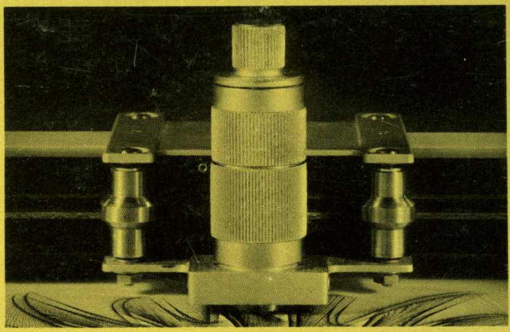
PROGRAMMING CALCOMP PEN PLOTTERS



PROGRAMMING CALCOMP PEN PLOTTERS

JUNE 1968

CALIFORNIA COMPUTER PRODUCTS, INC.
305 Muller Avenue, Anaheim, California 92803

INTRODUCTION

This manual provides the technical information that a FORTRAN-oriented programmer needs to write graphic computer programs utilizing a CalComp off-line or on-line pen plotting system. The specifications described herein apply to the CalComp Basic Software Package for Pen Plotters, which is supported for a large number of computing systems. Your local CalComp representative can supply details on availability.

Details of the principles used in the CalComp Basic Software may be found in the CalComp Software Reference Manual, No. 1005, which is provided to user installations.

TYPES OF SOFTWARE

CalComp provides three general classifications of plotting software.

The highest level is made up of Applications programs. These programs are the problem solvers that determine the output to be graphically displayed. CalComp can provide these proprietary programs, or the user may develop his own. Examples of applications packages available from CalComp include "General Purpose Contouring Package" and "FLOWGEN/F" for generation of plotter-drawn flowcharts from source programs written in FORTRAN.

The second type is classified as Functional Software. Functional programs or subprograms perform plotting functions frequently used in many different applications, and are offered in package form. Examples include CIRCLE, SHADE, GRID, and SMOOTH.

The third classification is called Basic Software, which is a set of closely related subroutines that generate output for a plotting system. The programmer is not required to communicate with the hardware in its own data structure. Instead, he communicates with the set of subroutines in a manner such as: "move the pen to a specified coordinate"; "place some characters at a certain location on the page"; "draw an annotated axis"; or "scale and draw a line through a series of points." This kind of communication reduces the problem of formatting graphic output to the same level as formatting the data for a printed report.

HARDWARE CAPABILITY

CalComp Basic Software converts the user's data to commands to drive CalComp pen plotters (drum or flatbed, incremental or ZIP MODE) with resolution ranging from 0.01 inch to 0.001 inch. These plotters may be driven on-line through CalComp interface controllers attached to the user's computer, or off-line from CalComp magnetic tape units. In addition, telecommunications hardware is available to permit plotters to be used remotely with time-shared computer systems.

CalComp pen plotting systems are designed to respond to digital incremental commands which cause either a lateral movement of the pen (Y-axis), a movement of the drum or flatbed beam (X-axis), or a combined movement of both (X, Y) in some ratio depending on the system. The straight line algorithm in the Basic Software determines the appropriate sequence of commands needed to move the pen from one point to another, and up or down.

COMPATIBILITY FEATURES

The modular design of CalComp hardware and software allows the user to upgrade or modify most CalComp plotting systems without the massive conversion problems usually associated with hardware modifications. In most cases, a change in the user's computer or plotting system requires little or no modification to operational application programs, if they have been written in a high-level language such as FORTRAN.

CALCOMP SOFTWARE SUPPORT POLICY

CalComp software is an integral and essential part of our product line. Like individual hardware items, each software package is given a product number and is supported with appropriate literature and documentation. Because of the proprietary nature of CalComp software, the packages are leased rather than sold outright. The lease agreement allows unrestricted use of the software with your CalComp system, and the lease price covers this usage for as long as you retain the system.

CalComp guarantees that its basic software will perform according to specifications at the time of installation. For certain on-line systems, CalComp will also assist in the installation and checkout of the Basic Software at your facility.

A SAMPLE PLOTTING PROGRAM

To illustrate the use of CalComp Basic Software, the sample program that produced the graph below is shown on the next page. The only assumptions made are that: (1) the 24 pairs of TIME and VOLTAGE data values are contained in a file of 24 records; and (2) the plotting pen is initially positioned at the extreme -Y side of the plotter. Notice that only 11 executable statements are required to complete the graph.

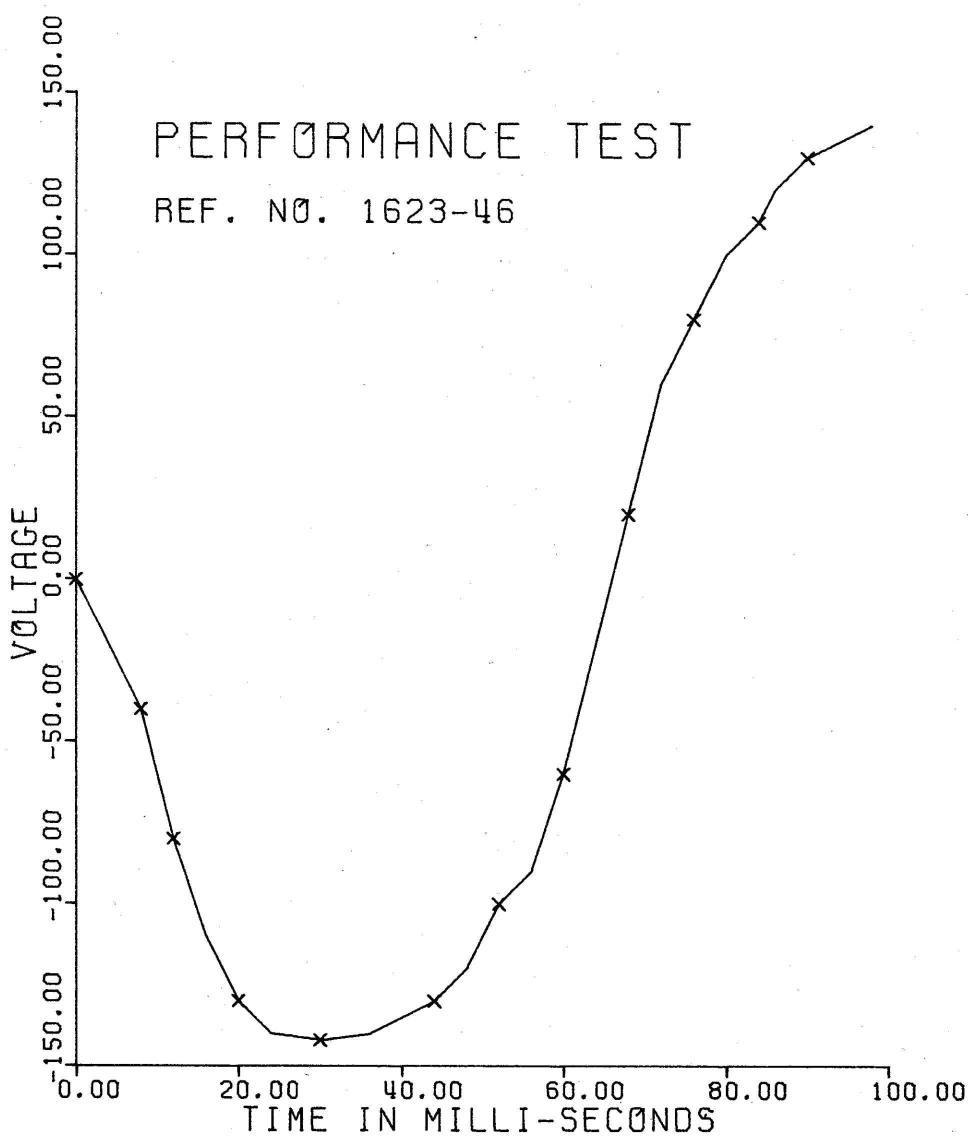


Figure 1

DIMENSION IBUF (1000), XARRAY (26), YARRAY (26)

Reserve 1000 locations for PLOT work area and buffer.
Reserve space for 24 data values plus two additional locations required by the SCALE, AXIS, and LINE subroutines.

10 CALL PLOTS (IBUF, 1000, 6)

Initialize the PLOT subroutine, giving the name of the PLOT work area, its size, and the logical number for the output device.

20 READ 25, (XARRAY (I), YARRAY(I), I=1, 24)

25 FORMAT (2F6.2)

Read 24 pairs of TIME and VOLTAGE from an input file into two arrays with names XARRAY and YARRAY.

30 CALL PLOT (0.0, 0.5, −3)

Establish a new origin one-half inch higher than the point where the pen was initially placed by the operator so that the annotation of the TIME axis will fit between the axis and the edge of the plotting surface.

40 CALL SCALE (XARRAY, 5.0, 24, 1)

Compute scale factors for use in plotting the TIME values within a five-inch plotting area.

50 CALL SCALE (YARRAY, 6.0, 24, 1)

Compute scale factors for use in plotting the VOLTAGE data values within a six-inch plotting area (i.e., the data pairs of TIME, VOLTAGE will plot within a five-by-six-inch area).

60 CALL AXIS (0.0, 0.0, 21HTIME IN MILLI-SECONDS, −21, 5.0, 0.0, XARRAY(25), XARRAY(26))

Draw the TIME axis (5 inches long), using the scale factors computed in statement 40 to determine the milli-seconds at each inch along the TIME axis.

70 CALL AXIS (0.0, 0.0, 7HVOLTAGE, 7, 6.0, 90.0, YARRAY(25), YARRAY(26))

Draw the VOLTAGE axis (6 inches long), using the scale factors computed in statement 50 to determine the voltage at each inch along the VOLTAGE axis.

80 CALL LINE (XARRAY, YARRAY, 24, 1, 2, 4)

Plot VOLTAGE vs TIME, drawing a line between each of the 24 scaled points and a symbol X at every other point.

90 CALL SYMBOL (0.5, 5.6, 0.21, 16HPERFORMANCE TEST, 0.0, 16)

Plot the first line of the graph title.

100 CALL SYMBOL (0.5, 5.2, 0.14, 16HREF. NO. 1623-46, 0.0, 16)

Plot the second line of the graph title.

110 CALL PLOT (12.0, 0.0, 999)

Advance the pen beyond the current plotting area, write a terminating record, and close the plot output device.

120 STOP

Terminate program execution.

END

PLANNING YOUR GRAPH

Graphs and plots, like computer listings, require some planning to achieve a pleasing and effective format. The following check list of standard plotting conventions is keyed to Figure 2 to help you in your planning.

A. The initial position of the pen when the plotting operation begins is assumed to be the logical origin (X=0, Y=0). All pen movements are defined in X-coordinate and Y-coordinate inches from the current origin. Subsequent origins can be established at other positions by appropriate programming to provide new reference points. (See PLOT.) When the next graph is started, a new origin should be defined far enough away to avoid any axis or annotation conflict with the previous graph.

B. The X-axis lies parallel to the side of the drum plotter, with the +X direction toward the back. The X-axis line should be at least ½ inch from the sides of the plotter to leave a margin for labeling.

   The maximum width of a graph (X-direction) is limited only by the length of the paper: On a drum plotter it is usually 120 feet; on flatbed plotter models 502, 602, and 702 it is 34 inches; on flatbed plotter models 618, and 718 it is 72 inches.

C. The Y-axis is parallel to the width of the plotter. The Y-axis line should also be at least ½ inch from the sides of the plotter to leave a margin for the label and annotation.

   The maximum height of a graph (Y-direction) is determined by the pen carriage movement: On drum plotter models 565, 575, 665, and 765 it is 11 inches; on models 563, 663, and 763 it is 29½ inches; on flatbed plotter models 502, 602, and 702 it is 31 inches; on models 618 and 718 it is 54 inches.

D. The angle of rotation about any point is determined by a vector which when extended in the +X direction represents 0 degrees. Any angle argument used in a calling sequence may be stated in plus or minus degrees relative to the X-axis.

E. When drawing several graphs in one program, it may be desirable to draw trim lines for the operator's convenience in separating the graphs.

F. After the last graph has been drawn, the pen should be moved to a position that permits easy removal of the graphs. On the drum plotters, this would be several inches beyond the end margin to allow the paper to be torn off the roll. On flatbed plotters, this position would be the extreme −X, −Y corner of the bed.

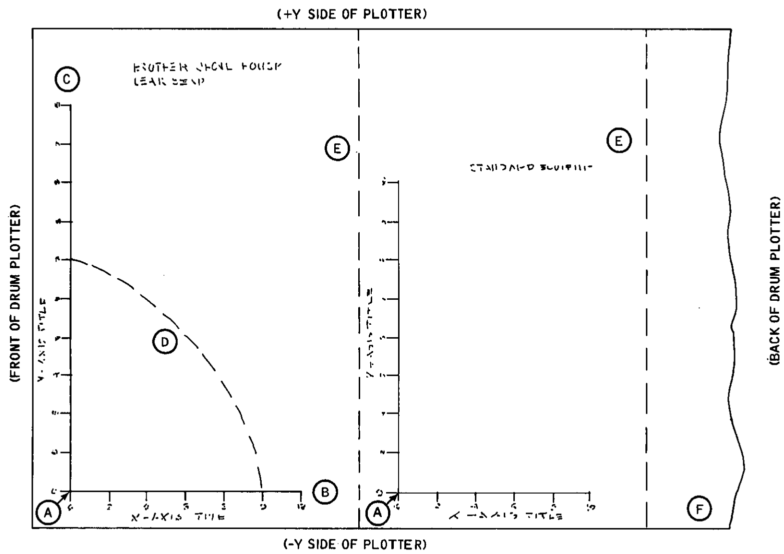


Figure 2

STANDARD SOFTWARE MAKES IT EASY

The following pages describe in detail the CalComp Basic Software, comprising six subroutines that can be called by your FORTRAN program. They, in turn, call upon each other for certain operations. A particular application may not need all of these subroutines, depending on the user's requirements.

PLOT converts all pen movement specifications from inches to actual plotter commands, and outputs these to the attached device.

SYMBOL draws any sequence of alphanumeric characters, as well as special point-plotting symbols. It calls only PLOT.

NUMBER draws the fixed decimal equivalent of an internal floating point number. It calls only SYMBOL, which then calls PLOT.

SCALE examines a data array to determine an optimum starting value and a scaling factor for use by AXIS and LINE in converting data units to plotter page dimensions. It is the only subroutine that does not call any other because it does no actual plotting.

AXIS draws an axis line with the appropriate scale annotation and title. It calls SYMBOL and NUMBER as well as PLOT.

LINE plots a series of scaled data points defined by two arrays (X and Y), connecting the points with straight lines if desired. It may call SYMBOL as well as PLOT.

The argument names used in the following subroutine descriptions conform to standard FORTRAN conventions: If the first letter is I, J, K, L, M, or N, the argument must be an integer value; if it is any other letter, the argument must be a real floating-point value. The description of each argument is generally consistent for most computers and for all CalComp pen plotting systems. However, the programmer should also refer to the CalComp Programming Supplement on the computing system being used, so that he will be informed of any special considerations.

BASIC PLOTTING

Most graphic applications require the generation of X-Y graphs to show the relationship between two or more sets of data. Usually these graphs can be produced easily and quickly by a suitably programmed combination of the five supporting subroutines SCALE, AXIS, LINE, SYMBOL, and NUMBER. These subroutines do not directly produce plotter commands; they only compute appropriate arguments that define pen positions, and then call the PLOT subroutine, which generates the actual plotter commands.

When unique plotting requirements cannot be satisfied by using the supporting subroutines, the user can resort to the PLOT subroutine, which gives him direct control of the pen movement (to any X, Y coordinates position), pen status (up or down), and generation of Search records.

By calling the PLOT subroutine with a different entry name (PLOTS, FACTOR, WHERE, or NEWPEN), the user also has control of: plot output record size; output device opening and closing; number of plotter steps per logical inch; locating the current pen position; and selecting any pen in a Model 618 or 718 multi-pen system. Each entry is described separately below.

PLOT

The PLOT entry to the PLOT subroutine is used primarily to move the pen in a straight line to a new position, with the pen either up or down during the movement. It converts the arguments to the appropriate sequence of plotter commands, and outputs these to the attached device (tape, disk, drum, or plotter controller).

The calling sequence has 3 arguments:

CALL PLOT (XPAGE, YPAGE, ± IPEN)

XPAGE, YPAGE — are the X, Y coordinates of the terminal position to which the pen is moved, in inches from the current reference point (origin). An origin (where both X, Y equal zero) may be established anywhere on (or off) the plotting surface, as explained below for negative IPEN values.

±IPEN — is a signed integer which controls the pen up/down status, origin definition, and the generation of Search records.

If IPEN=2, the pen is *down* during movement, thus drawing a visible line.

If IPEN=3, the pen is *up* during movement.

If IPEN=−2 or −3, a new origin is defined at the terminal position after the movement is completed as if IPEN were positive. The logical X, Y coordinates of the new pen position are set equal to zero, so that that position is the reference point for succeeding pen movements. In addition, all of the plotter commands accumulated in the output buffer area are transmitted to the output device.

A Search record with the next sequential Search address is also produced at this time if the output device is a tape unit. The Search records of a plot tape furnish the means of locating with the CalComp off-line tape unit, any desired plotting records.

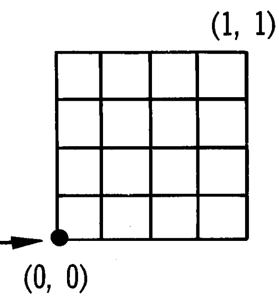
When operating the CalComp tape unit in Search mode or Multi-plot mode, a Search address is preset by means of a three-digit thumbswitch. When the corresponding Search record is found, the tape unit stops.

IPEN must be 999 in the last plotter call in a program. The effects are the same as if IPEN=−3; except that a "999" Search record is written if the plotting system is off-line, and the output device is then closed.

The examples in Figure 3 show the pen movements that result from a series of calls to the PLOT subroutine. The initial call to PLOTS and an appropriate Dimension statement for the PLOT buffer area, as well as a call to FACTOR, are included. Opposite each call is shown the Search-record address that would be produced for an off-line plotting system.

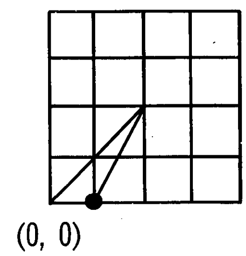
Search Address

DIMENSION IBUF(1000)
10 CALL PLOTS (IBUF, 1000, LDEV)
This indicates the terminal position of the pen.

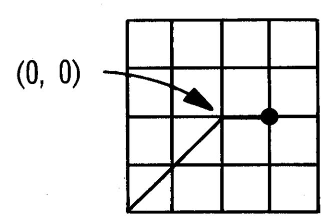


001

20 CALL PLOT (0.5, 0.5, 2)
30 CALL PLOT (.25, 0., 2)

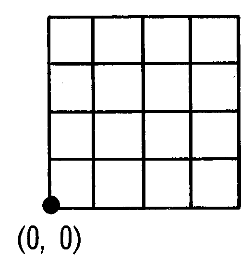


40 CALL PLOT (0., 0., 3)
Returns the pen to the origin.
50 CALL PLOT (0.5, 0.5, —2)
Defines a new origin.
60 CALL PLOT (.25, 0., 2)
Note that the X, Y values in statements #50, 60 are the same as those used in #20, 30.



002

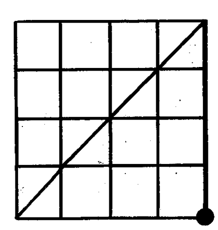
70 CALL PLOT (—0.5, —0.5, —3)
Positions the pen at the initial origin and defines this as the new origin.



003

80 CALL FACTOR (0.5)
Reduces the following pen movements by ½.
90 CALL PLOT (2.0, 2.0, 2)

100 CALL PLOT (2.0, 0., 999)



999

Figure 3

PLOTS

The PLOTS entry is used to initialize the PLOT subroutine. It must be called only once—before any other call to PLOT, SYMBOL, NUMBER, AXIS, or LINE is given. This entry sets up certain constants and the plot buffer area from which the plotter commands are written, and it opens the plot output device by performing standard file opening procedures through the computer's operating system. If the output device is a tape unit, the first Search record, with Search address #001, is written out. Figure 3 includes an example of using PLOTS.

This calling sequence also has 3 arguments:

CALL PLOTS (IBUF, NLOC, LDEV)

IBUF is the name of a large area of storage assigned to accumulate the plotter commands produced by PLOT and to buffer the output. This area should be defined by a DIMENSION statement as an array.

NLOC is the number of locations reserved for the buffer area IBUF. Consult your computer-oriented supplement for the particular manner of defining the size, which is specified in words for some computers and in characters or bytes for others. Typically, the size should be large enough to produce plot records of at least several hundred commands. This value should correspond to the array size specified in the DIMENSION statement for IBUF.

For users of 600-series plotters driven by either a Model 750 or 760 tape unit, the sign of NLOC is used to set the logical increment size in the program to match the actual switch setting used on the plotter. +NLOC corresponds to the small increment size, and −NLOC corresponds to the large increment size.

LDEV is the logical output-device number, which is assigned by the user. In some versions of PLOT this argument may not be applicable, depending on the characteristics of the operating system. Consult your computer-oriented supplement for details.

FACTOR

The FACTOR entry to the PLOT subroutine enables the user to enlarge or reduce the size of the entire plot by changing the effective number of plotter steps per inch of page coordinates. A sample FACTOR statement is shown in Figure 3.

CALL FACTOR (FACT)

FACT is the ratio of the new plot size to the normal plot size. For example, if FACT=2.0, all following pen movements will be twice their normal size. When FACT is reset to 1.0, all plotting returns to normal size. A saving in computer and plotter time is made possible during the debugging stage of a plotting application program by calling FACTOR with a value less than 1.0 (after calling PLOTS), so the entire plot output will be reduced in size. When debugging is completed, this call statement can be removed.

WHERE

The WHERE entry returns the current pen-position coordinates and the scaling factor (that are in use by the PLOT subroutine) to the three locations designated in the calling sequence. This permits user-written subroutines to know the current pen location for optimizing pen movement.

There are 3 arguments in the calling sequence:

CALL WHERE (RXPAGE, RYPAGE, RFACT)

RXPAGE, RYPAGE are locations that will be filled with the current pen position coordinates resulting from the previous call to PLOT (which may have been called by SYMBOL, NUMBER, AXIS, or LINE).

RFACT is filled with the current plot scaling factor; i.e., the value last supplied by a call to FACTOR, or 1.0 if FACTOR has not been called.

NEWPEN

The NEWPEN entry is provided only in PLOT subroutines for the Model 618 or 718 flatbed plotters. This entry enables program selection of any one of the four available pens. Pen # 1 is the bottom-most of the pens, which are spaced 0.6 inch apart in the Y-axis direction, and it is initially selected when the PLOTS entry is called.

The calling sequence is:

CALL NEWPEN (INP)

INP specifies the number of the pen to be selected (1-4). The old pen is raised, and the new pen is moved to the same physical location where the old pen was positioned. The user must guard against the situation where the new pen cannot be moved to the previous pen's position. This can happen if the distance between the new and old pens is greater than the distance from the old pen to either Y-axis limit (top or bottom edge of the plotter).

DRAWING TEXT AND SYMBOLS

The SYMBOL subroutine produces plot annotation at any angle and in practically any size. There are two SYMBOL call formats: (1) the "standard" call, which can be used to draw text such as titles, captions, and legends; and (2) the "special" call, which is used to draw special centered symbols such as a box, octagon, triangle, etc., for plotting data points.

The standard characters that are drawn by SYMBOL include the letters A-Z, digits 0-9, and special characters (+ − & = ( ) #, and .). See your computer-oriented supplement to this manual for other characters available in your particular SYMBOL routine.

Both forms of the SYMBOL calling sequence have 6 arguments. The "standard" call is:

CALL SYMBOL (XPAGE, YPAGE, HEIGHT, IBCD, ANGLE, +NCHAR)

XPAGE, YPAGE are the coordinates, in inches, of the lower left-hand corner (before rotation) of the first character to be produced. The pen is up while moving to this point.

Annotation may be continued from the position following where the last annotation ended. Continuation occurs when XPAGE and/or YPAGE equals 999.0, and may be applied to X or Y independently. (Calling WHERE to obtain the current pen position and using RXPAGE, RYPAGE in another call to SYMBOL would not give the same results as using 999.)

HEIGHT is the height, in inches, of the character to be plotted. For best results, it should be a multiple of seven times the plotter increment size (e.g., .07, .14, .21), but other values are acceptable. The width of a character, including spacing, is normally the same as the height (e.g., a string of 10 characters 0.14 inch high is 1.4 inches wide).

IBCD is the text to be used as annotation, in internal computer representation (usually BCD or A-type format). The character(s) must be left-justified and contiguous in a single variable, in an array, or in a Hollerith literal (if the compiler permits). Nonsignificant blanks at the end of the text do not cause any pen movement.

The text must be right-justified in IBCD if a single character is desired and NCHAR = 0.

ANGLE is the angle, in degrees, at which the annotation is to be plotted. If ANGLE = 0., the character(s) will be plotted right side up and parallel to the X-AXIS.

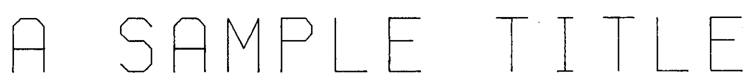
+NCHAR is the number of characters to be plotted from IBCD. If NCHAR > 0, the data must be left-justified in the first location of IBCD.

If NCHAR = 0, one alphanumeric character is produced, using a single character which is right-justified in the first location of IBCD.

Some examples of using the "standard" call to SYMBOL are shown in Figure 4.

```
DIMENSION IBCD (4)
CALL SYMBOL (X,Y,.28,IBCD,0.,16)
```

Note: IBCD is a 4-word array containing 16 characters.




```
CALL SYMBOL (X,Y,.14,IBCD,0.,16)
CALL SYMBOL (999.,999.,.14,IBCD,90.,16)
CALL SYMBOL (999.,999.,.14,IBCD,180.,16)
CALL SYMBOL (999.,999.,.14,IBCD,270.,16)
```

Note: two spaces follow "A SAMPLE TITLE"

Figure 4

The second form is the "special" call, which produces only a single symbol based on the index value of INTEQ — not on the BCD representation of a character.

CALL SYMBOL (XPAGE, YPAGE, HEIGHT, INTEQ, ANGLE, —ICODE)

XPAGE, YPAGE, HEIGHT, and ANGLE are the same as described for the "standard" call. If the symbol to be produced is one of the centered symbols (i.e., if INTEQ is less than 14), XPAGE, YPAGE represent the geometric center of the character produced.

INTEQ is the integer equivalent of the desired symbol, both of which are shown in the symbol table of the computer supplement. If INTEQ is 0 through 13, a centered symbol is produced. (See Figure 5A.)

—ICODE is negative and determines the status of the pen during the move to XPAGE, YPAGE.

When ICODE is:

—1, the pen is up during the move, after which a single symbol is produced;

—2, or less, the pen is down during the move, after which a single symbol is produced.

DRAWING NUMBERS

NUMBER functions as a pre-processor to the SYMBOL subroutine. It converts a real variable (floating-point number) to the appropriate fixed-decimal equivalent so that it may be plotted by subroutine SYMBOL in the FORTRAN F-type format.

CALL NUMBER (XPAGE, YPAGE, HEIGHT, FPN, ANGLE, ±NDEC)

XPAGE, YPAGE, HEIGHT, and ANGLE are the same as described for subroutine SYMBOL. The continuation feature, where XPAGE or YPAGE equals 999., may also be used.

FPN is the location of the floating-point number that is to be converted and plotted.

±NDEC controls the precision of the conversion of the number FPN. If the value of NDEC > 0, it specifies the number of digits to the right of the decimal point that are to be converted and plotted, after proper rounding. For example, assume an internal value (perhaps in binary form) of $-.12345678 \times 10^3$. If NDEC were 2, the plotted number would be —123.46.

If NDEC $= 0$, only the integer portion of the number and a decimal point are plotted, after rounding.

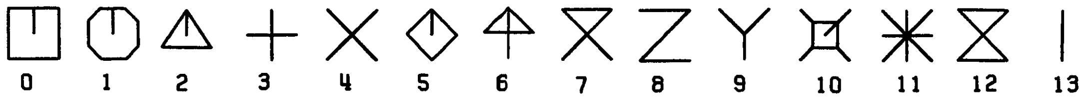
If NDEC $= -1$, only the integer portion of the number is plotted, after rounding. The above example would be plotted as —123 with no decimal point.

If NDEC < -1, $|NDEC| - 1$ digits are truncated from the integer portion, after rounding.

The magnitude of NDEC should not exceed 9.

The examples in Figure 5B-5D show various uses of SYMBOL and NUMBER.

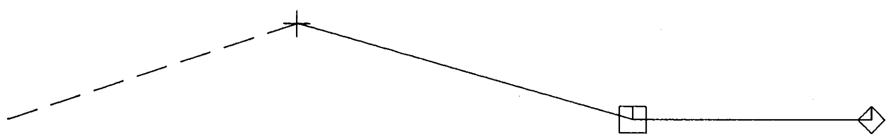
A: Centered Symbols



B: Plotting Data Points

```
CALL SYMBOL (1.5,0.5,.14,3,0.,−1)
CALL SYMBOL (3.25,0.,.14,0,0.,−2)
CALL SYMBOL (4.5,0.0,.14,5,0.,−2)
```

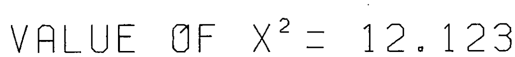
Pen is Up — — —
Pen is Down } ———
Pen is Down



C: Combining SYMBOL and NUMBER and Drawing a Superscript

```
CALL SYMBOL (X, Y,.14,10HVALUE OF X,0.,10)
CALL SYMBOL (999.,Y+.1,.07,2H2  ,0.,2)
CALL SYMBOL (999.,Y,.14,2H=  ,0.,2)
CALL NUMBER (999.,999.,.14,VALUE,0.,3)
```

Superscript



D: Drawing Text and Numbers at Various Angles

```
   DO 10 I=0,315,45
   ANGLE=I
   CALL SYMBOL (X,Y,.1,7H  ANGLE=,ANGLE,7)
10 CALL NUMBER (999.,999.,.1,ANGLE,ANGLE,−1)
```

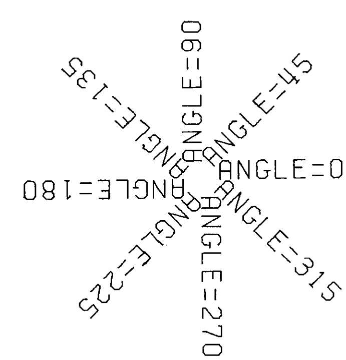


Figure 5

SCALING YOUR DATA

Typically, the user's program will accumulate plotting data in two arrays: (1) an array of independent variables, X_i; and (2) an array of dependent variables, $Y_i = f(X_i)$. It would be unusual, however, if the range of values in each array corresponded exactly with the number of inches available in the actual plotting area. For some problems the range of data is predictable and the programmer can pre-determine suitable conversion factors for use in drawing the axis scale values and plotting the data points on the graph. However, in general, these factors are not known in advance.

Therefore, the SCALE routine is used to examine the data values in an array and to determine a starting value, either minimum or maximum, and a scaling factor, positive or negative, such that: (1) the scale annotation drawn by the AXIS subroutine at each division will properly represent the range of real data values in the array, and (2) the data points, when plotted by the LINE subroutine, will fit in a given plotting area. These two values are computed and stored by SCALE at the end of the array.

The scaling factor (DELTAV) that is computed represents the number of data units per inch of axis, but is adjusted so that it is always an interval of 1, 2, 4, 5, or 8 x 10^n (where n is an exponent consistent with the original unadjusted scaling factor). Thus, an array may have a range of values from 301 to 912, to be plotted over an axis of 10 inches. The unadjusted scaling factor is (912-301)/10 = 61.1 units/inch. The adjusted DELTAV would be 8 x 10^1 = 80.

The starting value (FIRSTV), which will appear as the first annotation on the axis, is computed as some multiple of DELTAV which is equal to or outside the limits of the data in the array. For the example given above, if a minimum is wanted for FIRSTV, 240 would be chosen as the best value. If a maximum is desired instead, 960 would be selected.

There are 4 arguments in the calling sequence:

CALL SCALE (ARRAY, AXLEN, NPTS, ±INC)

ARRAY is the first location of the array of data points to be examined.

AXLEN is the length of the axis to which the data is to be scaled. Its value must be greater than 1.0 inch.

NPTS is the number of data values to be scanned in the array. The FORTRAN dimension statement should specify at least two locations more than the number of values being scanned, to allow room for SCALE to store the computed starting value and scaling factor at the end of the array.

±INC is an integer whose magnitude is used by SCALE as the increment with which to select the data values to be scanned in the array. Normally |INC| = 1; if it is 2, every other value is examined.

If INC is positive, the selected starting value (FIRSTV) approximates a minimum, and the scale factor (DELTAV) is positive.

If INC is negative, the selected starting value (FIRSTV) approximates a maximum, and the scaling factor (DELTAV) is negative.

If INC = ±1, the array must be dimensioned at least two locations larger than the actual number of data values it contains. If the magnitude of INC > 1, the computed values are stored at (INC) locations and (2*INC) locations beyond the last data point. The subscripted location for FIRSTV is ARRAY(NPTS*INC+1); for DELTAV it is ARRAY (NPTS*INC+INC+1).

Generally, SCALE is called to examine each array that will be plotted on the graph, as shown in the Sample Program. If the user knows the range of his data values he does not need to call SCALE for that array so long as he supplies an appropriate FIRSTV and DELTAV when AXIS and LINE are called.

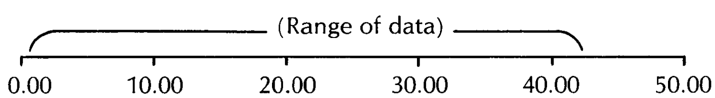
The following examples illustrate some typical uses of SCALE.

Example 1. — Given an array of 24 data values to be plotted over a 5-inch axis, assume the minimum value in the array is 1.00 and the maximum is 42.00. The statement CALL SCALE (ARRAY, 5.0, 24, +1) would give the following results.

Units/inch = (42.00 - 1.00)/5.0 = 8.2
DELTAV (next higher interval) = 10.
FIRSTV (minimum multiple) = 0.00

FIRSTV value is stored in ARRAY (25)
DELTAV value is stored in ARRAY (26)

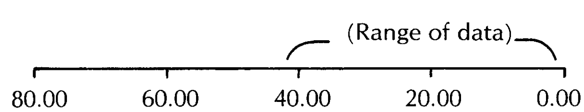
Using these values, AXIS would draw this axis line:



Example 2. — Assume the same array is to be plotted on a 4-inch axis, from maximum to minimum. CALL SCALE (ARRAY, 4.0, 24, −1) would give these results:

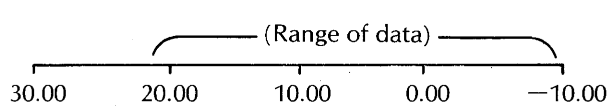
DELTAV = (1.00 - 42.00)/4.0 = −10.25 which is adjusted to −20.
Minimum multiple = 0.00; FIRSTV = Minimum + (AXLEN*|DELTAV|) = 80.00.

In this case the axis would be drawn:



Example 3. — Assume 100 points are to be plotted on a 4-inch axis from maximum (+22) to minimum (−9), using *every other* data value in the array. The DIMENSION statement should specify ARRAY (204), and the calling sequence is CALL SCALE (ARRAY, 4.0, 100, −2).

Initial DELTAV = (−9 −22)/4 = −7.75, adjusted to −8.
Initial FIRSTV = +16.00; last value on axis = −16.00.
The axis range is inadequate for the data range, so DELTAV is revised to the next higher interval.
Revised DELTAV = −10., stored in ARRAY(203).
Revised FIRSTV = 30.00, stored in ARRAY(201).



DRAWING THE AXIS

Most graphs require axis lines and scales to indicate the orientation and values of the plotted data points. The most common type of scaled axis is easily produced by the AXIS subroutine, which draws any length line at any angle, divides it into one-inch intervals, annotates the divisions with appropriate scale values, and labels the axis with a centered title. When both X and Y axes are needed, AXIS is called separately for each one.

There are 8 arguments in the calling sequence:

CALL AXIS (XPAGE, YPAGE, IBCD, ±NCHAR, AXLEN, ANGLE, FIRSTV, DELTAV)

XPAGE, YPAGE — are the coordinates of the starting point of the axis line. The entire line and terminal ends should be at least one-half inch from any side to allow space for the scale annotation and axis title. Usually, both X and Y axes are joined at the origin of the graph, where XPAGE and YPAGE equal zero; but other starting points can be used if desired. When using the LINE subroutine, however, at least one of the coordinates must be 0 ; i.e., for an X-axis, XPAGE = 0, and for a Y-axis, YPAGE = 0.

IBCD — is the title, which is centered and placed parallel to the axis line. This parameter may be an alphameric array, or it may be a Hollerith literal if the FORTRAN compiler being used permits it. The characters have a fixed height of 0.14 inch (about 7 characters per inch).

±NCHAR — specifies the number of characters in the axis title, and determines, by its sign, which side of the line the scale (tick) marks and labeling information shall be placed. Since the axis line may be drawn at any angle, the line itself is used as a reference.

If the sign is *positive*, all annotation appears on the positive (counterclockwise) side of the axis, which condition is normally desired for the Y-axis.

If the sign is *negative*, all annotation appears on the negative (clockwise) side of the axis, which condition is normally desired for the X-axis.

AXLEN — is the length of the axis line, in inches.

ANGLE — is the angle, in positive or negative degrees, at which the axis is to be drawn. Normally, this value is zero for the X-axis and 90.0 for the Y-axis.

FIRSTV — is the starting value (either minimum or maximum) which will appear at the first tick mark on the axis. This may be either the value computed by the SCALE subroutine, stored at subscripted location ARRAY(NPTS*INC+1), or a value determined by the user, stored anywhere.

This number and each scale value along the axis is always drawn with two decimal places. Since the digit size is 0.105 inch (about 10 characters per inch), and since a scale value appears every inch, no more than six digits and a sign should appear to the left of the decimal point.

DELTAV represents the ratio of data units per inch of axis. This increment (or decrement) is added to FIRSTV for each succeeding one-inch division along the axis. This may be the value computed by SCALE, stored beyond FIRSTV at ARRAY(NPTS*INC+INC+1), or a value determined by the user, stored anywhere.

In order to use a standard format of two decimal places, the size of DELTAV is adjusted so it is less than 100, but not less than 0.01. As a result, the decimal point may be shifted left or right in the scale values as drawn, and the axis title is then followed by "$*10^n$," where n is the power-of-ten adjustment factor. (See X-axis example in Figure 6.)

Figure 6 illustrates those typical X and Y axis elements controlled by the arguments of AXIS.

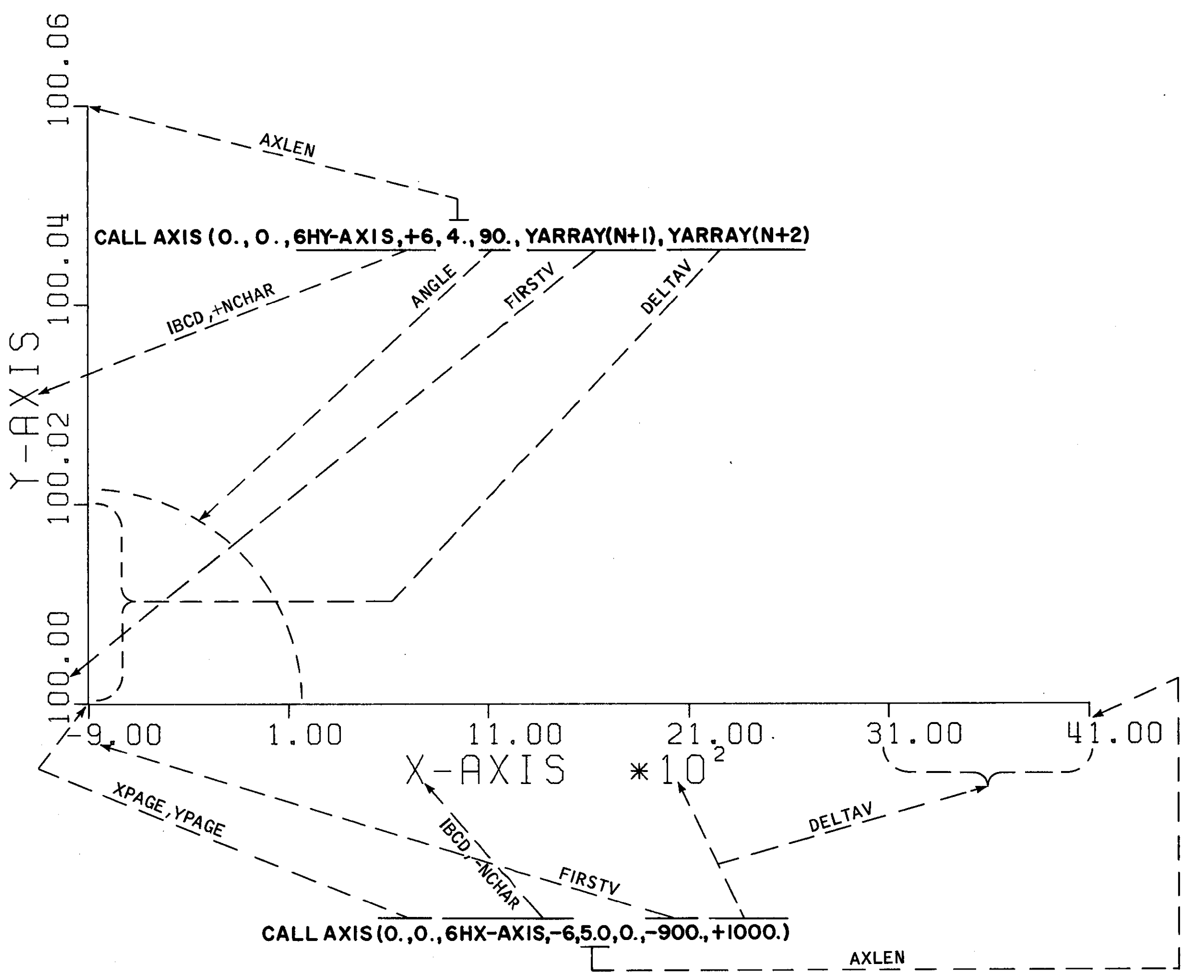


Figure 6

DRAWING THE DATA LINE

The LINE subroutine produces a line plot of the pairs of data values in two arrays (X and Y). LINE computes the page coordinates of each plotted point according to the data values in each array and the respective scaling parameters. The data points may be represented by centered symbols and/or connecting lines between points.

The scaling parameters corresponding to FIRSTV and DELTAV (see SCALE) must immediately follow each array. If these parameters have not been computed by the SCALE subroutine they must be supplied by the user.

The calling sequence has 6 arguments:

CALL LINE (XARRAY, YARRAY, NPTS, INC, ±LINTYP, INTEQ)

XARRAY is the name of the array containing the abscissa (X) values and the scaling parameters for the X-array.

YARRAY is the name of the array containing the ordinate (Y) values and the scaling parameters for the Y-array.

NPTS is the number of data points in one of the two arrays previously mentioned. The count does not include the extra two locations for the scaling parameters. The number of points in each array must be the same.

INC is the increment that the LINE subroutine is to use in gathering data from the two arrays, as described previously for the SCALE subroutine.

±LINTYP is a control parameter which describes the type of line to be drawn through the data points. The magnitude of LINTYP determines the frequency of plotted symbols; e.g., if LINTYP = 4, a special symbol (denoted by INTEQ) is plotted at every 4th data point.

If LINTYP is *zero*, no symbols are plotted.

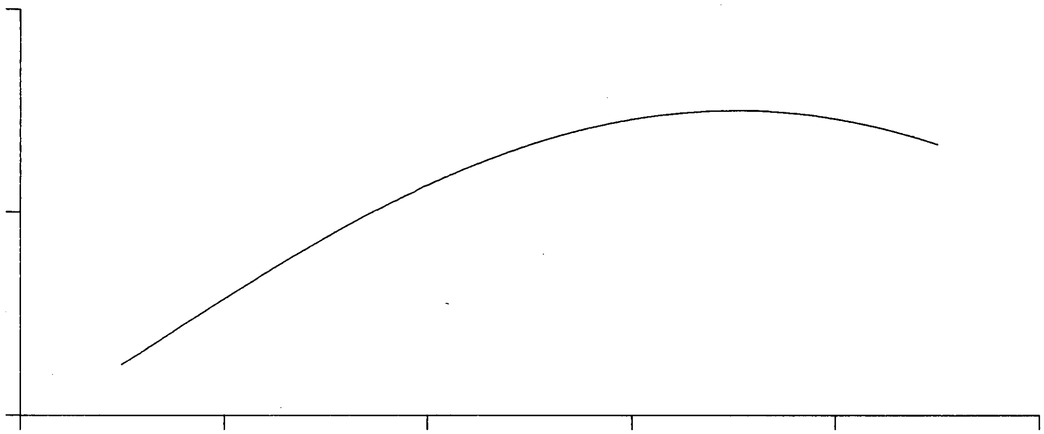
If LINTYP is *positive*, a straight line connects every data point defined in the array. (The pen is up when moving from its current position to the first point.)

If LINTYP is *negative*, no connecting lines are drawn; only the symbols are plotted.

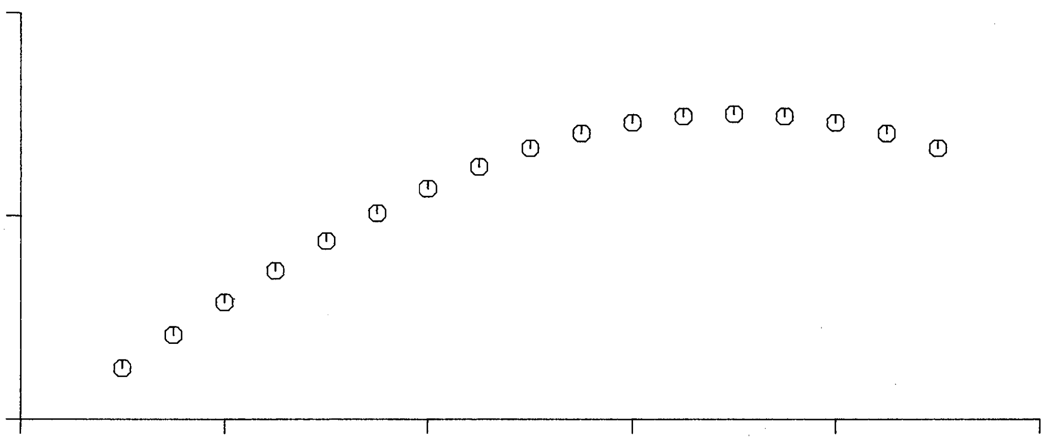
INTEQ is the integer equivalent of the special plotting symbol centered at each data point. This value normally can be 0 through 13, and has meaning only when LINTYP is not zero. Figure 5A lists the symbols that are available. Some of these symbols are: box, octagon, triangle, plus, X, diamond, and asterisk.

Figure 7 illustrates the types of lines drawn by various combinations of LINTYP and INTEQ. The dummy axes are shown for reference only.

A. CALL LINE (XARRAY,YARRAY,33,1,0,0)



B. CALL LINE (XARRAY,YARRAY,33,1,-2,1)



C. CALL LINE (XARRAY,YARRAY,33,1,1,2)

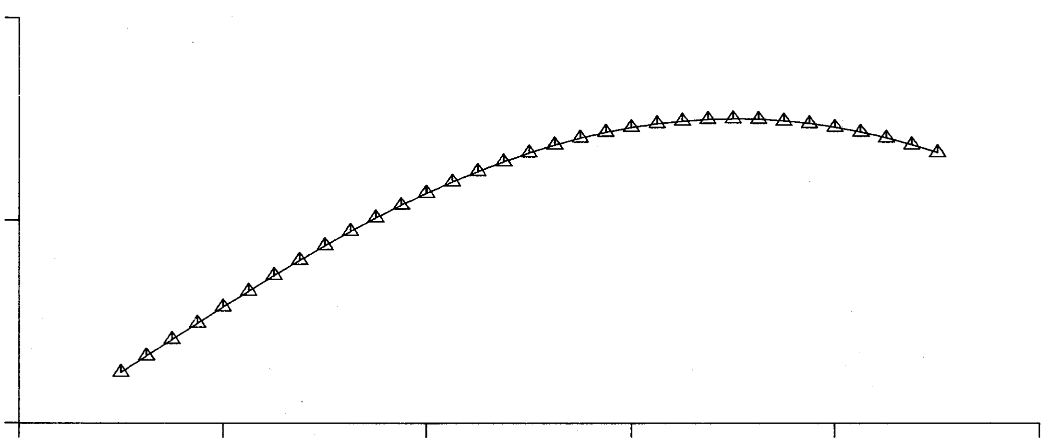


Figure 7

OPERATING CONSIDERATIONS

An understanding of some of the operating characteristics of CalComp plotting systems is desirable when you are planning your application program. The quality of appearance of the finished 'plot' is dependent on the combination of pen, paper, and ink. The various types of these items are described in the Software Reference Manual. The accuracy of the plot can be affected by the initial setting of the pen and the travel of the pen with respect to the limit switches.

LIMIT SWITCHES

The plotting pen carriage is driven by step motors. In order to avoid damage to a step motor when the pen is given commands to move beyond its maximum travel, limit switches are placed at the sides of the plotter. When a limit switch is actuated, the circuits that impulse the step motor to drive in the direction of the switch are disabled. All further commands to move in that direction are ignored until commands of the opposite direction move the pen away from the limit switch. Drum plotters have limit switches at the ends of the Y-axis only; flatbed plotters have limit switches in both the X and Y axes.

The limit switches may be used to initially position the pen at the start of a plotting run (as described below), if extreme accuracy is not required. They should not be used for origin control during a plot, however, since switch bounce can cause inaccuracies of from one to as many as five increments. Also, in some on-line plotter systems, actuating a limit switch can cause a system interrupt, which should be avoided if at all possible.

INITIAL ORIENTATION AND ORIGIN CONTROL

When PLOTS is called, a "logical" origin is defined with X = 0, Y = 0. It is assumed, when actual plotting is begun, that the initial physical position of the pen represents the desired logical origin. If form graph paper is being used, the operator should be instructed where to set the pen for its initial orientation, using the reticle attached to the plotting pen. This is a small magnifying glass with cross-hair lines that intersect at the same Y-axis position as the pen point and at one inch away in the +X direction. After this initial orientation, the program should avoid running the pen into any of the limit switches, as this could cause a shift of origin with respect to the grid lines.

If blank graph paper is being used, the program can control the correspondence between logical origin and pen position by purposely running the pen into a limit switch, thus establishing the initial orientation without an operator's assistance. When PLOTS defines the initial logical origin, the physical pen position need not be known. The following two calls to PLOT can position the pen where desired and defines a new logical origin. These calls are:

```
CALL PLOT (0.,−YMAX,−3)
CALL PLOT (0.,YMARGN,−3)
```

In the first call, YMAX represents the maximum travel permitted on the Y-axis of the plotter. On a Model 565 plotter the width is 11 inches; so YMAX = 11. A call for a move of 11 inches in the −Y direction will move the pen to the limit switch at the −Y side of the plotter. The third parameter value of −3 defines this new position as a new logical origin, where X = 0, Y = 0.

In the second call, YMARGN is the margin wanted between the side of the plotter and the desired origin of the plot. Usually this is *one-half inch* or more to allow for the labeling below the X-axis line. The value of −3 again defines the new position as the logical origin for the rest of the plot.

The program should not attempt to use the pen position at the limit switch as the origin for a plot, even if there are no −Y movements expected during plotting. Due to variations in the sensitivity of the mechanical limit switches, a later move to the Y-origin could possibly be stopped one or two steps shorter or longer than the initial orientation. This would effectively cause a shift in the correspondence between the physical origin and the logical origin.

The program should also avoid utilizing the full travel between opposite limit switches for the same reasons given above. About 0.1 inch of margin at each end should be allowed to avoid any possibility of accidentally running into a limit switch. Thus, 10.8 inches would be the safe Y-axis travel on an 11-inch plotter, and 29.3 inches would be safe on the 29½ inch plotter.

SEARCH RECORDS

Users should avoid excessively long plotting runs without providing some means of interruption and restart. This permits the operator to intervene when the pen, paper, or ink requires attention. A solution provided in off-line plotting systems is the use of Search records and the Search address feature on the CalComp tape unit. The PLOT subroutine produces a Search record when any of the following events occurs:

(1) PLOTS is called (Search address = 001).

(2) PLOT is called with a −IPEN value (Search address = next sequential number).

(3) An end-of-tape condition (Search address = 999 at end of tape and at beginning of next reel).

(4) PLOT is called with IPEN = 999 (Search address = 999).

When all of the plots on a tape file are to be plotted at one time, the tape unit Search address is set to 999 with the thumbswitch, and the unit is put in Multi-plot mode. When a Search record 999 is found at end-of-tape or end-of-file, the tape unit stops.

The Search record 999 at end-of-tape is necessary because the CalComp tape unit is insensitive to tape marks. When the next reel of the file is mounted, the tape unit is manually put in Search mode (no plotting occurs) until the first Search record 999 is found. Then the unit is manually switched back to Multi-plot mode to continue plotting.

If only a selected portion of the plot tape is wanted, it may be located by setting the appropriate Search address in the thumbswitch and running the tape unit in Search mode. When the desired section is found, the unit can be set to Multi-plot mode and another number set in the thumbswitch; or to Single-plot mode, which stops plotting when the next Search record is found (regardless of the Search address).

In on-line systems, no Search records are produced (except for spooling environments). In this case, if long runs are expected, the user should provide some means of plotter interruption and restart with his own program.

SUMMARY OF STANDARD CALLING SEQUENCES

```
CALL PLOT (XPAGE, YPAGE, ±IPEN)
CALL PLOTS (IBUF, NLOC, LDEV)
CALL FACTOR (FACT)
CALL WHERE (RXPAGE, RYPAGE, RFACT)
CALL NEWPEN (INP)
```

XPAGE, YPAGE are the X, Y coordinates of the terminal position of a pen movement, in inches from the current origin.
IPEN specifies the pen up/down status during movement (up=3; down=2), and, if negative, establishes a new origin at the new position.
IBUF names a large output buffer area.
NLOC is the number of locations reserved for IBUF.
LDEV is the logical number of the plot output device.
FACT is a scale factor that determines the enlargement or reduction of the entire plot.
RXPAGE, RYPAGE, RFACT are the locations that will contain the current values of XPAGE, YPAGE, and FACT after WHERE is called.
INP is the number of the selected pen.

```
CALL SYMBOL (XPAGE, YPAGE, HEIGHT, IBCD, ANGLE, +NCHAR)
CALL SYMBOL (XPAGE, YPAGE, HEIGHT, INTEQ, ANGLE, −ICODE)
CALL NUMBER (XPAGE, YPAGE, HEIGHT, FPN, ANGLE, ±NDEC)
```

XPAGE, YPAGE define the relative origin of the character string (usually the lower left corner of the first character position).
HEIGHT is the height (and width), in inches, of a character position.
IBCD is the location of a character string.
ANGLE is the angle at which the character string is to be plotted.
NCHAR is the number of characters in IBCD.
INTEQ is the integer equivalent of a special centered plotting symbol.
ICODE specifies the pen up/down status during movement to the relative origin.
FPN is the location of a real (floating-point) value.
NDEC specifies the number of decimal places to be printed.

```
CALL SCALE (ARRAY, AXLEN, NPTS, ±INC)
CALL AXIS (XPAGE, YPAGE, IBCD, ±NCHAR, AXLEN, ANGLE, FIRSTV, DELTAV)
CALL LINE (XARRAY, YARRAY, NPTS, INC, LINTYP, INTEQ)
```

ARRAY names an array of data values.
AXLEN is the length of the axis line.
NPTS is the number of entries in an array.
INC is the increment between entries in an array..
XPAGE, YPAGE is the relative origin of the axis line.
IBCD is the location of the alphanumeric axis title.
NCHAR is the number of characters in IBCD.
ANGLE is the angle of the axis line.
FIRSTV is the first scale value printed along the axis.
DELTAV is the increment between scale values on the axis.
XARRAY, YARRAY contain the pairs of data values to be plotted.
LINTYP specifies the type of line to be drawn through the data points.
INTEQ is the integer equivalent of a special centered plotting symbol.

CALCOMP

CALIFORNIA COMPUTER PRODUCTS, INC.
305 North Muller Street Anaheim, California 92803
Phone (714) 774-9141 TWX 910-591-1154

1006 10M 668