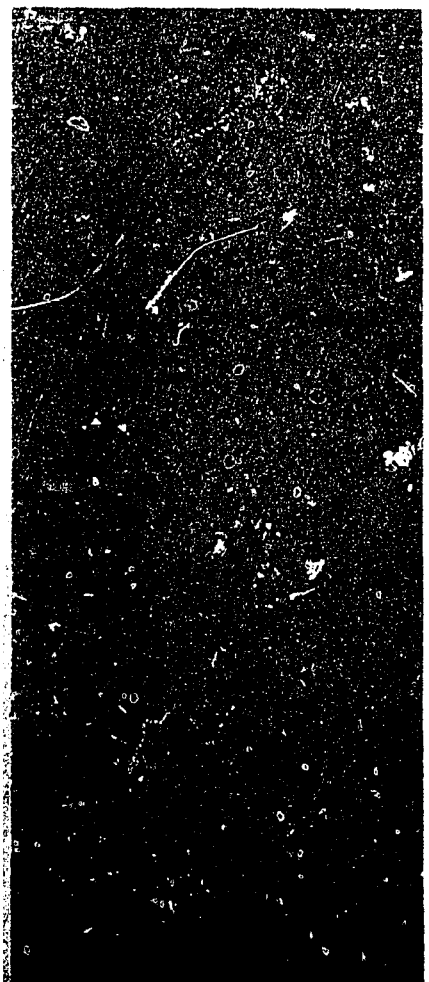
ORNL/TM-6546

TEK11 Graphics User's Guide

C. R. Stewart, Jr.
W. D. Joubert
D. R. Overbey
K. A. Stewart



OAK RIDGE NATIONAL LABORATORY
OPERATED BY UNION CARBIDE CORPORATION · FOR THE DEPARTMENT OF ENERGY

ORNL/TM-6546
Dist. Category UC-20

Contract No. W-7405-eng-26

FUSION ENERGY DIVISION

TEK11 GRAPHICS USER'S GUIDE

C. R. Stewart, Jr., W. D. Joubert,
D. R. Overbey, and K. A. Stewart

October 1978

NOTICE This document contains information of a preliminary nature. It is subject to revision or correction and therefore does not represent a final report.

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37830
operated by
UNION CARBIDE CORPORATION
for the
DEPARTMENT OF ENERGY

CONTENTS

ABSTRACT

The TEK11 graphics library was written for use on PDP-11 minicomputers running the RT-11 operating system to drive Tektronix 4010 graphics display terminals. Library subroutines are coded in FORTRAN and assembly language. The library includes routines to draw axes, either linear or semilog, to plot data in terms of logical values without first scaling to screen coordinates, to label graphs, and to plot in a maximum of four regions on the screen. Modes of plotting may be point plot with any character at the point, vector plot, or bar plot. Two features, automatic scaling and windowing, permit the researcher to use computer graphics without spending time first to learn about scaling or "Tek points" and preparing long parameter lists for subroutines. Regions on the screen are defined by specifying minima and maxima logical coordinates, i.e., °K or milliseconds, and a region number. After definition, a region may be activated for plotting by calling REGN with the region number as an argument.

1. INTRODUCTION

The TEK11 graphics library is intended for use on PDP-11 minicomputers driving Tektronix 4010 series graphics display terminals. Library subroutines are coded in FORTRAN and assembly language. The library includes routines to draw axes, either linear or semilog; to plot data in terms of logical values without first scaling to screen coordinates; to label graphs; and to plot into a maximum of four regions on the screen. Modes of plotting may be point plot with any character at the point, vector plot, or bar plot. Data arrays may contain either integer or real numbers. Plotting integer-type data, such as output from a scalar or ADC, saves memory and CPU time. Data may also be plotted in an autoincrement mode, in which either the x or y position for a point is automatically incremented after each point is plotted. Two features, automatic scaling and windowing, permit the researcher to use computer graphics without spending time first to learn about scaling or "Tek points" and preparing long parameter lists for subroutines. Regions on the screen are defined simply by specifying minima and maxima logical coordinates and may be activated for plotting by calling REGN with the region number as an argument.

To clarify use of the TEK11 graphics library, several liberally commented, self-explanatory sample programs are listed in Appendix A. Every attempt has been made to structure this library for ease of use while still permitting a great deal of flexibility for the user. The authors welcome suggestions for improvement.

BLANK PAGE

2. USER PROGRAM STRUCTURE

The TEK11 graphics library is very user-oriented and requires only a minimum knowledge of computer graphics. Most of the parameters required to produce a graph, such as plotting boundaries, scaling type, etc., are initialized to default values; subroutines are available to change these values, allowing the user a wide margin of flexibility.

In using the graphics library, certain plotting routines must be called in a given order. A typical procedure for producing a plot is:

1.	Initialize graphics system	CALL INITT
2.	Set the range of logical coordinates to be used for plotting, e.g., 0. < x < 15. and 0. < y < 100.	CALL SETMM (arguments)
3.	Perform preliminary operations, e.g., draw axes, labels	
4.	Plot data	CALL PLOT, PLOTX, OR PLOTY

Keep in mind that this is only an elementary outline; the TEK11 graphics system is flexible enough to allow any reasonable sequence of graphics subroutine calls.

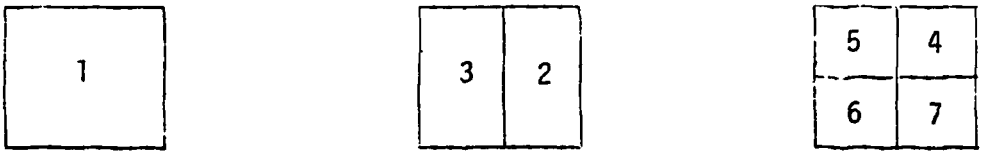
In step 3 above, the user has the option of tailoring the graphics facilities to suit his needs. This is done by modifying the state of the graphical environment (see Sect. 3). The following program and output illustrate the fundamentals of the graphics system.

```
      PROGRAM SINE
      DIMENSION X(46), Y(46)
      LINEAR = 0
C
C INITIALIZE GRAPHICS
      CALL INITT
C
C ERASE SCREEN
      CALL ERASE
C
C SET THE RANGE OF DATA VALUES TO BE PLOTTED
      XMIN = 0.0
      XMAX = 15.0
      YMIN =  -1.5
      YMAX = 1.5
      CALL SETMM(XMIN,XMAX,YMIN,YMAX)
C
C DRAW A SET OF AXES.  PASS A LABEL AND ITS LENGTH IN CHARACTERS
C TO THE SUBROUTINE, AS WELL AS SCALING TYPE
      NCHARS = 11
      CALL AXES('SINE VALUES',NCHARS,LINEAR)
C
C AND A GRID
      CALL GRID
C
C PUT THE SINE VALUES INTO THE ARRAYS
      DO 1   I = 1,46
          X(I) = (I-1)/3.0
          Y(I) = SIN(X(I))
    1 CONTINUE
C
C PLOT THE 46 (X,Y) PAIRS
      CALL PLOT(X,Y,46)
C
      STOP
      END
*
```

3. SCREEN REGIONS

The Tektronix screen limits are $0 < x < 1023$, $0 < y < 780$. When a plot is made, the displayed data values must be mapped into this screen so that a reasonable looking graph is produced. Consequently, the actual x and y values in the data need not (and usually do not) coincide with the screen coordinate system. Actual data values, which may be in units such as msec or percentage points, are known as logical values. A square defined by minimum and maximum x and y logical values can be scaled and placed anywhere on the display screen. This is called a window. Coordinate axes and numbers always fall directly outside the window.

Logical boundaries for the user's data are set by SETMM, and screen coordinates for the window are set by SWINDO. There are seven system-defined windows available.



Region 1 is the default window. (Region numbers are not associated with user-defined regions.)

Using a system-defined region is a three-step process. First, SETMM is called to set the logical boundaries of the graph. Second, WINDOW is called to associate the current set of logical boundaries with a given system window. This association lasts throughout the program, or until WINDOW is called again with the same region number. Note that WINDOW does not affect which region is currently being used for plotting. Finally, a window and its associated logical extents are activated by calling REGN

with the desired region number. This replaces any previous user-defined window and logical boundaries, e.g.,

```
          .
          .
          .
          .
          .
      CALL INITT
C
C  SET UP SOME WINDOWS
      CALL STEMM (-300., 300., 0., 250.)
      CALL WINDOW (4)
      CALL SETMM (0., 100., 01., 1.)
      CALL WINDOW (1)
        .
        .
        .
        .
      CALL REGN (4)
      CALL AXES ("PLOT", 4, 0)
      CALL PLOT (X, Y, N)
```

The call to REGN establishes system window 4 as the current plotting region. The x axis values for this plot will range from -300 to 300 and the y axis values from 0 to 250.

4. PROGRAMMING CONSIDERATIONS

The procedure for running a program using TEK11 varies from system to system, but in general the FORTRAN main program must be linked with the OBJ file "TEKLIB." For details concerning a particular site, consult system personnel.

Labeling of the display can be cone in two ways. The simplest way is to position the cursor at the desired position using LOCATE and displaying the string with a WRITE statement. However, for graphics terminals not addressable with FORTRAN unit numbers, NOTATE must be used, perhaps in conjunction with character string encoding.

5. SUBROUTINES

This section contains descriptions of the TEK11 subroutines. Subroutines are grouped by function. See Sect. 4 for methods of linking the TEK11 library into user programs.

5.1 PARAMETER SETTING ROUTINES

The following routines modify the graphics system environment. This environment is defined by the following parameters:

number type of input data	logical (data value) boundaries
plot mode	window boundaries
data point character	current window number
scaling type	

Windowing is discussed in Sect. 3.

INITT

Usage: CALL INITT

Function: Initializes the TEK11 graphics system, setting system parameters to their defaults. This must be the first call to the TEK11 system.

MODE

Usage: CALL MODE (ITYPE, ICHAR)

Function: Sets the plot modes for subsequent plots as well as the character to be plotted at each point while in point-plot mode.

Arguments: ITYPE – mode of plotting
=1 - vector plot (default). Points connected with line segements.

=2 - point plot. Each data point plotted as a single character at the point.

=3 - bar plot. Data is plotted in bar-chart fashion.

ICHAR – character used in point-plot mode, left-adjusted in data word.

NUM

Usage: CALL NUM (NUMTYP)

Function: Sets the data type for the arrays passed to PLOT, PLOTX and PLOTY.

Arguments: NUMTYP — data type of numbers
=0 - I integer,

=1 - (default).

SETMM

Usage: CALL SETMM (XMIN, XMAX, YMIN, YMAX)

Function: Sets the range of logical data values to be plotted. Since the values passed by this subroutine have no defaults, this routine must be called before any calls to axis or plotting routines.

Arguments: XMIN — minimum x value,

XMAX — maximum x value,

YMIN — minimum y value,

YMAX — maximum y value.

SCALE

Usage: CALL SCALE (ISCALE)

Function: Sets the type of scaling to be used for subsequent plots. (The scaling type may also be set by a call to AXES.)

Argument: ISCALE — type of scaling
=0 - linear (default),

=1 - semilog (linear x and log y).

5.2 WINDOW ROUTINES

See Sect. 3 for a full discussion of windowing.

SWINDO

Usage: CALL SWINDO (MINX, MINY, MAXX, MAXY)

Function: Sets the boundaries of the screen area where subsequent plotting is to take place. Default is the entire screen.

Arguments: MINX – leftmost x screen coordinate,

MINY – lowest y screen coordinate,

MAXX – rightmost x screen coordinate,

MAXY – topmost y screen coordinate.

WINDOW

Usage: CALL WINDOW (IREGN)

Function: Sets the logical boundaries associated with the specified system-defined region to the values of the current logical boundaries. Must be called before activating the desired region. See Sect. 3.

Argument: IREGN – region number, chosen between 1 and 7 inclusive.

REGN

Usage: CALL REGN (IREGN)

Function: Activates the specified system window by establishing that window's screen and logical boundaries as those to be used in subsequent plots. Must be preceded by a call to WINDOW.

Argument: IREGN – region number, between 1 and 7 inclusive.

5.3 AXIS ROUTINES

AXES

Usage: CALL AXES (XLABEL), NCHARS, ISCALE)

Function: Sets the scaling type, draws a box around the window, draws x and y axes, and puts a label above the top left corner of the window.

XAXIS

Usage: CALL XAXIS (XLABEL, NCHARS)

Function: Draws the x axis and labels it.

YAXIS

Usage: CALL YAXIS

Function: Draws a y axis.

BOX

Usage: CALL BOX

Function: Draws a grid covering the view window.

ORIGIN

Usage: CALL ORIGIN

Function: Draws a horizontal and vertical line across the window and through the logical origin of the plot

5.4 PLOT ROUTINES

The following routines plot a set of data, interpreting the arrays as either real (default) or integer. Each routine is subject to all plot parameters listed in Sect. 5.1.

PLOT

Usage: CALL PLOT (X, Y, N)
or CALL PLOT (IX, IY, N)

Function: Plots [x(i), y(i)] for i = 1 to N. MODE controls how the data are plotted.

Arguments: X, IX — independent variable, dimensioned to any size,

Y, IY — dependent variable, dimensioned to same size, as x,

N — number of points plotted.

PLOTX

Usage: CALL PLOTX (X, Y, N)
or CALL PLOTX (IX, IY, N)

Function: Plots x data, autoincrementing y from the given start value; i.e., plots [x(i), y+i-1] for i = 1 to N.

Arguments: X, IX — data array,

Y, IY — starting y value,

N — number of points plotted.

PLOTY

Usage: CALL PLOTY (X, Y, N)

Function: Plots y data, autoincrementing x from the given start value; i.e., plots [y+i-1, x(i)] for i = 1 to N.

Arguements: X, IX — starting x value,

Y, IY — data array,

N — number of points plotted.

5.5 TERMINAL CONTROL ROUTINES

The following subroutines directly control the state or action of the user's terminal. Unless otherwise noted, these routines may be called at any time after the call to INITT.

ALPHA

Usage: CALL ALPHA

Function: Causes the terminal to enter alphanumeric mode.

ERASE

Usage: CALL ERASE

Function: Erases the terminal screen.

HCOPY

Usage: CALL HCOPY

Function: Produces a hard copy of the screen display.

HOME

Usage: CALL HOME

Function: Positions cursor at the upper left corner of the screen.

LOCATE

Usage: CALL LOCATE (IX, IY)

Function: Positions the cursor at an absolute screen location.

Arguments: IX — x screen coordinate,

IY — y screen coordinate.

PAUSE

Usage: CALL PAUSE

Function: Moves cursor to home, outputs an asterisk to the terminal, and pauses. Continue program by typing <carriage return>.

5.6 MISCELLANEOUS ROUTINES

LINE

Usage: CALL LINE (IX, IY)

Function: Draws a line from the current cursor position to point (IX, IY).

NOTATE

Usage: CALL NOTATE (LABEL, IX, IY, NCHARS)

Function: Writes a label at a given screen position.

Arugements: LABEL — label array. Two characters per 16-bit word.

IX — the x screen position of the leftmost edge of the character string.

IY — the y screen position of the first character.

NCHARS — the number of characters in the label.

WDATE

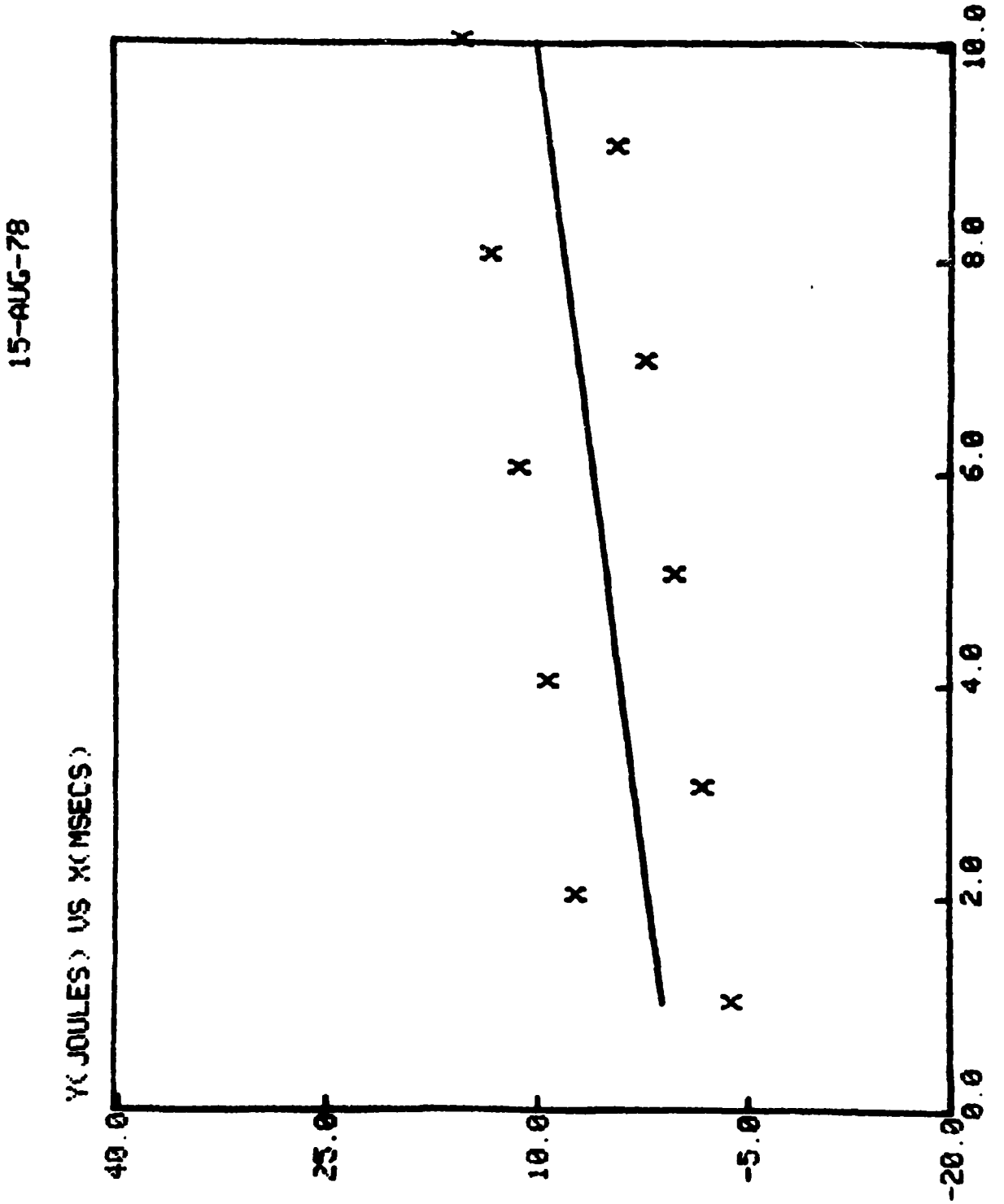
Usage: CALL WDATE

Function: Writes today's date at the top right corner of the screen.

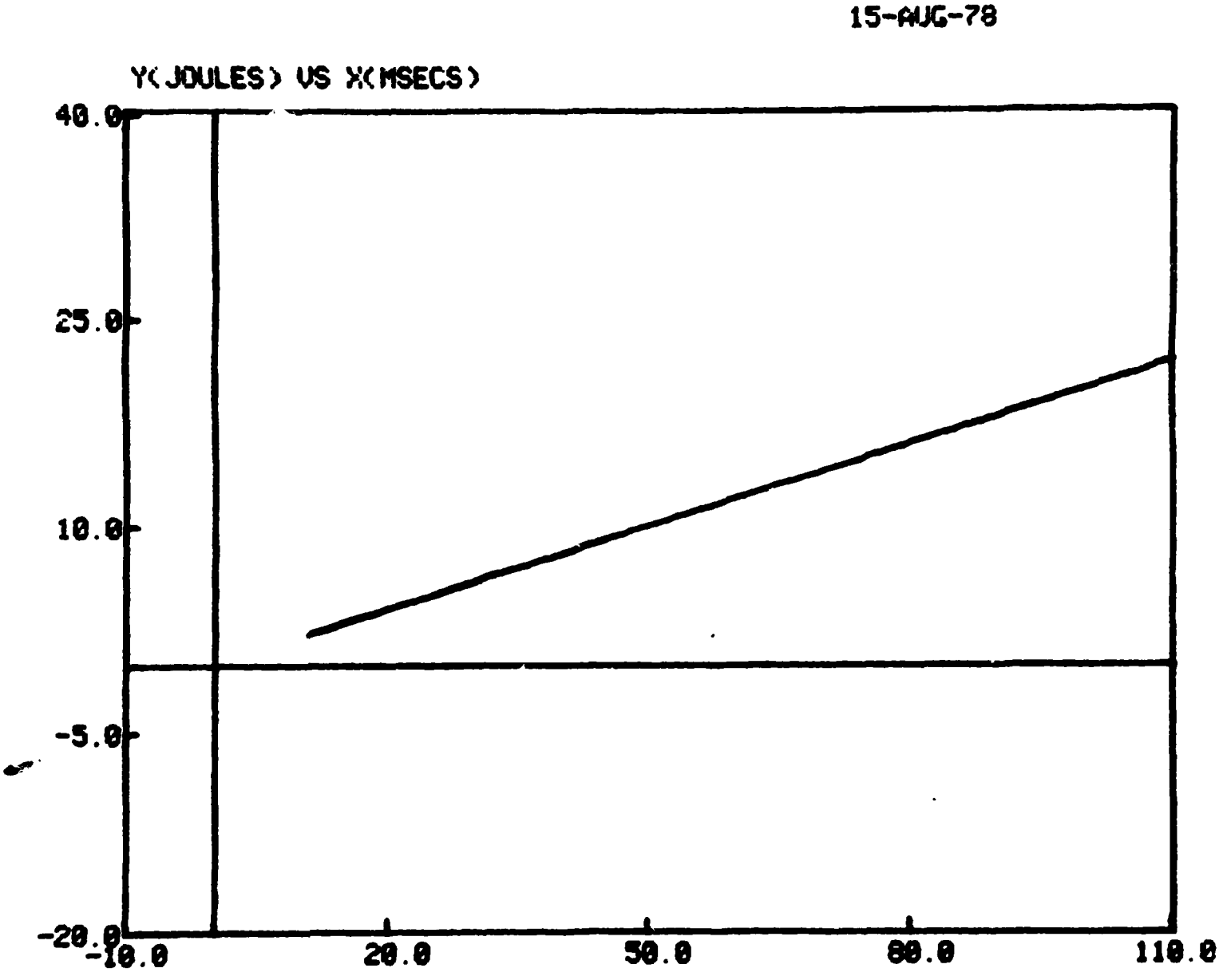
APPENDIX A

Sample programs

```
      DIMENSION X(10),Y(10)
C
      CALL INITT
C
C           ERASE SCREEN
      CALL ERASE
C
C           DEFINE WINDOW
      XMIN=0.
      XMAX=10.
      YMIN=-20.
      YMAX=40.
      CALL SETMM(XMIN,XMAX,YMIN,YMAX)
C
C           SET POINT PLOT MODE WITH "X" AT POINT
      CALL MODE(-1,'X')
C
C           DRAW AXES WITH LABEL AND SET LINEAR SCALING
      CALL AXES('SCATTER PLOT',12,0)
      CALL WDATE
C
C             DATA FOR FIRST CALL TO PLOT
      D=5.
      DO 10 I=1,10
      FI=I
      D=-D
      X(I)=FI
      Y(I)=FI+D
   10 CONTINUE
C           PLOT DATA
      CALL PLOT(X,Y,10)
C
C       SET PLOT MODE TO VECTOR
      CALL MODE(1,'X')
C
C          SUPERIMPOSE A VECTOR PLOT
      DO 15 I=1,10
      Y(I)=I
   15 CONTINUE
C
      CALL PLOT(X,Y,10)
      STOP
      END
```


15-AUG-78
Y(JOULES) VS X(MSECS)
40.0
25.0
10.0
-5.0
-20.0
0.0
2.0
4.0
6.0
8.0
10.0

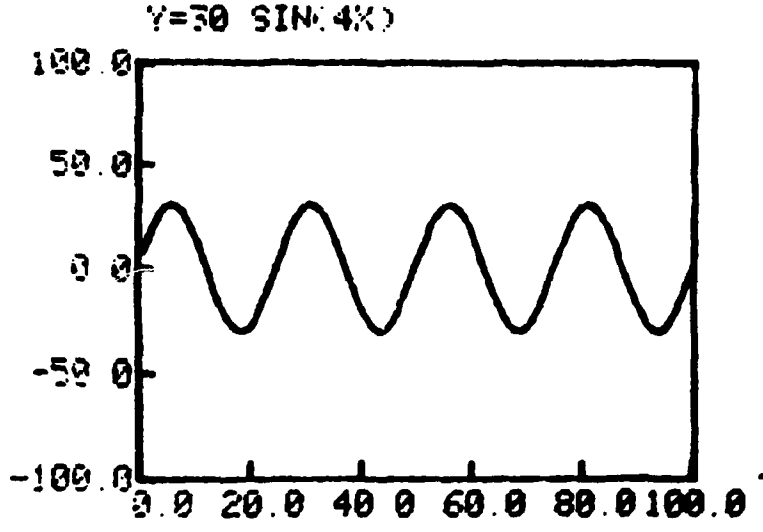
```
      DIMENSION X(100),Y(100)
        DOUBLE PRECISION XLBL(3)
        DATA XLBL(1)/'Y(JOULES'/,XLBL(2)/') VS X(M'/
        DATA XLBL(3)/'SECS)'/
C
      CALL INITT
C
C           ERASE SCREEN
      CALL ERASE
C
C           NO. OF CHARACTERS IN LABEL
      NWL=21
C
C           DEFINE WINDOW
      XMIN=-10.
      XMAX=110.
      YMIN=-20.
      YMAX=40.
      CALL SETMM(XMIN,XMAX,YMIN,YMAX)
C
      CALL WDATE
C
      CALL AXES(XLBL,NWL,0)
C
      DO 17 I=1,100
      X(I)=I+10.
      Y(I)=X(I)/5.
   17 CONTINUE
C
      CALL PLOT(X,Y,100)
      CALL ORIGIN
      CALL PAUSE
C
C           MAKE HARD COPY
      CALL HCOPY
      STOP
      END
```

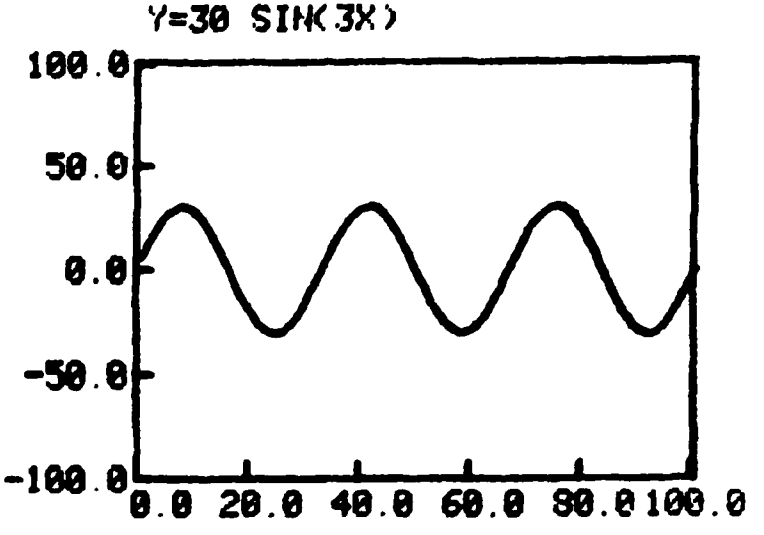

15-AUG-78
Y(JOULES) VS X(MSECS)
40.0
25.0
10.0
-5.0
-20.0
-10.0
20.0
50.0
80.0
110.0

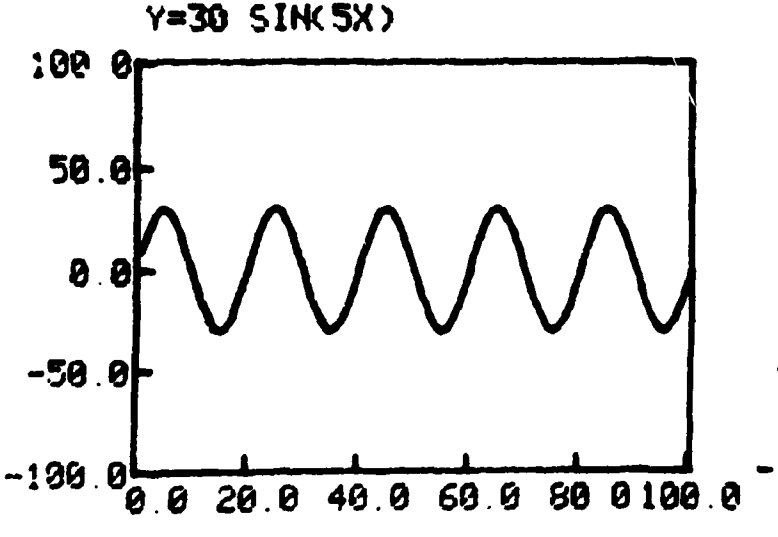
```
      DIMENSION X(100),Y(100)
      CALL INITT
      CALL ERASE
C
C           DEFINE REGIONS 4,5,6,7
      XMIN=0.
      XMAX=100.
      YMIN=-100.
      YMAX=100.
      CALL SETMM(XMIN,XMAX,YMIN,YMAX)
      CALL WINDOW(4)
      CALL WINDOW(5)
      CALL WINDOW(6)
      CALL WINDOW(7)
C
CCCCC     DEMONSTRATE PLOTTING IN REGIONS 4, 5, 6, 7   CCCCCCCCCCC
C           ALL PLOTS WILL BE LINEAR AS SET BY CALL TO AXES          C
C           ALL PLOTS WILL BE VECTOR AS SET BY CALL TO MODE          C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C           SET PLOT MODE TO VECTOR
      CALL MODE(1,'X')
C
      F1=6.28*3.
      F2=6.28*4.
      F3=6.28*5.
C
C           ACTIVATE REGION 4
      CALL REGN(4)
C
      CALL AXES('Y=30 SIN(3X)',12,0)
      CALL WDATE
C
C           DATA FOR REGION 4
      DO 40 I=1,100
      X(I)=I
   40 Y(I)=30.*SIN(F1*I/100)
      CALL PLOT(X,Y,100)
C
C
C           ACTIVATE REGION 5
      CALL REGN(5)
C
      CALL AXES('Y=30 SIN(4X)',12,0)
      DO 50 I=1,100
   50 Y(I)=30.*SIN(F2*I/100)
      CALL PLOT(X,Y,100)
C
C            ACTIVATE REGION 6
      CALL REGN(6)
```

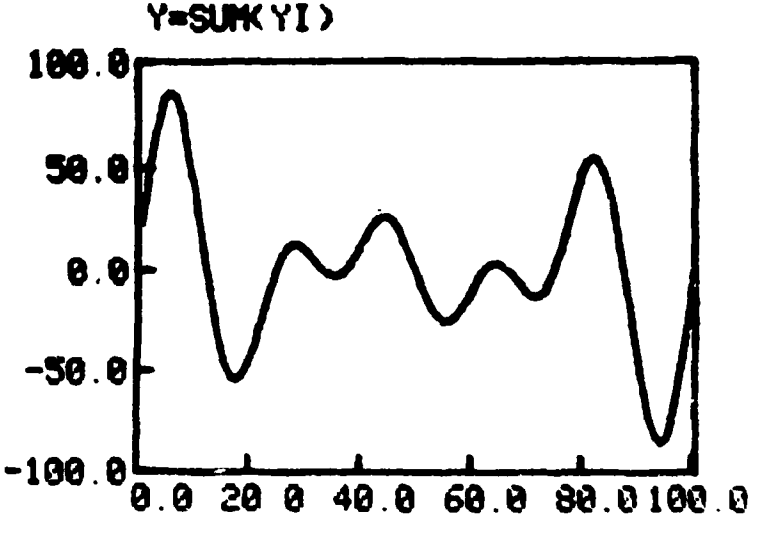
```
C
      CALL AXES('Y=30 SIN(5X)',12,0)
      DO 60 I=1,100
   60 Y(I)=30.*SIN(F3*I/100)
      CALL PLOT(X,Y,100)
C
C           ACTIVATE REGION 7
       CALL REGN(7)
C
      CALL AXES('Y=SUM(YI)',9,0)
      DO 70 I=1,100
      T=I/100.
   70 Y(I)=30.*(SIN(F1*T)+SIN(F2*T)+SIN(F3*T))
      CALL PLOT(X,Y,100)
      STOP
      END
```

17-AUG-78


Y=30 SIN(4X)
100.0
50.0
0.0
-50.0
-100.0
0.0 20.0 40.0 60.0 80.0 100.0


Y=30 SIN(3X)
100.0
50.0
0.0
-50.0
-100.0
0.0 20.0 40.0 60.0 80.0 100.0

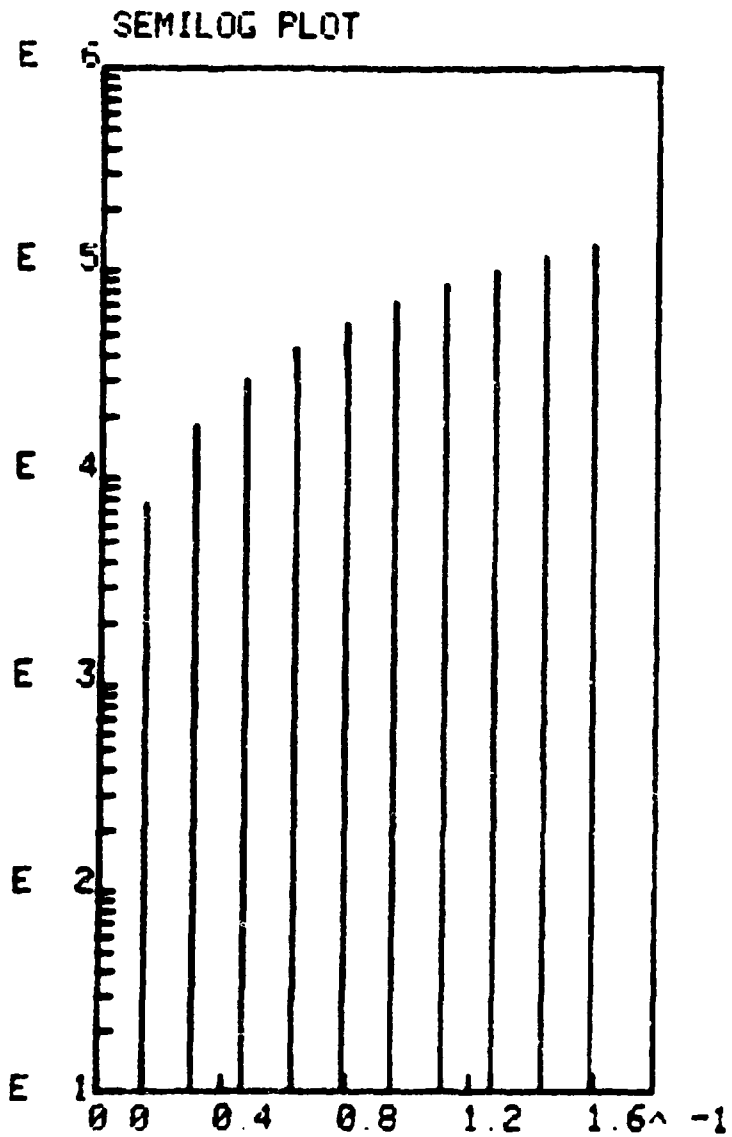

Y=30 SIN(5X)
100.0
50.0
0.0
-50.0
-100.0
0.0 20.0 40.0 60.0 80.0 100.0

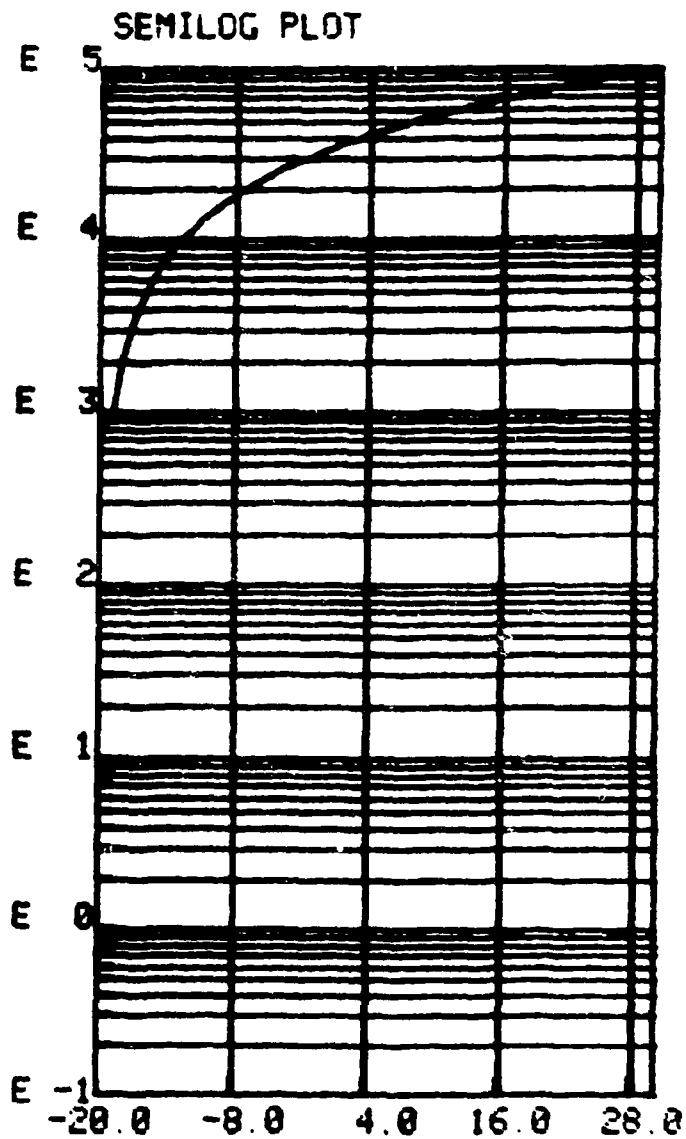

Y=SUM(YI)
100.0
50.0
0.0
-50.0
-100.0
0.0 20.0 40.0 60.0 80.0 100.0

```
      DIMENSION X(50),Y(50)
      DOUBLE PRECISION XLBL(2)
      DATA XLBL(1),XLBL(2) /'SEMILOG ','PLOT'/
      DATA NWL/12/
C
      CALL INITT
      CALL ERASE
C
C            DEFINE REGION 2
      XMIN=-20.
      XMAX=30.
      YMIN=1E-1
      YMAX=1E4
      CALL SETMM(XMIN,XMAX,YMIN,YMAX)
      CALL WINDOW(2)
C
C            DEFINE REGION 3
      XMIN=1.2E-3
      XMAX=1.8E-1
      YMIN=1E1
      YMAX=1E5
      CALL SETMM(XMIN,XMAX,YMIN,YMAX)
      CALL WINDOW(3)
C
CCCC         DEMONSTRATE PLOTTING IN REGIONS 2 AND 3
C            BOTH PLOTS WILL BE SEMILOG AS SET BY CALL TO AXES
C            PLOT 2 WILL BAR PLOT AS SET BY CALL TO MODE
C              GRID WILL BE FILLED IN ON PLOT 1 BY CALL TO GRID
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      CALL WDATE
C
C            ACTIVATE REGION 2
      CALL REGN(2)
C
C            DRAW SEMILOG AXES WITH LABEL AND SET LOG SCALING
      CALL AXES(XLBL,NWL,1)
      CALL GRID
C
C            DATA FOR REGION 2
      DO 20 I=1,50
      FI=I
      X(I)=FI-20.
   20 Y(I)=1.1**200.
      CALL PLOT(X,Y,50)
C
C            ACTIVATE REGION 3
      CALL REGN(3)
C
      CALL AXES(XLBL,NWL,1)
```

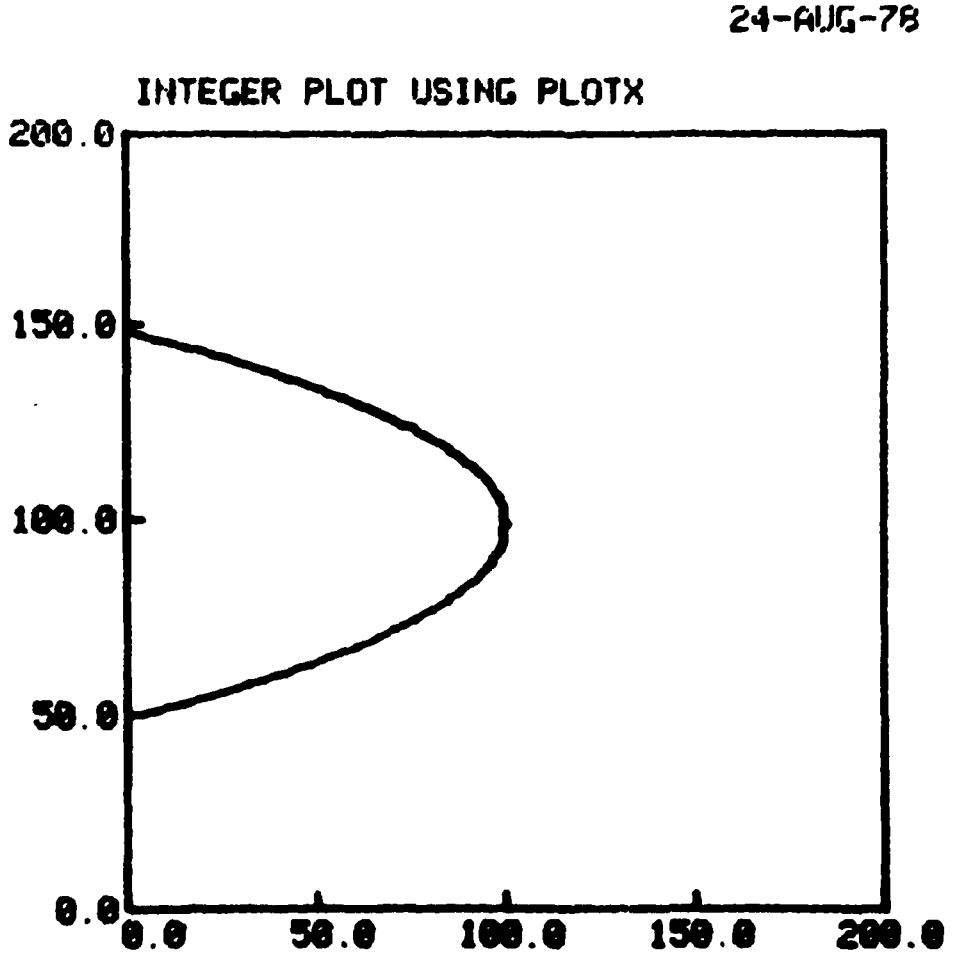
```
C           DATA FOR REGION 3
      DO 30 I=1,10
      FI=I
      X(I)=FI/10.*.1A
   30 Y(I)=10.+2E3*FI
C           SET PLOT MODE TO BAR PLOT
      CALL MODE(2,'X')
C
C           PLOT DATA
      CALL PLOT(X,Y,10)
      STOP
      END
```

24-AUG-78


SEMILOG PLOT
E 6
E 5
E 4
E 3
E 2
E 1
0.0
0.4
0.8
1.2
1.6^ -1


SEMILOG PLOT
E 5
E 4
E 3
E 2
E 1
E 0
E -1
-20.0
-8.0
4.0
16.0
28.0

```
      DIMENSION IX(100)
      CALL  INITT
      CALL  ERASE
      CALL  WAIT
C
C
CCCCC    DEMONSTRATE USER DEFINEABLE SCREEN REGION
C          SCREEN DEFINITIONS:
      MINX=300
      MINY=200
      MAXX=800
      MAXY=700
C
C           LOGICAL  DEFINITION
      XMIN=0.
      YMIN=0.
      XMAX=200.
      YMAX=200.
C
C           DEFINE REGION
      CALL SETMM(XMIN,XMAX,YMIN,YMAX)
      CALL SWINDO(MINX,MINY,MAXX,MAXY)
C
C           ALL LIBRARY PLOTTING ROUTINES WILL ACCEPT EITHER
C           INTEGER OR FLOATING POINT ARRAYS AS SET BY CALL TO
C           NUM.  PRECEEDING ARRAY INPUTS TO PLOTTING ROUTINES
C          HAVE BEEN FLOATING POINT BY DEFAULT.  THE FOLLOWING
C           PLOTTING EXAMPLE WILL USE AN INTEGER ARRAY.
C
C           SET INTEGER INPUT MODE
      CALL NUM(0)
CCCCCCCCCCCCCCC   DEMONSTRATE AUTOINCREMENT PLOTTING.  CCCCCCCCCC
C                 AXES AND SCALING MAY BE EITHER LINEAR OR LOG.  C
C                 PREVIOUSLY DEFINED SPECIAL USER SCREEN REGION  C
C                 WILL BE USED FOR THE FOLLOWING EXAMPLE.        C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C            X DATA ARRAY
      DO 10 I=1,100
   10 IX(I) = 100-(I-50)**2/25.
C
C           DRAW AXES
      CALL AXES('INTEGER PLOT USING PLOTX',24,0)
C           PLOT DATA
      CALL PLOTX(IX,50,100)
C
      STOP
      END
```


24-AUG-78
INTEGER PLOT USING PLOTX
200.0
150.0
100.0
50.0
0.0
0.0
50.0
100.0
150.0
200.0

APPENDIX B

Index to Subroutines:

ALPHA

AXES (LABEL, NCHARS, ISCALE)

BOX

ERASE

GRID

HCOPY

HOME

INITT

LINE (IX, IY)

LOCATE IX, IY)

MODE (ITYPE, ICHAR)

NOTATE (LABEL, IX, IY, NCHARS)

NUM (NUMTYP)

ORIGIN

PAUSE

PLOT (X, Y, N)

PLOTX (X, Y, N)

PLOTY (X, Y, N)

REGN (IREGN)

SCALE (ISCALE)

SETMM (XMIN, XMAX, YMIN, YMAX)

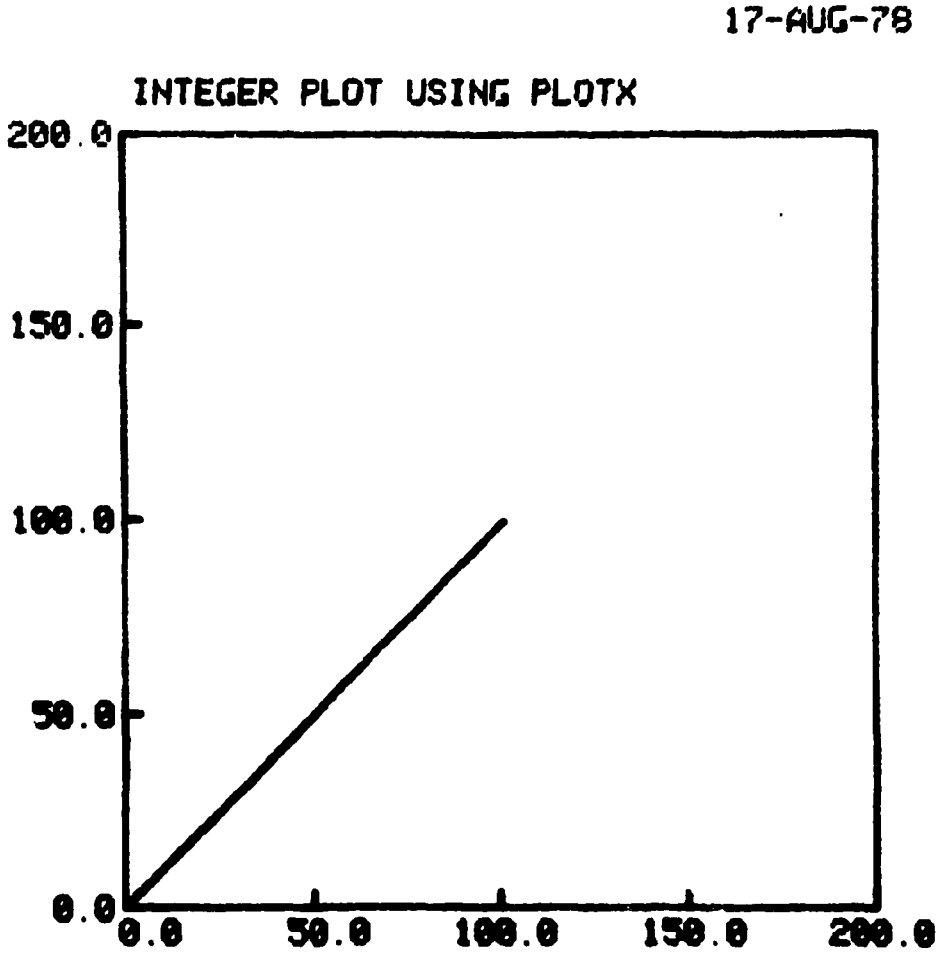
SWINDO (MINX, MINY, MAXX, MAXY)

WDATE

WINDOW (IREGN)

XAXIS (XLABEL, NCHARS)

YAXIS


17-AUG-78
INTEGER PLOT USING PLOTX
200.0
150.0
100.0
50.0
0.0
0.0
50.0
100.0
150.0
200.0