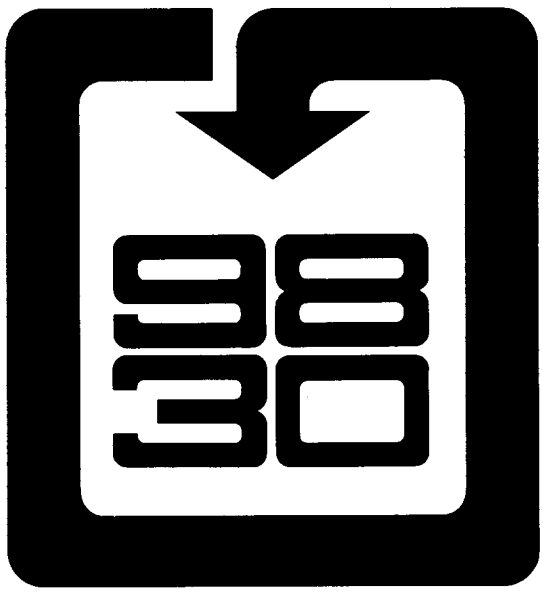
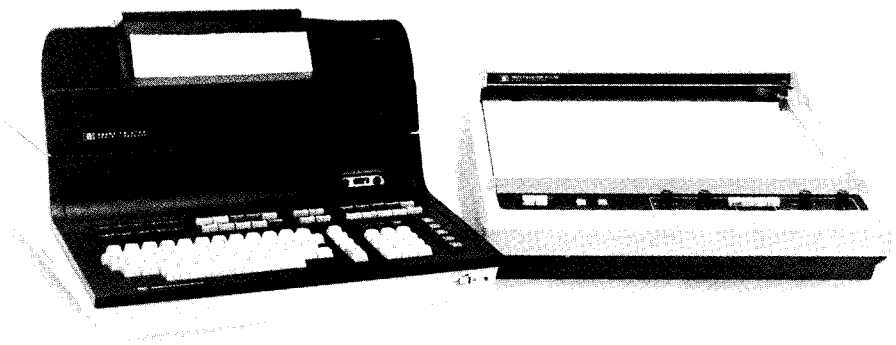


HEWLETT PACKARD 9830A CALCULATOR
PLOTTER CONTROL ROM

OPERATING MANUAL

PLOTTER CONTROL ROM
11271B & OPTION 271



9830A Calculator shown with 9866A Printer and 9862A Plotter

HEWLETT–PACKARD CALCULATOR PRODUCTS DIVISION
P.O. Box 301, Loveland, Colorado 80537, Tel. (303) 667-5000
(For World-wide Sales and Service Offices see rear of manual.)

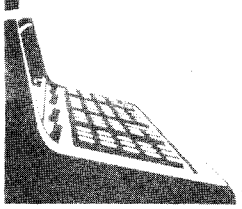


TABLE OF CONTENTS

PREFACE

The Plotter Control Read-Only-Memory (ROM) can be purchased as an accessory plug-in block or as an internal modification to the calculator.

The Plug-in Version:

The 11271B Plotter Control ROM block is installable by the user. It plugs into any of the five slots behind the ROM door on the left side of the calculator.

The Calculator Modification:

The Option 271 Plotter Control ROM must be installed by qualified HP personnel. When it is installed, a decal showing the option number (Option 271) is attached to the inside of the ROM door.

Should you wish to add the option after you have received your calculator, please order accessory number HP 11271F from the sales office nearest to you (see the back of this manual). The Option 271 will then be installed for you by our field personnel.

Once either version of the ROM (the plug-in block or the internal modification) has been installed, the operation is identical. Therefore, this manual makes no further distinction between the two types of ROM.

Chapter 1

GENERAL INFORMATION

DESCRIPTION OF THE ROM

The Plotter Control ROM enables the HP 9830A Calculator to control an HP 9862A Calculator Plotter, providing hard copy of graphic solutions to problems solved by the calculator.

The plotter command set consists of plotting commands and printing commands.

'Plotting' commands are used to:

- Scale the units selected by the user and establish the origin of the coordinates (0,0) anywhere on or off the plotting area.
- Draw X and Y axes, of any length, anywhere on the plotting area.
- Segment the axes by drawing tic marks, at intervals selected by the user.
- Plot points with respect to a previously established origin.
- Raise or lower the pen either before or after moving the pen to the point to be plotted.
- Offset the origin of the coordinates to any position and then plot with respect to that offset origin.
- Plot in increments – plot each new point with respect to the last point plotted instead of with respect to the origin.

'Printing' commands are used to:

- Print letters, digits and other symbols.
- Specify character height and width and the printing angle.
- 'Character-plot' – before labeling a point, position the pen, with respect to the point to be labeled, using character-spaces as the plotting units.
- Reference FORMAT statements to format the printing.
- Establish a unique 'typewriter' mode to control printing entirely from the keyboard of the calculator.

EQUIPMENT SUPPLIED

The following items are supplied with each Plotter Control ROM:

Operating Manual
-hp- Part No. 09830-90003, quantity 1

Plotter Pack (contains programs for the plotter)
-hp- Part No. 09830-76000 , quantity 1

Other equipment is packaged with the HP 9862A Calculator Plotter.

INSPECTION PROCEDURE

Refer to Appendix A in the 9830A Calculator Operating and Programming Manual for the procedures to verify operation of the ROM.

INSTALLING THE PLUG-IN BLOCK

The complete procedure to install a plug-in block is in the Operating and Programming Manual for the 9830A Calculator. Following are some reminders:

The block can be installed in any of the five ROM slots.

Switch the calculator off before installing or removing a block.

The label on the block should be 'right-side-up' and facing the ROM door when the block is properly installed.

Ensure that the block is properly mated to the connector at the back of the slot before switching the calculator on.

THE 9862A PLOTTER

Information concerning the HP 9862A Calculator Plotter is contained in the Peripheral Manual (HP Part No. 09862-90012) for the plotter. Refer to that manual for information regarding plotter installation in the system, initial turn-on procedure, plotter maintenance, etc. The following plotter 'set-up' information is included here to keep you from continuously having to refer to the plotter manual for information regarding the general use of your plotter once you have installed it, and learned to use it in the 9830A System.

Before plotting, the plotter must be prepared and the physical limits of the plotting area must be established. The front-panel controls on the plotter are used for this purpose.

LINE AND CHART HOLD

The LINE pushbutton is the power switch for the plotter; press it to apply power, and press it again to remove power. The white LINE indicator is lit whenever the plotter is ON.

Pressing CHART HOLD activates the electro-static paper hold-down mechanism. Pressing CHART HOLD again deactivates it. The plotter will not plot or letter, and the pen holder and arm will move freely when CHART HOLD is deactivated.

LOADING PAPER

To load paper, release CHART HOLD and manually move the pen arm all the way to one side of the plotter. Lay a sheet of paper on the plotting surface and smooth out any irregularities in the paper. Make sure that the paper is squarely against the ridge at the bottom of the plotting surface. Activate CHART HOLD and smooth the paper to the platen.

GRAPH LIMITS

The graph limit controls are used to determine the physical size of the plot.

LOWER LEFT and the two knobs to its left are used to determine the physical location of the lower left-hand corner of the plotting area.

UPPER RIGHT and the two knobs to its right are used to determine the physical location of the upper right-hand corner of the plotting area. Together, the upper right-hand corner and the lower left-hand corner determine the size of the plotting area.

Altering the lower left-hand setting will translate the upper right-hand setting by the same direction and amount.

To specify the lower left-hand corner of the plotting area, press LOWER LEFT; the pen will move, without touching the paper, to the lower left-hand corner of the plotting area. This point can be set anywhere within the lower left-hand part of the plotting surface (platen) by adjusting the two knobs associated with LOWER LEFT. Once the lower left-hand corner has been set, the upper right-hand corner is set in the same way by pressing UPPER RIGHT and adjusting the two knobs associated with it. Once the plotting area has been determined, it can be relocated by moving the position of the lower left-hand corner — the upper right-hand corner will 'track' the change.

NOTES

Chapter 2

PLOTTING

INTRODUCTION

This chapter describes the commands used to make plots — scaling, drawing and marking axes, plotting functions, lines and points. Example programs illustrating the various plotting commands are included at the end of this chapter. Labeling and lettering of graphs is described in Chapter 3.

It is assumed that you are already familiar with BASIC programming and with the operating procedures for the HP 9830A Calculator. Also that you have read the peripheral manual for the 9862A Plotter.

The following conventions are assumed:

1. All commands can be activated either from the keyboard or from a program except where noted.
2. All parameters shown in statements are values (i.e., they can be numbers, variables or expressions). The only statement containing a parameter which is not a value is the LABEL statement, discussed in Chapter 3.
3. Any parameter shown enclosed in square brackets is optional as far as the statement containing it is concerned. However, program sense may dictate that the parameter be present in specific cases.

USER-UNITS

The user can designate his own units for plotting. The calculator automatically scales (converts) the 'user-units' to 'plotter-units', which it uses to position the pen on the plotter surface.

Units are specified, indirectly, by means of the SCALE statement (described below). First the user adjusts the front-panel controls on the plotter to establish graph limits which fit the size of his plotter paper. Next he uses a SCALE statement to specify the values of the graph limits. Specifying the limit values also determines the user-units; all subsequent plotting commands then use those units so that the user never has to concern himself with 'plotter-units'. As an example, you might specify that the left edge of the graph represents −10 and the right edge 100. This has the effect of dividing the horizontal axis into (in this case) 110 of your units. What the user-units represent is determined by you — they could be feet, centimeters, hours or any other unit.

Once the units for a plot have been established, the size of the plot can be changed, to fit a larger or smaller sheet of plotter paper, simply by resetting the GRAPH LIMIT controls on the plotter. It is not necessary to change units because the calculator will automatically rescale them to fit the new plotting area.

THE SCALE STATEMENT

SCALE $value_1$, $value_2$, $value_3$, $value_4$
SCALE X_{min}, X_{max}, Y_{min}, Y_{max}

Example:
SCALE −10, 10, −5, 5
SCALE −4*PI, 4*PI, − 0.3, 1.1

The SCALE statement establishes the full-scale values, in user-units, for the plot. X_{min} to X_{max} and Y_{min} to Y_{max} correspond exactly to the respective limits of the horizontal and vertical edges of the plotting area (the area is established mechanically, as previously described). This also establishes the point, on or off the plotting area, where the origin (point 0,0) of the coordinate system is located.

A SCALE statement must be executed before any plotting can occur. Once established, the scale remains established until one of the following occurs:

A new SCALE statement is executed.

The program is initialized.

A SCRATCH or SCRATCH A or SCRATCH V is executed.

The calculator is switched off.

The parameters (X, Y, etc.) in a SCALE statement must be given in the correct order. If the minimum and maximum values are switched, subsequent plotting commands may not be executed properly.

THE PEN STATEMENT

PEN

The PEN statement is a 'stand-alone' instruction requiring no parameter. It raises the pen without otherwise changing its position relative to the plotting area.

Instructions to raise or lower the pen, either before or after movement, can be included in several other statements (see PLOT and IPLOT).

THE OFFSET STATEMENT

OFFSET $value_1$, $value_2$
OFFSET X, Y

Example:
OFFSET 3, −3

The OFFSET statement moves the origin (point 0,0) of the coordinate system to the point specified by the values of X and Y. Subsequent plotting commands are then made with respect to the new origin until such time as that origin is again changed by means of, for example, a new OFFSET or a new SCALE statement.

OFFSET statements are not cumulative; that is, a new 'offset' is made with respect to the original origin and not with respect to the last offset origin.

Offsetting greatly simplifies plotting from the user's point of view. For example, it sometimes becomes necessary to divide the plotting area into smaller segments and make a separate plot in each segment. While plotting in each segment, it is not necessary for the user to 'correct' each point before plotting it; instead, the OFFSET statement moves the origin to some convenient point within that segment, so that the calculator automatically makes the necessary 'corrections' for each point to be plotted. Example 2 at the end of this chapter shows the use of OFFSET to enable two plots to be made, over the same range, side-by-side, on the same sheet of paper.

THE AXIS STATEMENT

XAXIS $value_1$ [, $value_2$ [, $value_3$, $value_4$]]
XAXIS Y-offset [, ±tic [, start point, end point]]

or

YAXIS $value_1$ [, $value_2$ [, $value_3$, $value_4$]]
YAXIS X-offset [, ±tic [, start point, end point]]

Example:

```
XAXIS 3, 1, -4, 4
XAXIS 0
YAXIS -3, PI/8
```

The AXIS statement draws an X-axis or Y-axis according to the parameters given in the statement. The pen is automatically raised before moving to the start point, and again after drawing the axis.

NOTE

The following describes the X-axis; the same information is applicable to the Y-axis if 'left' and 'right' for the X-axis are read as 'bottom' and 'top', respectively, for the Y-axis.

1. If no optional parameters are given, draws a straight line from left to right across the complete plotting area (from X_{min} to X_{max}). The line crosses the Y-axis at the point specified by the value of 'Y-offset'.
2. If a 'tic' parameter is included then tic marks are made along the axis as it is drawn; the value of 'tic' determines the spacing, in user-units, between tics. The first tic is drawn at the starting point of the line. The tic parameter is usually positive (the 'plus' sign is not required), but a negative tic spacing can also be used – see 4, below.
3. If the start point and end point parameters are given, then the axis is drawn only between those points; that is, from the start point to the end point. If the start point parameter is less than the end point parameter, then the axis is drawn from left to right. If the start point parameter is greater than the end point parameter, then the axis is drawn from right to left.

(continued)

THE AXIS STATEMENT
(continued)

4. A tic parameter which has a positive value results in:
   (i) Normal tic spacing if the axis is drawn from left to right;
   (ii) A tic only at the right end of the axis if the axis is drawn from right to left.

   A negative tic value results in:
   (i) A tic at the left end of the axis if the axis is drawn from left to right;
   (ii) Normal tic spacing if the axis is drawn from right to left.
5. If an OFFSET statement has previously been given, then all of the optional parameters must be included in any AXIS statement.

In the SCALE statement, the values for Xmin and Xmax, Ymin and Ymax are defined. If, in the AXIS statement, the value for the optional parameters, "start point" and "end point" are omitted, the values used are calculated from the values defined in the SCALE statement. A scale factor (S.F.) and an adjustment number (A) are calculated in the SCALE statement.

$$S.F. = [9999 / (X_{max} - X_{min})] \qquad A = S.F. * X_{min}$$

When the AXIS statement is executed, the value of Xmin is recalculated using the formula

$$X_{min} = A / S.F.$$

If the recalculated value generated by the calculator is less than the actual Xmin defined in the SCALE statement, an error 82 will occur.

To avoid this possible problem, define all the parameters in the AXIS statement.

THE PLOT STATEMENT

PLOT $value_1$, $value_2$ [, $value_3$]
PLOT X, Y [, control pen]

Example:
PLOT SIN(X), COS(X), −2

The PLOT statement moves the pen to the point specified by the value of the X and Y parameters.

When no optional 'control pen' parameter is given:

If the pen was raised, it moves to the point specified and then lowers, marking a point on the paper.

If the pen was lowered, it remains lowered while moving to the point specified, thus drawing a straight line on the paper.

THE 'CONTROL PEN' PARAMETER

The value and sign of this parameter in the PLOT (and IPLOT) statement determines whether the pen will be raised or lowered before or after it moves to the specified point.

If the parameter is:

positive – control occurs before movement;
negative – control occurs after movement;
odd – raises pen if it was lowered;
even– lowers pen if it was raised;
zero – leaves pen unchanged.

The value of the control parameter can be any number in the range ±32767. If the value is not an integer, then it is automatically rounded up or down according to the value of the fractional part of the number; that is, up for .5 or greater, or down for less than .5. (Rounding is the same as the standard rounding in the calculator; it is not the same as the INT function, where the value becomes that of the next lower integer.)

When plotting a function, it is sometimes useful to use a control pen parameter of zero in your PLOT statements so that you can make a 'dummy run' of your plot. First, raise the pen by pressing the PEN UP control on the plotter. Then run the part of the program which plots the function and watch the course of the pen over the plotter paper. If the pen movement does not appear to be correct, you can make any necessary changes to your program. When you are satisfied that the program appears to be plotting correctly, stop the program, press PEN DOWN on the plotter and rerun the program to actually draw the plot.

THE IPLOT STATEMENT

IPLOT $value_1$, $value_2$ [, $value_3$]
IPLOT X, Y [, control pen]

Example:
IPLOT 2, –3A/4, 1

The IPLOT statement moves the pen (from its current position) in the X direction and in the Y direction, by the amounts specified by the values of X and Y.

The 'control pen' parameter is optional and operates exactly as described previously – see the PLOT STATEMENT.

Notice that the IPLOT statement plots a point with respect to the previously plotted point and not with respect to the origin (0,0) of the graph.

The IPLOT statement is very useful when drawing regular geometric shapes such as a cross. In cases like this, it is easier to plot each point relative to the current position of the pen than it is to plot each point relative to the origin of the graph (see the third example at the end of this chapter).

When an IPLOT statement is executed immediately after printing a letter or other character on the plotter (see Chapter 3) the pen movement is related to the last 'plotted position'† and not to the 'current position' of the pen. This feature is used to advantage in the second example program (Plotting with X's) at the end of Chapter 3 and in the program which appears in the Appendix to this manual.

† 'Plotted position' refers to a point plotted either by a PLOT statement or by an IPLOT statement.

EXAMPLE PROGRAMS-PLOTTING

1. PLOTTING A FUNCTION

This program plots the function (SIN X)/X. The scale is chosen to suit standard 7 X 10 inch plotter paper. However, the plot can be made on any size of paper, up to the limits allowed by the plotter.

Before running this program, ensure that the GRAPH LIMITS controls on the plotter are properly set to suit the size of paper which you are using. Figure 2-1 shows the plot resulting from this program — the height to width ratio of your plot may vary from that shown, depending on the dimensions of your paper.

```
10 SCALE -5*PI,5*PI,-0.3,1.1
20 XAXIS 0,PI/2
30 YAXIS 0,0.1
40 FOR X=-5*PI TO 5*PI STEP PI/20
50 PLOT X,SIN(X)/X
60 NEXT X
70 END
```

When you have run the program, you will see that the last point was not plotted. This is not an error in the plotter block but results because the user-unit chosen, PI, is not an exact value. In this case, in line 40, the method used to obtain the value of 5 * PI for the FOR loop (i.e., by incrementing, in steps of PI/20, from −5*PI) results in a value which is slightly larger than the value obtained by simply multiplying 5 by PI. Thus the last point is assumed to be outside the range of the FOR loop and is not plotted. The last required point (5*PI) will be plotted if line 40 is changed to:

```
40 FOR X=-5*PI TO 5*PI+PI/20 STEP PI/20
```

Whether or not this type of 'over-range' is required for other plots, depends upon the units used and can be easily ascertained by experimenting.

Another version of this program is used in Chapter 3 to illustrate labeling graphs.

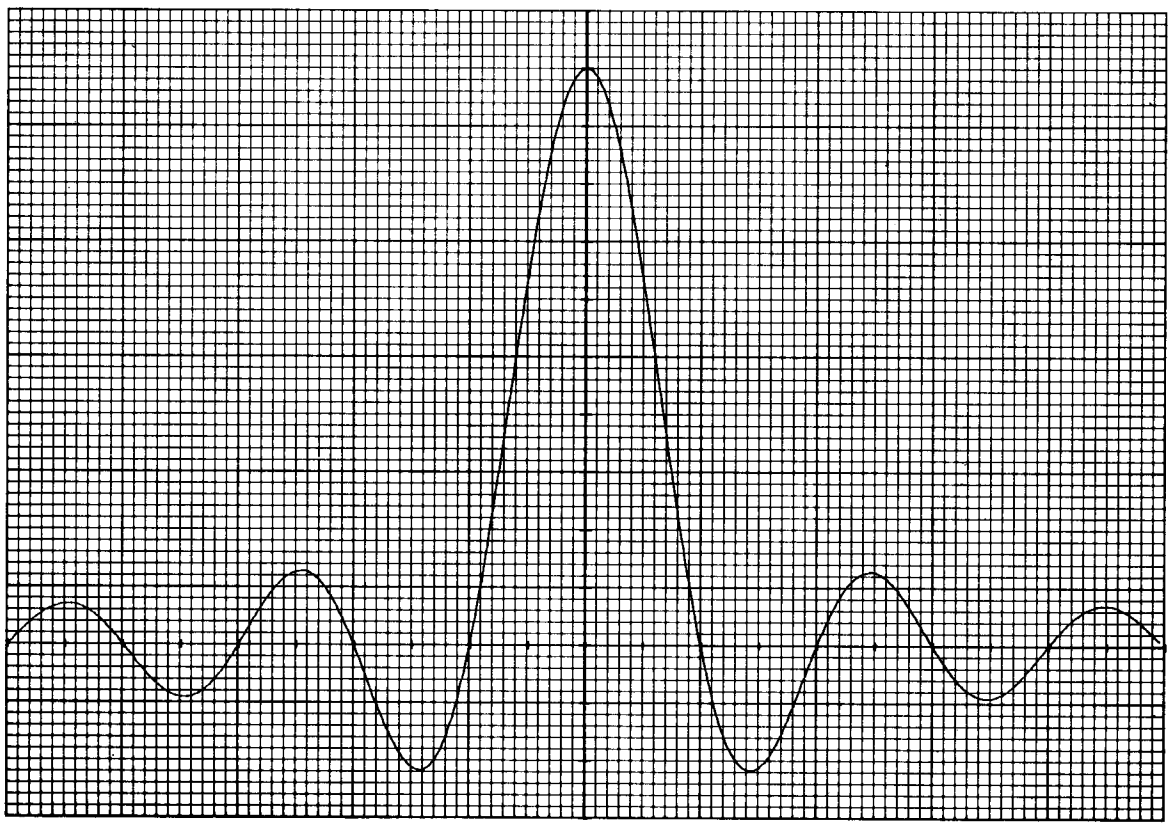


Figure 2-1. Plot of (SIN X)/X

EXAMPLE PROGRAMS-PLOTTING
(continued)

2. PLOTTING WITH OFFSET

The second example program illustrates use of the OFFSET statement. The purpose of this program is to make two plots, side-by-side, of different functions over the same range — in this case, SIN X and COS X over the range of −180° to 180° (see Figure 2-2). OFFSET is used in lines 30 and 80 to move the origin (0,0) of the graph. This enables the same AXIS statements (lines 150 and 160) to be used for both plots; also the two FOR loops (starting at lines 50 and 100) can both have the same range.

The program is scaled to plot on a standard 7 X 10 inch sheet of paper; as in the previous example, other sizes of paper can be used.

```
10 SCALE 0,750,0,2.2
20 DEG
30 OFFSET 750/4,1.1
40 GOSUB 150
50 FOR X=-180 TO 180 STEP 5
60 PLOT X,SIN(X)
70 NEXT X
80 OFFSET 750*3/4,1.1
90 GOSUB 150
100 FOR X=-180 TO 180 STEP 5
110 PLOT X,COS(X)
120 NEXT X
130 PEN
140 STOP
150 XAXIS 0,22.5,-180,180
160 YAXIS 0,0.2,-1,1
170 RETURN
180 END
```

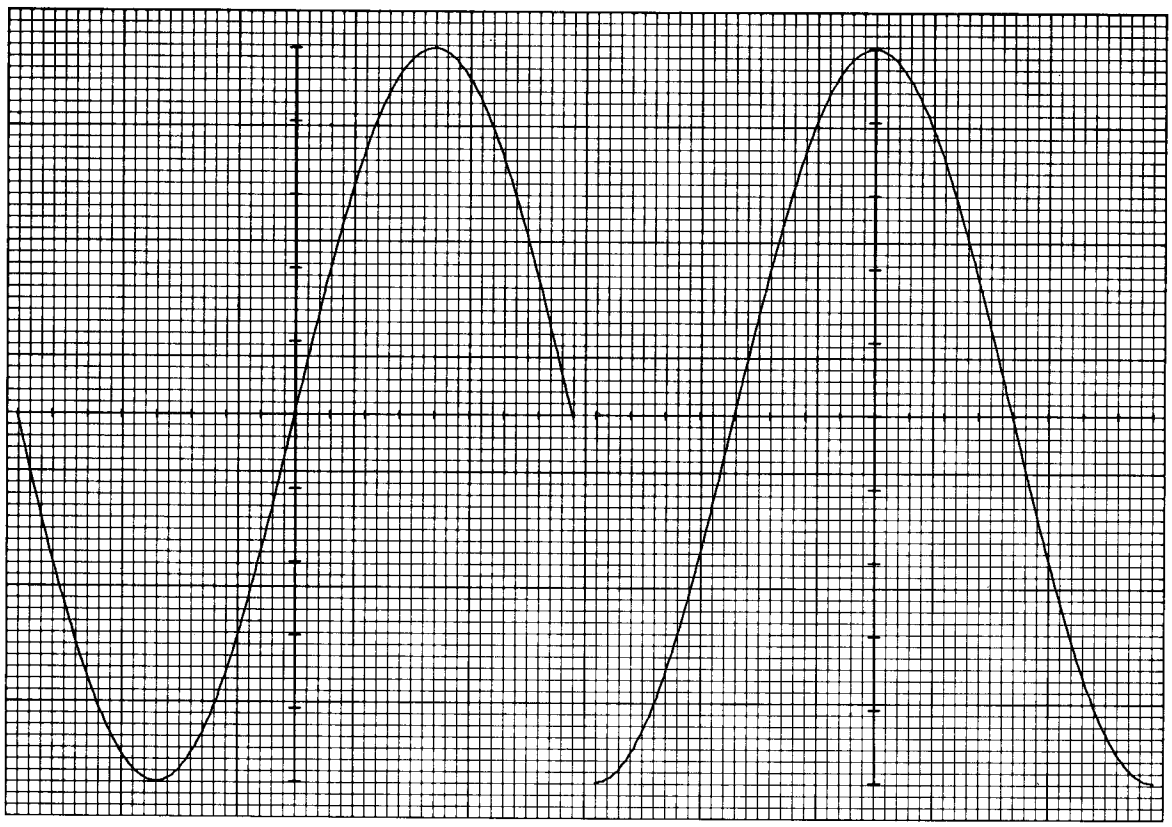


Figure 2-2. Plotting with OFFSET

EXAMPLE PROGRAMS-PLOTTING
(Continued)

3. INCREMENTAL PLOTTING

The third example program illustrates use of the IPLOT statement to plot incrementally. The program uses a subroutine to plot crosses at different locations on the paper (see Figure 2-3). Before each cross is plotted, an OFFSET statement moves the pen to the required positon. IPLOT statements are then used to plot the coordinates of each point of the cross relative to the previous plotted point. As can readily be seen from this example, plotting a regular shape is much easier to program when increments can be used than it would be if the coordinates of each point had to be calculated with respect to some fixed origin.

The scale assumes standard 7 X 10 inch plotter paper. If the height to width ratio of the paper to be used differs from 7:10, then the crosses will not be square. However, they will be drawn square if you change the parameters in the SCALE and OFFSET statements to suit the dimensions of your paper.

```
10 SCALE 0,100,0,70
20 OFFSET 20,20
30 GOSUB 130
40 OFFSET 20,50
50 GOSUB 130
60 OFFSET 80,50
70 GOSUB 130
80 OFFSET 80,20
90 GOSUB 130
100 OFFSET 50,35
110 GOSUB 130
120 STOP
130 PLOT -3,3,1
140 IPLOT 0,7,2
150 IPLOT 6,0
160 IPLOT 0,-7
170 IPLOT 7,0
180 IPLOT 0,-6
190 IPLOT -7,0
200 IPLOT 0,-7
210 IPLOT -6,0
220 IPLOT 0,7
230 IPLOT -7,0
240 IPLOT 0,6
250 IPLOT 7,0,-1
260 RETURN
270 END
```

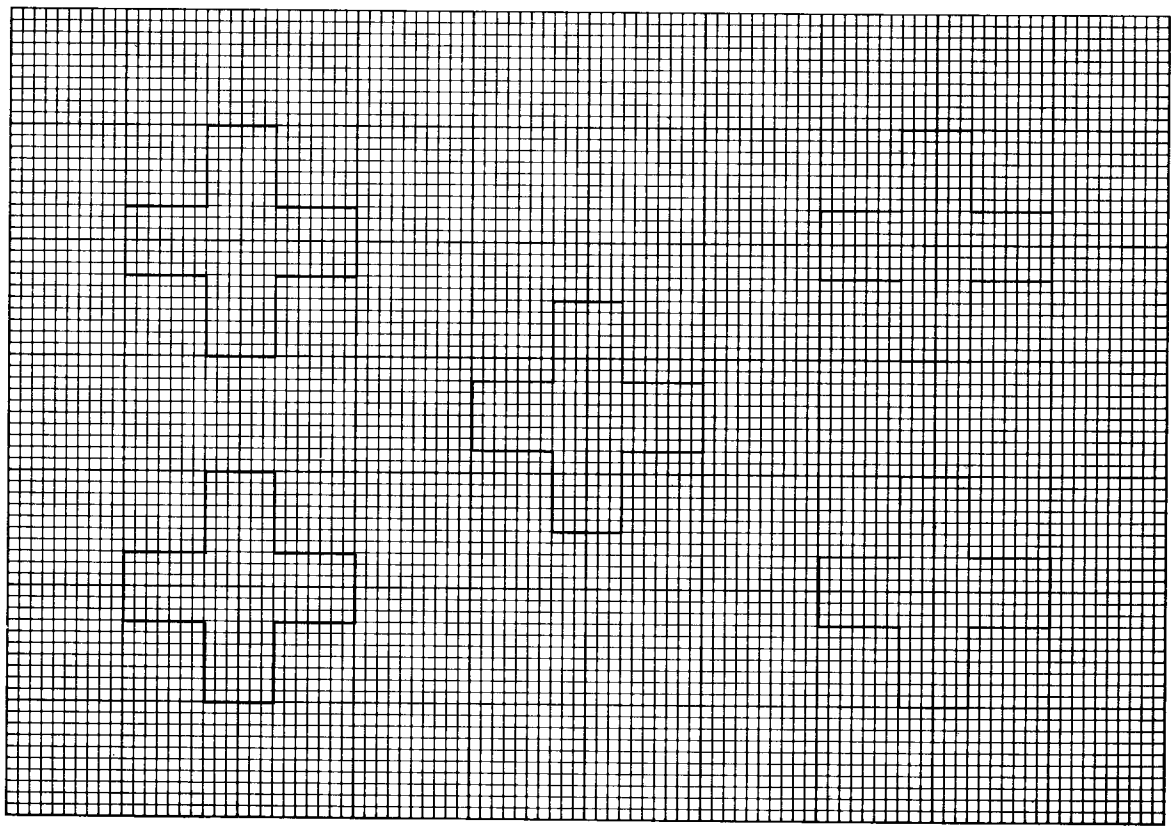


Figure 2-3. Plotting with IPLOT

NOTES

Chapter 3

PRINTING

INTRODUCTION

Three commands – LABEL, LETTER and CPLOT – are used to print characters (letters, numbers, symbols) on the plotter. One important aspect of these commands is that they are almost entirely independent from the plotting commands, described in the previous chapter. The only interaction between plotting and printing is that the pen must usually be positioned by means of a PLOT or IPLOT statement before printing is possible.

Letter sizes are specified by the user as a percentage of the height of the plotting area, which is established mechanically by means of the controls on the front panel of the plotter. The direction in which characters are to be printed, relative to the horizontal axis of the plotting area, is specified as an angle, in degrees, radians or grads. If the plotting area is rectangular, as opposed to square, then a simple compensation factor can be specified to prevent characters from appearing distorted.

Example programs illustrating the various printing commands are included at the end of this chapter.

THE LABEL STATEMENT

LABEL (Specifications) [Print List]

LABEL (line number or * [, $value_1$, $value_2$, $value_3$ [, $value_4$]])[list]

'Specifications' consist of:

line number of a FORMAT statement, or an asterisk;
$value_1$ – character height;
$value_2$ – character aspect ratio;
$value_3$ – angle of rotation;
$value_4$ – paper height/paper width.

'List' consists of:

values, or literals (in quotation marks), or both.

Example:

```
LABEL (140, 3, 1.7, 0) "F(X)=" I
LABEL (*, 5, 2, PI/4, 8/11)
LABEL (*) "PLOT OF (SINX)/X"
LABEL (70) X "PI"
```

The LABEL statement is similar to the WRITE statement; it is used to print alpha-numeric characters on the plotter and to reference FORMAT statements to format the printout. In addition, LABEL can be used to specify the height and width of the printed characters and the direction in which they are printed.

The parameters in the LABEL statement are grouped into two parts, as shown above: the 'specifications' (at least one of which must always be included) and the optional 'print list'.

(Continued)

THE LABEL STATEMENT
(Continued)

SPECIFICATION PARAMETERS

The specification parameters must be enclosed in parentheses. The line number of a FORMAT statement, or, if no FORMAT statement is to be referenced, an asterisk (*), must be included. The character parameters (height, aspect ratio and angle of rotation) are optional, but, if one of them is needed then all three must be included. The last specification parameter, the ratio of paper height to paper width, is also optional; it can be included only if the other character parameters are also there.

1. The first parameter is the line number of the FORMAT statement used to format the printout (see 'Print List', below). The FORMAT statement referenced can contain any parameters normally allowed in FORMAT statements.

   This parameter is a line number, so it must be a positive integer (it cannot be a variable or an expression). Like any line number, it is subject to being automatically changed any time the program line numbering is changed by means of a REN ('renumber') statement.

   If no FORMAT statement is to be referenced, then an asterisk (*) must be used instead of a line number.

2. The character parameters determine the height and aspect ratio of the characters, and the direction in which they will be printed. Normally, it is assumed that the plotting area, set by the GRAPH LIMITS controls on the plotter, is square. If the plotting area is rectangular rather than square, then the fourth character parameter, paper height/paper width, should be included.

   Once character specifications have been established, by means of a LABEL statement, they remain established until a new LABEL statement with new character specifications is encountered, or until a program is initialized, or until a SCRATCH is executed. At turn-on, or if no character specifications have been established, the calculator automatically assumes the following:

   character height = 1.5%
   aspect ratio = 2
   rotation = 0°
   paper height/paper width = 1

   a. 'Character Height' is expressed as a percentage of the height of the plotting area, up to a maximum of 18.4%. If the height of the plotting area is subsequently changed, the height of the characters will change accordingly.

   b. 'Aspect Ratio' determines the character width. It is expressed as the ratio of character height divided by character width. (An aspect ratio of 1.7 gives a reasonably pleasing character shape.)

   c. 'Angle of Rotation' specifies the direction of printing. The angle is in degrees, radians or grads — whichever was selected prior to execution of the LABEL statement. 0° results in printing in the direction of the X-axis, from left to right; 90° results in printing in the direction of the Y-axis, upwards, and so on. Any angle can be selected.

   d. 'Paper Height/Paper Width' is a compensating parameter required only if the plotting area is rectangular, as opposed to square. Without this parameter, the calculator will assume a square plotting area and will therefore distort the characters if they are printed in a non-square area. The distortion becomes particularly noticeable when printing at angles other than zero. The compensating parameter ensures that characters retain their proper shape, regardless of the angle of rotation.

THE 'PRINT LIST'

The 'print list' is an optional part of the LABEL statement; it contains the characters, numbers, variables, etc., which are to be printed on the plotter. A LABEL statement containing the line number of a FORMAT statement and containing a print list is similar to a WRITE statement. The items in the print list are printed on the plotter in the format determined by the parameters in the referenced FORMAT statement. That FORMAT statement can contain any of the parameters normally allowed in FORMAT statements (field width, fixed-float, spaces, carriage return/line feed, messages, etc.). FORMAT statements are fully described in the Operating and Programming Manual for the 9830A. (A table showing the code for each character for use with FORMAT B is shown in the appendix to this manual. The first example at the end of this chapter includes a B parameter.)

When a LABEL statement has an asterisk (i.e., it does not reference a FORMAT statement) it acts like a PRINT statement. Items such as commas, semi-colons and TAB's have the same definitions they normally have in PRINT statements. Also, the form of any numerical printout depends upon whether the statement is executed in keyboard or program mode.

Printing starts at the current pen position. When a carriage return/line feed is encountered the pen lifts and returns to the character position directly below the first character position of the current line.

THE LETTER STATEMENT

LETTER

LETTER is a 'stand-alone' instruction requiring no parameters. When encountered in a program, or executed from the keyboard, it establishes the 'typewriter' mode – with the calculator keyboard acting as the typewriter keyboard and the plotter as the printing device. This command enables the user to add any comments or labeling to his graphs without having to first program them.

As soon as the LETTER mode is established, a question mark (?) appears in the display. Any character or symbol can now be immediately printed by pressing the key which represents that character or symbol. The SHIFT key can be used to obtain the upper case characters. The dimensions of the characters and the direction of printing are determined by the current character specifications, derived from a previously encountered LABEL statement or from the 'assumed' specifications if there was no previous LABEL statement (see 'THE LABEL STATEMENT', above).

If the LETTER statement is encountered in a program the program halts. The program automatically resumes at the next statement when the LETTER mode is terminated, by pressing the STOP key.

When in the 'typewriter' mode:

1. The question mark reappears in the display as soon as the plotter has finished printing the previous character and is ready for the next one.
2. The pen lifts after printing each character and moves to the next character position.
3. All non-character keys are deactivated, except the REWIND key and those mentioned below.

(Continued)

THE LETTER STATEMENT
(Continued)

4. The four DISPLAY keys, with vertical and horizontal arrows on them, can be used to move the pen one character space in the direction indicated by the arrow on the key pressed (the direction being subject to any angle of rotation previously specified). If the SHIFT key is pressed at the same time that a DISPLAY key is pressed, the pen will move only one tenth of a character space. (See 'THE CPLOT STATEMENT', below, for an exact description of a character space).
5. The '@' symbol is obtained by pressing the SHIFT key and the RESULT key at the same time.
6. A 'carriage return, line feed' instruction is simulated by pressing the EXECUTE key.
7. The LETTER mode is terminated by pressing the STOP key.

THE CPLOT STATEMENT

CPLOT $value_1$, $value_2$
CPLOT spaces wide, spaces high

(where 'spaces' refers to 'character spaces'.)

Example:
CPLOT –6, –0.3

CPLOT ('character plot') lifts the pen and moves it, from its current position, in the directions and by the amounts specified by the two parameters. The horizontal movement is determined by 'spaces wide' and the vertical movement by 'spaces high'. Since the size of a space is determined by the size of the characters (see 'The Character Space', below), one space wide will probably have a different length from one space high. In the CPLOT statement, the horizontal, and vertical directions are subject to any angle of rotation established previously by a LABEL statement.

CPLOT is used to position the pen prior to printing labels for tic marks or for other specific points on the graph. It enables the user to immediately position the pen 'so many characters spaces away'. Without the CPLOT command it would be necessary to first calculate the distance in user-units, based on character size, and then position the pen that distance away. (Imagine trying to do that, for instance, at an angle of rotation of 48.73°!) Following is an example:

It is required that the word 'FOCUS' be used to identify a point on a graph. As shown in Figure 3-1, the word is to be located one character space to the left of the point and is to be centered vertically on the point (the angle of rotation is zero).

a. Use any convenient method, such as a PLOT statement, to position the pen exactly at the point to be labeled (point A in Figure 3-1).
b. The following CPLOT statement will now move the pen six character spaces to the left and half a character height downward (to point B in Figure 3-1), ready to print:

CPLOT –6, –0.3

The reason the second parameter is 0.3, and not 0.5, becomes clear once it is understood that a 'space' consists of the height (or width) of the character plus the actual space left between it and the next character (see 'The Character Space', below).

c. A LABEL statement (or a LETTER statement) can now be used to print the word 'FOCUS':

```
LABEL (*) "FOCUS"
```

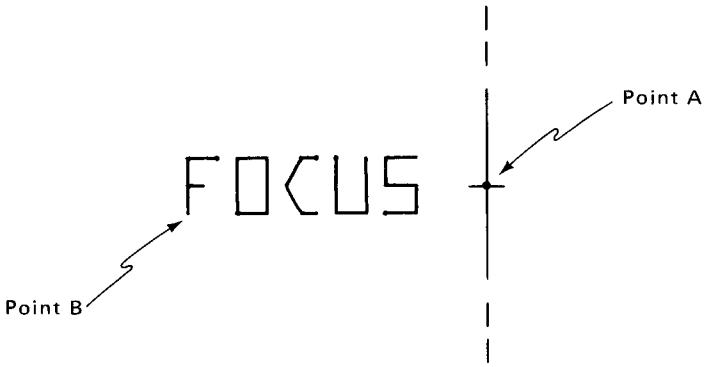


Figure 3-1. Using CPLOT

THE CHARACTER SPACE

The character space is defined as shown in Figure 3-2. Notice that the height and width are not necessarily equal and that each character dimension is six tenths of the corresponding space dimension. The parameters in a LABEL statement specify character height and aspect ratio, thus indirectly determining the size of the character space.

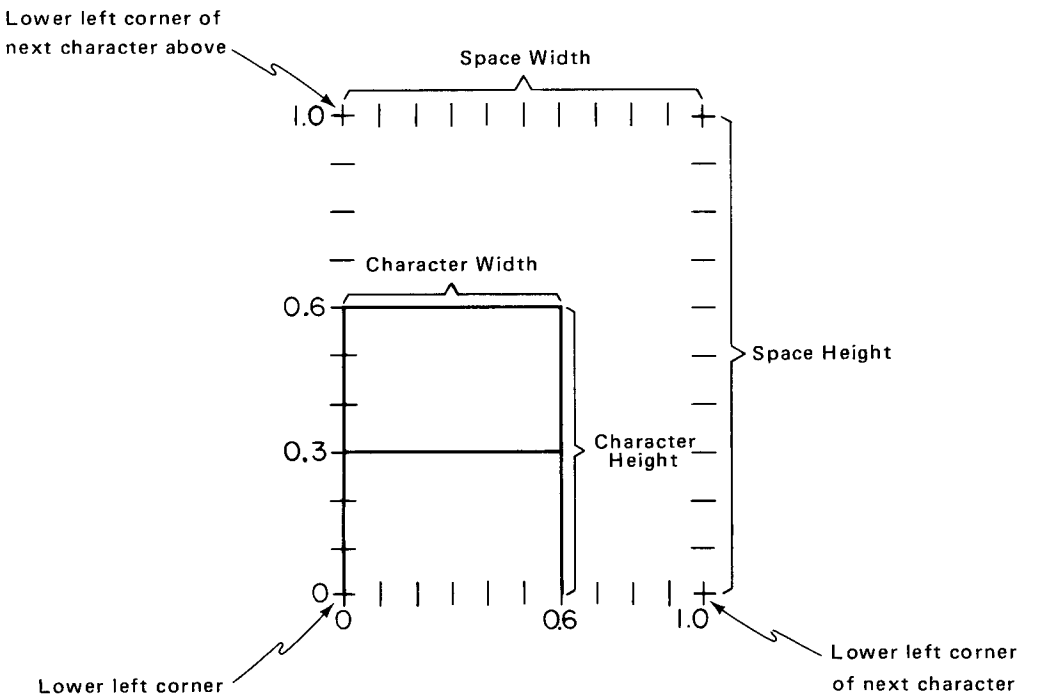


Figure 3-2. The Character Space (with the letter 'A')

EXAMPLE PROGRAMS-PRINTING

1. LABELING THE AXES

This program illustrates the use of the LABEL and CPLOT statements for labeling graphs (see Figure 3-3); the plot is similar to the one used in the first example at the end of Chapter 2 and requires the same size paper (7 by 10 inches).

a. The program has four main segments:

   Lines 20 to 60 print the title;
   Lines 90 to 150 label the Y-axis;
   Lines 160 to 220 label the X-axis;
   Lines 240 to 260 plot the function.

b. The printout in lines 30 through 50 uses the 'assumed', or 'turn-on', character specifications. Line 50 references the 'B' format to print the brackets.

c. Lines 90 and 160 establish the character specifications for the labeling along the Y-axis and the X-axis, respectively. The only difference between these two statements is that line 90 specifies zero rotation whereas line 160 rotates the printout by PI/2 radians (90°).

d. Lines 110 and 180 prevent point 0,0 from being labeled.

e. The PLOT and CPLOT statements in lines 120 and 130 (and in lines 190 and 200) act as a pair: first, to move the pen to the point to be identified, and then to move the pen the required number of character spaces away from that point before printing.

f. The LABEL statements in lines 140 and 210 both reference the FORMAT statement in line 230. The parameter in the FORMAT statement is doubled to prevent a carriage return and line feed between the value for X and the word 'PI' when line 210 is executed.

```
10 SCALE -5*PI,5*PI,-0.3,1.1
20 PLOT -4*PI,0.96,1
30 LABEL (*)"PLOT OF"
40 LABEL (*)"(SIN X)/X"
50 LABEL (60)91"FROM -4PI TO 4PI"93
60 FORMAT 2B
70 XAXIS 0,PI/2,-4*PI,4*PI
80 YAXIS 0,0.1,-0.2,1
90 LABEL (*,1,1.7,0,7/10)
100 FOR Y=-0.2 TO 1 STEP 0.1
110 IF Y=0 THEN 150
120 PLOT 0,Y,1
130 CPLOT 2,-0.3
140 LABEL (230)Y
150 NEXT Y
```

```
160 LABEL (*,1,1.7,PI/2,7/10)
170 FOR X=-4 TO 4 STEP 1/2
180 IF X=0 THEN 220
190 PLOT X*PI,0,1
200 CPLOT 2,-0.3
210 LABEL (230)X"PI"
220 NEXT X
230 FORMAT 2F4.1
240 FOR X=-4*PI TO 4*PI+PI/20 STEP PI/20
250 PLOT X,SIN(X)/X
260 NEXT X
270 END
```

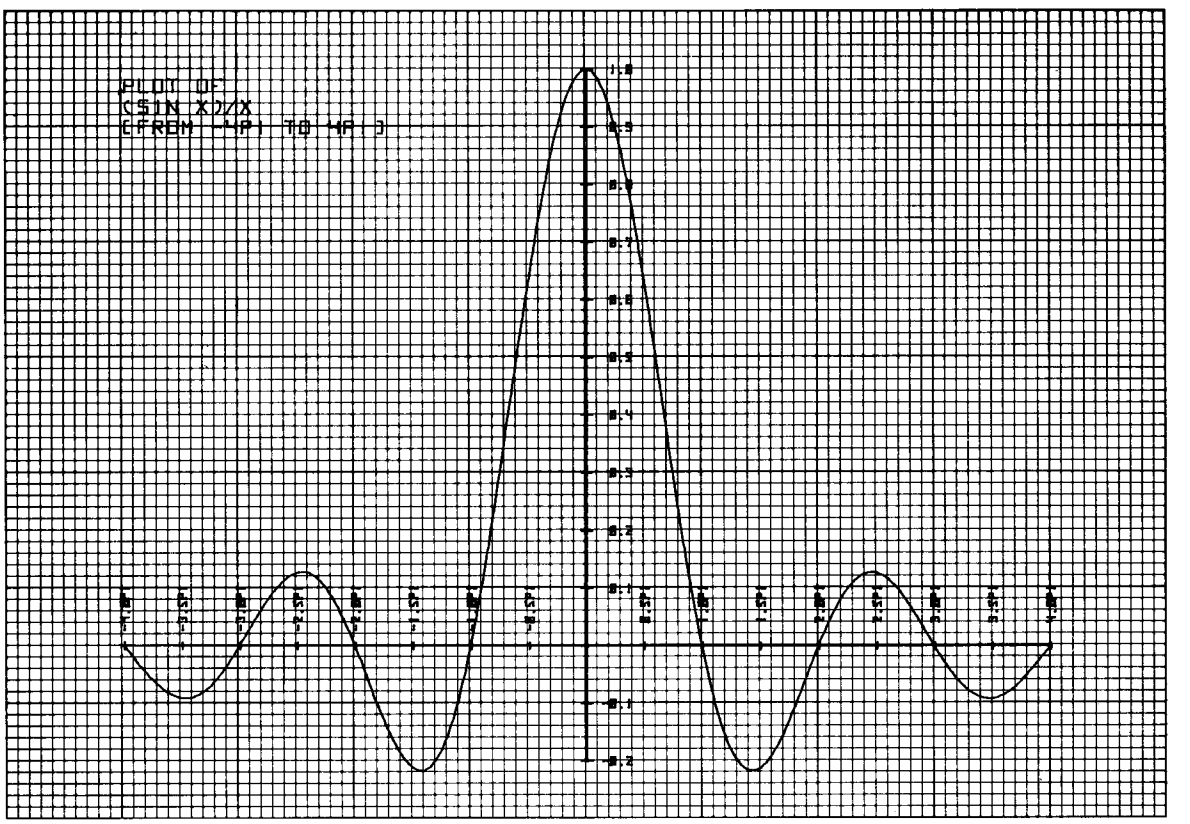


Figure 3-3. Illustrating LABEL and CPLOT Statements

EXAMPLE PROGRAMS-PRINTING
(Continued)

2. PLOTTING WITH X'S

This program uses the CPLOT and LABEL commands to draw an 'X' at each plotted point; at the same time a line is drawn between points. The program plots the function Y=2*X+3, for values of X from 1 to 6 (Figure 3-4).

The program illustrates a very important feature of the IPLOT command, namely, that it moves the pen with respect to the last point plotted and is not affected by any intermediate CPLOT and LABEL statements.

After each point has been plotted (line 50), the CPLOT and LABEL statements draw an 'X' centered on that point. The next statement (line 80) is 'IPLOT 0,0' which, instead of leaving the pen exactly where it is, as might perhaps be expected, returns the pen to the last point plotted.

The program assumes that the GRAPH LIMITS on the plotter are set for a square plotting area.

```
 10 SCALE 0,10,0,20
 20 XAXIS 0,1
 30 YAXIS 0,1
 40 FOR X=1 TO 6
 50 PLOT X,2*X+3
 60 CPLOT -0.3,-0.3
 70 LABEL (*)"X"
 80 IPLOT 0,0
 90 NEXT X
100 PEN
110 END
```

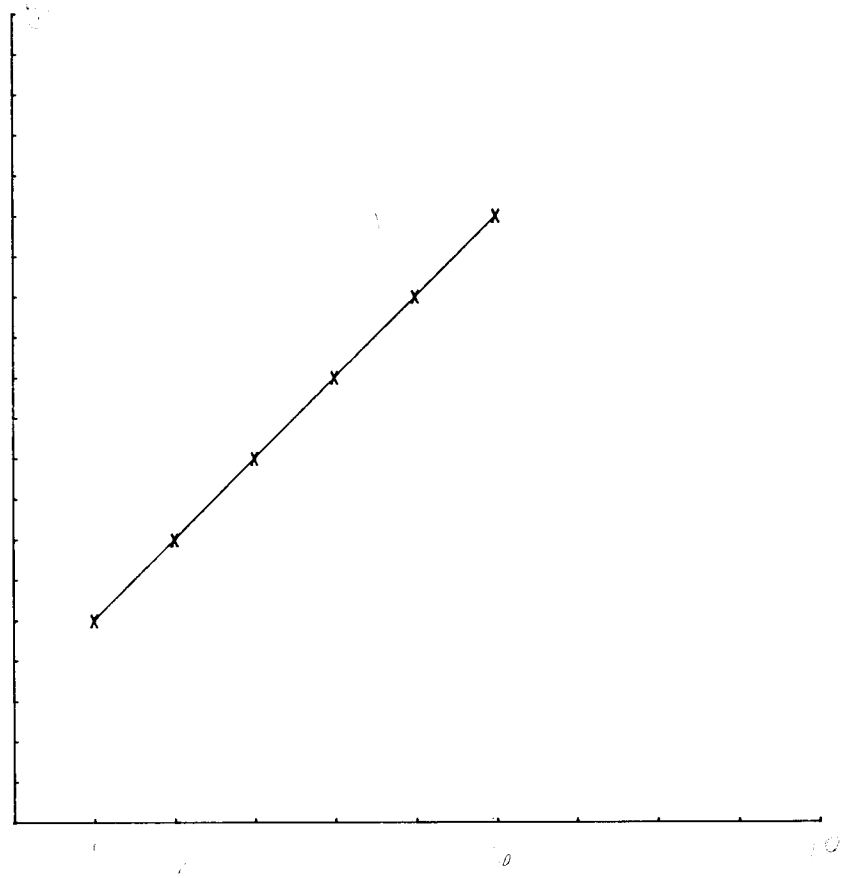


Figure 3-4. Plotting X's

EXAMPLE PROGRAMS-PRINTING
(Continued)

3. ROTATING THE PRINTOUT

This program demonstrates that a print-out can be made at any angle simply by changing the value of the angle-of-rotation parameter in a LABEL statement (Figure 3-5).

Standard 7 by 10 inch plotter paper is required; if your paper has different dimensions, change the parameters in the SCALE statement and also the paper height/paper width parameter in the LABEL statement (line 50), to suit the size of your plotter paper.

```
10 DEG
20 SCALE 0,10,0,7
30 FOR D=0 TO 350 STEP 10
40 PLOT 5,3.5,1
50 LABEL (*,2,1.7,D,7/10)
60 CPLOT 8,-0.3
70 LABEL (90)D
80 NEXT D
90 FORMAT F4.0,X,"DEGREES"
100 END
```

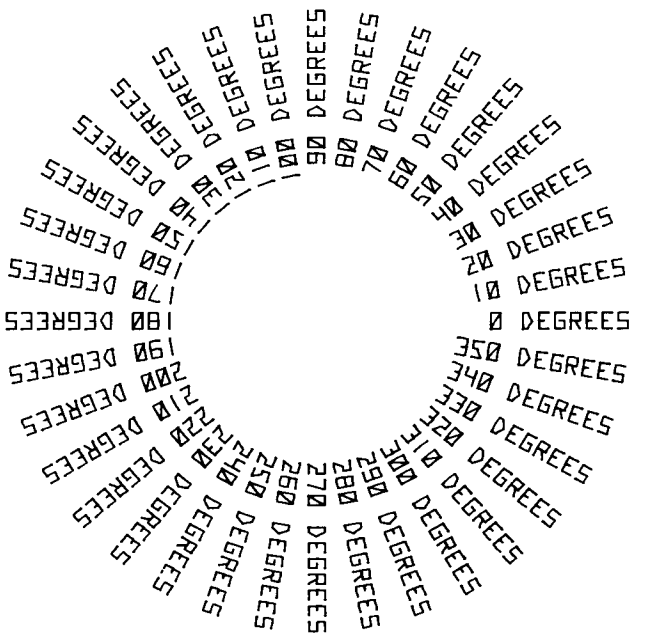


Figure 3-5. Angle of Rotation

4. DEMONSTRATING THE 'LETTER' STATEMENT

The LETTER statement is best demonstrated by asking you to experiment. Type the word LETTER into the display and then press the EXECUTE key. This establishes the 'typewriter' mode; the plotter is now ready to print a character, as indicated by the question mark in the display.

You can print any character by pressing its key, the size and direction of the character printed being determined by the current character specifications. You can also move the pen by using the four DISPLAY keys. Any time you wish to change the character specifications, press the STOP key, to terminate the typewriter mode, and execute a new LABEL statement.

Read 'The LETTER Statement', earlier in this chapter, for a full description of the typewriter mode.

NOTES

APPENDIX

ERROR MESSAGES

Many of the errors associated with plotting are similar to those in the calculator and result in the same error messages appearing. However, the Plotter block does have three unique errors associated with it.

ERROR 80	Attempt to execute an AXIS, OFFSET, PLOT or IPLOT statement before executing a SCALE statement.
ERROR 81	– 'Character height' specification in a LABEL statement greater than 18.4% of the height of the plotting area. – 'Aspect ratio' in a LABEL statement specifies a character width greater than 18.4% of the height of the plotting area. – The X or Y parameter in a CPLOT statement requires a pen movement greater than 18.4% of the height of the plotting area.
ERROR 82	Attempt to execute an AXIS statement a. with the 'start point' specified to be out of the plotting area, or b. with the tic mark spacing too small (i.e., space between tics is less than 1/9999 of the maximum width or height of the plotting area).

CHARACTER CODES

Shown below are the characters (together with their associated numeric codes) which are printed when FORMAT statements with 'B' parameters are referenced. Some of the characters, such as the square brackets and the quotation marks, cannot be printed on the plotter unless the 'B' format is used. For an example, see the first program (Labeling the Axes) at the end of Chapter 3; line 50 in that program uses codes 91 and 93 to print the square brackets.

The following program can be used to duplicate the list of characters and numeric codes. The program assumes a plotting area 10 inches high and 7 inches wide.

```
10 SCALE 0,70,0,100
20 PLOT 0,90,1
30 FOR J=32 TO 104 STEP 24
40 IPLOT 14,0,0
50 FOR K=J TO J+23
60 LABEL (90,1,1.7,0,100/70)K,K
70 NEXT K
80 NEXT J
90 FORMAT F4.0,2X,B,/
100 END
```

32	(space)	56	8	80	P	104	H
33	!	57	9	81	Q	105	I
34	"	58	:	82	R	106	J
35	#	59	;	83	S	107	K
36	$	60	<	84	T	108	L
37	%	61	=	85	U	109	M
38	&	62	>	86	V	110	N
39	'	63	?	87	W	111	O
40	(	64	@	88	X	112	P
41	)	65	A	89	Y	113	Q
42	*	66	B	90	Z	114	R
43	+	67	C	91	[	115	S
44	,	68	D	92	\	116	T
45	-	69	E	93	]	117	U
46	.	70	F	94	↑	118	V
47	/	71	G	95	←	119	W
48	0	72	H	96	@	120	X
49	1	73	I	97	A	121	Y
50	2	74	J	98	B	122	Z
51	3	75	K	99	C	123	[
52	4	76	L	100	D	124	\
53	5	77	M	101	E	125	]
54	6	78	N	102	F	126	↑
55	7	79	O	103	G	127	←

FOLD OUT ➡

SCALE

SCALE $value_1$, $value_2$, $value_3$, $value_4$
SCALE X_{min}, X_{max}, Y_{min}, Y_{max}

Scales plotting area to user units and establishes the origin (0,0) of the coordinate system.

PEN

PEN Raises Pen

OFFSET

OFFSET $value_1$, $value_2$
OFFSET X , Y

Redefines specified point (X,Y) to be the temporary origin (0,0) of the coordinate system.

XAXIS

XAXIS $value_1$ [, $value_2$ [, $value_3$, $value_4$]]
XAXIS Y-offset [, ±tic [, start point , end point]]

- If no optional parameters – draws horizontal line from X_{min} to X_{max}; crosses Y-axis at point defined by 'Y-offset'.

- ±tic – specifies spacing between tic marks; first tic at start point of axis.

- start point, end point – specifies start point and end point of axis other than X_{min} and X_{max}.

YAXIS

YAXIS $value_1$ [, $value_2$ [, $value_3$, $value_4$]]
YAXIS X-offset [, ±tic [, start point , end point]]

Similar to XAXIS except draws vertical line.

PLOTTER CONTROL CO

(All values can be a number,

PL

PLOT $value_1$, $value_2$ [, value
PLOT X, Y [, pen control]

- Moves pen to specified poin

- Pen control:

no parameter	– if per
	– if pe ment
positive	– pen c
negative	– pen c
odd	– raises
even	– lower
zero	– pen s

IPL

IPLOT $value_1$, $value_2$ [, value
IPLOT X, Y [, pen control]

Moves pen in X-direction amounts specified by the va control' see PLOT.)

LAB

LABEL (specifications) [list]
LABEL (line number or * [$value_4$]]) [list]

- 'Specifications' are:

  line number of a FORMAT
  $value_1$ – character heig plotting area).
  $value_2$ – character aspect
  $value_3$ – angle of rotation
  $value_4$ – distortion cor width).

- 'List' consists of:

  values, or literals (in quota

Establishes size of character prints contents of optional li FORMAT statement (if referer

PLOTTER CONTROL COMMANDS-A SUMMARY

(All values can be a number, a variable or an expression)

PLOT

PLOT $value_1$, $value_2$ [, $value_3$]

PLOT X, Y [, pen control]

- Moves pen to specified point (X,Y)

- Pen control:

no parameter	– if pen down, leaves pen down – if pen up, lowers pen after movement
positive	– pen control before movement
negative	– pen control after movement
odd	– raises pen
even	– lowers pen
zero	– pen status unchanged

IPLOT

IPLOT $value_1$, $value_2$ [, $value_3$]

IPLOT X, Y [, pen control]

Moves pen in X-direction and in Y-direction by the amounts specified by the values of X and Y. (For 'pen control' see PLOT.)

LABEL

LABEL (specifications) [list]

LABEL (line number or * [, $value_1$, $value_2$, $value_3$ [, $value_4$]]) [list]

- 'Specifications' are:

  line number of a FORMAT statement, or an asterisk.
  $value_1$ – character height (up to 18.4% height of plotting area).
  $value_2$ – character aspect ratio (height/width).
  $value_3$ – angle of rotation (degrees, radians or grads).
  $value_4$ – distortion correction (paper height/paper width).

- 'List' consists of:

  values, or literals (in quotation marks), or both.

Establishes size of characters and direction of printout; prints contents of optional list (if included) according to FORMAT statement (if referenced).

If no optional character para

character height – 1.5%
aspect ratio – 2
angle of rotation – 0°
paper height/paper width

LET

LETTER

- Establishes 'typewriter' each key pressed.
- DISPLAY keys move pe
- EXECUTE gives carriage
- STOP terminates the mo

CPL

CPLOT $value_1$, $value_2$

CPLOT character spaces wide,

Raises pen and moves it hori number of character spaces determined by previous LABE specifications if there was non

one character height = 0.6 *
one character width = 0.6 *

Horizontal and vertical direct of rotation previously specifie

L COMMANDS-A SUMMARY

number, a variable or an expression)

PLOT

[, value$_3$]

trol]

ed point (X,Y)

– if pen down, leaves pen down
– if pen up, lowers pen after movement
– pen control before movement
– pen control after movement
– raises pen
– lowers pen
– pen status unchanged

IPLOT

[, value$_3$]

ntrol]

ection and in Y-direction by the the values of X and Y. (For 'pen

LABEL

) [list]

or * [, value$_1$, value$_2$, value$_3$ [,

ORMAT statement, or an asterisk.
er height (up to 18.4% height of

aspect ratio (height/width).
otation (degrees, radians or grads).
n correction (paper height/paper

quotation marks), or both.

racters and direction of printout;
nal list (if included) according to
eferenced).

If no optional character parameters, following are assumed:

character height – 1.5%
aspect ratio – 2
angle of rotation – 0°
paper height/paper width – 1

LETTER

LETTER

- Establishes 'typewriter' mode – prints character of each key pressed.
- DISPLAY keys move pen without printing.
- EXECUTE gives carriage return/line feed.
- STOP terminates the mode.

CPLOT

CPLOT value$_1$, value$_2$

CPLOT character spaces wide, character spaces high

Raises pen and moves it horizontally and vertically by the number of character spaces specified. Size of 'space' is determined by previous LABEL statement (or by 'assumed' specifications if there was none) as follows:

one character height = 0.6 * height of one character space
one character width = 0.6 * width of one character space

Horizontal and vertical directions are subject to any angle of rotation previously specified in a LABEL statement.