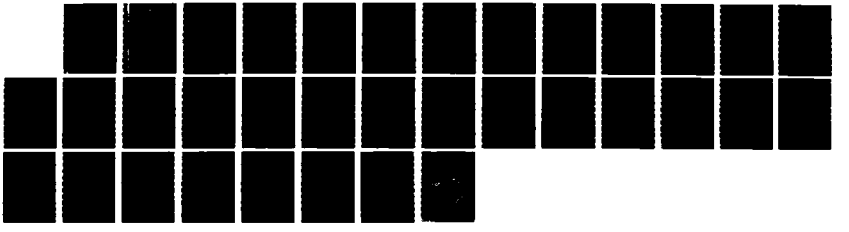
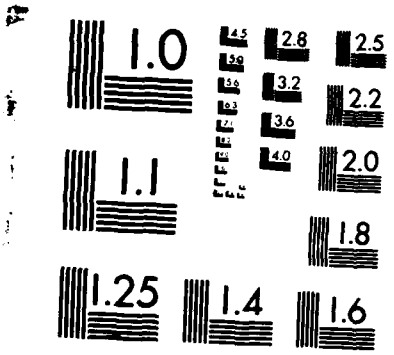
AD-A173 838 THE IMPLEMENTATION OF THE PLOTTING PROGRAM PLOTEZ(U) 1/1
MATERIALS RESEARCH LABS ASCOT VALE (AUSTRALIA)
P J RYAN MAY 86 MRL-TN-502

UNCLASSIFIED F/G 9/2 NL





MRL-TN-502

AR-004-441

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

MATERIALS RESEARCH LABORATORIES

MELBOURNE, VICTORIA

TECHNICAL NOTE

M R L - T N - 5 0 2

THE IMPLEMENTATION OF THE PLOTTING PROGRAM PLOTEZ

Peter J. Ryan

MAY 1986

DEPARTMENT OF DEFENCE

MATERIALS RESEARCH LABORATORIES

TECHNICAL NOTE

M R L - T N - 5 0 2

THE IMPLEMENTATION OF THE PLOTTING PROGRAM PLOTEZ

Peter J. Ryan

ABSTRACT

PLOTEZ is a versatile plotting routine which produces two-dimensional X-Y plots of data on suitable graphics terminals. PLOTEZ uses the extensive PLOT 10 IGL graphics library and is thus device-independent. A considerable variety of plot characters is available making PLOTEZ highly useful for comparing different sets of data. This report describes the historical development of PLOTEZ, how to run PLOTEZ on a VAX computer and some sample plots produced. A listing of the code is included in an Appendix.

POSTAL ADDRESS: Director, Materials Research Laboratories
P.O. Box 50, Ascot Vale, Victoria 3032, Australia

SECURITY CLASSIFICATION OF THIS PAGE **UNCLASSIFIED**

DOCUMENT CONTROL DATA SHEET

REPORT NO.	AR NO.	REPORT SECURITY CLASSIFICATION
MRL-TN-502	AR-004-441	Unclassified

TITLE

The implementation of the plotting program PLOTEZ

AUTHOR(S)

Peter J. Ryan

CORPORATE AUTHOR

Materials Research Laboratories
P.O. Box 50,
Ascot Vale, Victoria 3032

REPORT DATE	TASK NO.	SPONSOR
May 1986	DST 85/153	DSTO

FILE NO.	REFERENCES	PAGES
G6/4/8-3152	3	32

CLASSIFICATION/LIMITATION REVIEW DATE

CLASSIFICATION/RELEASE AUTHORITY
Superintendent, MRL
Physical Chemistry Division

SECONDARY DISTRIBUTION

Approved for Public Release

ANNOUNCEMENT

Announcement of this report is unlimited

KEYWORDS

Plotting	Fortran 77
Graphics	Plot 10
Computer Programs	IGL Graphics Library

COSATI GROUPS 1402

ABSTRACT

PLOTEZ is a versatile plotting routine which produces two-dimensional X-Y plots of data on suitable graphics terminals. PLOTEZ uses the extensive PLOT 10 IGL graphics library and is thus device-independent. A considerable variety of plot characters is available making PLOTEZ highly useful for comparing different sets of data. This report describes the historical development of PLOTEZ, how to run PLOTEZ on a VAX computer and some sample plots produced. A listing of the code is included in an Appendix.

C O N T E N T S

THE IMPLEMENTATION OF THE PLOTTING PROGRAM PLOTEZ

1. INTRODUCTION

PLOTEZ is a plotting program which produces graphical output of data from a VAX computer. It utilizes the Tektronix PLOT10 IGL library of graphics subroutines [1] to create two-dimensional plots of data on Tektronix-type graphics terminals. The horizontal (X) axis usually represents the independent data (X-values) while the vertical (Y) axis represents the dependent (Y-values) data set. It should be emphasized that PLOTEZ is a plotting program and not a *graphics package*; that is, no manipulation of the data plotted on the screen is possible.

PLOTEZ has had a rich history. It had its origins as the simple plotting program XYPLOT used for plotting PHA (Pulse Height Analysis) gamma-ray spectra on a PDP-15 minicomputer located in the Physics School at Melbourne University in the early 1970's. The output device used was a Tektronix 611 storage display scope and graphical output achieved using some primitive pen-up, pen-down commands which came supplied with the PDP-15 system. Hard copy was achieved by doing a screen-dump using a photocopy-type device.

With the introduction of a VAX-11/780 computer in 1979 at Melbourne University this program was ported to the VAX, considerably enhanced (and renamed PLOT). Graphical output to TEK-4010 terminals was achieved by sending special character strings to the terminal to put it into graphics mode, whereas text output (for numbers on the axes and titles) was achieved by simply exiting from graphics mode and using formatted FORTRAN WRITE instructions. This was the only way to proceed at the time since the VAX had no plotting package installed. PLOT ran in this fashion for several years on the VAX-VMS system very successfully and has now been adapted to run on the Data General MV-8000 in the Physics School at Melbourne University.

In 1981, the author accepted a position at the University of Massachussetts at Amherst (USA) and found that there was no suitable plotting program for the RSX-11 system running on the PDP-11/35 which the Nuclear

Physics group used for data analysis. The PLOT program was then converted to run on this machine using the early PLOT10 Terminal Control System (TCS) graphics library [2]. Later, in 1983, when the group upgraded their computer facility to a VAX-11/750 the code was adapted to run on this machine. This program was very successful and used widely. It is still running at the University of Massachussetts and elsewhere. Among the modifications applied at UMass was the ability to produce semi-log plots which were needed for plotting electron scattering cross sections. These typically varied by orders of magnitude for small changes in momentum transferred to the nucleus.

2. DISCUSSION

The most recent changes to PLOTEZ have been made during the period September, 1985 till March, 1986. The author had started working on the MRL VAX-11/780 and had a need for a callable plotting routine to do plots of ship magnetic signatures. By using a set of magnetic dipoles to represent a ship's magnetization the geomagnetic field perturbations in the vicinity of the ship could be calculated [3].

The code PLOT was ported to the MRL VAX, renamed PLOTEZ and upgraded considerably to run with the PLOT10 IGL library [1]. The PLOT10 IGL library provides a more advanced set of graphics routines than the earlier PLOT10 TCS library. The main enhancements implemented in this new version are: (a) software generation of text and (b) user control of the plot aspect ratio. Most of the routines have been thoroughly rewritten (in FORTRAN 77) so that the code runs faster, is more efficient and better documented internally. Further the type of plot output (e.g dashed lines, special symbols such as stars) has been greatly expanded using the capabilities of the IGL routines.

3. USING PLOTEZ ON A VAX COMPUTER

PLOTEZ is called as a subroutine from a FORTRAN program which contains the X,Y values stored in arrays. For example the following test program first generates sets of X,XE,Y,YE values (XE,YE are the errors in X,Y) and then calls the plotting routine for output to a graphics terminal.

```
      PROGRAM TEST
      INTEGER NPT, ICHAR, IAX
      PARAMETER (NPT=51)
      REAL X(NPT),XE(NPT),Y(NPT),YE(NPT)
C
C     GENERATE THE X,Y,XE,YE VALUES
      DO 100 I = 1,NPT
      X(I) = FLOAT(2*I-2)
      Y(I) = 100.*SIN(X(I)/3.0)
```

```
            XE(I) = 0.0
            YE(I) = 0.1*ABS(Y(I))
100         CONTINUE
C
C           NOW PLOT THE DATA
            TYPE *,' ICHAR,IAX ?'
            ACCEPT *,ICHAR,IAX
            CALL PLOTEZ(X,Y,XE,YE,NPT,ICHAR,IAX)
C           STOP
            END
```

In the call to PLOTEZ the arrays X,Y,XE,YE carry the data and NPT refers to the number of points of the arrays to be plotted. The parameters ICHAR,IAX specify characteristics of the plot required and will be described in the next section.

For a user running on the MRL VAX the program TEST must now be compiled and linked with PLOTEZ and the IGL library:

```
$FORTRAN TEST
$LINK TEST,[RYANP.PLOT]PLOTEZ,SYS$LIBRARY:IGL/LIB
```

To run the program it is necessary to be logged on to a suitable graphics terminal and type:

```
$RUN TEST
```

For a user at a different site it would be necessary to first install PLOTEZ and the IGL library, then compile and link PLOTEZ and the IGL library with the user program. A listing of the program is given in an appendix at the end of this report.

3.1 The Parameters ICHAR and IAX

The parameters ICHAR and IAX included in the program listing above are used to specify the type of plot requested. ICHAR controls the plotting symbol used. There are 3 main ranges of integer values ICHAR can take:

1. 0<ICHAR<17 the IGL special symbols are used. For example for ICHAR=4 triangles are plotted for the points; for ICHAR=12 stars are plotted for the points.

2. ICHAR=17,18 a semi- or full histogram plot is done.

3. ICHAR>=20 the points are joined up as either a continuous line (ICHAR=20) or a dashed line (ICHAR>20).

Note that for ICHAR < 0 error bars are not plotted.

The parameter IAX specifies the graphic action requested. Currently IAX can take all integer values from -2 to +4. For example, if IAX=-1 the program asks for scaling information, titles etc. and then erases the screen drawing a new frame with tick marks etc. In contrast, for IAX=1 the screen is not erased, the same scale as was determined by the last plot is used and the data overplotted on to the same frame.

Further information on the action specified by the parameters ICHAR and IAX is given in Appendix A.

3.2 Response to PLOTEZ Prompts

If IAX is negative, PLOTEZ prompts the user for various input parameters. These must be entered (and in the correct format) or the program will wait indefinitely (or crash if the wrong type of data is input).

Question 1. 'PLOT TYPE? [P=AUTO SCALE,S=SET SCALE,E=EXIT,DEF=P]>>

If 'P' or return is entered the program scales the input data automatically whereas if 'S' is entered the program asks for the minimum and maximum X,Y values to be plotted. If 'E' is entered the program exits without plotting. The single character input here can be in either upper or lower case.

Question 2. 'FOR SEMI-LOG PLOT TYPE L [DEF=LINEAR]>>'

If 'L' or 'l' is entered the Y data points are scaled logarithmically; if anything else is entered the Y data points are scaled linearly.

Question 3. If 'S' or 's' has been entered for Q1, then the program asks:' MIN X, MAX X, MIN Y, MAX Y VALUES TO PLOT [NO DEF.] >>'

These can be entered in free format. For example a typical response would be : 0.0,10.0,-100.0,100.0

Questions 4,5,6. 'TYPE TOP TITLE'>>
'TYPE X-AXIS TITLE'>>
'TYPE Y-AXIS TITLE'>>

Here the 3 titles are prompted for - the TOP title which is printed above the frame, the X-AXIS title printed beneath the X-axis, and the Y-AXIS title which is printed to the left of the frame and rotated through 90 degrees. After each title is entered the return key must be pressed.

The code will accept character strings of up to 80 characters in length as input to these prompts.

Question 7. 'CHARACTER SIZE FOR TOP, X-, Y-AXIS TITLES [DEF=1.,1.,1.] >>'

The size of the characters used for the 3 titles can be specified separately here. If return is entered the 3 titles are printed using the default character size. The values input scale the title sizes from the default. For example if 2.0 is entered for the X-axis character size and 1.0 for the Y-axis character size, the X-axis title characters will be twice as big as the Y-axis characters. The character sizes are restricted to be a maximum value of 2.0. The default input values are 1.0,1.0,1.0 which will be set if the return key is pressed.

Question 8. 'NUMBERS OF X- AND Y-AXIS INCREMENTS [DEF=5,5]'

The numbers of X- and Y-axis increments between large tick marks are entered here. If NTX or NTY exceeds 10 they are reset to 10. If return is entered NTX and NTY are both set to their default values 5. A typical response to this prompt would be : 6,10 specifying 6 increments along the X-axis and 10 along the Y-axis. Note that if log scaling is requested for the Y-data points the value of NTY will be calculated automatically and the value input here will be ignored.

Question 9. 'X,Y VIEWPORT FACTORS [DEF=1.,1.]'

Here the factors XVW,YVW which define the size of the viewport are entered. The viewport is the area on the display (screen or plotter) on to which the plot is drawn. A typical response would be : 1.5,1.2 specifying that the viewport is compressed by a factor of 1.5 in the X-direction and by 1.2 in the Y-direction. In this way the aspect ratio of the plot can be altered. The defaults (entered by pressing return) are both equal to 1.0.

Question 10. 'DEVICE, OPTION [DEF=4010,1]'

Here the two parameters IDEV,IOPT required by the IGL initialization routine GRSTRT (see ref. [1]) are entered. The default values are IDEV=4010, IOPT=1 (TEK4010 or compatible terminal). Note that if hard copy output is required on to a TEK4662 or compatible plotter the program will pause after this question has been answered to give the user time to set up the plotter etc. When this has been done typing any integer or return will enable the program to continue. For a TEK4662 plotter it is necessary to type in several more returns to enable it to start plotting.

4. PROGRAM DETAILS

At present PLOTEZ contains about 900 lines of code including about 350 lines of comments. The actual code contains roughly 550 active lines of FORTRAN 77 and its incorporation into a user program should not seriously degrade performance. The maximum number of points which can be plotted in a single call to PLOTEZ is set to 4096 using a PARAMETER statement. PLOTEZ is written to conform to the FORTRAN 77 standard and frequent use is made of FORTRAN 77 features such as IF-THEN-ELSE-ENDIF constructs, CHARACTER type variables etc.

PLOTEZ projects the plot vectors on to a Tektronix terminal using the default system of GDUs (Graphic Display Units) as the display surface units. On a Tektronix 4010 terminal, for example, the default window in GDUs is roughly 130 units along the X-axis and 100 units along the Y-axis. The use of GDUs means that the output is independent of the physical size of the screen since each GDU is 1/100th the distance of the shorter axis on the display surface.

PLOTEZ has a main routine PLOTEZ and 6 subsidiary routines. The main routine accepts the input data from the calling program, decides what type of plot is required then initializes the device and calls the subsidiary routines to do the scaling and plotting. The function of these routines is as follows.

(a) FRAME - draws the frame with tick marks and numbers and also draws the titles. Note that since the text is software generated it is all scaled to the same size as the plot.

(b) PLOTXY - plots the points on the frame drawn by FRAME. The type of plot symbol is specified by the parameter ICHAR.

(c) LOGTRAN - converts the Y-values into their LOG10 representation if a semi-log plot is requested.

(d) LINSCALE - determines the increments between the tick marks and the starting position on each axis.

(e) LOGSCALE - determines increments between tick marks etc. in the Y-axis for semi-log plots.

(f) XPOW - decomposes the input real variable into its two scientific-notation components and returns these values to the calling routine. For example, if -30.0 is input the real number -3.0 and the integer 1 are returned. This is used by the scaling routines.

(g) TICK - draws tick marks or line segments between two specified points.

5. SAMPLE OUTPUT

The program in section 3 produces the output shown in Fig. 1 when the input parameters 6,-2 are specified for the parameters ICHAR,IAX. The program calculates a set of sinusoidal Y-values with the Y-error bars proportional to the absolute Y-values. The program plots the data points as diamonds with vertical error bars.

A more complicated example is given in Fig.2 where the three-axis magnetic field perturbations due to a naval vessel passing over a mine located at a depth of 20.0 m are plotted on the same graph. The X-field component is represented by the continuous line, the Y-field component by the dotted line, and the Z-field component by the dash-dot line. The ship magnetization is modelled by a set of 20 three-axis magnetic dipoles spaced 6 m apart along a midships line at water level. The X,Y,Z field components are calculated by summing the contributions from each dipole in each axis using the equations for a magnetic dipole. The set of dipoles used is a test set of data given in Table 3.1 of [3] to examine the validity of the technique of representing a ship's magnetization by a set of discrete three-axis dipoles. The dipole field equations are taken from the program listing given in the Appendix of [3].

This plot is produced by 3 calls to PLOTEZ. The first call uses the ICHAR value of 20 to produce a continuous line plot of the X-field component and IAX=-1 to set the scale of the plot manually. Then the Y and Z components of the field are overplotted on the same plot by calls to PLOTEZ with ICHAR,IAX values of 21,+1 (dotted line), and 22,+1 (dash-dot line). The FORTRAN code fragment which produced this output is:

```
CALL PLOTEZ(XDISP,BX,ERRX,ERRY,NUMSTEP,20,-1)
CALL PLOTEZ(XDISP,BY,ERRX,ERRY,NUMSTEP,21,+1)
CALL PLOTEZ(XDISP,BZ,ERRX,ERRY,NUMSTEP,22,+1)
```

The arrays XDISP,BX,BY,BZ,ERRX,ERRY carry the X-values, the X,Y,Z field components and the errors respectively, while NUMSTEP specifies the number of points plotted to make the 3 curves on the plot.

6. CONCLUSION

The graphics program PLOTEZ is now available as a callable subroutine on the MRL VAX 11/780. It is a device-independent program and will run on any TEK-4010 compatible terminal. To interface PLOTEZ to a user program, it is necessary to link the program with PLOTEZ and the IGL PLOT10 graphics library on the VAX system.

7. REFERENCES

1. TEK User's manual for PLOT10 4010C01 Interactive Graphics Library (1982). Part No. 070-2685-02, Tektronix, Inc.

2. PLOT10 4010A01 Terminal Control System User's Manual (1979). Manual Part No. 070-2241-00. Tektronix, Inc.

3. Saiva, G. (1983). "Magnetic Dipole Modelling of Ship Magnetization", Aust. RAN Research Lab. Tech. Note No. 6/80.

APPENDIX A

A. Action Taken By Values of ICHAR Parameter

The ICHAR parameter is used to give different characters for the plotted points:

1. For ICHAR < 17 one of the characters from the default software font with Graphics Text Emulation (3D) (see [1]) is used. These are:

ICHAR	CHARACTER
1	Dot or bullet
2	Square
3	Octagon
4	Triangle
5	+
6	Diamond
7	Square with x-cross
8	Lozenge
9	Octagon with x-cross
10	Square with x-cross
11	Nabla or del
12	Star
13	Asterisk
14	X-cross
15	Up arrow
16	Down arrow

2. For ICHAR = 17 a semi-histogram plot is requested; for ICHAR=18 a full histogram plot is drawn.

3. For ICHAR = 20 the points are joined as a continuous line. For ICHAR>20 one of the possible dashed lines is drawn:

ICHAR	LINE TYPE
21	.....
22	- - - - - -
23	- - - - - -
24	——— ———
25	- - - -

etc. as given in the IGL user manual [1].

B. Action Taken By IAX Parameter

IAX specifies which type of plot is wanted.

IAX	ACTION
+4	No plot is done; only the scale is calculated

+3	Draws the frame only using the scale input from the calling program
+2	Erases the screen, then using the same scale as was calculated previously for the last plot, draws the frame and plots the data
+1	No frame drawn, screen not erased; data overplotted on the frame drawn by last plot
0	Auto-scaled plot
-1	Prompts for scaling parameters etc; recycles after plotting
-2	Same as -1 but returns after plotting

APPENDIX B

```
      SUBROUTINE PLOTEZ(X,Y,XE,YE,NPT,ICHAR,IAX)
C
C     PLOTTING ROUTINE FOR TEKTRONIX-TYPE TERMINALS ON VAX 11/780
C     WHICH USES THE PLOT10 IGL LIBRARY.
C
C     X    --  X VALUES
C     Y    --  Y VALUES
C     XE   --  ERROR IN X VALUES
C     YE   --  ERROR IN Y VALUES
C     LB   --  BYTE ARRAY - IF TRUE PLOT POINT, ELSE NOT
C     NPT  --  NUMBER OF POINTS
C
C     ICHAR   --      CONTROLS PLOT CHARACTER
C             0  --   ERROR BARS ONLY
C             1  --   POINTS
C             2  --   SQUARES
C             3  --   OCTAGONS
C             4  --   TRIANGLES
C             5  --   +'S
C             6  --   DIAMONDS
C             7  --   SQUARE WITH X-CROSS
C             8  --   LOZENGE
C             9  --   OCTAGON WITH X-CROSS
C             10 --   SQUARE WITH +-CROSS
C             11 --   NABLA OR DEL
C             12 --   STAR
C             13 --   ASTERISK
C             14 --   X-CROSS
C             15 --   UP ARROW
C             16 --   DOWN ARROW
C
C             17 --   GIVES SEMI-HISTOGRAM
C             18 --   GIVES FULL HISTOGRAM
C
C             20 -- CONTINUOUS LINE
C            >20 -- DASHED LINES :
C             21 -- DOTTED LINE (.....)
C             22 -- DASHED-DOT LINE (-•-••-••)
C             23 -- DASHED LINE (-----)
C             24 -- LONG DASHED LINE
C             25 -- DASH - TWO DOT LINE (-•••-•••-••)
C             ETC. SEE IGL MANUAL
C
C     IF ICHAR -VE ERROR BARS ARE NOT PLOTTED
C
C     IAX --  TYPE OF PLOT
C             4 --    NO PLOT, CALCULATES SCALE ONLY
C             3 --    PLOTS AXES ONLY. SCALE INPUT FROM CALL.
C             2 --    PLOTS AXES, SAME SCALE AS LAST PLOT.
C             1 --    NO AXES, NO ERASE, SAME SCALE AS LAST PLOT.
C             0 --    NORMAL PLOT (AUTO SCALE,NO TITLE OPTIONS ETC).
```

```
C              -1 --     ASKS FOR PLOT, WITH OPTIONS FOR TITLE
C                        AND USER SCALING. RECYCLES AFTER PLOTTING.
C              -2 --     ASKS FOR PLOT, WITH OPTIONS FOR TITLE
C                        AND USER SCALING. RETURNS AFTER PLOTTING.
C
C
C-----------------------------------------------------------------
C       FOR  IAX = <3,4>  USE:
C
C       XMIN -- MIN. X VALUE TO PLOT
C       XMAX -- MAX. X VALUE TO PLOT
C       YMIN -- MIN. Y VALUE TO PLOT
C       YMAX -- MAX. Y VALUE TO PLOT
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       LATEST VERSION AS OF APR-1986 P.RYAN
C         PHYSICAL CHEMISTRY DIVISION
C         MATERIALS RESEARCH LABORATORIES
C         P.O. BOX 50, ASCOT VALE 3032, VICTORIA, AUSTRALIA
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       SET MAXIMUM NUMBER OF POINTS TO PLOT
        PARAMETER (MAXPTS = 4096)
C
C       DECLARATION STATEMENTS
        DIMENSION X(*),Y(*),XE(*),YE(*)
        LOGICAL LB(MAXPTS), LOGPLOT, SETSCALE
        REAL LY(MAXPTS),LYE(MAXPTS)
        CHARACTER * 80  TITLE,XTITLE,YTITLE
        CHARACTER * 7 TOP,X_AXIS,Y_AXIS
        CHARACTER * 1 ANS, ANSLOG
        COMMON /XYBLK/ XMIN,XMAX,YMIN,YMAX,XDELT,YDELT,XSIZE,XWIDTH
        DATA XLEFT,XRIGHT,YBOT,YTOP/4*0.0/
        DATA XVW, YVW / 1.0, 1.0/
        DATA IDEV, IOPT/ 4010, 1/
        DATA LOGPLOT, SETSCALE /.FALSE., .FALSE./
        DATA TOP,X_AXIS,Y_AXIS/' TOP',' X-AXIS',' Y-AXIS'/
C
C       SET DEFAULTS :
C         NTT,NXT,NYT - NO. OF CHARACTERS IN TOP,X-,Y-AXIS TITLES
C         NTX,NTY - NO. OF INCREMENTS ALONG X-,Y-AXES
C         XSIZE,XWIDTH - CHARACTER WIDTH AND CELL WIDTH (=XSIZE+0.385)
100     NTT = 0
        NXT = 0
        NYT = 0
        NTX = 5
        NTY = 5
        XSIZE = 1.410
        XWIDTH = 1.795
        IF(IAX .EQ. 4) SETSCALE = .FALSE.
C
C       FOR AUTO-SCALED PLOT OR OVERPLOTTING TRANSFER CONTROL TO 900
        IF(IAX.GE.0) GO TO 900
```

```
C
C       EXIT IF IAX IS OUT OF RANGE
        IF(IABS(IAX) .GT. 4) THEN
          TYPE *,' ICHAR OUTSIDE RANGE: ICHAR > 4 '
          GO TO 10000
        END IF
C
C       PAUSE IF OUTPUT DEVICE IS A TEK4662 PLOTTER
        IF(IDEV.EQ.4662) ACCEPT 150,NUM
150     FORMAT(I10)
C
C       IAX IS NEG.... ASK FOR GRAFIX PARAMS
        TYPE 200
200     FORMAT(' PLOT TYPE ?',
     + ' [P=AUTO SCALE, S=SET SCALE, E=EXIT, DEF=P] >>',$)
        ACCEPT 800,ANS
        IF(ANS.EQ.'E'.OR.ANS.EQ.'N'.OR.ANS.EQ.'e'.OR.ANS.EQ.'n')
     +  GO TO 10000
        IF(ANS.NE.'E'.OR.ANS.NE.'N'.OR.ANS.EQ.'S'.OR.
     +     ANS.ne.'e'.or.ANS.ne.'n'.or.ANS.eq.'s')GO TO 250
        GO TO 10000
250     TYPE 300
300     FORMAT(' FOR SEMI-LOG PLOT TYPE L [DEF=LINEAR] >>',$)
        ACCEPT 800,ANSLOG
        LOGPLOT = .FALSE.
        IF(ANSLOG.EQ.'L'.OR.ANSLOG.EQ.'l') LOGPLOT = .TRUE.
        IF (ANS.EQ.'S'.OR.ANS.EQ.'s') THEN
          SETSCALE = .TRUE.
          TYPE 350
350       FORMAT($' MIN X, MAX X, MIN Y, MAX Y VALUES TO PLOT',
     +           ' [NO DEF.] >> ')
          ACCEPT *,XMIN,XMAX,YMIN,YMAX
        ELSE
          SETSCALE = .FALSE.
        END IF
C
C       INPUT TITLING INFORMATION
C         NTT,NXT,NYT - NO OF CHARACTERS IN TITLES
C         TITLE, XTITLE, YTITLE - TITLE CHARACTER STRINGS
C         CHTOP,CHAXIS,CHYAXIS  - TITLE CHARACTER SIZES
        TYPE 400,TOP
400     FORMAT(' TYPE ',A,' TITLE >>',$)
        ACCEPT 850,NTT,TITLE
        TYPE 400,X_AXIS
        ACCEPT 850,NXT,XTITLE
        TYPE 400,Y_AXIS
        ACCEPT 850,NYT,YTITLE
        TYPE 450
450     FORMAT(' CHARACTER SIZES FOR TOP, X-,Y-AXIS TITLES ',
     +         ' [DEF=1.,1.,1.] >>',$)
        ACCEPT 500,CHTOP,CHXAXIS,CHYAXIS
500     FORMAT(3G20.7)
        CHTOP =   MIN(2.0,CHTOP)
        CHXAXIS = MIN(2.0,CHXAXIS)
        CHYAXIS = MIN(2.0,CHYAXIS)
```

```
        IF(CHTOP.LE.0.0)   CHTOP   = 1.0
        IF(CHXAXIS.LE.0.0) CHXAXIS = 1.0
        IF(CHYAXIS.LE.0.0) CHYAXIS = 1.0
        TYPE 550
550     FORMAT($,' NUMBERS OF X AND Y AXIS INCREMENTS [DEF= 5,5] >>',$)
        ACCEPT 600,NTX,NTY
600     FORMAT(2I10)
        NTX = MIN(10,NTX)
        NTY = MIN(15,NTY)
        IF(NTX.LE.0) NTX = 5
        IF(NTY.LE.0) NTY = 5
        TYPE 650
650     FORMAT(' X,Y VIEWPORT FACTORS [DEF=1.,1.] >> ',$)
        ACCEPT 700,XVW,YVW
700     FORMAT(2G20.7)
        XVW = MAX(1.0,XVW)
        YVW = MAX(1.0,YVW)
        TYPE 750
750     FORMAT(' DEVICE, OPTION [DEF=4010,1] >>',$)
        ACCEPT 600,IDEV,IOPT
        IF(IDEV.LE.0) IDEV = 4010
        IF(IOPT.LE.0) IOPT = 1
C
C       FORMATS USED FOR CHARACTER DATA INPUT
800     FORMAT(A1)
850     FORMAT(Q,A)
900     IF (SETSCALE .EQ. .TRUE.) GO TO 1100
        IF (SETSCALE .EQ. .FALSE.) THEN
          IF(IAX.GE.1 .AND. IAX.LE.3) THEN
            TYPE 950
950         FORMAT('**** WARNING - SCALES NOT YET SET SO RETURN ****')
            GO TO 10000
          ELSE
            GO TO (1150,1150,1100,1000) IAX
          END IF
        END IF
C
C       SET SCALE BY DETERMINING MAX, MIN X,Y VALUES TO PLOT
C
1000    CONTINUE
        XMIN = X(1)
        XMAX = X(1)
        YMIN = Y(1)
        YMAX = Y(1)
        DO 1050 I=1,NPT
        IF (ICHAR .LT. 0) THEN
          XMIN = MIN(XMIN,X(I))
          XMAX = MAX(XMAX,X(I))
          YMIN = MIN(YMIN,Y(I))
          YMAX = MAX(YMAX,Y(I))
        ELSE IF (ICHAR .GE. 0) THEN
          XMIN = MIN(XMIN,X(I)-XE(I))
          XMAX = MAX(XMAX,X(I)+XE(I))
          YMIN = MIN(YMIN,Y(I)-ABS(YE(I)))
          YMAX = MAX(YMAX,Y(I)+ABS(YE(I)))
```

```
        END IF
1050    CONTINUE
        SETSCALE = .TRUE.
C
C       LINEAR SCALE FOR X-VALUES
1100    CALL LINSCALE(XMIN,XMAX,XMV,XDIV,IPX,NTX)
C
C       LINEAR OR LOG SCALE FOR Y-VALUES
        IF(LOGPLOT .EQ. .FALSE.) CALL LINSCALE(YMIN,YMAX,YMV,YDIV,IPY,NTY)
        IF(LOGPLOT .EQ. .TRUE. ) CALL
LOGSCALE(YMIN,YMAX,YMV,NTY,IPY,INC,ISPEC)
C
C       FOR LOG SCALING CONVERT Y-VALUES TO LOG10 BASE
1150    IF(LOGPLOT .EQ. .TRUE. )  CALL LOGTRAN(X,Y,XE,YE,LY,LYE,NPT)
C
C       RETURN IF IAX = 4
        IF(IAX.EQ.4)GO TO 10000
C
C       INITIALIZE DEVICE - SET REQUIRED DEVICE DRIVER
C       ALSO ENSURE SOFTWARE PRODUCED TEXT AND SET TEXT SIZE
C
1200    IF(IDEV.EQ.4662) ACCEPT 1250,NUM
1250    FORMAT(I10)
        CALL GRSTRT(IDEV,IOPT)
        CALL TXQUAL(4)
        IF (IAX .NE. 1) THEN
          XSIZE  = XSIZE/XVW
        END IF
        CALL TXSIZE(0,XSIZE,0.0)
        CALL BELL
        IF(IAX.NE.1) CALL NEWPAG
        IF(IAX.EQ.1) GO TO 1350
C
C       SET WINDOW FOR PLOTTING
C       XLEFT,XRIGHT,YBOT,YTOP CORRESPOND TO THE LIMITS OF THE X,Y
C       VALUES WHICH CAN BE REFERENCED ON THE SCREEN. THESE ARE SET
C       TO ALLOW AMPLE ROOM TO WRITE TITLES, NUMBERS ON THE AXES ETC.
C
1300    XDELT = ABS(XMAX-XMIN)
        XLEFT = XMIN - 0.20*XDELT
        XRIGHT= XMAX + 0.10*XDELT
        IF(.NOT. LOGPLOT) THEN
          YDELT = ABS(YMAX-YMIN)
          YBOT  = YMIN - 0.15*YDELT
          YTOP  = YMAX + 0.15*YDELT
        ELSE IF(LOGPLOT)  THEN
          YDELT = ABS(LOG10(YMAX)-LOG10(YMIN))
          YBOT = LOG10(YMIN) - 0.15*YDELT
          YTOP = LOG10(YMAX) + 0.15*YDELT
          YMIN = LOG10(YMIN)
          YMAX = LOG10(YMAX)
        END IF
1350    CALL WINDOW(XLEFT,XRIGHT,YBOT,YTOP)
C
C       XGDUMAX, YGDUMAX CORRESPOND TO THE LIMITS OF THE DISPLAY SURFACE IN
```

```
C       GDUS WHERE THE PLOT IS TO BE DRAWN. THE DISPLAY SURFACE IS ROUGHLY
C       130 GDUS IN THE X-AXIS AND 100 IN THE Y-AXIS.
        XGDUMAX = 130./XVW
        YGDUMAX = 100./YVW
        CALL VWPORT(0.0,XGDUMAX,0.0,YGDUMAX)
C
C       CALL SUBROUTINE PLOTXY TO PLOT POINTS
C
1400    IF(IAX.NE.3 ) THEN
          IF (.NOT. LOGPLOT) CALL PLOTXY(X,Y,XE,YE,LB,NPT,ICHAR,LOGPLOT)
          IF (LOGPLOT)       CALL PLOTXY(X,LY,XE,LYE,LB,NPT,ICHAR,LOGPLOT)
        END IF
C
C       CALL SUBROUTINE FRAME TO DRAW AXES AND NOS. ON AXES.
C
1450    IF(IAX.NE.1)CALL FRAME(XMV,YMV,XDIV,YDIV,IPX,IPY,NTX,NTY,
     +  LOGPLOT,INC,ISPEC,
     +  TITLE,XTITLE,YTITLE,NTT,NXT,NYT,CHTOP,CHXAXIS,CHYAXIS)
C
C       RETURN ONCE FRAME ETC. DRAWN FOR IAX=3
        IF(IAX .EQ. 3) GO TO 9000
C
1500    IF(IAX.EQ.-1)THEN
          CALL HOME
          CALL GRSTOP
          GO TO 100
        END IF
C
9000    CONTINUE
        CALL HOME
        CALL GRSTOP
C
10000   CONTINUE
        RETURN
        END

C*************************************************************************
C
C       SUBROUTINE FRAME
C
C       DRAWS AXES, TICK MARKS AND TEXT
C*************************************************************************
        SUBROUTINE FRAME(XMV,YMV,XDIV,YDIV,IPX,IPY,NTX,NTY,
     +  LOGPLOT,INC,ISPEC,TITLE,XTITLE,YTITLE,NTT,NXT,NYT,
     +  CHTOP,CHXAXIS,CHYAXIS)
C
        DIMENSION YLOG(20), YMARK(20)
        REAL XVALUE(20),YVALUE(20),XPOSX(20),YPOSY(20)
        INTEGER NBEFOREX(20), NBEFOREY(20)
        CHARACTER * 80 TITLE, XTITLE, YTITLE
        LOGICAL LOGPLOT
        REAL XDEL, YDEL
        COMMON /XYBLK/ XMIN,XMAX,YMIN,YMAX,XDELT,YDELT,XSIZE,XWIDTH
C
C       FIRST DRAW FRAME
```

```
      CALL SKIP
      CALL MOVE(XMIN,YMIN)
      CALL DRAW(XMAX,YMIN)
      CALL DRAW(XMAX,YMAX)
      CALL DRAW(XMIN,YMAX)
      CALL DRAW(XMIN,YMIN)
C
C     CALCULATE X, Y INCREMENT UNITS CONVENIENTLY SCALED.
      XDEL = XDELT/100.
      YDEL = YDELT/100.
C
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
C     DRAW X-AXIS TICK MARKS + NOS
C      NOTE : XV IS THE ACTUAL X-VALUE
C             XVALUE() IS THE SCALED X-VALUE
C             XINC IS THE X-INCREMENT BETWEEN LARGE TICK MARKS
C             XDIV IS THE SCALED INCREMENT BETWEEN LARGE TICK MARKS
C             XMV IS THE SCALED MINIMUM VALUE
C             IPX, IPXM ARE EXPONENTS WHICH SCALE THE X-VALUES
C
      IF(IPX.LT.0) IPXM = IPX
      IF(IPX.GE.0.AND.IPX.LE.3) IPXM=0
      IF(IPX.GE.4) IPXM = IPX
      XINC = XDIV * 10.**IPX
C
C     FIRST DRAW X-AXIS TICK MARKS
      DO 100 I=1,NTX+1
      XV =(XMV+XDIV*FLOAT(I-1))*10.**IPX
      IF (I.GT.1 .AND. I.LT.NTX+1) THEN
        CALL TICK(XV,YMIN,XV,YMIN+3.0*YDEL)
        CALL TICK(XV,YMAX-3.0*YDEL,XV,YMAX)
      END IF
      IF (I.LE.NTX) THEN
        CALL TICK(XV+0.5*XINC,YMIN,XV+0.5*XINC,YMIN+1.5*YDEL)
        CALL TICK(XV+0.5*XINC,YMAX-1.5*YDEL,XV+0.5*XINC,YMAX)
      END IF
100   CONTINUE
C
C     NOW SCAN X-VALUES TO DETERMINE THE MAX. NO OF SIGNIFICANT FIGS
C     REQUIRED (NSIGX) AND STORE THE XVALUES, THE STARTING POSITIONS AND
C     THE NO. OF FIGS BEFORE THE DECIMAL PT. IN ARRAYS XVALUE(), XPOSX(),
C     AND NBEFOREX RESPECTIVELY. NOTE X-VALUES ROUNDED TO 2 DECIMAL PLACES
      NSIGX = -1
      DO 125 I = 1,NTX+1
      XV = (XMV+XDIV*FLOAT(I-1))*10.0**IPX
      XVAL = XV / 10.**IPXM
      XVAL = FLOAT(NINT(100.0*XVAL)) / 100.0
C
C     DETERMINE HOW MANY FIGS. AFTER DECIMAL PT.
C      NFRAC = FRACTIONAL PART OF XVAL * 100
C      IDIGIT1, IDIGIT2 = DIGITS OF NFRAC
C      NBEFOREX = NO. OF FIGS BEFORE DECIMAL PT.
C      NAFTERX  = NO. OF FIGS AFTER DECIMAL PT.
      NFRAC = ABS(NINT(100.*XVAL)-100*INT(XVAL))
      IDIGIT1 = NFRAC / 10
```

```
      IDIGIT2 = NFRAC - 10*IDIGIT1
      NAFTERX = -1
      IF (IDIGIT1 .GT. 0) NAFTERX = 1
      IF (IDIGIT2 .GT. 0) NAFTERX = 2
      NSIGX = MAX(NAFTERX,NSIGX)
C
C     DETERMINE START OF CHARACTER STRING USING KNOWLEDGE THAT CHARACTER
CELL
C     IS XWIDTH*XDEL GDU'S WIDE, THEN STORE NBEFOREX(),XPOSX(),XVALUE()
      CALL XPOW(XVAL,BX,IPXVAL)
      NBEFOREX(I) = MAX(1,1+IPXVAL) + (1-SIGN(1.0,XVAL))/2
      XPOSX(I) = XV - FLOAT(NBEFOREX(I))*XWIDTH*XDEL
      XVALUE(I) = XVAL
125   CONTINUE
C
C     NOW PRINT X-VALUES
      DO 150 I = 1,NTX+1
      CALL MOVE(XPOSX(I),YMIN-4.0*YDEL)
      NCHARX = NBEFOREX(I) + NSIGX + 1
      CALL RNUMBR(XVALUE(I),NSIGX,NCHARX)
150   CONTINUE
C
C     WRITE 10 RAISED TO POWER IPXM IF IPXM IS NON-ZERO
      IF (IPXM.NE.0) THEN
        CALL MOVE (XMAX-2.*XDEL,YMIN-12.0*YDEL)
        CALL TEXT(3,'X10')
        CALL MOVE(XMAX+1.0*XDEL,YMIN-9.0*YDEL)
        CALL INUMBR(IPXM,3)
      END IF
C
CYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY
C     NOW DRAW Y-AXIS TICK MARKS AND NOS.
C     SEE COMMENTS ABOVE FOR X-TICK MARKS ETC. THAT IS
C     LOOP 200 DRAWS TICK MARKS, LOOP 225 SCANS THE Y-VALUES TO
C     DETERMINE THE NO. OF SIG. FIGS REQUIRED, AND THEN LOOP 250
C     PRINTS THE Y-VALUES
C
      IF(LOGPLOT)  GO TO 300
      IF(IPY.LT.0) IPYM = IPY
      IF(IPY.GE.0.AND.IPY.LE.4) IPYM = 0
      IF(IPY.GE.5) IPYM = IPY
      YINC = YDIV * 10.**IPY
C
C     FIRST DRAW TICK MARKS
      DO 200 J=1,NTY+1
      YV = (YMV+YDIV*FLOAT(J-1))*10.**IPY
      IF (J.GT.1 .AND. J.LT.NTY+1) THEN
        CALL TICK(XMIN,YV,XMIN+2.*XDEL,YV)
        CALL TICK(XMAX-2.*XDEL,YV,XMAX,YV)
      END IF
      IF (J.LE.NTY) THEN
        CALL TICK(XMIN,YV+0.5*YINC,XMIN+XDEL,YV+0.5*YINC)
        CALL TICK(XMAX-XDEL,YV+0.5*YINC,XMAX,YV+0.5*YINC)
      END IF
200   CONTINUE
```

```
C
C       NOW SCAN Y-VALUES ROUNDED TO 2 DECIMAL PLACES. SEE COMMENTS IN LOOP
C       125 ABOVE
        NSIGY = -1
        DO 225 J = 1,NTY+1
        YV = (YMV+YDIV*FLOAT(J-1))*10.0**IPY
        YVAL = YV / 10.**IPYM
        YVAL = FLOAT(NINT(YVAL*100.0)) / 100.0
        NFRAC = ABS(NINT(100.*YVAL)-100*INT(YVAL))
        IDIGIT1 = NFRAC / 10
        IDIGIT2= NFRAC - 10*IDIGIT1
        NAFTERY = -1
        IF (IDIGIT1 .GT. 0) NAFTERY = 1
        IF (IDIGIT2 .GT. 0) NAFTERY = 2
        NSIGY = MAX(NAFTERY,NSIGY)
        CALL XPOW(YVAL,BY,IPYVAL)
        NBEFOREY(J) = MAX(1,IPYVAL+1) + (1-SIGN(1.0,YVAL))/2
        YPOSY(J) = YV
        YVALUE(J) = YVAL
225     CONTINUE
C
C       NOW PRINT Y-VALUES
        DO 250 J = 1,NTY+1
        NCHARY = NBEFOREY(J) + 1 + NSIGY
        XPOS = XMIN - (FLOAT(NCHARY)+0.5)*XWIDTH*XDEL
        CALL MOVE(XPOS,YPOSY(J))
        CALL RNUMBR(YVALUE(J),NSIGY,NCHARY)
250     CONTINUE
C
C       WRITE 10 RAISED TO THE POWER IPYM IF IPYM NON-ZERO
        IF (IPYM.NE.0) THEN
          CALL MOVE(XMIN-10.*XDEL,YMAX+4.0*YDEL)
          CALL TEXT(3,'X10')
          CALL MOVE(XMIN-7.0*XDEL,YMAX+6.0*YDEL)
          CALL INUMBR(IPYM,3)
        END IF
        GO TO 9999
C
CLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL
C       DRAW Y-AXIS TICK MARKS AND NOS FOR LOG SCALING
C
300     CONTINUE
C
        IF(ISPEC.EQ.0) THEN
C         DRAW TICK MARKS, NOS FOR ISPEC = 0 I.E. IN POWERS OF 10
          DO 400 ILOG = 1,NTY+1
          IVAL = IPY + (ILOG-1)*INC
          YVALOG = FLOAT(IVAL)
          CALL MOVE(XMIN-8.*XDEL,YVALOG)
          CALL TEXT(2,'10')
          CALL MOVE(XMIN-6.5*XDEL,YVALOG+3.5*YDEL)
          CALL INUMBR(IVAL,3)
          IF(ILOG.GT.1 .AND. ILOG.LT.NTY+1) THEN
            CALL TICK(XMIN,YVALOG,XMIN+2.*XDEL,YVALOG)
            CALL TICK(XMAX,YVALOG,XMAX-2.*XDEL,YVALOG)
```

```
            END IF
            IF(ILOG.LT.NTY+1) THEN
              MARK = (2*INC-3)/2 + 1
              IF(INC.EQ.1) YFAC = 5.0
              IF(INC.NE.1) YFAC = 10.0
              DO 350 IMARK = 1,MARK
              YVALOG = IVAL + LOG10((YFAC**IMARK))
              CALL TICK(XMIN,YVALOG,XMIN+1.0*XDEL,YVALOG)
              CALL TICK(XMAX,YVALOG,XMAX-1.0*XDEL,YVALOG)
350           CONTINUE
            END IF
400         CONTINUE
C
C         FOR CASE OF ISPEC = 1
C         FIRST CALCULATE YLOG(), YMARK() VALUES WHERE
C         YLOG  - THE LOGS OF THE Y-LOCATIONS OF THE MAIN TICK MARKS
C         YMARK - THE VALUE 1.,2.,, OR 5 WRITTEN NEXT TO THE TICK MARK
          ELSE IF (ISPEC .EQ. 1) THEN
            YLOG(1) = YMIN
            YLOG(NTY+1) = YMAX
            YMARK(1) = YMV
            DO 450 IMARK=2,NTY+1
            YMARK(IMARK) =YMARK(IMARK-1)*2.0
            IF(YMARK(IMARK).NE.2..AND.YMARK(IMARK).NE.10.)YMARK(IMARK)=5.
            IF(YMARK(IMARK) .EQ. 10.)THEN
              YMARK(IMARK) = 1.0
              IPY =IPY + 1
            END IF
            YTEMP = YMARK(IMARK) * 10.**IPY
            YLOG(IMARK) = LOG10(YTEMP)
450         CONTINUE
C
C         NOW DRAW TICK MARKS
          DO 550 JMARK = 1,NTY
            IF(YMARK(JMARK).EQ.1.0 .AND. JMARK.GT.1 .AND. JMARK.LT.NTY+1) THEN
              CALL TICK(XMIN,YLOG(JMARK),XMIN+2.0*XDEL,YLOG(JMARK))
              CALL TICK(XMAX,YLOG(JMARK),XMAX-2.0*XDEL,YLOG(JMARK))
            ELSE IF(YMARK(JMARK) .NE. 1.0 .AND. YLOG(JMARK) .LT. YMAX) THEN
              MARK1 = NINT(YMARK(JMARK))
              MARK2 = 2*MARK1 - MOD(MARK1,2)
              DO 500 IMAR = MARK1,MARK2
              YVALOG = LOG10(FLOAT(IMAR)/FLOAT(MARK1)) + YLOG(JMARK)
              CALL TICK(XMIN,YVALOG,XMIN+XDEL,YVALOG)
              CALL TICK(XMAX,YVALOG,XMAX-XDEL,YVALOG)
500           CONTINUE
            END IF
550         CONTINUE
C
C         NOW DRAW NUMBERS ON Y-AXIS FOR LOG SCALING FOR ISPEC =1
            DO 600 KMARK=1,NTY+1
            CALL XPOW(10.**YLOG(KMARK),YMANT,IEXP)
            MANTISS = NINT(YMANT)
            IF (MANTISS.NE.1) THEN
              IF(NTY.LE.10) THEN
                CALL MOVE(-3.0*XWIDTH*XDEL+XMIN,YLOG(KMARK))
```

```
            CALL INUMBR(MANTISS,2)
          END IF
        ELSE IF (MANTISS.EQ.1) THEN
          CALL MOVE(XMIN-3.5*XWIDTH*XDEL,YLOG(KMARK))
          CALL TEXT(2,'10')
          CALL MOVE(XMIN-2.5*XWIDTH*XDEL,YLOG(KMARK)+3.0*YDEL)
          CALL INUMBR(IEXP,2)
        END IF
600     CONTINUE
C
C       SPECIAL CASE FOR SCALING FROM 2.*10.**IPY TO 5.*10.**IPY
        IF(NTY.EQ.1 .AND. YMV.EQ.2.0)  THEN
          IF(IPY .NE. 0) THEN
            CALL MOVE(XMIN-14.*XDEL,YMAX-10.*YDEL)
            CALL TEXT(3,'X10')
            CALL MOVE(XMIN-10.*XDEL,YMAX-7.*YDEL)
            CALL INUMBR(IPY,2)
          END IF
          DO 650 IP = 3,4
          YIP = LOG10(FLOAT(IP)) + IPY
          CALL MOVE(XMIN-3.5*XWIDTH*XDEL,YIP)
          CALL INUMBR(IP,2)
650       CONTINUE
        END IF
C
      END IF
9999  CONTINUE
C
CTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
C     DRAW TITLES
C
C     MAIN TITLE
      IF(NTT.GT.0) THEN
        CALL TXANGL(0.)
        CALL TXSIZE(0,CHTOP*XSIZE,0.0)
        CALL MOVE(XMIN+5.0*XDEL,YMAX+2.0*YDEL)
        CALL TEXT(NTT,TITLE)
      END IF
C
C     X-TITLE - TITLE BELOW X-AXIS
      IF(NXT.GT.0) THEN
        CALL MOVE(XMIN+20.*XDEL,YMIN-12.5*YDEL)
        CALL TXANGL(0.)
        CALL TXSIZE(0,XSIZE*CHXAXIS,0.0)
        CALL TEXT(NXT,XTITLE)
      END IF
C
C     Y-TITLE - TITLE TO LEFT OF Y-AXIS ROTATED THROUGH 90 DEG.
      IF(NYT.GT.0) THEN
        CALL TXANGL(90.)
        IF(LOGPLOT) CALL MOVE(XMIN-17.5*XDEL,YMIN+15.*YDEL)
        IF(.NOT. LOGPLOT) THEN
          XLOC = XMIN - FLOAT(IPY-IPYM+NSIGY+6)*XDEL*XWIDTH
          CALL MOVE(XLOC,YMIN+15.*YDEL)
        END IF
```

```
          CALL TXSIZE(0,XSIZE*CHYAXIS,0.0)
          CALL TEXT(NYT,YTITLE)
        END IF
C
        RETURN
        END
C*************************************************************************
C       SUBROUTINE PLOTXY
C
C       PLOTS THE (X,Y) POINTS ON THE FRAME DRAWN BY FRAME
C
C*************************************************************************
        SUBROUTINE PLOTXY(X,Y,XE,YE,LB,NPT,ICHAR,LOGPLOT)
        REAL X(*),Y(*),XE(*),YE(*)
        LOGICAL LB(*), LOGPLOT
        COMMON /XYBLK/ XMIN,XMAX,YMIN,YMAX,XDELT,YDELT,XSIZE,XWIDTH
C
C       TEST WHICH POINTS LIE WITHIN RANGE AND SET LB VALUES TO
C       TRUE OR FALSE FOR EACH (X,Y) PAIR
C
        DO 50 IP = 1,NPT
        LB(IP) = .FALSE.
50      CONTINUE
C
        DO 100 JP = 1,NPT
C
C       FOR ICHAR < 0 EXCLUDE ERROR BARS FROM CONSIDERATION
        IF (ICHAR .LT. 0) THEN
          IF (X(JP).GE.XMIN .AND. X(JP).LE.XMAX .AND.
     1        Y(JP).GE.YMIN .AND. Y(JP).LE.YMAX)
     2        LB(JP) = .TRUE.
C
C       FOR ICHAR GE 0 AND LINEAR Y-SCALE INCLUDE SIZE OF X,Y ERROR BARS
        ELSE IF (ICHAR .GE. 0 .AND. LOGPLOT .EQ. .FALSE.) THEN
          IF (X(JP)-XE(JP).GE.XMIN .AND. X(JP)+XE(JP).LE.XMAX .AND.
     1        Y(JP)-YE(JP).GE.YMIN .AND. Y(JP)+YE(JP).LE.YMAX)
     2        LB(JP) = .TRUE.
C
C       FOR ICHAR GE 0 AND LOG Y-SCALE INCLUDE SIZE OF X-ERROR BARS ONLY
        ELSE IF (ICHAR .GE. 0 .AND. LOGPLOT .EQ. .TRUE.)  THEN
          IF (X(JP)-XE(JP).GE.XMIN .AND. X(JP)+XE(JP).LE.XMAX .AND.
     1        Y(JP).GE.YMIN .AND. Y(JP) .LE.YMAX)
     2        LB(JP) = .TRUE.
C
        END IF
100     CONTINUE
C
C       TEST WHICH TYPE OF PLOT IS REQUIRED
        ITYPE = ABS(ICHAR)
        IF(ITYPE.EQ.0) GO TO 550
        IF(ITYPE.GT.0 .AND. ITYPE.LE.16) GO TO 150
        IF(ITYPE.GE.17.AND. ITYPE.LE.18) GO TO 250
        IF(ITYPE.GE.20                 ) GO TO 350
C
C       PLOT SPECIAL SYMBOLS USING THE MARKER IGL ROUTINE
```

```
C       FOR THE SYMBOLS PLOTTED SEE COMMENTS AT START OF MAIN PROGRAM
C       OR TABLE F-8 IN PLOT10 IGL USER MANUAL 4010C01 (PART NO. 070-2685-02)
C       FOR THE DEFAULT ADE CHARACTERS
C
150     CONTINUE
        MARK = ITYPE-1
        DO 200 IP = 1,NPT
        IF(LB(IP)) CALL MARKER(X(IP),Y(IP),MARK)
200     CONTINUE
        GO TO 550
C
C       PLOTS SEMI-HISTOGRAM (ICHAR=17) OR FULL HISTOGRAM (ICHAR=18)
250     CONTINUE
        DO 300 IP = 1,NPT
        IF(.NOT.LB(IP)) GO TO 300
        I1=MAX(IP-1,1)
        I2=MAX(IP,2)
        I3=MIN(IP,NPT-1)
        I4=MIN(IP+1,NPT)
        XDEL1=(X(I2)-X(I1))/2.0
        XDEL2=(X(I4)-X(I3))/2.0
        X1=MAX(X(IP)-XDEL1,XMIN)
        X2=MIN(X(IP)+XDEL2,XMAX)
        IF(IP.LT.NPT)Y2=Y(IP+1)
        IF(IP.EQ.NPT)Y2=Y(IP)
        IF(ICHAR.EQ.17) THEN
          CALL MOVE(X1,Y(IP))
          CALL DRAW(X2,Y(IP))
          CALL DRAW(X2,Y2)
        ELSE IF(ICHAR.EQ.18) THEN
          CALL MOVE(X1,YMIN)
          CALL DRAW(X1,Y(IP))
          CALL DRAW(X2,Y(IP))
          CALL DRAW(X2,YMIN)
          CALL MOVE(X2,Y(IP))
          CALL DRAW(X2,Y2)
        END IF
300     CONTINUE
        GO TO 650
C
C       DRAW LINES
C       ICHAR = 20 - CONTINUOUS LINE
C       ICHAR > 20 - DASHED LINE
C
350     CONTINUE
        LINE = ITYPE-20
        CALL DASHPT(LINE)
C
C       FIND FIRST POINT WITHIN FRAME
        DO 400 IP = 1,NPT
        IF(LB(IP)) THEN
          CALL MOVE(X(IP),Y(IP))
          IFIRST = IP
          GO TO 425
        END IF
```

```
400     CONTINUE
C
C       NOW PLOT THE LINE
425     CONTINUE
        IF(IFIRST.GE.NPT) GO TO 650
        DO 450 IP = IFIRST+1,NPT
        IF(LB(IP)) CALL DRAW(X(IP),Y(IP))
450     CONTINUE
C
C       RESET DEFAULT LINE TYPE TO CONTINUOUS
        CALL DASHPT(0)
        GO TO 650
C
C
C       ERROR BARS
C
550     CONTINUE
        DO 600 IP = 1,NPT
        IF(.NOT.LB(IP)) GO TO 600
        IF(XE(IP).GT.0.0) THEN
          X1=MAX(X(IP)-XE(IP),XMIN)
          X2=MIN(X(IP)+XE(IP),XMAX)
          CALL TICK(X1,Y(IP),X2,Y(IP))
        END IF
        IF(YE(IP).GT.0.0) THEN
          IF(LOGPLOT.EQ..FALSE.) THEN
            Y1=MAX(Y(IP)-YE(IP),YMIN)
            Y2=MIN(Y(IP)+YE(IP),YMAX)
            CALL TICK(X(IP),Y1,X(IP),Y2)
          ELSE IF(LOGPLOT.EQ..TRUE.) THEN
            Y1 = MAX(Y(IP)-LOG10(YE(IP)),YMIN)
            Y2 = MIN(Y(IP)+LOG10(YE(IP)),YMAX)
            CALL TICK(X(IP),Y1,X(IP),Y2)
          END IF
        END IF
600     CONTINUE
C
650     CONTINUE
        RETURN
        END
C*************************************************************************
C       SUBROUTINE LOGTRAN
C*************************************************************************
        SUBROUTINE LOGTRAN(X,Y,XE,YE,LY,LYE,NPT)
C
C       CONVERTS DATA INTO LOG10 REPRESENTATION
C
C       INPUT  : Y,YE VALUES
C       OUTPUT : LY - THE LOG10 VALUES OF THE Y VALUES
C              : LYE - RETURNED AS THE YE( ) VALUES
C
        REAL X(*),Y(*),XE(*),YE(*),LY(*),LYE(*)
        INTEGER NPT
C
        DO 1000 IPT = 1,NPT
```

```
          IF(Y(IPT) .GT. 0.0) THEN
            LY(IPT) = LOG10(Y(IPT))
          ELSE
            LY(IPT) = -10.0
          END IF
C
          LYE(IPT) = YE(IPT)
C
1000      CONTINUE
          RETURN
          END
C*************************************************************************
C         SUBROUTINE LINSCALE
C*************************************************************************
          SUBROUTINE LINSCALE(XMIN,XMAX,XMV,XDIV,IPX,NTX)
C
C         INPUT  : XMIN, XMAX - THE MIN, MAX VALUES TO SCALE
C         OUTPUT : XMV  - THE SCALED MINIMUM VALUE (NO. FROM 0 TO 10)
C                  XDIV - THE SCALED INCREMENT BETWEEN LARGE TICK MARKS
C                  IPX  - THE POWER OF 10 MULTIPYING THE VALUES
C                  NTX  - THE NUMBER OF INCREMENTS ON THE AXIS
C
C         FIRST RESET XMIN, XMAX IF XMIN GE XMAX
          IF (XMIN .GE. XMAX) THEN
            IF (XMIN .EQ. 0.0) XMAX = 1.0
            IF (XMIN .NE. 0.0) XMAX = 2.0 * ABS(XMIN)
          END IF
C
          DIFF = (XMAX-XMIN)/FLOAT(NTX)
          CALL XPOW(XMIN,B1,IP1)
          CALL XPOW(XMAX,B2,IP2)
          IPX=MAX(IP1,IP2)
          IF(IP1 .EQ. 0)IPX = IP2
          XMV  = XMIN/10.**IPX
          XDIV = DIFF/10.**IPX
          RETURN
          END
C*************************************************************************
C         SUBROUTINE LOGSCALE
C*************************************************************************
          SUBROUTINE LOGSCALE(YMIN,YMAX,YMV,NTY,IPY,INC,ISPEC)
C
C         SETS VERTICAL SCALE FOR LOG PLOTTING
C         INPUT  : YMIN,YMAX - THE MIN, MAX VALUES TO SCALE
C                  NOTE THESE VALUES MAY BE RESET BY LOGSCALE
C         OUTPUT : YMV - THE SCALED MIN. VALUE (NO. FROM 1 TO 10)
C                  NTY - THE NUMBER OF INCREMENTS ON THE AXIS
C                  IPY - THE MIN. POWER OF 10
C                  INC - THE EXPONENT INCREMENT FOR THE LARGE TICK MARK
C                  ISPEC = 0 - SCALES IN POWERS OF 10 ONLY
C                  ISPEC = 1 - SCALES 1.,2.,5. X POWERS OF 10
C
          INTEGER IY(2)
          REAL Y(2)
C
```

```
      ISPEC=0
      INC=1
      YMV=1.0
      IF (YMIN .LE. 0.0)  YMIN=1.E-10
      IF (YMAX .LE. 0.0)  YMAX=1.E10
      IF (YMAX .LE. YMIN) YMAX=YMIN*10.0
      CALL XPOW(YMIN,Y1,IY1)
      CALL XPOW(YMAX,Y2,IY2)
      IF(Y2.GT.1.5) IY2=IY2+1
      NTY=IY2-IY1
C
C     BRANCH FOR DIFFERENT VALUES OF NTY
      IF(NTY .LE. 5) GO TO 100
      IF(NTY .GT. 10) THEN
        INC=(NTY+9)/10
        NTY=(NTY+INC-1)/INC
      END IF
      IPY = IY1
      YMIN=10.**IPY
      YMAX=10.**(NTY*INC+IPY)
      RETURN
C
C     ISPEC=1 CASE - SCALES 1.,2.,5. X POWERS OF 10
100   ISPEC=1
      CALL XPOW(YMIN,Y(1),IY(1))
      CALL XPOW(YMAX,Y(2),IY(2))
      DO 200 I=1,2
      IF(Y(I).GT.7.)IY(I)=IY(I)+1
      IF(Y(I).LE.1.5.OR.Y(I).GT.7.)  Y(I)=1.0
      IF(Y(I).LE.7..AND.Y(I).GE.3.5) Y(I)=5.0
      IF(Y(I).LT.3.5.AND.Y(I).GT.1.5)Y(I)=2.0
200   CONTINUE
C
      YMV=Y(1)
      IPY=IY(1)
      YMIN=Y(1)*10.**IY(1)
      YMAX=Y(2)*10.**IY(2)
      IT1=0
      IT2=0
      IF(Y(1).NE.1.0) IT1=(+Y(1)+1.1)/3.0
      IF(Y(2).NE.1.0) IT2=(+Y(2)+1.1)/3.0
      NTY=3*(IY(2)-IY(1)) + (IT2-IT1)
      IF(Y(1).EQ.2..AND.Y(2).EQ.5..AND.IY(1).EQ.IY(2))  NTY=1
      IF(Y(1).EQ.5..AND.Y(2).EQ.1..AND.IY(1).EQ.IY(2)-1)NTY=1
      RETURN
      END
C*********************************************************************
C     SUBROUTINE XPOW
C*********************************************************************
      SUBROUTINE XPOW(XVAL,XNORM,IEXP)
C
C     INPUT  : XVAL  - ANY REAL NO.
C     OUTPUT : XNORM - 10 RAISED TO THE MANTISSA OF LOG10(XVAL)
C              IEXP  - THE CHARACTERISTIC OF LOG10(XVAL)
C     NOTE DOUBLE PRECISION USED TO AVOID ROUNDING ERRORS
```

```
C
      DOUBLE PRECISION DA, DB, DG
C
      DA = DABS(DBLE(XVAL))
      IF(XVAL .EQ. 0.0) THEN
        XNORM = 0.
        IEXP  = 0
        RETURN
      END IF
C
      DG=DLOG10(DA) + 1.D-6
      IEXP = IDINT(DG)
      IF(DG .LT. 0.D0) IEXP = IEXP-1
      DB = DA / DBLE(10.0**IEXP)
      IF(DB .LT. 1.D0) DB = 1.D0
      IF(XVAL .GE. 0.0) XNORM = SNGL(DB)
      IF(XVAL .LT. 0.0) XNORM = -1.0 * SNGL(DB)
      RETURN
      END
C*********************************************************************
C     SUBROUTINE TICK
C     DRAWS TICK MARKS OR LINE SEGMENTS BETWEEN TWO SPECIFIED POINTS
C*********************************************************************
      SUBROUTINE TICK (X1,Y1,X2,Y2)
      REAL X1,Y1,X2,Y2
C
C     X1,Y1 ARE COORDS. OF START PT, X2,Y2 COORDS OF STOP PT.
      CALL MOVE(X1,Y1)
      CALL DRAW(X2,Y2)
      RETURN
      END
```

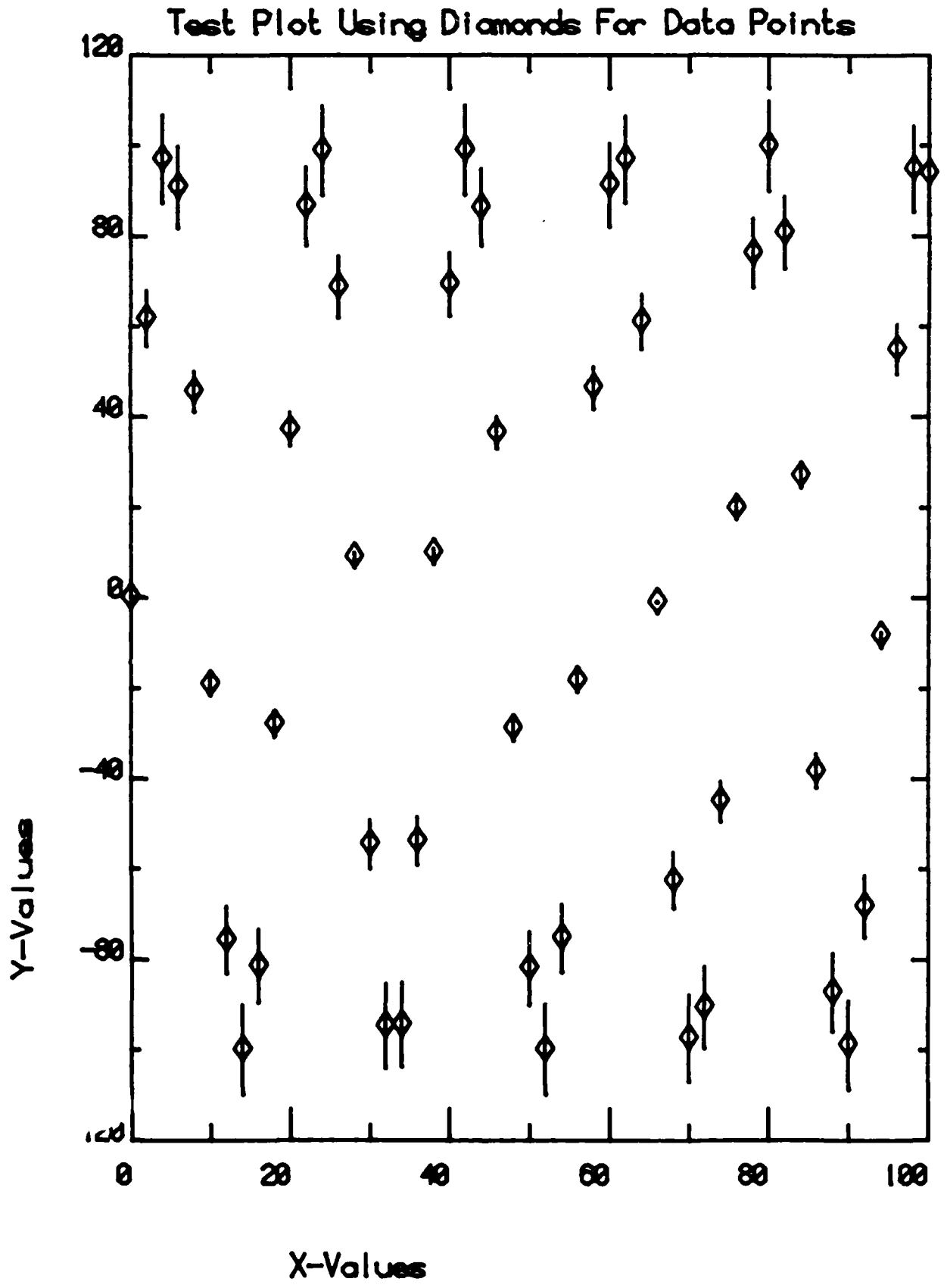

Test Plot Using Diamonds For Data Points
120
80
40
0
-40
-80
-120
Y-Values
0
20
40
60
80
100
X-Values

FIGURE 1

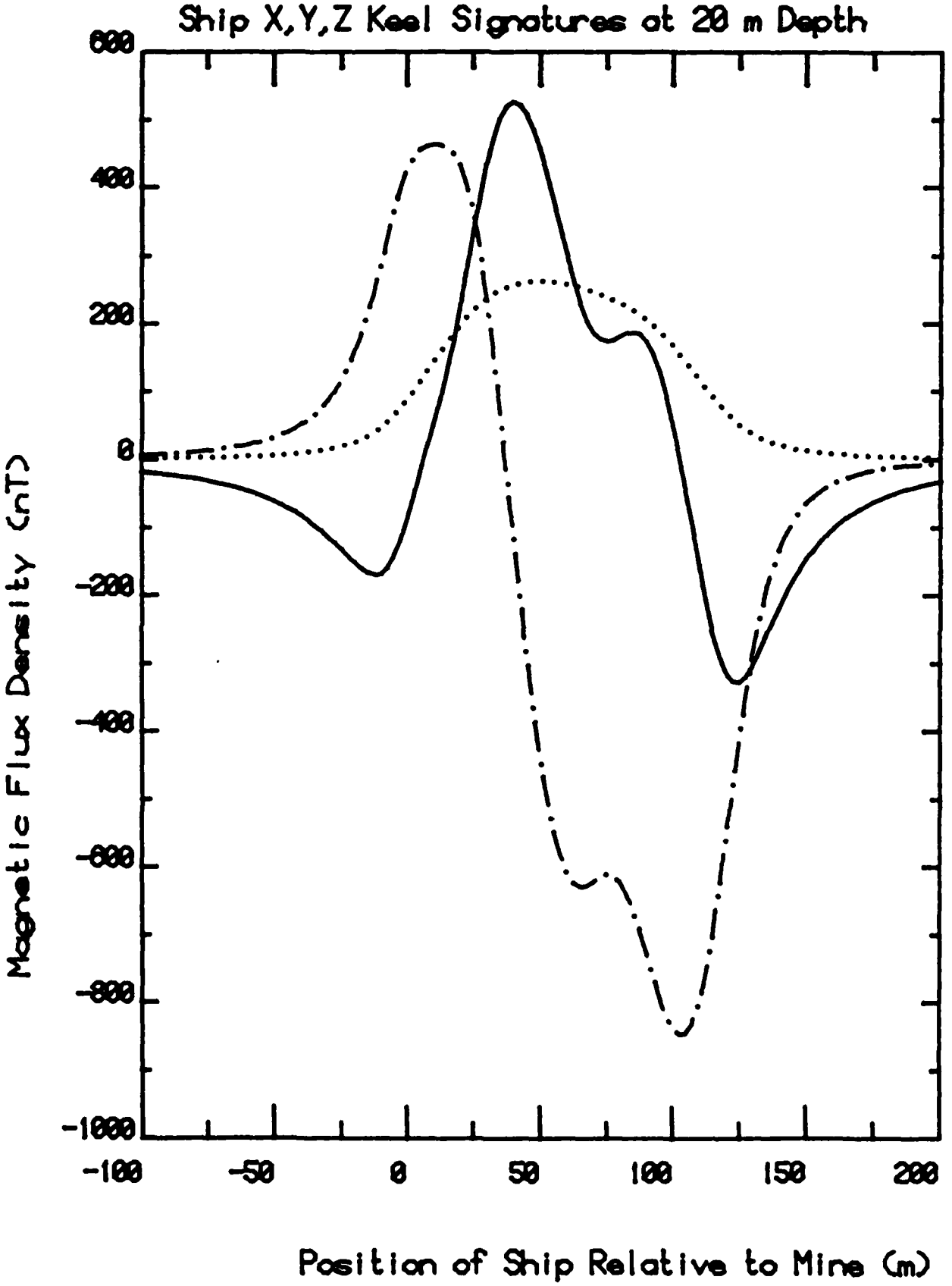

Ship X,Y,Z Keel Signatures at 20 m Depth
600
400
200
0
-200
-400
-600
-800
-1000
-100
-50
0
50
100
150
200
Magnetic Flux Density (nT)
Position of Ship Relative to Mine (m)

FIGURE 2

END

12-86

DTIC