ORNL/TM-10148

ornl

OAK RIDGE NATIONAL LABORATORY

MARTIN MARIETTA

RT-11 FORTRAN Subroutines for X-Y Plotting on Hewlett-Packard 7470A/7475A Graphics Plotters

D. E. Goeringer

OPERATED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
NTIS price codes—Printed Copy: A03; Microfiche A01

ORNL/TM-10148

Contract No. DE-AC05-840R21400

Analytical Chemistry Division

RT-11 FORTRAN Subroutines for X-Y Plotting on Hewlett-Packard 7470A/7475A Graphics Plotters

D. E. Goeringer

Date Published: August 1986

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831
operated by
MARTIN MARIETTA ENERGY SYSTEMS, INC.
for the
U. S. Department of Energy

TABLE OF CONTENTS

RT-11 FORTRAN Subroutines for X-Y Plotting on Hewlett-Packard 7470A/7475A Graphics Plotters

Abstract

A package of FORTRAN IV subroutines has been developed for x-y plotting with Hewlett-Packard 7470A/7475A graphics plotters on Digital Equipment Corporation PDP-11 series computers running the RT-11 operating system. This report describes the function of each subroutine in detail, gives examples of their use, and shows sample output plots.

1. Introduction

Graphics can be very useful for the presentation of data. Because these data are often stored in computers in digital form, digital plotters are well suited as graphics peripheral devices. As the use of personal computers has proliferated, the production of inexpensive graphics plotters with "intelligence" has also increased. One of the most widely used digital plotters in small computer systems is the Hewlett-Packard 7470A/7475A series. The widespread popularity of the IBM PC personal computer and its clones has resulted in the availability of large amounts of 7470A/7475A applications software. These programs enable users to easily utilize the powerful capabilities of these new generation plotters without mastering the plotter command language known as Hewlett Packard Graphics Language (HP-GL). The Digital Equipment Corporation PDP-11 series of computers is also in wide usage, especially among the scientific community. However, there seems to be a dearth of application software for the Hewlett-Packard 7470A/7475A graphics plotters. This report describes a FORTRAN graphics package, QKPLOT, which enables PDP-11 users running the RT-11 operating system to make x-y plots on Hewlett-Packard 7470A/7475A graphics plotters via simple subroutine calls. Because the software consists of a set of FORTRAN subroutines, the user need not know HP-GL and can incorporate the code in application programs by storing the assembled version of QKPLOT in a suitable library.

2. Hardware/Communications/Software Requirements

The subroutines described in this report are specifically for use with the Hewlett-Packard 7470A and 7475A series graphics plotters. Software development was done on an LSI-11/23-Plus microcomputer running RT-11 (Version 5.0) and FORTRAN IV (Version 2.1) with virtual array support. Although the subroutines were not tested on other combinations of hardware/software, they should run with little or no modification on PDP-11 series computers running RT-11 and FORTRAN IV with virtual array support.

The subroutines assume that the plotter is interfaced to the computer via an eavesdrop connection (see plotter manual), i.e., a special cable is inserted in series with the cable linking the console terminal communications port with the terminal. Therefore, all plotter commands are coded as output to the console terminal. If the plotter is to be interfaced to a serial port other than the console terminal, all program lines performing plotter input/output must be modified accordingly.

Communications parameters, i.e., the baud rate, parity, and number of stop bits, for the console terminal RS-232-C serial port must correspond to those established for the graphics plotter. The plotter 'D/Y' rocker switch must also be in position 'Y' so that the plotter will be in the on-line, programmed-off state when powered up. In this mode the plotter continually monitors communication between the computer and terminal, but it does not interpret any computer data as plotter commands until it receives a plotter-on instruction. Subsequent to receiving the plotter-on instruction, the plotter switches to on-line, programmed-on state; all data from the computer are then processed as plotter commands until a plotter-off command is received. Consequently, two rules must be followed when using this subroutine package: (1) all console terminal input/output must be preceded by a plotter-off instruction (2) all QKPLOT subroutine calls must be preceded by a plotter-on instruction. Examples

for including QKPLOT subroutine calls in FORTRAN programs which perform console terminal input/output are given in the description of subroutine COMM and the sample program in Section 4.1.

3. Subroutine Descriptions

Individual subroutines in the QKPLOT package are described in this section. Examples of their use are also included for clarity.

3.1 Subroutine COMM

This subroutine should be called before any other from the QKPLOT package. It issues the plotter-on command, sets the plotter to default conditions with the initialize instruction (see plotter manual), and establishes the following communications parameters:

1. XON/XOFF handshake (XON = DC1, XOFF = DC3)

   The RT-11 Keyboard Monitor uses this protocol for synchronizing data transfer.

2. Normal data transmission mode

   All plotter instructions are put in an execution buffer where they are parsed and executed in order.

3. Disabled monitor mode

   Plotter instructions are not displayed on terminal as they are received or parsed from the buffer.

4. XOFF threshold level = 50

   The plotter will send the XOFF character to the computer when 50 empty bytes remain in the plotter input buffer.

5. Output terminator = CR (carriage return)

The plotter sends CR at the end of each response to a data request from the computer.

6. Echo terminate character = LF (line feed)

   Because the RT-11 monitor echoes all responses from the plotter (full duplex), the echoed data must be ignored by the plotter until the echo terminate character is received. It then begins accepting plotter data again.

7. Turnaround delay = 10 milliseconds

   The plotter will wait 10 ms after receiving a request for data from the computer before it responds.

The example below shows how to perform console terminal input/output via plotter-off/plotter-on commands when using the QKPLOT subroutine package. In the example, the plotter is set up for serial communication and initialized with subroutine COMM. A message is sent to the console terminal, then subroutine QPSIZE from the QKPLOT package is called. After returning from QPSIZE the program prompts the user for input from the terminal. Subroutine PRINT, which is used to send character strings to the terminal, and GETSTR, which is used to get character strings from the terminal, are in the RT-11 subroutine library SYSLIB.

```
LOGICAL*1 ON(4),OFF(4),ANS(2),ERR
DATA ON/"33,'.','Y',"200/,OFF/"33,'.','Z',"200/

      CALL COMM

      CALL PRINT (OFF)
100   TYPE 1000
1000  FORMAT(' TESTING CONSOLE TERMINAL I/O')
```

```
      IOPT = 3
      XLEN = 6.
      YLEN = 4.
      IROT = 0

      CALL PRINT(ON)
      CALL QPSIZE(IOPT,XLEN,YLEN,IROT)

      CALL PRINT (OFF)
200   TYPE 1500
1500  FORMAT(' REPEAT(Y/N)?'$)
      CALL GETSTR(5,ANS,1,ERR)
      IF(ERR)GOTO 200
      IF(ANS(1) .EQ. 'Y')GOTO 100

      STOP
      END
```

3.2 Subroutine QPSIZE(IOPT,XLEN,YLEN,IROT)

Subroutine QPSIZE establishes the physical size and orientation of the plot. Argument IOPT determines the method of entry for axes lengths: 1) selects scaling points P1 and P2 (see plotter manual) in plotter units under QPSIZE control; 2) selects P1, P2 under QPSIZE control via plotter keys P1 and P2; 3) selects numerical size entry in inches from calling program via arguments XLEN and YLEN; 4) selects numerical size entry in plotter units from calling program via arguments XLEN and YLEN. Axes lengths refer to the physical distance between minima and maxima of the abscissa and ordinate. Because the extrema of the axes are slightly offset from the corners of the plot, the actual plot area is slightly larger than that set by the axes lengths. Subroutine arguments for QPSIZE are defined below:

IOPT - Mode of plot size entry (integer)
1 = numerical P1,P2 entry (plotter units) under QPSIZE control
2 = manual P1,P2 entry under QPSIZE control via plotter control keys
3 = numerical size entry (inches) under calling program control via arguments XLEN and YLEN
4 = numerical size entry (plotter units) under calling program control via arguments XLEN and YLEN
Note: When IOPT=3 or 4, the plot is automatically centered.

XLEN - Abscissa length in selected units (real)
YLEN - Ordinate length in selected units (real)
Note: Use dummy arguments for XLEN and YLEN when IOPT = 1 or 2.

The last argument, IROT, defines the orientation of the plotter coordinate system. However, the plotter commands 'RO 0' and 'RO 90', which perform this function, are only present in the 7475A. Therefore, 7470A users must modify subroutines QPAXES by removing the argument IROT and all references to 'RO 0' and 'RO 90' plotter commands.

IROT - orientation of coordinate system (integer)
0 = no rotation
1 = 90 degree rotation
Note: See manual for definite of coordinate system rotation.

An example of QPSIZE use is presented in the description of subroutine COMM. It sets an abscissa length of 6 inches and an ordinate length of 4 inches via program-controlled numerical mode. The plot is not rotated.

3.3 Subroutine QPAXES(XMIN,XMAX,YMIN,YMAX,XLABEL,YLABEL, TITLE)

Subroutine OPAXES calculates the tick mark values and spacing, draws and labels the axes and tick marks, and scales the axes for the plot. After subroutine QPAXES has been executed, the plotter coordinate system is mapped in user units so subsequent data can be plotted in units convenient for the application.

QPAXES calculates minima and maxima for the abscissa and ordinate from the argument values (XMIN, XMAX, YMIN, YMAX) defined below. However, the calculated tick mark values for the extrema of the axes may not exactly correspond to those values input via the scaling arguments. Therefore, the actual values for the tick marks at the ends of both axes are returned to their respective arguments (XMIN,XMAX,YMIN,YMAX). These actual values are useful if the user desires to define a plotting window (see subroutine QPWIND). After user-unit scaling is established, non-integer data values such as 123.456 can be plotted. Data (user units) sent to the plotter are limited to the range -32768 to 32767. Therefore, to plot values outside these limits, data must be reduced to acceptable ranges by arithmetic conversion.

The axes can be labeled by storing character strings in byte arrays corresponding to arguments XLABEL (abscissa) and YLABEL (ordinate). A title may also be drawn above the plot by using a byte array corresponding to argument TITLE. The arguments XLABEL, YLABEL, and TITLE are dimensioned in subroutine QPAXES as shown below:

```
BYTE XLABEL(1),YLABEL(1),TITLE(1)
```

Therefore, the length of each character string is determined by its array dimension in the calling program. Definitions for QPAXES arguments are given below:

XMIN - minimum value for abscissa data (real)
XMAX - maximum value for abscissa data (real)
YMIN - minimum value for ordinate data (real)
YMAX - maximum value for ordinate data (real)
Note: The actual values calculated for the corresponding tick marks are returned in the respective arguments.

XLABEL - character string for abscissa label (byte array)
YLABEL - character string for ordinate label (byte array)
TITLE - character string for plot title (byte array)
Note: Each character string must be terminated with a null byte.

An example program section using QPAXES is listed below. The corresponding plot is shown in Figure 1. The program code scales the abscissa (x-axis) from 0 to 5 and the ordinate (y-axis) from 0 to 10. The x-axis and the y-axis are labeled 'X-axis' and 'Y-axis', respectively, and the plot is titled 'Test Plot'. Subroutine SCOPY, which copies a character string into an array, is in the RT-11 subroutine library SYSLIB.

```
BYTE XLABEL(10),YLABEL(10),TITLE(10)

XMIN = 0.
XMAX = 5.
YMIN = 0.
YMAX = 10.

CALL SCOPY('X-axis',XLABEL)
CALL SCOPY('Y-axis',YLABEL)
CALL SCOPY('Test Plot',TITLE)

CALL QPAXES(XMIN,XMAX,YMIN,YMAX,XLABEL,YLABEL,TITLE)
```

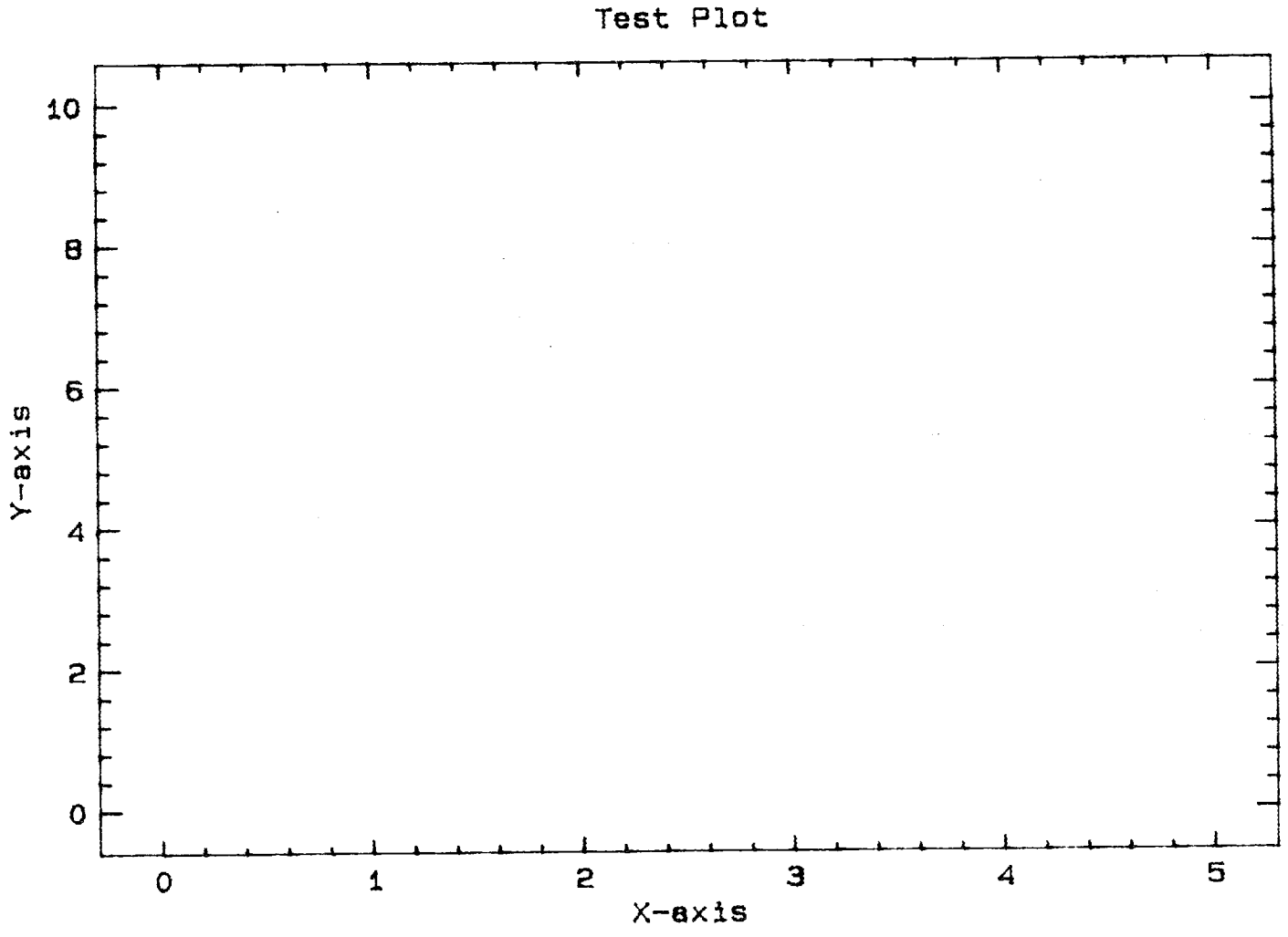

Figure 1

3.4 Subroutine QPWIND(IOPT,X1W,Y1W,X2W,Y2W)

Subroutine QPWIND enables the user to restrict plotting to a defined window, i.e., to restrict programmed pen motion to a rectangular area of the plotting surface called a window. Argument IOPT is used as a switch for the window: 0) off, 1) on. Arguments X1W, Y1W, X2W, and Y2W delineate limits for the window; they are in user units, so scaling must be in effect before using this subroutine. Scaling is activated by calling subroutine QPAXES. QPWIND argument definitions are given below:

IOPT - Window on/off switch (integer)
0 = turn window off with the maximum plotting area available for pen motion
1 = turn window on with limits set by arguments given below

X1W - Minimum abscissa value in user units (real)
Y1W - Minimum ordinate value in user units (real)
X2W - Maximum abscissa value in user units (real)
Y2W - Maximum ordinate value in user units (real)
Note: Use dummy arguments for window limits when IOPT = 0.

The programming example below establishes a plotting window from 0.5 to 4.0 for the abscissa and from 2.5 to 7.5 for the ordinate. It assumes the plot axes have been scaled by the program code shown in the example for subroutine QPAXES.

```
IOPT = 1
X1W  = 0.5
X2W  = 4.0
Y1W  = 2.5
Y2W  = 7.5

CALL QPWIND(IOPT,X1W,Y1W,X2W,Y2W)
```

3.5 Subroutine PLOT(NPTS,X,Y,IOPT,SYM)

This subroutine enables the user to plot a set of x-y data pairs and optionally connect the points with a line selected from a set of seven types. PLOT assumes that the axes have been scaled, so the data must be input in user units previously defined by subroutine QPAXES. The symbol plotted at each point, assigned by argument SYM, may be any ASCII printing character except ';' (see range below). A null character (octal 000) will only draw a dot at each point. Arguments X and Y are real arrays containing data to be plotted and are dimensioned in subroutine PLOT as shown below:

```
VIRTUAL X(1),Y(1)
```

Therefore, the size of the arrays containing data to be plotted is determined by the calling program. Additionally, the data arrays must be dimensioned with VIRTUAL statements. The number of data points to be plotted is passed to PLOT via argument NPTS. Arguments for PLOT are defined in more detail below:

NPTS - Number of x-y data pairs to plot (integer)

X - Array of points for abscissa (real,virtual)
Y - Array of points for ordinate (real,virtual)
Note: The data must be in user units and stored in virtual arrays.

IOPT - Line type for connecting points (integer)
0 = no line
1-6 = see plotter manual
7 = solid line

SYM - ASCII code for symbol to be drawn and centered on each point (byte)
Note: This can be any printing character in the range decimal 33-126, except 59 (';'). A null character, decimal 0, will only draw a point.

This programming example assumes the axes have been scaled as in the QPAXES example and a window established as in the QPWIND example. Six x-y data pairs are in the data set to be plotted. Because a window is in effect, only two points will be drawn. Each of these points is to be marked with the symbol '+' and connected with line type 1 (dotted line). The resulting plot is shown in Figure 2.

```
      VIRTUAL X(10),Y(10)
      BYTE SYM,NOTE(20)

      DO 100 J = 0,5
      X(J+1) = J
      Y(J+1) = 2*J
100   CONTINUE

      LINTYP = 1
      SYM = '+'
      NPTS = 6

      CALL PLOT(NPTS,X,Y,LINTYP,SYM)

      MODE = 1
      XPOS = 4.
      YPOS = 5.
      CALL SCOPY('ANNOTATION TEST',NOTE)

      CALL QPNOTE(MODE,XPOS,YPOS,NOTE)
```

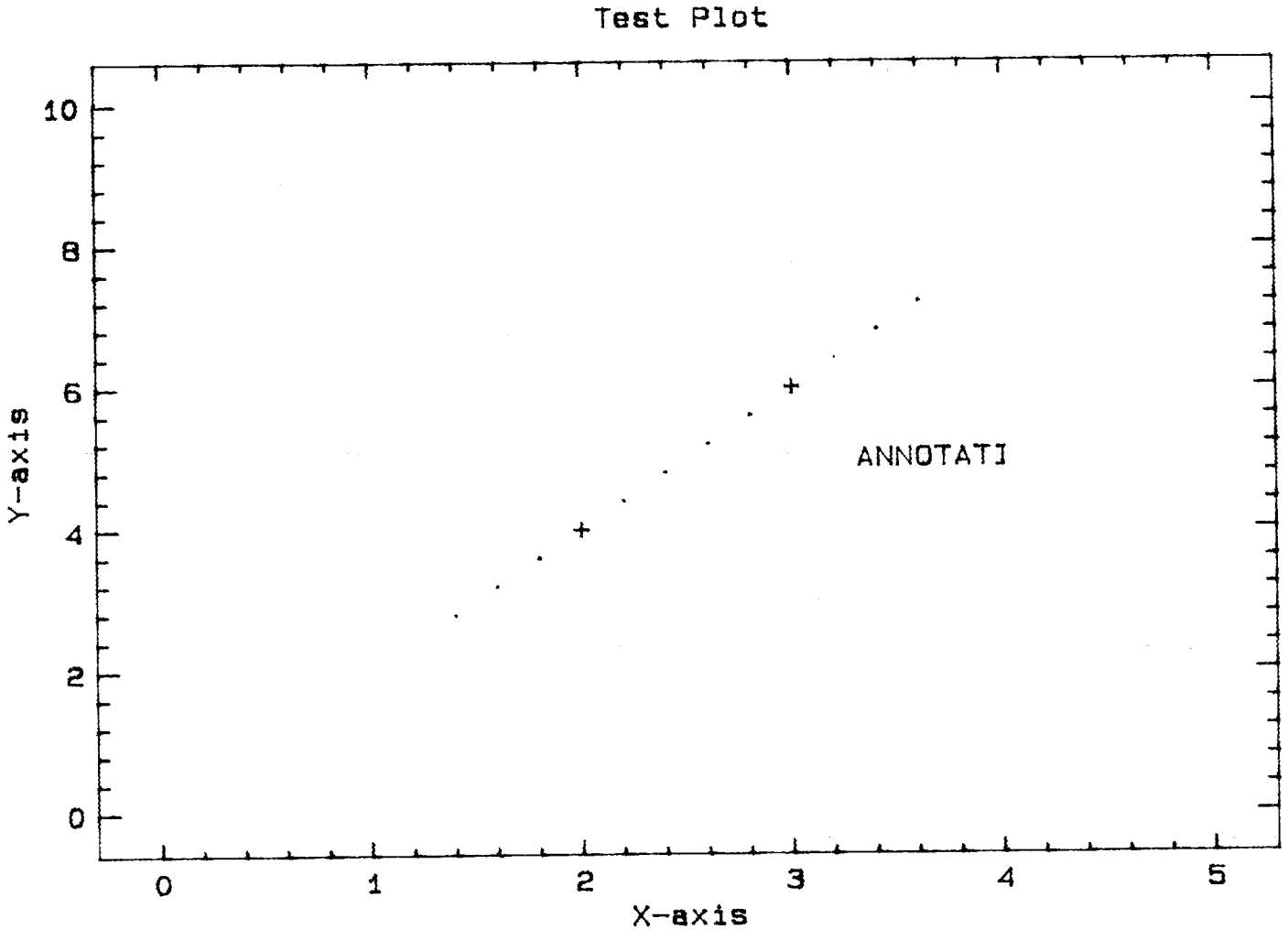

Figure 2

3.6 Subroutine QPNOTE (MODE,XPOS,YPOS,NOTE)

Subroutine QPNOTE enables the user to annotate the plot at a selected position. Argument MODE determines the method for selecting the center position for the annotation: 0) location manually set under QPNOTE control via plotter P1 and P2 controls; 1) location numerically set under calling program control via arguments XPOS and YPOS. The coordinates for the annotation center are returned in arguments XPOS and YPOS when manual positioning is used. When using program-controlled centering the user transfers numerical values for the annotation center in those same arguments. If scaling is in effect, the values for XPOS and YPOS are in user units; otherwise they are in plotter units. The annotation character string is passed in the argument NOTE. It is dimensioned as a byte array in QPNOTE as shown below:

```
BYTE NOTE(1)
```

Therefore, the length of the annotation is determined by the dimension of its byte array in the calling program. Definitions of the arguments are given below:

MODE - Method for positioning annotation center (integer)
0 = manual entry under QPNOTE control via plotter P1 and P2 controls with coordinate values returned in arguments XPOS and YPOS
1 = numerical entry under calling program control via arguments XPOS and YPOS

XPOS - Abscissa coordinate for center of annotation (real)
YPOS - Ordinate coordinate for center of annotation (real)
Note: The coordinates are in user units if scaling is in effect, otherwise in plotter units.

NOTE - Character string for annotation (byte array)
Note: The string must be terminated by a null byte.

An example of the use of QPNOTE is shown in the description of subroutine PLOT. It assumes scaling and windowing are in effect as set by the code shown for subroutines QPAXES and QPWIND. The character string 'ANNOTATION TEST' is to be centered at coordinate x=4,y=5. Figure 2 shows a plot corresponding to the programming example. Note that because windowing is in effect, only a portion of the string is actually drawn.

4. Programming/Plot Examples

The FORTRAN program listed below demonstrates the use of the QKPLOT subroutine package for general purpose plotting. It is followed by an example printout of the dialog between the computer and user during actual operation. Figure 3 shows the plotter output resulting from the parameters selected in the example.

4.1 General Purpose FORTRAN Plotting Program

```
      PROGRAM XYPLOT

      VIRTUAL X(50),Y(50)
      BYTE SPEN1(5),DSPEN(5),ON(4),OFF(4),SYM(2),ANS(2),ERR

      DATA SPEN1/'S','P','1',';',"200/,ON/"33,'.','Y',"200/,
      +OFF/"33,'.','Z',"200/,DSPEN/'S','P','0',';',"200/

C
C     ESTABLISH COMMUNICATION PARAMETERS
C

      CALL COMM
C
C     INPUT PLOT SIZE AND ORIENTATION
C
100   CALL PRINT(OFF)
      TYPE 3000
3000  FORMAT('INPUT PLOT SIZE OPTION'/' 1)P1,P2 (PLOTTER UNITS)
      +2)P1,P2 MANUAL'/' 3)LENGTH (INCHES) 4)LENGTH (PLOTTER UNITS):'$)
      ACCEPT *,ISIZE
      IF(ISIZE .EQ. 1 .OR. ISIZE .EQ. 2)GOTO 220
```

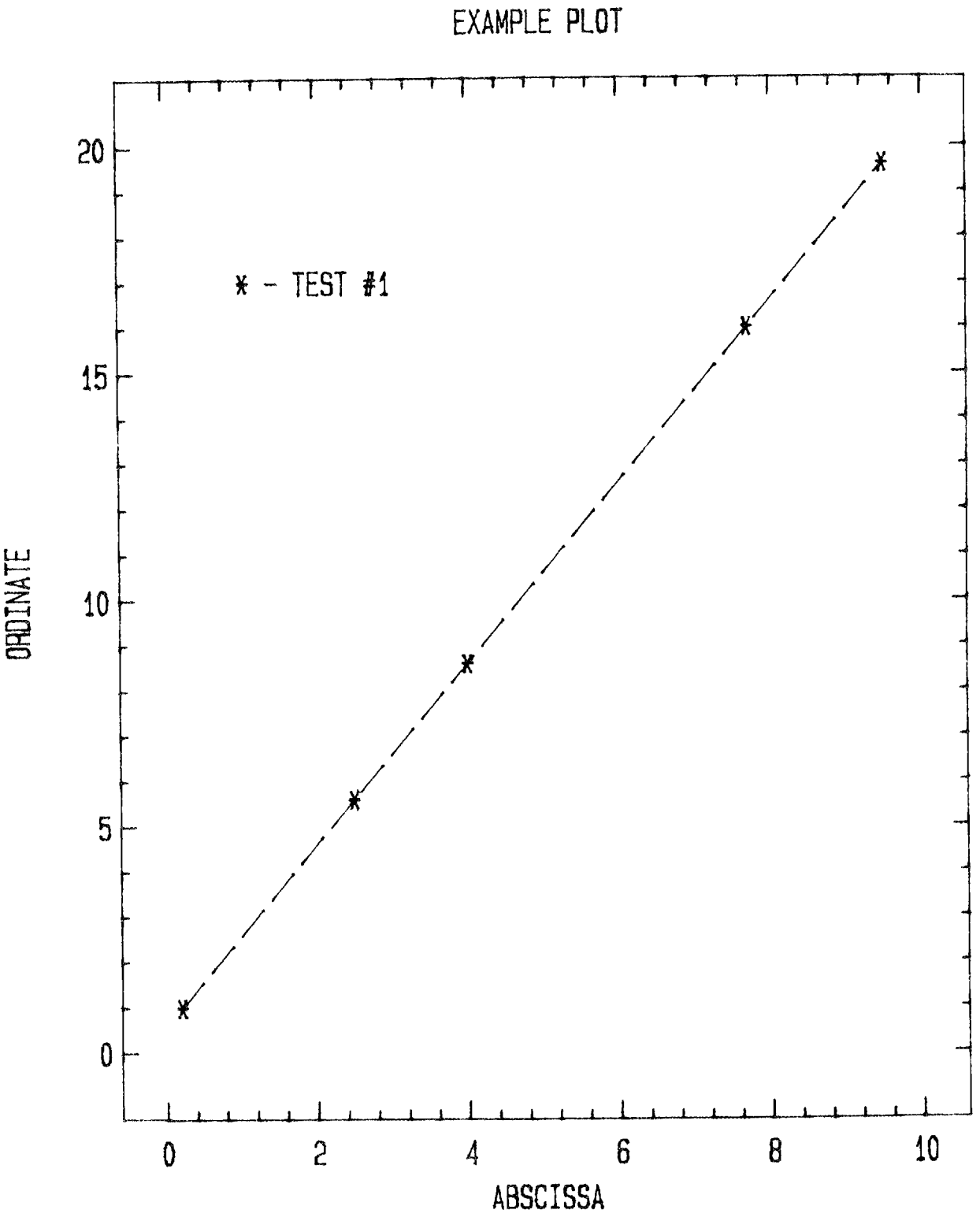

Figure 3

```
      IF(ISIZE .EQ. 3)TYPE 3210
3210  FORMAT(' ABSCISSA LENGTH (INCHES):'$)
      IF(ISIZE .EQ. 4)TYPE 3220
3220  FORMAT(' ABSCISSA LENGTH (PLOTTER UNITS):'$)
      ACCEPT *,XLEN

      IF(ISIZE .EQ. 3)TYPE 3260
3260  FORMAT(' ORDINATE LENGTH (INCHES):'$)
      IF(ISIZE .EQ. 4)TYPE 3270
3270  FORMAT(' ORDINATE LENGTH (PLOTTER UNITS):'$)
      ACCEPT *,YLEN

220   TYPE 3300
3300  FORMAT(' 0)NO ROTATION OR 1)90 DEGREE ROTATION:'$)
      ACCEPT *,IROT

C
C     SET PLOT SIZE AND ORIENTATION
C
      CALL PRINT(ON)
      CALL QPSIZE(ISIZE,XLEN,YLEN,IROT)
      CALL PRINT(OFF)
C
C     INPUT ABSCISSA, ORDINATE, AND TITLE LABELS
C
240   TYPE 3400
3400  FORMAT(' ABSCISSA LABEL (<26 CHARS):'$)
      CALL GETSTR(5,XLABEL,25,ERR)
245   TYPE 3450
3450  FORMAT(' ORDINATE LABEL (<26 CHARS):'$)
      CALL GETSTR(5,YLABEL,25,ERR)
250   TYPE 3500
3500  FORMAT(' PLOT TITLE (<26 CHARS):'$)
      CALL GETSTR(5,TITLE,25,ERR)
```

```
C
C       INPUT MINIMUM AND MAXIMUM VALUES
C       FOR X AND Y DATA SET
C
300     TYPE 4000
4000    FORMAT(' XMIN,XMAX:'$)
        ACCEPT *,XMIN,XMAX
        TYPE 4050
4050    FORMAT(' YMIN,YMAX:'$)
        ACCEPT *,YMIN,YMAX
C
C       CALCULATE TICK MARKS, DRAW AND LABEL AXES,
C       AND SCALE AXES IN USER UNITS
C
325     CALL PRINT(ON)
        CALL PRINT(SPEN1)
        CALL QPAXES(XMIN,XMAX,YMIN,YMAX,XLABEL,YLABEL,TITLE)
        CALL PRINT(OFF)
C
C       DEFINE WINDOW IF DESIRED
C
        TYPE 4200
4200    FORMAT(' DEFINE WINDOW (Y/N)?'$)
        CALL GETSTR(5,ANS,1,ERR)
        IF(ANS(1) .NE. 'Y')GOTO 350
        TYPE 4250
4250    FORMAT(' XMIN,YMIN FOR WINDOW:'$)
        ACCEPT *,X1W,Y1W
        TYPE 4300
4300    FORMAT(' XMAX,YMAX FOR WINDOW:'$)
        ACCEPT *,X2W,Y2W
C
C       INPUT X-Y DATA
C
```

```
350     TYPE 4400
4400    FORMAT(' # X-Y DATA PAIRS (<51):'$)
        ACCEPT *,NPTS
        DO 375 J=1,NPTS
        TYPE 4450
4450    FORMAT(' X('I2'),Y('I2'):'$)
        ACCEPT *,X(J),Y(J)
375     CONTINUE
C
C       INPUT PLOTTING SYMBOL AND LINE TYPE
C
400     TYPE 4600
4600    FORMAT(' PLOTTING SYMBOL:'$)
        CALL GETSTR(5,SYM,1,ERR)
        TYPE 4650
4650    FORMAT(' LINE TYPE (0-7):'$)
        ACCEPT *,LINTYP
C
C       PLOT DATA
C
        CALL PRINT(ON)
        CALL PLOT(NPTS,X,Y,LINTYP,SYM(1))
        CALL PRINT(OFF)
C
C       ANNOTATE PLOT IF DESIRED
C
425     TYPE 4800
4800    FORMAT(' ANNOTATE PLOT (Y/N)?'$)
        CALL GETSTR(5,ANS,1,ERR)
        IF(ANS(1) .NE. 'Y')GOTO 600
        TYPE 4850
4850    FORMAT(' ANNOTATION (<26 CHARS):'$)
        CALL GETSTR(5,NOTE,25,ERR)
        TYPE 4900
```

```
4900    FORMAT(' 0)MANUAL OR 1)NUMERICAL POSITIONING:'$)
        ACCEPT *,MODE
        IF(MODE .EQ. 1)GOTO 450
        PAUSE 'POSITION PEN, HIT RETURN'
        GOTO 475
450     TYPE 4950
4950    FORMAT(' X,Y COORDINATE FOR CENTER:'$)
        ACCEPT *,XPOS,YPOS
475     CALL PRINT(ON)
        CALL QPNOTE(MODE,XPOS,YPOS,NOTE)
        CALL PRINT(OFF)
500     TYPE 5000
5000    FORMAT(' NEW PLOT (Y/N)?'$)
        CALL GETSTR(5,ANS,1,ERR)
        IF(ANS(1) .EQ. 'Y')GOTO 100
        CALL PRINT(DSPEN)
        END
```

4.2 Example Program Dialog

The following section is a printout of the dialog between the computer and user for the sample program listed above. The user responses are underlined.

```
.RUN XYPLOT
INPUT PLOT SIZE OPTION
1)P1,P2 (PLOTTER UNITS) 2)P1,P2 MANUAL
3)LENGTH (INCHES) 4)LENGTH (PLOTTER UNITS):3

ABSCISSA LENGTH (INCHES):5

ORDINATE LENGTH (INCHES):6

0)NO ROTATION OR 1)90 DEGREE ROTATION:1
```

```
ABSCISSA LABEL (<26 CHARS):ABSCISSA

ORDINATE LABEL (<26 CHARS):ORDINATE

PLOT TITLE (<26 CHARS):EXAMPLE PLOT

XMIN,XMAX:0.0,10.

YMIN,YMAX:0,20

DEFINE WINDOW (Y/N)?N

# X-Y DATA PAIRS (<51):5

X( 1),Y( 1):.2,1

X( 2),Y( 2):2.5,5.6

X( 3),Y( 3):4,8.6

X( 4),Y( 4):7.7,16

X( 5),Y( 5):9.5,19.6

PLOTTING SYMBOL:*

LINE TYPE (0-7):3

ANNOTATE PLOT (Y/N)?Y

ANNOTATION (<26 CHARS):* - TEST #1

0)MANUAL OR 1)NUMERICAL POSITIONING:1
```

X,Y, COORDINATE FOR CENTER:2,17

NEW PLOT (Y/N)?N

Acknowledgment

I thank Peter Todd for the procedure to calculate tick mark positions and format axes labels.

****** WARNING ******

Internal Distribution

1. J. A. Carter
2. W. H. Christie
3. G. L. Glish
4-8. D. E. Goeringer
9. J. M. Keller
10. L. N. Klatt
11. H. S. McKown
12. S. A. McLuckey
13. T. R. Mueller
14. J. M. Ramsey
15. W. D. Shults
16. D. H. Smith
17. P. J. Todd
18. R. E. Valiga
19. R. L. Walker
20. W. B. Whitten
21. A. Zucker
22. Central Research Library
23. Document Reference Section
24-25. Laboratory Records Department
26. Laboratory Records Department - RC
27. ORNL Patent Office

External Distribution

28. Office of Assistant Manager for Energy Research and Development, Department of Energy, Oak Ridge Operations Office, Oak Ridge, TN 37831
29-55. Technical Information Center, Oak Ridge, TN 37831