{(DIF)(\Sigma X \Sigma Y) - (\Sigma X \Sigma Y)}{(DIF)(\Sigma X)^2 - (\Sigma X)^2}$$

$$\text{Y-Intercept} = \frac{\Sigma Y - (SLOPE)(\Sigma X)}{DIF}$$

$$\text{Standard Deviation of Slope} = \sqrt{\frac{Y^2 - \frac{(\Sigma Y)^2}{DIF} - \frac{\left((\Sigma X \Sigma Y) - \frac{\Sigma X \Sigma Y}{DIF}\right)^2}{\Sigma X^2 - \frac{(\Sigma X)^2}{DIF}}}{(DIF - 2)(\Sigma X^2) - \frac{(\Sigma X)^2}{DIF}}}$$

$$\text{Standard Deviation of Y Intercept} = \sqrt{(\text{SD of SLOPE})^2 \frac{\Sigma X^2}{DIF}}$$

DIF = the interval over which the calculation is computed (FROM-TO+1)

ΣX = summation of X values

ΣY = summation of Y values

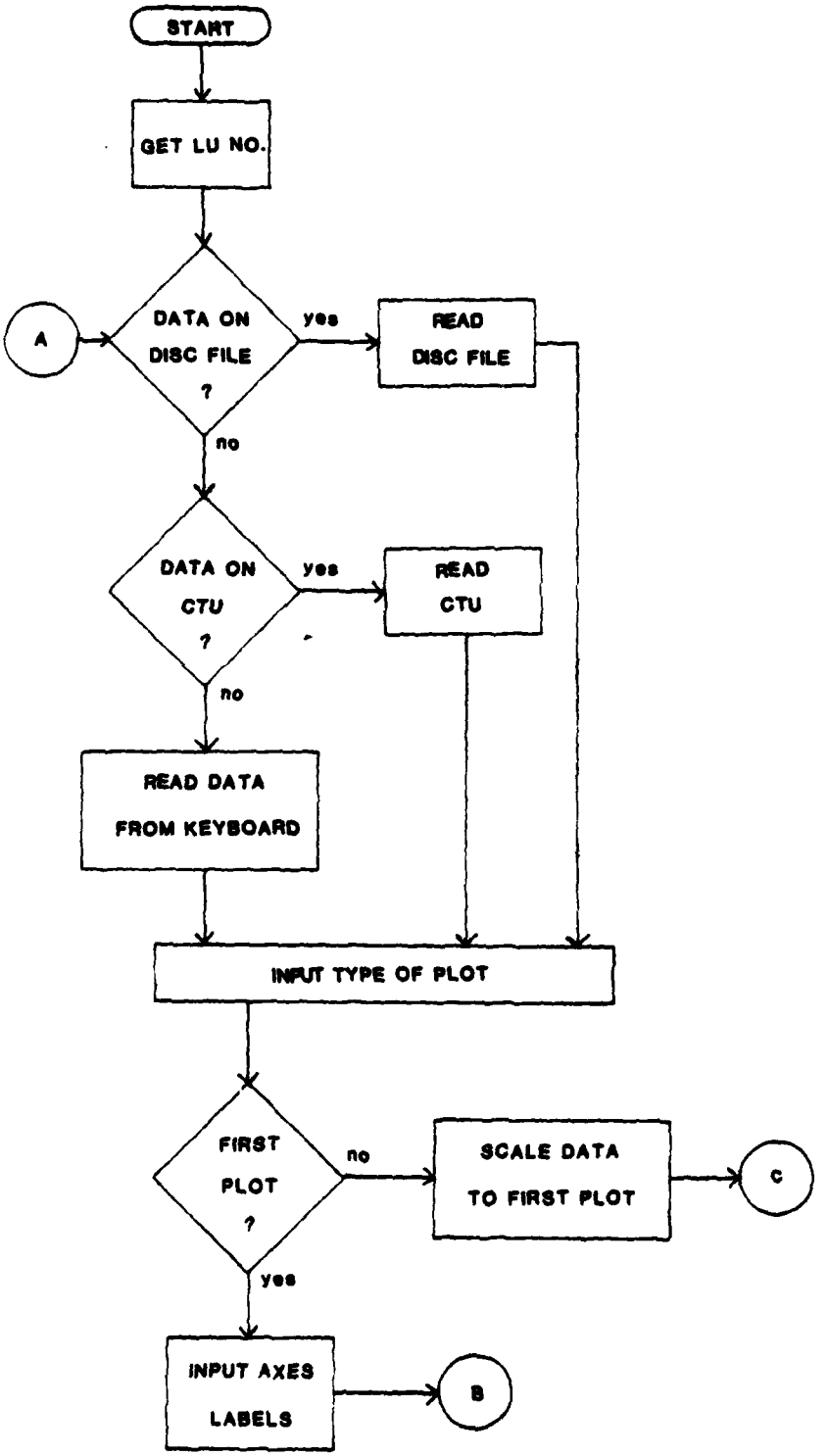


Fig. 1 — Program GPLOT (Continues)

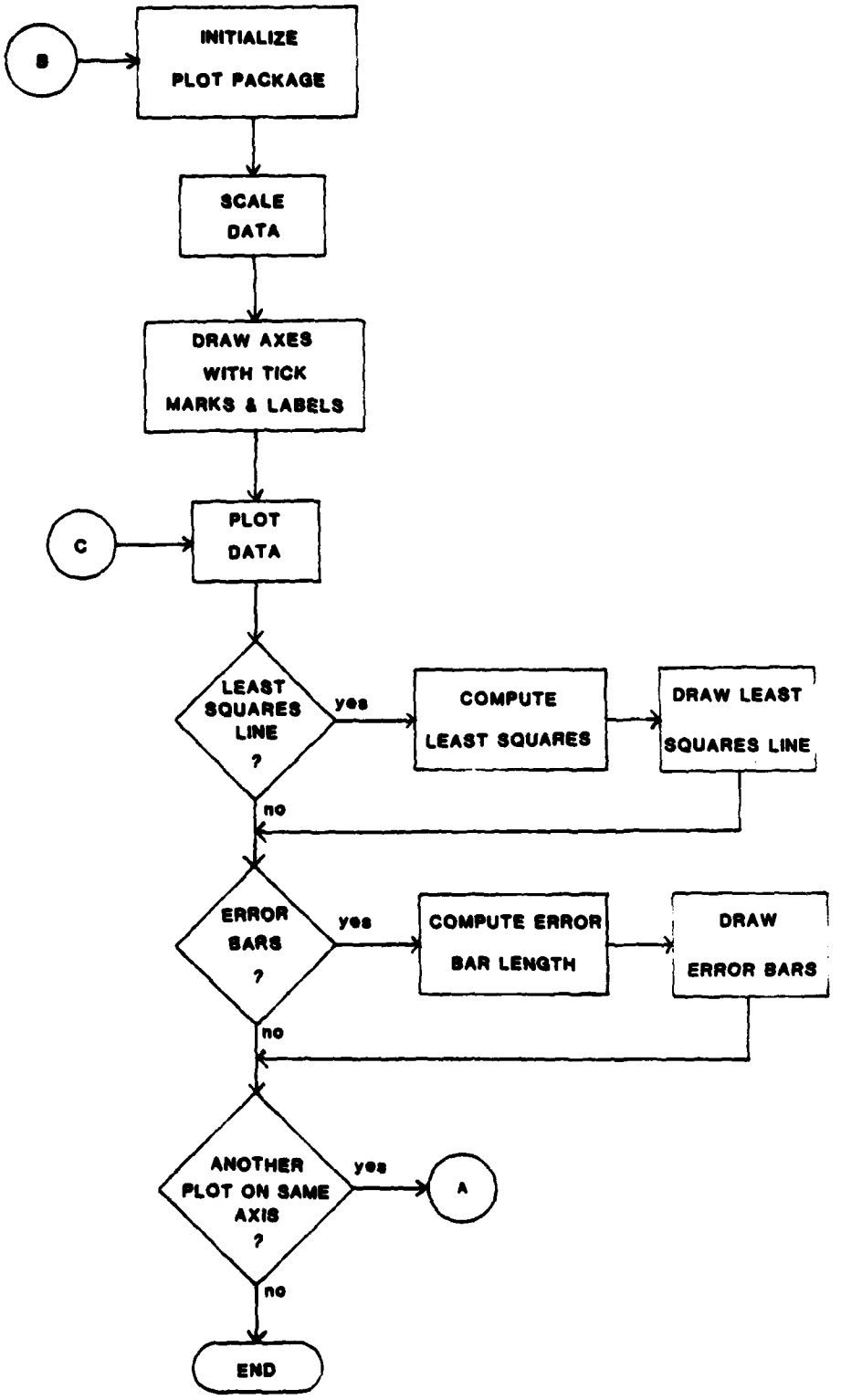


Fig. 1 (Continued) — Program GPLOT

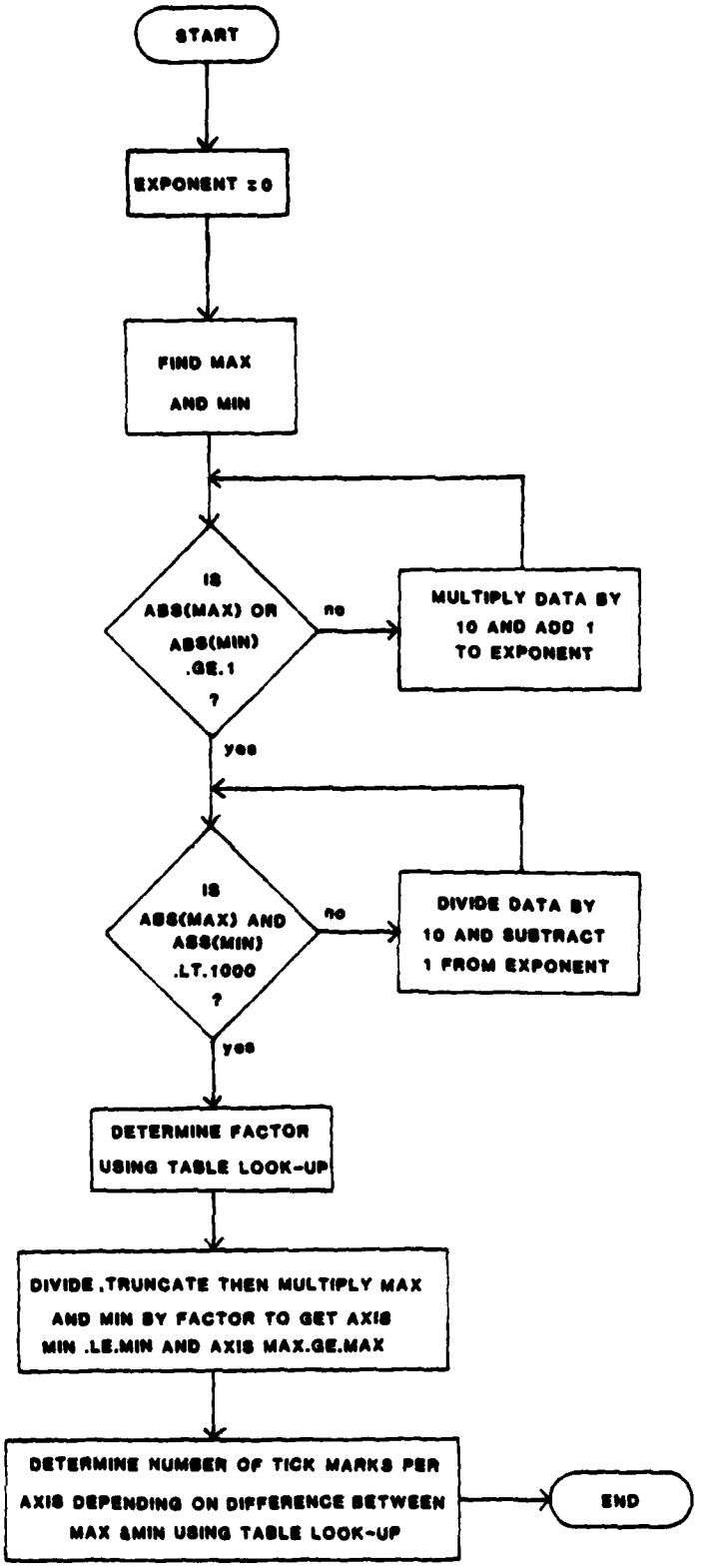


Fig. 2 — Program scale

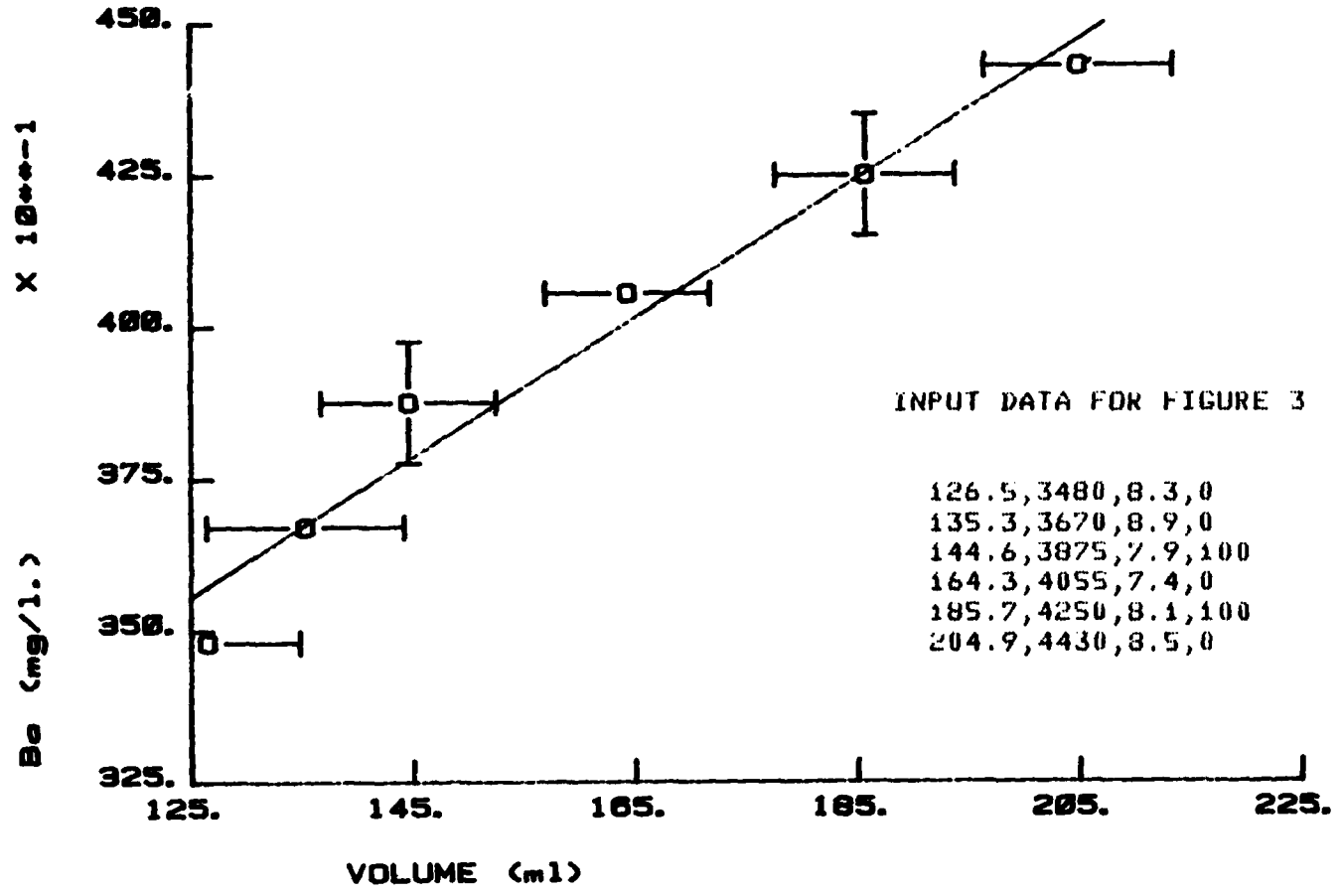


Fig. 3 — Barium present in wash water

INPUT FOR FIGURE 4

391.52234	.34266162
397.41956	.44873518
403.31677	.57228398
409.21399	.69693267
415.11115	.77269948
421.00836	.81021631
426.90558	.82622504
432.80280	.81192720
438.70001	.77404380
444.59723	.70316505
450.49445	.62104368
456.39166	.52572405
462.28888	.43847001
468.18610	.35781497
474.08331	.29903460
479.98053	.26176220
485.87775	.24086520
491.77496	.23149 45
497.67218	.22974458
503.56940	.23414394
509.46655	.24098742
515.36377	.24440914
521.26099	.25149703
527.15820	.26249540
533.05542	.27337164
538.95264	.28766954
544.84985	.31137723
550.74707	.33948427
556.64429	.36918002
562.54150	.40791887
568.43872	.43468165
574.33594	.45912260
580.23315	.47635341
586.13037	.48246366
592.02759	.47574240
597.92480	.45777833
603.82202	.43223757
609.71924	.39129901
615.61646	.34864962
621.51367	.30037886
627.41089	.25235242
633.30811	.19369423
639.20532	.14542344
645.10254	.10436270
650.99976	.72345108E-01
656.89697	.51814735E-01
662.79419	.28718073E-01
668.69128	.16253207E-01
674.58850	.91653429E-02
680.48572	.47657781E-02
686.38293	.47657781E-02

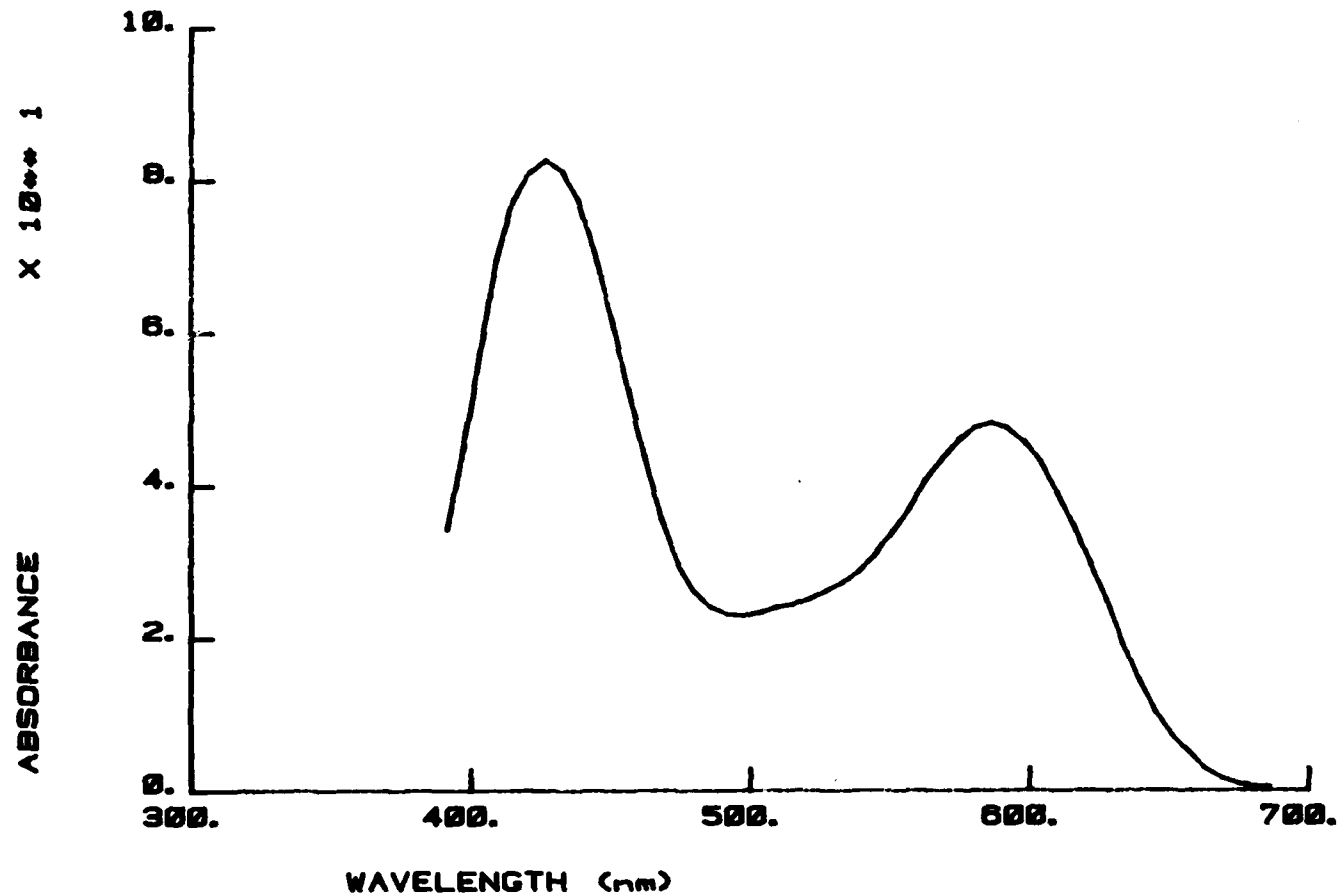


Fig. 4 — Plot of a uv spectrum

INPUT DATA FOR FIGURE 5

PLOT 1

5810,82.4
9140,127.3
12240,163.5
15580,209.1
19130,256.8

PLOT 2

8450,97.2
11720,130.2
14690,167.1
17860,199.6
18810,231.6

PLOT 3

7108,188.5
11500,220.1
16040,253.8

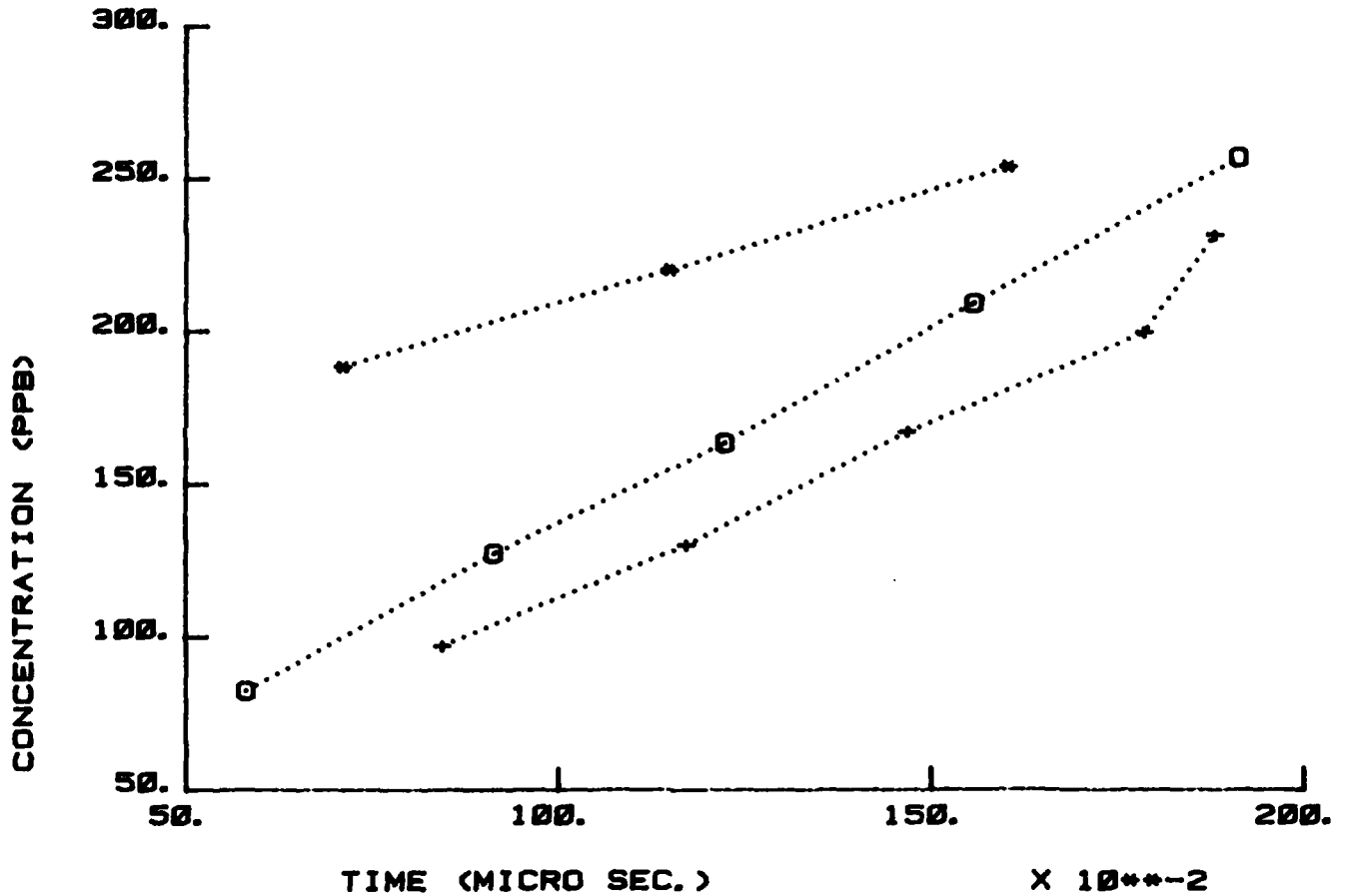


Fig. 5 — Sample # R3267

References

1. A proposed graphics standard developed by the Graphics Standard Planning Committee of the Association of Computing Machinery. AMC COMP. SURVEYS, 10, 363-502, 1978.

2. Hewlett-Packard, RTE Fortran IV Reference Manual, January 1980, p. 7-9.

3. Hewlett-Packard, HP 92840A Graphics Plotting Software User's Manual, January 1980. p. 5-1.

4. Hewlett-Packard, HP 92068A RTE-IVB Terminal User's Reference Manual, January 1980, pp. 1-2, 4-56.

5. Hewlett-Packard, RTE Relocatable Library Reference Manual, December 1978, p. 3-7.

6. Hewlett-Packard, HP 92068A RTE-IVB Terminal User's Reference Manual, January 1980, p. 3-7.

Acknowledgments

The author would like to thank Dr. John C. Cooper, Naval Research Laboratory, for his influence and advice in writing and debugging this program.

The author would also like to thank Dr. Noel H. Turner, Naval Research Laboratory, for helpful discussions in developing the algorithm which became the basis for the scale routine.

Appendix A

SAMPLE DIALOGUE OF PROGRAM EXECUTION

```
:RU,GPLOT
 OUTPUT ON GRAPHICS TERMINAL TYPE 0
        ON PLOTTER           TYPE 1
1
   ***   SOURCE OF DATA   ***
 DISC FILE           TYPE 0
 KEYBOADRD OR TAPE   TYPE 1
0
 ENTER NAME OF DATA FILE
 TK01
   ***   TYPE OF PLOT   ***
 LINE PLOT                          TYPE 1
 SYMBOLS CONNECTED WITH LINES       TYPE 2
 SYMBOLS PLOT                       TYPE 3
1
 ENTER X-LABEL,Y-LABEL & TITLE ON 3 SEPARATE LINES
WAVELENGTH (nm)
ABSORBANCE
FIGURE 4 : PLOT OF A UV SPECTRUM
   ***   ORIGIN LOCATION   ***
 TO FORCE ORIGIN TO START AT (0,0)     TYPE 0
 TO LET SCALE DETERMINE ORIGIN         TYPE 1
1
   ***   SCALING   ***
 AUTOMATIC              TYPE 0
 MANUAL                 TYPE 1
0
 LEAST SQUARES LINE ?
                        NO  TYPE 0
                        YES TYPE 1
0
 ERROR BARS ?
             NONE          TYPE 0
             X             TYPE 1
             Y             TYPE 2
             X & Y         TYPE 3
0
 TO EXIT                                          TYPE 0
 IF YOU WANT ANOTHER PLOT ON SAME AXIS
 AND ALL X & Y VALUES ARE WITHIN THE
 SCALES OF THE FIRST PLOT                         TYPE 1
0
```

Appendix B
PROGRAM LISTINGS

```
FTN4,L
      PROGRAM GPLOT(4,79)
C     GENERALIZED GRAPH PLOTTING ROUTINE FOR THE HP 9872B PLOTTER
C     MAXIMUM 100 PTS. ALLOWED, X AND Y MAY BE INPUT FROM A DISC
C     FILE, CARTRIDGE TAPE OR KEYBOARD FORMATTED IN (X,Y) PAIRS
C     FOR EXPERIMENTAL ERROR BARS FORMAT IS (X,Y,DELTAX,DELTAY).
C
      DIMENSION IDCB(144),NAME(3),IAX(15),IAY(15),IHED(20),IBUF(40)
      DIMENSION X(100),Y(100),IPRAM(5),IGCB(192),IOBUF(20),JCHAR(6)
      DIMENSION DELX(100),DELY(100)
      DATA JCHAR/1HO,1H+,1H*,1HX,1H#,1H$/
      CALL RMPAR(IPRAM)
      LU=IPRAM(1)
      LUG=IPRAM(2)
      IF(LUG.NE.0)GOTO 15
      WRITE(LU,10)
10    FORMAT(" OUTPUT ON GRAPHICS TERMINAL TYPE 0",/,
     1       "         ON PLOTTER           TYPE 1")
      READ(LU,*)ILUG
C     DEFINE LU AND ID NUMBERS OF PLOTTER AND GRAPHICS TERMINAL
      LUG=24
      IF(ILUG.EQ.1)LUG=20
15    KK=0
      IF(LU.LE.0)LU=1
      ID=1
      IF(LUG.EQ.20)ID=2
C     TO CENTER CHARACTER SET HALF WIDTH & HALF HEIGHT
C     DEPENDING ON IF USING PLOTTER OR CRT
      HW=0.5
      HH=0.5
      IF(ID.EQ.1) GOTO 20
      HW=0.333
      HH=0.25
20    KK=KK+1
25    WRITE(LU,30)
30    FORMAT( "   ***   SOURCE OF DATA   ***",/
     1 " DISC FILE          TYPE 0",/
     2 " KEYBOADRD OR TAPE  TYPE 1")
      READ (LU,*) IFMT
      IF(IFMT.EQ.0) GOTO 60
35    WRITE(LU,40)
40    FORMAT(" ENTER LU NUMBER OF CTU OR KEYBOARD")
      READ(LU,*)LUT
      IF(LUT.GT.30) GOTO 35
45    WRITE(LU,50)
50    FORMAT(" ENTER NUMBER OF DATA POINTS (MAX=100)")
      READ(LU,*)NP
      IF(NP.GT.100) GOTO 45
      DO55 I=1,NP
55    READ (LUT,*) X(I),Y(I),DELX(I),DELY(I)
      GOTO 100
60    WRITE(LU,65)
65    FORMAT(" ENTER NAME OF DATA FILE")
      READ(LU,70)NAME
70    FORMAT(3A2)
```

```
      CALL OPEN(IDCB,IERR,NAME,3)
      IF(IERR.LT.0)WRITE(LU,95)NAME
75    FORMAT(" ERROR CODE = ",I4)
      IF(IERR.LT.0) GOTO 60
      K=1
80    DO 82 J=1,40
82    IBUF(J)=0
      CALL READF(IDCB,IERR,IBUF,40,LEN)
      IF(LEN.EQ.-1) GOTO 85
      IF(IERR.LT.0)GOTO 90
      CALL CODE
      READ(IBUF,*)X(K),Y(K),DELX(K),DELY(K)
      K=K+1
      GO TO 80
85    NP=K-1
      CALL CLOSE(IDCB,IERR)
      IF(IERR.LT.0)WRITE(LU,75)IERR
      GOTO 100
90    WRITE(LU,95)NAME
95    FORMAT(" FILE ",3A2," DOESN'T EXIST OR IS ALREADY OPEN")
      GOTO 25
C     IF DOING MULTIPLE PLOTS, SCALE DATA TO FIRST PLOT
100   IF(KK.EQ.1) GOTO 135
      DO130  K=1,NP
      IF(IEXP)105,115,110
105   X(K)=X(K)/(10.**IABS(IEXP))
      GOTO 115
110   X(K)=X(K)*(10.**IABS(IEXP))
115   IF(JEXP)120,130,125
120   Y(K)=Y(K)/(10.**IABS(JEXP))
      GOTO 130
125   Y(K)=Y(K)*(10.**IABS(JEXP))
130   CONTINUE
135   WRITE(LU,140)
140   FORMAT("   ***   TYPE OF PLOT   ***",/
     1 " LINE PLOT                        TYPE 1",/
     2 " SYMBOLS CONNECTED WITH LINES     TYPE 2",/
     3 " SYMBOLS PLOT                     TYPE 3",)
      READ(LU,*)JJ
C
C     READ LABELS AND LEGEND
C
      IF(KK.GT.1) GOTO 210
      WRITE(LU,145)
145   FORMAT(" ENTER X-LABEL,Y-LABEL & TITLE ON 3 SEPARATE LINES")
      READ(LU,150) IAX,IAY,IHED
150   FORMAT(15A2/,15A2/,20A2)
C
C     INITIALIZE PLOT PACKAGE & DEFINE VIEWPORT
C
      CALL PLOTR(IGCB,ID,4,LUG,IOBUF,20)
      CALL SETAR(IGCB,2.0)
      CALL VIEWP(IGCB,0.,135.,0.,100.)
C
C     SCALE DATA & DEFINE WINDOW
```

```
C
      WRITE(LU,155)
155   FORMAT("   ***   ORIGIN LOCATION   ***",/
     1 " TO FORCE ORIGIN TO START AT (0,0)    TYPE 0",/
     2 " TO LET SCALE DETERMINE ORIGIN        TYPE 1")
      READ(LU,*)IZERO
      WRITE(LU,160)
160   FORMAT("   ***   SCALING   ***",/
     1       " AUTOMATIC             TYPE 0",/
     2       " MANUAL                TYPE 1")
      READ(LU,*)ISCAL
      IF(ISCAL.EQ.0)GOTO 170
      WRITE(LU,165)
165   FORMAT(" ENTER XMIN,XMAX,YMIN,YMAX")
      READ(LU,*)XMIN,XMAX,YMIN,YMAX
170   CALL PEN(IGCB,1)
      CALL SCALE (X,NP,XMIN,XMAX,XTIC,IZERO,IEXP,LU,ISCAL)
      CALL SCALE (Y,NP,YMIN,YMAX,YTIC,IZERO,JEXP,LU,ISCAL)
      CALL WINDW(IGCB,0.,150.,0.,100.)
C
C     SPECIFY CHARACTER SIZE
C
      CALL CSIZE(IGCB,3.)
C
C     DRAW AXES AND LABELS
C
      CALL FXD(IGCB,0)
      SXTIC=ABS(XMAX-XMIN)/XTIC
      SYTIC=ABS(YMAX-YMIN)/YTIC
      CALL MOVE(IGCB,35.,1.)
      IF(IEXP NE.0) GOTO 180
      CALL LABEL(IGCB)
      WRITE(LUG,175)IAX
175   FORMAT(15A2)
      GOTO 190
180   CALL LABEL(IGCB)
      WRITE(LUG,185)IAX,IEXP
185   FORMAT(15A2," X 10**",I2)
190   CALL MOVE(IGCB,3.,12.)
      CALL LDIR(IGCB,+1.57)
      IF(JEXP.NE.0) GOTO 195
      CALL LABEL(IGCB)
      WRITE(LUG,175)IAY
      GOTO 200
195   CALL LABEL(IGCB)
      WRITE(LUG,185)IAY,JEXP
200   CALL MOVE(IGCB,40.,90.)
      CALL LDIR(IGCB,0.)
      CALL LABEL(IGCB)
      WRITE(LUG,205)IHED
205   FORMAT(20A2)
      CALL VIEWP(IGCB,17.,120.,10.,80.)
      CALL WINDW(IGCB,XMIN,XMAX,YMIN,YMAX)
      CALL LAXES(IGCB,-SXTIC,SYTIC,XMIN,YMIN)
C
```

```
C       PLOT DATA POINTS
C
210     MM=0
        IF(JJ.EQ.2)MM=1
        CALL LINE(IGCB,MM)
        DO220 K=1,NP
        IF(K.EQ.1.OR.JJ.EQ.3) GOTO 215
        CALL DRAW(IGCB,X(K),Y(K))
C       MOVE TO (X,Y),CENTER CHAR.,PLOT CHAR.,MOVE "CP" BACK TO (X,Y)
215     CALL MOVE(IGCB,X(K),Y(K))
        IF(JJ.EQ.1) GOTO 220
        CALL CPLOT(IGCB,-HW,-HH,-2)
        CALL LABEL(IGCB)
        WRITE(LUG,225)JCHAR(KK)
        CALL MOVE(IGCB,X(K),Y(K))
220     CONTINUE
225     FORMAT(1A1)
        CALL PENUP(IGCB)
        CALL LINE(IGCB,0)
        WRITE(LU,230)
230     FORMAT(" LEAST SQUARES LINE ?",/,22X,"NO  TYPE 0",/,
       1        22X,"YES TYPE 1")
        READ(LU,*) ILINE
        WRITE(LU,235)
235     FORMAT(" ERROR BARS ?",/,
       1        "              NONE          TYPE 0",/,
       2        "              X             TYPE 1",/,
       3        "              Y             TYPE 2",/,
       4        "              X & Y         TYPE 3")
        READ(LU,*)IBAR
        IF(ILINE.EQ.0) GOTO 270
C
C       CALCULATE START AND END POINTS OF LEAST SQUARES LINE
C
        CALL LSREG(X,Y,1,NP,SLOPE,YINT,S1,S2)
        XST=XMIN
        YST=YINT
        YST=(SLOPE*XMIN)+YINT
240     IF(YINT.LE.YMAX) GO TO 245
        XST=(YMAX-YINT)/SLOPE
        YST=YMAX
245     IF(YINT.GE.YMIN) GO TO 250
        XST=(YMIN-YINT)/SLOPE
        YST=YMIN
250     XEND=XMAX
        YEND=(SLOPE*XMAX)+YINT
        IF(YEND.LE.YMAX) GO TO 255
        XEND=(YMAX-YINT)/SLOPE
        YEND=YMAX
255     IF(YEND.GE.YMIN) GO TO 260
        YEND=YMIN
        XEND=(YMIN-YINT)/SLOPE
C
C       PLOT LEAST SQUARES LINE
C
```

```
260   IF(ILINE.EQ.0) GOTO 270
      CALL MOVE(IGCB,XST,YST)
      CALL DRAW(IGCB,XEND,YEND)
      CALL PENUP(IGCB)
      LUX=LU
      IF(ID.EQ.2) LUX=6
      WRITE(LUX,265)SLOPE,S1,YINT,S2
265   FORMAT(" SLOPE="G12.5", +-"G12.6/" Y-int="G12.5", +-"G12.6)
C
C     COMPUTE AND PLOT ERROR BARS
C
270   IF(IBAR.EQ.0) GOTO 335
C     SCALE DEVIATIONS BY VALUE OF EXPONENT
      DO300  K=1,NP
      IF(IEXP)275,285,280
275   DELX(K)=DELX(K)/(10.**IABS(IEXP))
      GOTO 285
280   DELX(K)=DELX(K)*(10.**IABS(IEXP))
285   IF(JEXP)290,300,295
290   DELY(K)=DELY(K)/(10.**IABS(JEXP))
      GOTO 300
295   DELY(K)=DELY(K)*(10.**IABS(JEXP))
300   CONTINUE
      NOBAR=0
      BARX=ABS(XMAX-XMIN)/100.
      BARY=ABS(YMAX-YMIN)/67.
      DO325  K=1,NP
      XBAR1=X(K)-DELX(K)
      XBAR2=X(K)+DELX(K)
      YBAR1=Y(K)-DELY(K)
      YBAR2=Y(K)+DELY(K)
      GOTO(310,320,310) IBAR
305   GOTO 335
C     DRAW X ERROR BARS
C     IF ERROR IS SMALL DON'T DRAW ERROR BARS
310   IF(DELX(K).LE.BARX) NOBAR=1
      IF(DELX(K).LE.BARX) GOTO 315
      CALL MOVE(IGCB,X(K),Y(K))
      CALL CPLOT(IGCB,0.8,0.,-2)
      CALL DRAW(IGCB,XBAR2,Y(K))
      CALL MOVE(IGCB,XBAR2,Y(K)+BARY)
      CALL DRAW(IGCB,XBAR2,Y(K)-BARY)
      CALL MOVE(IGCB,X(K),Y(K))
      CALL CPLOT(IGCB,-0.8,0.,-2)
      CALL DRAW(IGCB,XBAR1,Y(K))
      CALL MOVE(IGCB,XBAR1,Y(K)+BARY)
      CALL DRAW(IGCB,XBAR1,Y(K)-BARY)
315   IF(IBAR.EQ.1) GOTO 325
C     DRAW Y ERROR BARS
320   IF(DELY(K).LE.BARY) NOBAR=1
      IF(DELY(K).LE.BARY) GOTO 325
      CALL MOVE(IGCB,X(K),Y(K))
      CALL CPLOT(IGCB,0.,0.5,-2)
      CALL DRAW(IGCB,X(K),YBAR2)
      CALL MOVE(IGCB,X(K)-BARX,YBAR2)
```

```
      CALL DRAW(IGCB,X(K)+BARX,YBAR2)
      CALL MOVE(IGCB,X(K),Y(K))
      CALL CPLOT(IGCB,0.,-0.5,-2)
      CALL DRAW(IGCB,X(K),YBAR1)
      CALL MOVE(IGCB,X(K)-BARX,YBAR1)
      CALL DRAW(IGCB,X(K)+BARX,YBAR1)
325   CONTINUE
      CALL PENUP(IGCB)
      IF(NOBAR.EQ.1) WRITE(LU,330)
330   FORMAT(" *** SOME ERROR BARS WERE TOO SMALL TO PLOT ***")
335   WRITE(LU,340)
340   FORMAT(" TO EXIT                                          TYPE 0",/,
     1       " IF YOU WANT ANOTHER PLOT ON SAME AXIS",/,
     2       " AND ALL X & Y VALUES ARE WITHIN THE",/,
     3       " SCALES OF THE FIRST PLOT                        TYPE 1")
      READ(LU,*)IDONE
      IF(KK.EQ.6) GOTO 360
C     SELECT PEN COLOR IF USING PLOTTER
      IF(ID.NE.2.OR.IDONE.NE.1) GOTO 355
345   WRITE(LU,350)
350   FORMAT("  *** PEN COLOR SELECTION ***",/,
     1       "    BLACK              TYPE 1",/,
     2       "    RED                TYPE 2",/,
     3       "    GREEN              TYPE 3",/,
     4       "    BLUE               TYPE 4")
      READ(LU,*) IPEN
      IF(IPEN.LT.1.OR.IPEN.GT.4) GOTO 345
      CALL PEN(IGCB,IPEN)
355   IF(IDONE.EQ.1) GOTO 20
360   CALL PEN(IGCB,0)
      CALL PLOTR(IGCB,ID,0)
      END
```

```
FTN4,L
      SUBROUTINE SCALE (X,NP,XMIN,XMAX,TIC,IZERO,IEXP,LU,ISCAL)
C     THIS ROUTINE COMPUTES MAX & MIN VALUES,SCALES DATA TO
C     E FORMAT AND DETERMINES THE NUMBER OF TICK MARKS PER AXIS
C     X      - ARRAY TO BE SCALED
C     NP     - NUMBER OF POINTS IN ARRAY X
C     XMIN   - MINIMUN VALUE OF X
C     XMAX   - MAXIMUM VALUE OF X
C     TIC    - NUMBER OF TICK MARKS ON AXIS
C     IZERO  - SET TO 0 TO FORCE ORIGIN TO (0,0),NORMALLY = 1
C     IEXP   - EXPONENT OF BASE 10 TO WHICH X IS RAISED
C     LU     - LOGICAL UNIT NUMBER OF TERMINAL
C     ISCAL  - SCALING, 0 FOR AUTOMATIC, 1 FOR MANUAL
      DIMENSION X(100)
C     FIND MAX AND MIN VALUES OF X
      IF(ISCAL.EQ.1)GOTO 15
      XMAX=X(1)
      XMIN=X(1)
      DO5  I=2,NP
      IF(X(I).GT.XMAX)XMAX=X(I)
      IF(X(I).LT.XMIN)XMIN=X(I)
5     CONTINUE
      IF(IZERO.NE.0.OR.XMIN.GE.0) GOTO 15
      WRITE(LU,10)
10    FORMAT(" ** DATA HAS NEGATIVE VALUES ORIGIN CAN'T BE (0,0) **"/)
      IZERO=1
15    IEXP=0
C     SCALE DATA USING E FORMAT AND SAVING EXPONENT (IEXP)
20    IF(ABS(XMAX).GE.1.OR.ABS(XMIN).GE.1) GOTO 30
      DO25  K=1,NP
25    X(K)=X(K)*10.
      XMIN=XMIN*10.
      XMAX=XMAX*10.
      IEXP=IEXP+1
      GOTO 20
30    IF(ABS(XMAX).LE.1000.AND.ABS(XMIN).LE.1000) GOTO 40
      DO35  K=1,NP
35    X(K)=X(K)/10.
      XMIN=XMIN/10.
      XMAX=XMAX/10.
      IEXP=IEXP-1
      GOTO 30
C     DETERMINE INTERVAL FACTOR
40    DIF=ABS(XMAX-XMIN)
      IF(DIF.GT.5.)GOTO 45
      FACTR=1.0
      GOTO 80
45    IF(DIF.GT.10.)GOTO 50
      FACTR=2.0
      GOTO 80
50    IF(DIF.GT.25.)GOTO 55
      FACTR=5.0
      GOTO 80
55    IF(DIF.GT.50)GOTO 60
      FACTR=10.0
```

```
      GOTO 80
60    IF(DIF.GT.125) GOTO 65
      FACTR=25.
      GOTO 80
65    IF(DIF.GT.250)GOTO 70
      FACTR=50.
      GOTO 80
70    IF(DIF.GT.500)GOTO 75
      FACTR=100.
      GOTO 80
75    FACTR=200.
80    IF(XMIN.GE.0.)GOTO 85
C  **** FOR NEGATIVE NUMBERS ****
      MIN=(XMIN/FACTR)-0.999999
      XMIN=MIN*FACTR
      IF(XMAX.GE.0) GOTO 90
      MAX=(XMAX/FACTR)
      XMAX=MAX*FACTR
      GOTO 95
C  **** FOR POSITIVE NUMBERS ****
85    MIN=XMIN/FACTR
      XMIN=MIN*FACTR
90    MAX=(XMAX/FACTR)+0.999999
      XMAX=MAX*FACTR
C  ********************************
95    IF(IZERO.EQ.0) XMIN=0.
      DIF=XMAX-XMIN
C     DETERMINE NUMBER OF TICK MARKS PER AXIS
      IF(DIF.NE.1.)GOTO 100
      TIC=1.
      GOTO 130
100   IF(DIF.GT.10)GOTO 105
      DIF=DIF*10.
      GOTO 100
105   IF(DIF.LE.100)GOTO 110
      DIF=DIF/10.
      GOTO 105
110   TIC=3.
      IF(DIF.EQ.100.)GOTO 125
      IF(DIF.EQ.80.)GOTO 115
      IF(DIF.GE.60.)GOTO 130
      IF(DIF.EQ.50.)GOTO 125
      IF(DIF.EQ.40.)GOTO 115
      IF(DIF.EQ.30.)GOTO 130
      IF(DIF.EQ.25.)GOTO 125
      IF(DIF.EQ.20.)GOTO 120
      IF(DIF.EQ.15.)GOTO 130
      IF(DIF.EQ.12.5)GOTO 125
115   TIC=4.
      GOTO 130
120   TIC=2.
      GOTO 130
125   TIC=5.
130   CONTINUE
      RETURN
      END
```

```
FTN4,L
      SUBROUTINE LSREG (X,Y,IFRM,ITO,SLOPE,YINT,SDSLOP,SDYINT)
C     PROGRAM CALCULATES LEAST SQUARES REGRESSION
C     X       - X ARRAY
C     Y       - Y ARRAY
C     IFRM    - STARTING POINT OF INTERVAL
C     ITO     - ENDING POINT OF INTERVAL
C     SLOPE   - RISE OVER RUN OF L.S. LINE
C     YINT    - Y INTERCEPT OF L.S. LINE
C     SDSLOP  - STANDARD DEVIATION OF SLOPE
C     SDYINT  - STANDARD DEVIATION OF Y INTERCEPT
      DIMENSION X(100),Y(100)
      FN=ITO-IFRM+1
      TX=0
      ZY=0
      XY=0
      WY=0
      SY=0
      DO 100 K=IFRM,ITO
      SY=SY+(Y(K)**2)
      WY=WY+(X(K)*Y(K))
      XY=XY+X(K)
      ZY=ZY+Y(K)
  100 TX=TX+(X(K)**2)
      TY=(ZY**2)/FN
      XZY=(XY*ZY)/FN
      XYN=(WY-XZY)**2
      XYD=TX-((XY**2)/FN)
      SX=(XY**2)/FN
      SDSLOP=SQRT((SY-TY-(XYN/XYD))/((FN-2.)*(TX-SX)))
      SDYINT=SQRT((SDSLOP**2)*(TX/FN))
      SLOPE=((FN*WY)-(XY*ZY))/((FN*TX)-(XY**2))
      YINT=(ZY-(SLOPE*XY))/FN
      RETURN
      END
```