REPORT R-366 SEPT. 1967

CSL COORDINATED SCIENCE LABORATORY

CCSYSTEM
CALCOMP PLOTTER COMPUTER PROGRAMMING SYSTEM

W. JACK BOUKNIGHT

UNIVERSITY OF ILLINOIS - URBANA, ILLINOIS

This work was supported in whole by the Joint Services Electronics Program (U.S. Army, U.S. Navy, and U.S. Air Force) under Contract DAAB-07-67-C-0199.

CCSYSTEM

Description of the CSL CDC 1604
Calcomp Plotter Operating Package

by

Jack Bouknight

July 1967

Abstract

A complete description is given of a programming package written at CSL for the CDC 1604 computer system allowing usage of the California Computer Products (CALCOMP) digital incremental plotter system for a graphical output medium.

ACKNOWLEDGEMENTS

Thanks are due to Paul Tucker for his efforts in the checkout of all programs in Group 3 and for writing two of those programs, CCGRAPH and CCELIPSE.

Much appreciation is expressed to the operators of the CALCOMP plotter system at the Department of Computer Science for their patient services in running our myriad and confusing results until at last, the pieces fit in place.

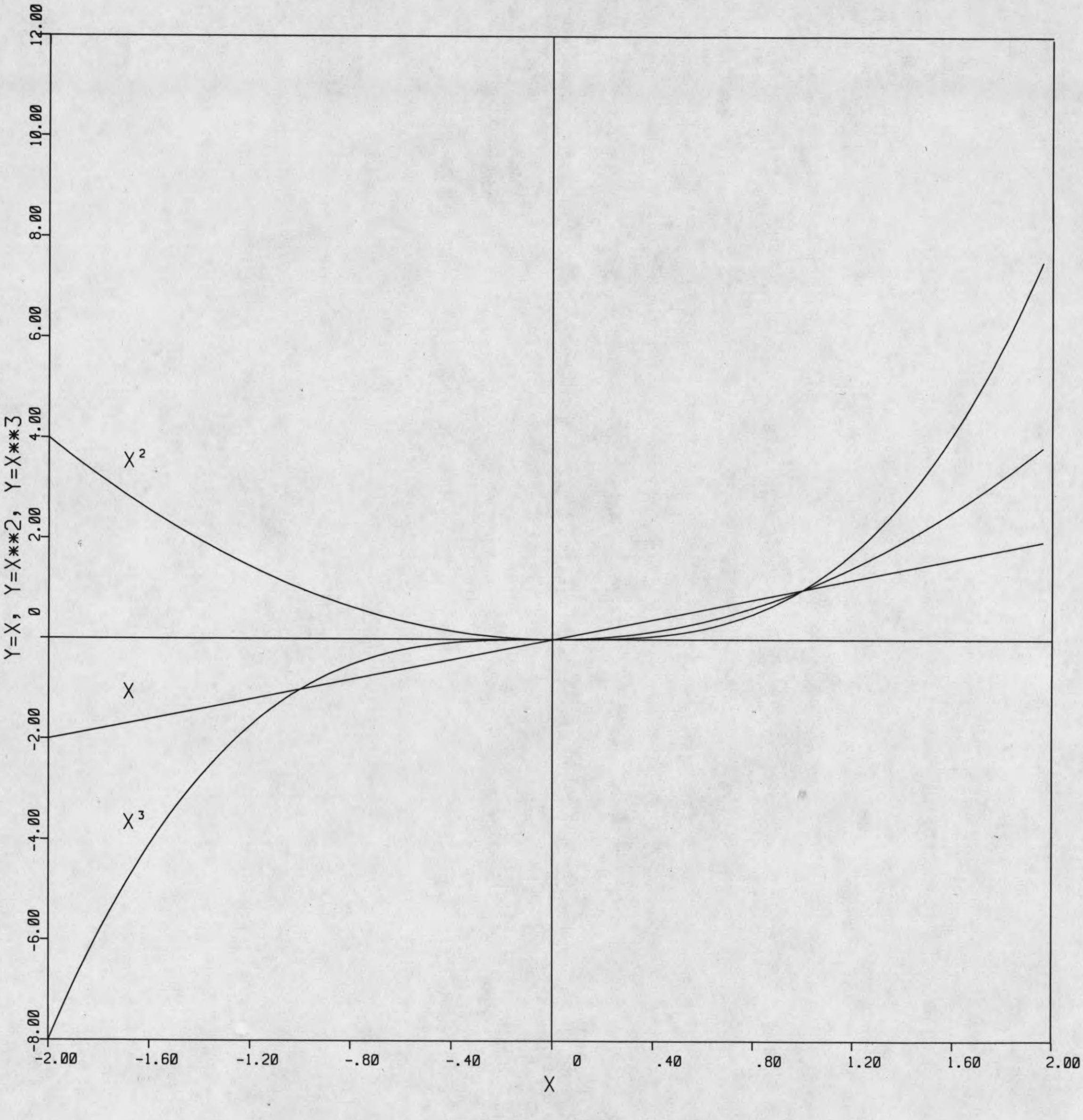
Acknowledgement is made of the several programs written at Calcomp and Oak Ridge (by D. K. Gavin) that served as useful examples for our development.

Thanks also to Joan Curtis for the typing of this manuscript.

Table of Contents

Page

List of Figures


12.00
10.00
8.00
6.00
4.00
2.00
0
-2.00
-4.00
-6.00
-8.00
Y=X, Y=X**2, Y=X**3
X²
X
X³
-2.00
-1.60
-1.20
-.80
-.40
0
.40
.80
1.20
1.60
2.00
X

1.0 Introduction

In the summer of 1965, a Calcomp Plotter System consisting of a 30" Model 564 plotter and tape drive Model 670 was purchased by the Department of Computer Science to provide the facility for large-scale graphical output from computer systems on the campus of the University of Illinois.

To utilize this facility for the benefit of researchers and students of the Coordinated Science Laboratory, a comprehensive and flexible programming package has been developed and implemented in the new CSL 1604 operating system for production of multi-faceted graphical output via an incremental digital plotting device.

Emphasis on design of the package was to reduce the amount of petty detail that the package user need be concerned with. Coupling of the plotter system with operation of the 1604 driven CRT system was also desired.

2.0 Package Description

In describing the overall design of the plotter operating package, we may group the individual routines into three categories: 1) basic plotter control routines, 2) old and new systems compatibility routines, and 3) special "function" routines for driving the basic plotter control routines.

2.1 Group 1: Basic Plotter Control Routines

There are three routines in this group:

1. CCABSPLT

2. CCPLOT

3. CCSYMBOL

CCABSPLT is "the" basic plotter control routine. It produces the actual plotter control magnetic tape. It is driven by CCPLOT to form the "hard core" of the plotter control package. CCSYMBOL generates the necessary calls to CCPLOT to draw designated symbols, usually BCD character strings. These three routines provide all the necessary "basic" functions for use of the plotter system.

2.2 Group 2: Old and New System Compatibility Routines

In this group are three routines:

1. CCPBCDOT

2. CALCOMP

3. SCPTOPLT

CCPBCDOT produces a BCD character string for plotting in exactly the same manner as do the routines of name XXXBCDOT for printing, punching, etc. The character string is controlled by a format string and all forms of data conversion are available.

CALCOMP is a routine available to maintain compatibility with routines written in the old Fortran '60 system at CSL prior to Sept. 1967. The calling sequences to CALCOMP are identical to the old CALCOMP routine in the Fortran '60 library.

A new routine, SCPTOPLT, couples the CSL 1604 CRT display system with the plotter output system and provides the user with the ability of reproducing the scope display on the plotter. This routine will probably be of great use to design oriented applications of the display system.

2.3 Group 3: Special Function Routines

As of July 1967, there were thirteen (13) routines included in this category:

1. CCLINSCL

2. CCLOGSCL

3. CCLINAX
4. CCLOGAX
5. CCLINGRD
6. CCLOGGRD
7. CCLINE
8. CCGRAPH
9. CCIRCLE
10. CCSPIRAL
11. CCPARLEL
12. CCELIPSE
13. CCDASHLN

These represent the first entries in this group which promises to grow and expand to cover new and diverse areas of graphical output requirements.

The first eight routines are basically concerned with the graphing of data. CCLINSCL and CCLOGSCL scale an array of data and provide range data for drawing graph axes. CCLINAX and CCLOGAX draw linear and logarithmic axes for graph plotting. CCLINGRD draws a linear grid pattern. CCLOGGRD draws a logarithmic grid pattern. CCLINE plots an array of data in graphical form. CCGRAPH produces complete graphs.

CCIRCLE draws a circle on the plotter paper. CCSPIRAL draws a spiraling figure. CCPARLEL draws a parallelogram under complete specification by the user. CCELIPSE draws ellipses in the same manner as CCIRCLE draws circles. CCDASHLN plots dashed lines. These are only the beginnings of a potentially very useful library package designed to couple the output of the 1604 computer system to the user in a form more easily interpreted by him.

3.0 Description of the Individual Routines

In the following sections, we will describe each of the routines of the plotter operating package. For reference, Appendix I will contain a list of the calling sequences of each of the routines.

3.1 Group 1: Basic Plotter Control Routines

3.1.1 CCABSPLT

This routine contains the following sections, each denoted by its entry point:

1. CCABSPLT
2. CCBLKADD
3. CCENDPLT

CCABSPLT is entered with the following calling sequence:

```
CALL      CCABSPLT(X,Y,IC)                                     (Fortran)
CALL      CCABSPLT,X,Y,IC                                      (Illar)
```

The necessary plotter commands will be generated and placed in the plotter tape buffer to move the plotter pen from its present position to the new position (X,Y) where X and Y are specified in inches on the plotter paper.

Whether the pen is moved to the new position in the raised or lowered position is determined by the value of IC: if IC = 2, the pen is lowered, and if IC = 3, the pen is raised. If IC = 1, the pen is left in the position found.

A word of caution should be injected at this point. CCABSPLT is essentially a system type computer routine and should be used by itself only when absolutely necessary and then only by users who have a good knowledge of the Calcomp plotter system. The routine CCPLOT should provide all the control necessary for the use of CCABSPLT.

Associated with the generation of blocks of data containing plotter action commands, CCBLKADD generates special data records on the output tape for providing stopping points during plotting for changing paper, color or type of ink pen, etc. The calling sequences are:

```
CALL      CCBLKADD (IFLAG)                                     (Fortran)
CALL      CCBLKADD,IFLAG                                       (Illar)
```

Because a block address writes data on magnetic tape, an option is available to allow the user to write out the unwritten contents of the internal tape buffer before writing the block address record or to just write the block address. If IFLAG is positive, the current buffer will be written out before the block address. If IFLAG is negative, only the block address is output. The last option is primarily intended for the purpose of initializing plotter data tapes with a block address in order that the Calcomp 670 tape drive may find the data for processing.

For signaling the end of the data on the plotter output tape, a universal convention has established that a block address record containing code number 799 be written on the tape. A call to CCENDPLT accomplishes this, after seeing that all remaining data are written on the tape, and then unloads the output tape.

The calling sequences are:

```
CALL      CCENDPLT                                              (Fortran)

CALL      CCENDPLT                                              (Illar)
```

No parameters are needed. The plotter buffer will be emptied and the plotter package will be reset to the same status as if it were just loaded into memory.

For each call to CCENDPLT, the following information about the data contained on the Calcomp output tape will be listed on the console typewriter.

```
Calcomp tape finished.
This plot will run for approx.  7.8 minutes.
Length of plot is  3.71 feet
violation(s)    xmin  xmax  ymin  ymax
```

The time for plotting and amount of paper needed are useful information that should be given to the DCS Calcomp Plotter operator for each tape to be plotted. For the edification of the user, one or more of the violations will be output if they occurred during the computer run.

3.1.2 CCPLOT

The main driving functions of the Calcomp plotter package are embodied in CCPLOT. In addition to driving CCABSPLT for basic plotter pen motion, the following functions are available:

1. Limited movement of the pen within user supplied boundaries.

2. User tracking of the current position of the "user" pen.

3. Ability to automatically scale input parameters in either X or Y directions.

4. Ability to translate (offset) the plotting area in either X or Y directions.

The basic calling sequences are:

```
PLOT,X,Y,IPEN                                                   (Fortran)

CALL      CCPLOT,X,Y,IPEN                                       (Illar)
```

All coordinates of positions to which the "user" pen must move are specified in units of linear length in the X and Y directions. The actual position of the plotter pen may be different from the "user" pen due to boundary violations or from the effects of the scaling and/or offset factors.

X and Y are the coordinates of the position to which the pen must move, subject to offset and scaling factors. Whether the pen is up or down depends on the value of |IPEN|. If |IPEN| = 2, the pen will be moved in the lowered position. If |IPEN| = 3, the pen will move in the raised position. If |IPEN| = 1, the pen remains in the previous position.

If IPEN is positive, a normal exit from CCPLOT will take place when the "user" pen has reached its destination. If IPEN is negative, the tape buffer will be written on the plotter output tape and a block address record written as soon as the pen motion is complete. It is good practice to write block address records after major sections of a plot to enable recovery procedures to be applied should misfortune strike and the plotter system fail to operate properly. WARNING: If a call to CCPLOT with IPEN negative violates a boundary (as explained later), the starting position of the pen in the next drawing will be unpredictable.

CCPLOT establishes what is known as the "user limited plotting area." This is a completely bounded area on the plotter surface outside of which the Calcomp plotter pen is not allowed to travel. To the user, however, the pen appears to continue to plot the required data. When discussing the action of this pseudo pen, it will be referred to as the "user" pen.

To ascertain the present position of the "user" pen, the user calls the routine CCWHERE with the following calling sequences:

```
CALL      CCWHERE(X,Y)                                          (Fortran)

CALL      CCWHERE,X,Y                                           (Illar)
```

The position of the "user" pen is returned as X and Y in floating point units. All offset and scaling effects are removed.

To set up the limits of the "user limited plotting area," the user calls the routine CCLIMITS with the following calling sequence:

```
CALL      CCLIMITS(XMIN,XMAX,YMIN,YMAX)                         (Fortran)

CALL      CCLIMITS,XMIN,XMAX,YMIN,YMAX                          (Illar)
```

XMIN,XMAX,YMIN, and YMAX are the minimum and maximum limits in the X and Y coordinate directions on the actual plotter surface. That is, they are specified in absolute dimensions of floating point inches and are not affected by the offset and scaling factors. Initially, when the Calcomp plotter package is loaded into memory, the following limits are in force:

XMIN = -0.505 inches

XMAX = 10.505 inches

YMIN = -0.505 inches

YMAX = 10.505 inches

To set new offset values, the routine CCOFFSET is called as follows:

```
CALL      CCOFFSET(XOFFSET,YOFFSET)                             (Fortran)

CALL      CCOFFSET,XOFFSET,YOFFSET                              (Illar)
```

XOFFSET,YOFFSET are specified in floating point units of linear length.

To set new scaling factors, the routine CCFACTOR is called as follows:

```
CALL      CCFACTOR(XFACTOR,YFACTOR)                             (Fortran)

CALL      CCFACTOR,XFACTOR,YFACTOR                              (Illar)
```

XFACTOR and YFACTOR are scale factors used to convert pen motion units into plotter pen distance inches. They are expressed in units of inches/unit.

The conversion of X and Y with the offset and scaling factors into floating point inches (absolute plotter scale) is:

$$X_{abs} = (X + XOFFSET) * XFACTOR$$

$$Y_{abs} = (Y + YOFFSET) * YFACTOR$$

Either CCFACTOR or CCOFFSET will move the pen to the corresponding point in the "new" picture.

Initially, when the Calcomp plotter package is loaded into memory, the following values are assigned to the offset and scaling factors:

XOFFSET = 0.0 units

YOFFSET = 0.0 units

XFACTOR = 1.0 inch/unit

YFACTOR = 1.0 inch/unit

A routine named CCFCTRS has been provided to allow the user to obtain the scaling and offset values presently in use. The calling sequence is:

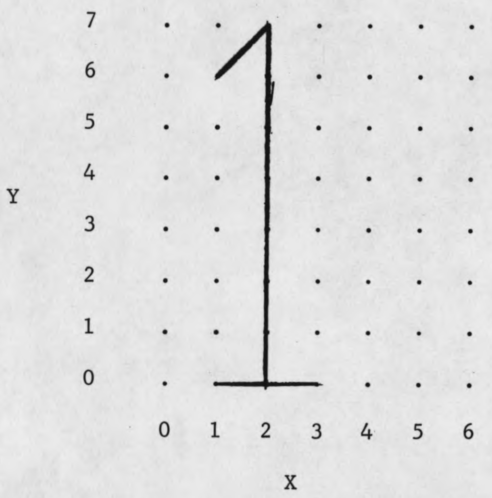
```
CALL        CCFCTRS(XOFFSET,YOFFSET,XFACTOR,YFACTOR)                    (Fortran)

CALL        CCFCTRS,XOFFSET,YOFFSET,XFACTOR,YFACTOR                     (Illar)
```

The parameters are explained in the above sections. This facility allows some routines to use the plotting package in an unbiased manner.

3.1.3 CCSYMBOL

CCSYMBOL provides the "basic" function of character generation for the Calcomp plotter control package. In addition to providing the facility for plotting a standard set of 64 BCD characters corresponding to the character set of the CDC 1612 line printer, there is the ability to plot an additional set of 64 special characters (includes the entire Greek alphabet) and also the ability to allow the user to create and use his own characters.

Table I lists both the table of 64 standard characters and the table of the 64 special characters. All of the symbols are constructed in the following manner. Lines are drawn between points of a 7 × 8 grid as shown below:



For example, the character "1" is constructed by plotting lines between the points

(1,6) (2,7) (2,0) (1,0) (3,0)

A special point (7,0), not on the grid, is used to interrupt the sequence of lines where it is necessary to do so. For example,

(0,2) (4,2) (7,0) (0,4) (4,4)

will plot the character = (equal to).

Associated with each character to be plotted is the character keyword. The keyword is made up of two parts. The first 24 bits contain the number of points to which the pen must be moved to plot the desired character. Included in this count are any (7,0) points in the list.

The list of points is called the coordinate string and starts in memory at some location, the address of which is contained right-justified in the second 24 bits of the keyword. For example, we present the keyword and coordinate string for the character "1":

keyword	ØØØØØØØ5ØØØØØØØ1b
coordinate string beginning in loc. 1	16272Ø1Ø3ØØØØØØØb

Appendix II will list all of the standard and special character coordinate strings, a count of the grid points, and a brief description of each character.

Associated with the 7 × 8 character grid is the term "base point" of the grid. The characters are all positioned by a pair of coordinates in the calling sequence. The point in the grid corresponding to this locating pair of coordinates is called the "base point." When plotting standard characters, the "base point" is (0,0) always. In the special user character modes, the "base point" is specified by the user with special offset factors to be explained later.

The user enters CCSYMBOL by using the following calling sequences:

```
CALL     CCSYMBOL(X,Y,HEIGHT,BCD,ANGLE,N)                          (Fortran)

CALL     CCSYMBOL,X,Y,HEIGHT,BCD,ANGLE,N                           (Illar)
```

The type of character plotting is determined by N. If N is a positive integer, standard characters will be plotted. If N = -1 or -2, a special symbol will be plotted. If N = -3 or -4, a user supplied symbol will be plotted.

In the standard character mode, a string of N characters, starting at location BCD in a packed BCD buffer (left-justified for words of less than eight characters) are plotted. The "base point" of the first character in the string is X,Y. X and Y are given in floating point units of linear length. The "base point" of each succeeding character is point (6,0) of the preceding character. When the complete string is plotted, the pen will be left at the "base point" of the (N + 1)st character.

The height of the character grid is given by HEIGHT and is specified in floating point units of linear length. The string of characters is plotted at an ANGLE with the long edge of the plotter paper. ANGLE is specified in floating point degrees. Both HEIGHT and ANGLE denote the same quantities in all three character modes.

In the special character mode, a single character is plotted with its "base point" at X,Y. If N = -1, the pen is moved to X,Y in the raised position. If N = -2, the pen is moved in the lowered position. When the character is finished, the pen is left at the "base point."

BCD is the address of a location in memory where a full word decimal integer in the range Ø $\leq$ integer $\leq$ 127 is to be found. This integer is used to index Table I to select the special character to be plotted. Note that the first 64 special characters are the same characters in the standard character set. Table I gives the decimal integer index associated with each letter.

In the user supplied character mode, BCD is the address of the location in memory where the character keyword for the user symbol is to be found. Again, the "base point" is set at X,Y. When the character is complete, the pen will be left at the "base point." If N = -3, the pen will be moved to X,Y in the raised position. If N = -4, the pen will be moved in the lowered position.

The adjustment of the "base point" of the character grid is performed by calling CCSOFSET in the following manner:

```
CALL        CCSOFSET(IXOFSET,IYOFSET)                                  (Fortran)

CALL        CCSOFSET,IXOFSET,IYOFSET                                   (Illar)
```

IXOFSET and IYOFSET are full word integer constants measured in grid steps. They are offset factors necessary to move the "base point" of a character grid from some position other than (Ø,Ø) to (Ø,Ø). For example, to move the "base point" for our example character "1" from point (Ø,Ø) to (2,3), the following statements would be executed:

```
CALL        CCSOFSET(-2,-3)                                            (Fortran)

CALL        CCSOFSET,=-2,=-3                                           (Illar)
```

All plotting by CCSYMBOL is performed via call statements to CCPLOT, and therefore all characters are scaled and offset in the normal manner. The user should be aware that if XFACTOR and YFACTOR are not equal when characters are plotted, distorted results will occur.

3.2 Group 2: Old and New System Compatibility Routines

3.2.1 CCPBCDOT

CCPBCDOT is a member of the XXXBCDOT family of system routines. These routines convert binary data into BCD data for output on some output medium. CCPBCDOT is the routine used to drive the Calcomp plotter as an output medium.

The basic calling statement to CCPBCDOT is as follows:

```
CALL        CCPBCDOT,FORMAT,X,Y,HEIGHT,ANGLE                           (Illar)
```

X and Y specifies the starting "base point" for the resulting BCD character string in floating point units of linear length. HEIGHT and ANGLE are also transmitted to CCSYMBOL for plotting of the character string.

FORMAT is the address of the format string to be used to direct the data conversion. Inclusion of this parameter in the standard call to CCPBCDOT, however, will lead to awkward statement writing for the user. For that reason, a new statement in the FORTRAN system has been designed to make the use of CCPBCDOT more natural in the compiler environment. We will discuss the special FORTRAN statement in a moment.

The initial call to CCPBCDOT sets up the general system data conversion routines. To introduce data for conversion, the following calling sequence is used:

```
CALL        WRDBCDOT,DATAWORD                                          (Illar)
```

There must be one call to WRDBCDOT for each field desired to be converted and plotted.

Upon the completion of all desired data conversion, the user calls the cleanup routine ENDBCDOT as follows:

```
CALL      ENDBCDOT                                                          (Illar)
```

There are no parameters. When control is returned to the user's program, the desired character string will have been plotted. The special FORTRAN statement for the use of CCPBCDOT is of the form:

```
PLOTLINE #(X,Y,HEIGHT,ANGLE)/list
```

# is the standard FORTRAN statement label corresponding to the FORMAT statement. X, Y, HEIGHT, and ANGLE were described above. The parentheses "(" and ")" <u>must</u> be present. "list" is the list of data words to be converted and is separated from the rest of the statement by the "/".

To give an example for both ILLAR and FORTRAN users of CCPBCDOT, let us plot the values of A, B, and IC in F12.10, E16.8, and I4 fields, respectively, as follows:

<u>Illar program</u>

```
          Call    CCPBCDOT,FORMAT,X,Y,HEIGHT,ANGLE
          Call    WRDBCDOT,A
          Call    WRDBCDOT,B
          Call    WRDBCDOT,IC
          Call    ENDBCDOT
            .
            .
            .

            .
   FORMAT BCD     3(f12.10,1X,e16.8,1X,I4)
```

<u>Fortran Statements</u>

```
          PLOTLINE 1ØØ(X,Y,HEIGHT,ANGLE)/A,B,IC
   100    FORMAT(F12.10,1X,E16.8,1X,I4)
```

3.2.2 <u>CALCOMP</u>

In the old FORTRAN '60 system, CALCOMP was the complete plotter operating package. The inclusion of CALCOMP in the new system plotter operating package is due to the need for some measure of compatibility between systems to allow old system programs to be run in the new system with a minimum of modification.

With respect to CALCOMP, no modifications are necessary at all. Calling sequences are the same in both systems.

For basic plotting (CCPLOT), use:

```
CALL      CALCOMP(1,X,Y,IPEN)                                             (Fortran)
```

For character generation (CCSYMBOL), use:

```
CALL      CALCOMP (2,X,Y,HEIGHT,BCD,ANGLE,N)                              (Fortran)
```

For single word data conversion (CCPBCDOT), use:

```
CALL      CALCOMP(3,X,Y,HEIGHT,ARG,ANGLE,FORMAT)                          (Fortran)
```

To obtain the current pen position (CCWHERE), use:

```
CALL        CALCOMP(4,X,Y)                                        (Fortran)
```

To terminate the plotter output tape (CCENDPLT), use:

```
CALL        CALCOMP(5)                                            (Fortran)
```

3.2.3 SCPTOPLT

Continuing on the theme of maximum flexibility and compatibility between systems and various pieces of hardware, SCPTOPLT provides the user with the means of plotting a CRT displayed picture on the Calcomp plotter.

The plotter "scope face" is an area of 10 x 10 units of linear length. For smaller or larger drawings, the user simply scales the plotter system by calling CCFACTOR. Facility is provided to map "scope faces" together to build larger pictures in mosaic form.

SCPTOPLT is called in the following manner:

```
CALL        SCPTOPLT(BUFFER,NUM,ENDX,ENDY)                        (Fortran)

CALL        SCPTOPLT,BUFFER,NUM,ENDX,ENDY                         (Illar)
```

BUFFER is the base address of the buffer containing the scope commands for the picture to be plotted. There are NUM words in BUFFER.

When the scope picture has been transcribed to the plotter paper, the plotter pen is moved to position ENDX,ENDY. The coordinates are expressed in floating point units and are affected by the offset and scaling factors of CCPLOT.

When the pen motion is completed, a block address record is written on the Calcomp output tape.

For more information on the CRT display system, see report R-357 by Jack Stifle. For our purposes, we will assume that the user reading this section is familiar with the CRT display system and its modes of operation.

To plot the points for modes 1, 2, 3, and 4, the character, asterisk (54b), is plotted in such a manner as to make the character 0.01 units wide when the actual plot is a 10" x 10" picture on the Calcomp plotter output. Modes 5 and 6 are line drawings with provision made during mode 5 to move the Calcomp pen minimum distance when traveling from line to line.

Mode 7 is character generation and characters are generated via CCSYMBOL with the following heights:

Ø	0.0312 units
1	0.0936 units
2	0.1560 units
3	0.2808 units

Mode Ø is ignored by SCPTOPLT since it is meaningless to Calcomp plotting.

3.3 Group 3: Special Function Routines

3.3.1 CCLINSCL

CCLINSCL is used to scan an array of linear valued elements and extract pertinent data for use by curve and axis plotting routines in the production of graphical output. The routine in use at CSL is an outgrowth of a routine, SCALE, supplied by California Computer Products.

Two equations must be satisfied simultaneously to obtain the correct scaling data.

$$A = R*(10.0**N) \leq \text{minimum value}$$

$$A + (S*DX) \geq \text{maximum value of the array}$$

A is to be the minimum marking of an axis drawn for the array. S is the length of the axis in floating point units of linear length. N is some integer of arbitrary value chosen only to help satisfy the equation by placing A as close to the minimum value of the array as possible.

DX will be the step size of the axis for the array. The linear axis is marked off in 1.0 linear unit intervals. R is a member of the following set of values:

1.0 1.5 2.0 2.5 3.0 4.0 5.0 6.0 8.0

The calling sequence for CCLINSCL is as follows:

```
CALL      CCLINSCL(ARRAY,NUM,NFLAG,S,T)                               (Fortran)

CALL      CCLINSCL,ARRAY,NUM,NFLAG,S,T                                (Illar)
```

ARRAY is the element array to be scaled and contains NUM elements. NFLAG specifies the type of element. If NFLAG is a positive integer, then ARRAY is assumed to be in floating point form. If NFLAG is a negative integer, then ARRAY is assumed to be in fixed point form.

The length of the axis to be drawn for the ARRAY is S units of floating point length. The results of the scaling are returned to the user's program in the two (2) word array, T, as follows:

$$T(1) = A$$

$$T(2) = DX$$

The array T will be referred to later in the writeup as the "scale factor array."

3.3.2 CCLOGSCL

CCLOGSCL performs the scaling operation upon arrays of information to be plotted in logarithmic (base 1Ø) form. The scaling criteria are the power of 1Ø closest to and less than or equal to the minimum value of the array, and the number of cycles necessary to cover the range of the array.

The calling sequence for CCLOGSCL is as follows:

```
CALL      CCLOGSCL(ARRAY,NUM,FLAG,T)                                  (Fortran)

CALL      CCLOGSCL,ARRAY,NUM,NFLAG,T                                  (Illar)
```

ARRAY is the array of elements to be scaled and contains NUM elements. The type of element is required to be floating point in form. NFLAG specifies whether the array is in log form or not. If NFLAG is a positive integer, the ARRAY is assumed to be in "true form" (i.e., log form). If NFLAG is a negative integer, the ARRAY is assumed to be in linear form and must be converted. Note -- no actual conversion is performed. Only the fact is taken into account during scaling.

T is the "scale factor array" and returns the following values to the user's program:

T(1) = power of 1∅ less than minimum of array ($10^{T(1)} \leq$ min. of array)

T(2) = number of covering cycles

3.3.3 CCLINAX

CCLINAX is the routine for plotting a linear form axis complete with tic marks and lable. It may be used in conjunction with the routine CCLINSCL or singly to serve the user's needs. Figure 1 shows a typical axis.

The calling sequence for CCLINAX is as follows:

```
CALL      CCLINAX(X,Y,BCD,NC,SIZE,ANGLE,T)                         (Fortran)

CALL      CCLINAX,X,Y,BCD,NC,SIZE,ANGLE,T                          (Illar)
```

X and Y are the coordinates of the minimum value end of the axis in floating point units of linear length. The length of the axis is SIZE floating point units of linear length (rounded to nearest whole unit). The base line of the axis makes an ANGLE in floating point degrees with the long edge of the plotter paper.

T is the two (2) word "scale factor array" described under the section on CCLINSCL. The axis will be labeled with BCD which contains |NC| characters.

At each end of the axis and at every unit length, a tic mark is placed and lettered with an appropriate value as obtained from T. The sign of NC specifies which side of the axis these tic marks, the lettering and the label will be drawn. If NC is positive, the marking will be placed on the counter-clockwise side of the base line. If NC is negative, the marking will be placed on the clockwise-side of the axis.

All lettering is 0.14 units of linear length in height. The tic marks are also 0.14 units in height. This routine performs the same functions as the routine AXIS in use by members of user groups for Calcomp plotters.

3.3.4 CCLOGAX

CCLOGAX plots a logarithmic (base 1∅) axis complete with major and minor tic marks and labeling. It may be used in conjunction with the routine CCLOGSCL or singly to serve the user's needs. Figure 4 shows a typical axis.

The calling sequence for CCLOGAX is as follows:

```
CALL      CCLOGAX(X,Y,BCD,NC,SIZE,ANGLE,T)                         (Fortran)

CALL      CCLOGAX,X,Y,BCD,NC,SIZE,ANGLE,T                          (Illar)
```

X and Y are the coordinates of the minimum value end of the axis in floating point units of linear length. The length of the axis is SIZE floating point units of linear length. The base line of the axis makes an ANGLE in floating point degrees with the long edge of the plotter paper.

T is the two (2) word "scale factor array" described under the section on CCLOGSCL. The axis will be labeled with BCD which contains |NC| characters.

In accordance with standard log graphical techniques, the tic marks on the log axis are of two heights. For all major cycle marks, the tic marks may be up to 0.1 units of linear length in height. All minor divisions between major cycle marks are made with tic marks of one-half (½) the height of the major tic marks.

The lettering and major tic marks may be up to 0.1 units of linear length in height. The guiding factor in determining this height is the SIZE of the axis and the number of major cycles. The lettering of the label will be 0.15 units or whatever height is necessary to insure that the length of BCD is less than or equal to SIZE.

All major cycles are lettered with 1Ø^K where K is the appropriate power of 1Ø for that major cycle. NC specifies upon which side of the axis the lettering and tic marks will be plotted. If NC is positive, the lettering and tic marks will appear on the counter-clockwise side of the base line. If NC is negative, the lettering and tic marks will be plotted on the clockwise side of the base line.

As an example for drawing x and y axes of a graph, we list below the FORTRAN statements necessary to produce the graphs in Figures 1, 2, 3 and 4.

Figure 1

```
CALL      CCLINAX(0.0,0.0,6HX-AXIS,-6,10.0,0.0,TX)

CALL      CCLINAX(0.0,0.0,6HY-AXIS,6,10.0,90.0,TY)
```

Figure 2

```
CALL      CCLINAX(0.0,0.0,6HX-AXIS,-6,10.0,0.0,TX)

CALL      CCLOGAX(0.0,0.0,6HY-AXIS,6,10.0,90.0,TB)
```

Figure 3

```
CALL      CCLOGAX(0.0,0.0,6HX-AXIS,-6,10.0,0.0,TA)

CALL      CCLINAX(0.0,0.0,6HY-AXIS,6,10.0,90.0,TY)
```

Figure 4

```
CALL      CCLOGAX(0.0,0.0,6HX-AXIS,-6,10.0,0.0,TA)

CALL      CCLOGAX(0.0,0.0,6HY-AXIS,6,10.0,90.0,TB)
```

3.3.5 CCLINGRD

CCLINGRD draws a one unit spacing grid perpendicular to some chosen axis base line. Each of the grids in Figures 1, 2, 3 and 4 were drawn by either CCLINGRD or CCLOGGRD.

The calling sequence for CCLINGRD is as follows:

```
CALL      CCLINGRD(X,Y,XSIZE,YSIZE,ANGLE)                              (Fortran)

CALL      CCLINGRD,X,Y,XSIZE,YSIZE,ANGLE                               (Illar)
```

X and Y are the coordinates of the minimum value end of the axis against which the grid will be plotted. They are expressed in floating point units of linear length. The base line of the adjoining axis is plotted at an angle in floating point degrees with the long edge of the plotter paper. The grid will be plotted at an angle of ± 90.0 degrees with the base line of the axis.

|XSIZE| is the length of the axis against which the grid will be drawn. YSIZE is the length of the grid in the direction perpendicular to the adjoining axis. Both distances are expressed in floating point units of linear length.

The sign of XSIZE specifies on which side of the axis the grid will be drawn. If XSIZE is positive, the grid will be placed on the clockwise side of the axis. If XSIZE is negative, the grid will be placed on the counter-clockwise side of the axis.

In addition to calling CCLINGRD and/or CCLOGGRD, the user must provide his own calls to CCPLOT to draw the grid "cap" lines. In Figures 1, 2, 3 and 4, these "cap" lines are the right-hand and top borders of each graph.

3.3.6 CCLOGGRD

CCLOGGRD draws a log scale grid with grid lines placed at each major cycle mark. Figures 1, 2, 3 and 4 show examples of the use of CCLOGGRD and CCLINGRD.

The calling sequence for CCLOGGRD is as follows:

```
CALL      CCLOGGRD(X,Y,XSIZE,YSIZE,ANGLE,T)                            (Fortran)

CALL      CCLOGGRD,X,Y,XSIZE,YSIZE,ANGLE,T                             (Illar)
```

X and Y are the coordinates of the minimum value end of the axis against which the grid will be plotted. They are expressed in floating point units of linear length. The base line of the adjoining axis plotted at an ANGLE in floating point degrees with the long edge of the plotter paper. The grid will be plotted at an angle of ± 90.0 degrees with the base line of the adjoining axis.

|XSIZE| is the length of the axis against which the grid will be drawn. YSIZE is the length of the grid in the direction perpendicular to the adjoining axis. Both distances are expressed in floating point units of linear length.

The sign of XSIZE specifies on which side of the axis the grid will be drawn. If XSIZE is positive, the grid will be placed on the clockwise side of the axis. If XSIZE is negative, the grid will be placed on the counter-clockwise side of the axis.

The spacing of the grid lines is based on the number of major cycles on the axis adjoining the grid. This information is obtained via the "scale factor array" T. A description of this array may be found under the section describing the routine CCLOGSCL.

As an example for drawing linear and logarithmic grids, we list below the FORTRAN statements necessary to produce the grids in Figures 1, 2, 3 and 4.

Figure 1

```
CALL      CCLINGRD(0.0,0.0,-10.0,10.0,0.0)

CALL      CCLINGRD(0.0,0.0,10.0,10.0,90.0)
```

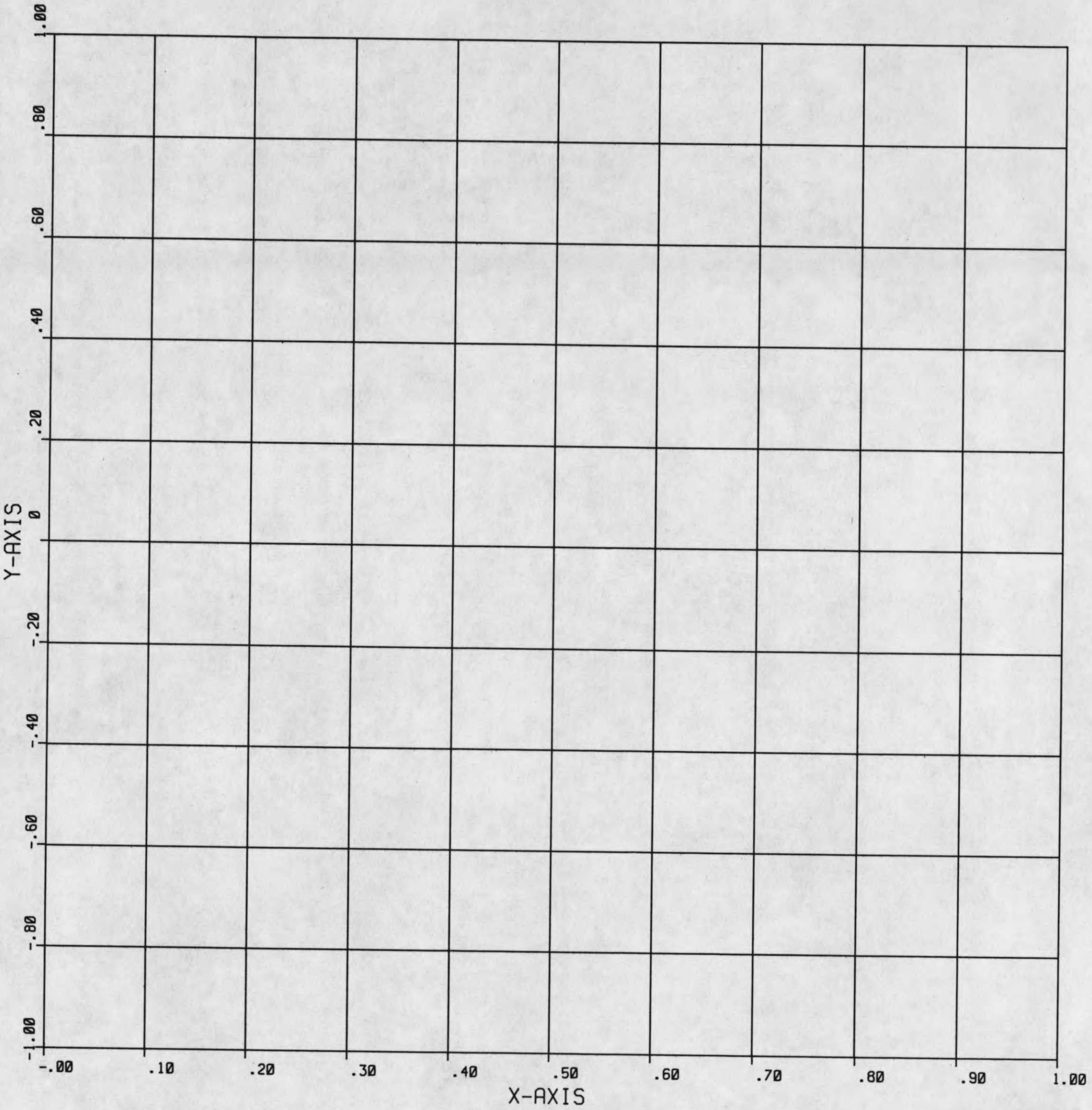


Figure 1 - Linear vs Linear Graph

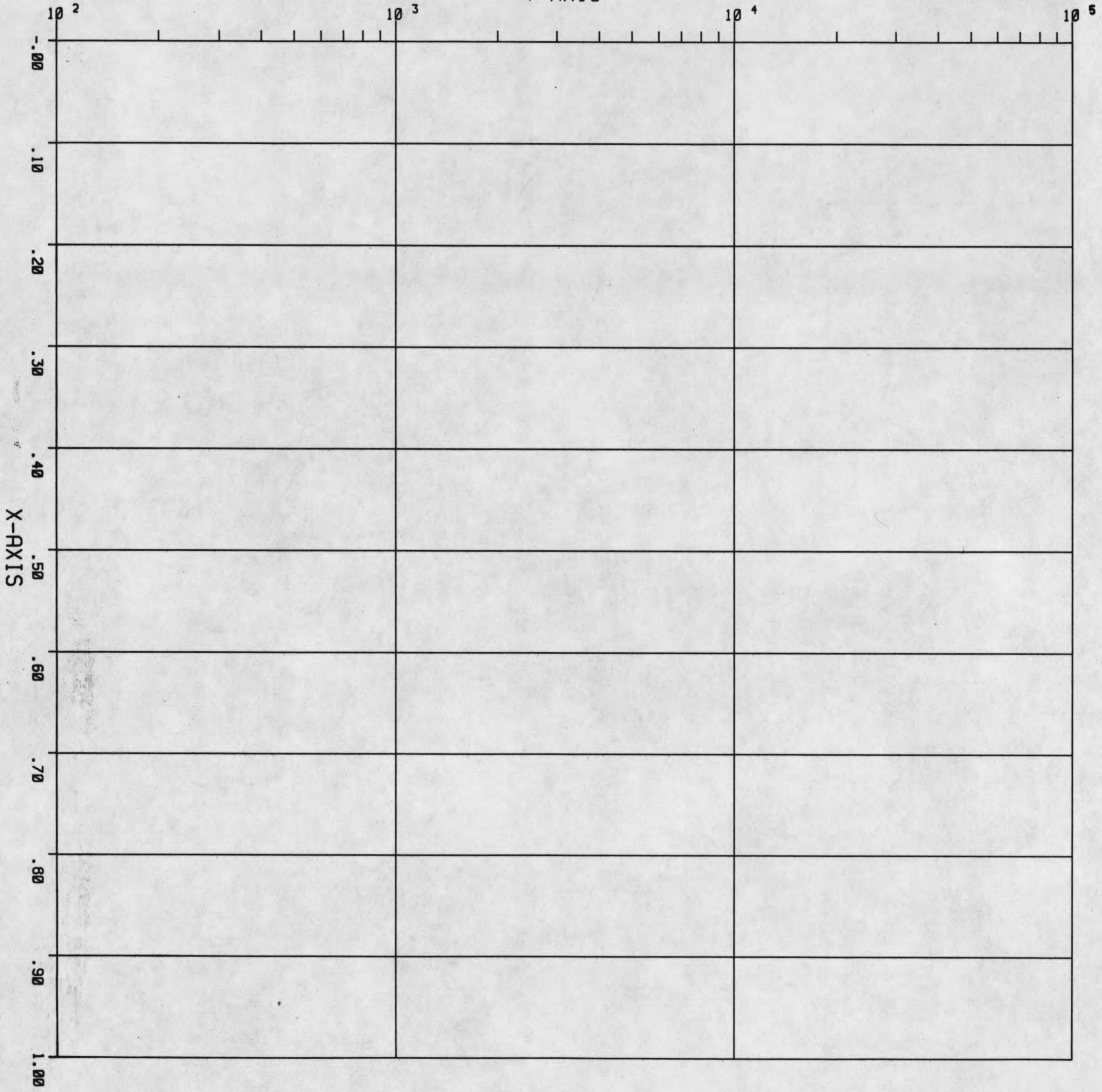


Figure 2 - Linear vs Log Graph

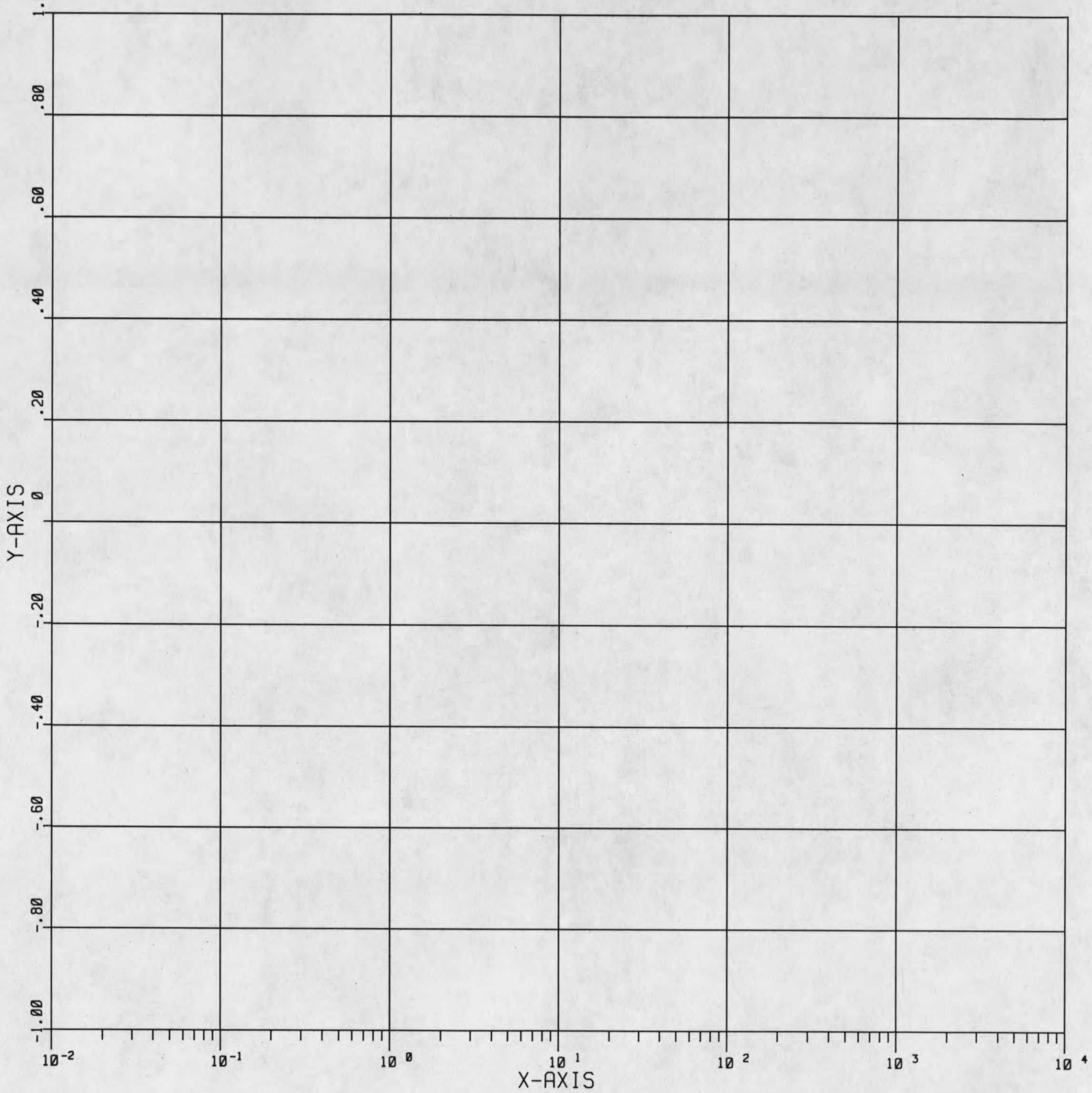


Figure 3 - Log vs Linear Graph

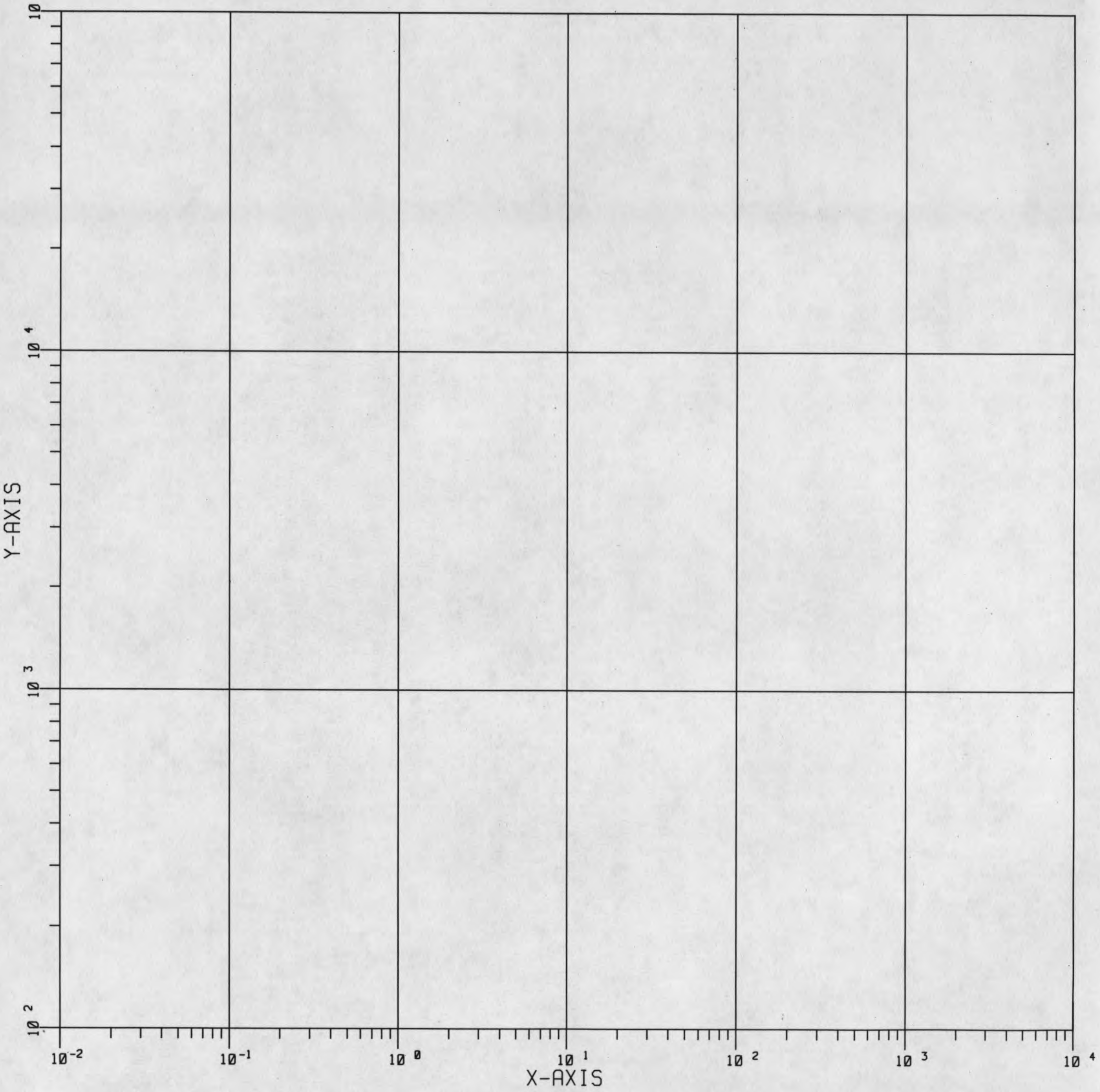


Figure 4 - Log vs Log Graph

Figure 2

```
CALL        CCLINGRD(0.0,0.0,-10.0,10.0,0.0)

CALL        CCLOGGRD(0.0,0.0,10.0,10.0,90.0,TB)
```

Figure 3

```
CALL        CCLOGGRD(0.0,0.0,-10.0,10.0,0.0,TA)

CALL        CCLINGRD(0.0,0.0,10.0,10.0,90.0)
```

Figure 4

```
CALL        CCLOGGRD(0.0,0.0,-10.0,10.0,0.0,TA)

CALL        CCLOGGRD(0.0,0.0,10.0,10.0,90.0,TB)
```

The values of the "scale factor arrays" are as follows:

TX(1) = 0.0	TY(1) = -1.0
TX(2) = 0.1	TY(2) = 0.2
TA(1) = -2.0	TB(1) = 2.0
TA(2) = 6.0	TB(2) = 3.0

3.3.7 CCLINE

Completing the family of general purpose graphical plotting routines is CCLINE. CCLINE produces complete curves plotted from an X and Y array in many formats.

The calling sequence for CCLINE is as follows:

```
CALL        CCLINE(X,Y,NC,NFLAG,TX,TY,XSIZE,YSIZE)                 (Fortran)

CALL        CCLINE,X,Y,NC,NFLAG,TX,TY,XSIZE,YSIZE                  (Illar)
```

X and Y are the arrays of elements governing the construction of the curve. The format of the elements may be floating point or fixed point. If floating point, they may be logarithmic or linear.

NC is a four (4) word array describing the size of the arrays and the manner in which the graph curve is drawn. NC(1) is the number of elements in X. NC(2) is the number of elements in Y. For plotting, elements in X and Y are matched on a 1-1 basis. If one array is shorter than the other, it is rescanned as many times as is necessary to match all the elements in the other array. Thus, for example, an X array of 10 elements will be rescanned 4 times to match a Y array containing 40 elements. Each time a rescan is initiated for one of the arrays, the pen will be raised during travel to the new point. Thus, four curves would be plotted in our above example.

The type of curve drawn may be a continuous line, a special symbol drawn at every K^{th} position, or a combination of both. The selection of the type is specified by the sign of NC(3). If NC(3) is negative, symbols only will be plotted. If NC(3) is positive and non-zero, both the curve and the symbols will be drawn. For the case of NC(3) = ± Ø, only the curve will be drawn. When the curve is drawn, all of the element pairs taken from X and Y are used to provide the smoothest curve possible.

The value of NC(3) determines the spacing of the special symbols where called for. If NC(3) = ± k, then at every k^{th} element pair taken from X and Y, a special symbol will be plotted. The value of NC(4), lying in the range of 0 to 31, selects one of a selected list of symbols. Appendix III lists the allowable symbols. Figure 5b shows a set of curves displaying all three combinations of curves and symbols. Figure 5a shows the FORTRAN program used to generate Figure 5b.

NFLAG is a six (6) word array used to describe the format of the graph and of the data. For simplicity, we will simply list the parameters and their description in the following table:

NFLAG(1)	+	linear axis for X
	-	log axis for X
NFLAG(2)	+	linear axis for Y
	-	log axis for Y
NFLAG(3)	+	X elements in floating point
	-	X elements in fixed point
NFLAG(4)	+	Y elements in floating point
		Y elements in fixed point
NFLAG(5)	+	X elements in true form
	-	X elements must be converted to log form
NFLAG(6)	+	Y elements in true form
	-	Y elements must be converted to log form

Note that NFLAG(5) and NFLAG(6) apply only when X and/or Y are in floating point format.

The remaining parameters relate the scaling of X and Y to the curve drawing section. TX is the two (2) word "scale factor array" usually obtained from CCLINSCL or CCLOGSCL for the array X. Likewise, TY is the "scale factor array" for Y. XSIZE and YSIZE are the lengths of the X-axis and Y-axis, respectively. They should be the same as the values supplied to the scaling routines.

3.3.8 CCGRAPH

The ultimate facility for graph production using the routines of the Calcomp operating system is provided by CCGRAPH. CCGRAPH produces a single graph or curve overlays with complete freedom of design by the user. CCGRAPH is basically a driver program and uses CCLINSCL, CCLOGSCL, CCLINAX, CCLOGAX, CCLINGRD, CCLOGGRD, and CCLINE to produce the required graph(s).

The calling sequence for CCGRAPH is:

```
CALL     CCGRAPH(X,Y,NC,NFLAG,XSIZE,YSIZE,BCDX,BCDY,BCDQ,N)          (Fortran)

CALL     CCGRAPH,X,Y,NC,NFLAG,XSIZE,YSIZE,BCDX,BCDY,BCDQ,N           (Illar)
```

X and Y are the coordinate arrays. Each may be fixed or floating point but not mixed. If they are floating point, they may be linear or logarithmic.

NC is a six (6) word array describing the size of the arrays, the design of the curves, and whether or not grids will be drawn. NC(1)-NC(4) are described under the section

```
CSL FORTRAN OF NOV. 1966, DATE 5/26/67
 --FORTRAN
        PROGRAM REPORT1
        DIMENSION XRAY(600),YRAY(600),NC(4),NFLAG(6),TX(2),TY(2)
        PI=3.1415
        Q=-2.*PI
        DO 1 L=1,600
        XRAY(L)=EXPF(-Q/12.)*SINF(Q)
        YRAY(L)=Q
 1      Q=Q+12.*PI/600.
        NC(1)=600
        NC(2)=600
        NC(3)=0
        NC(4)=0
        NFLAG(1)=1
        NFLAG(2)=1
        NFLAG(3)=1
        NFLAG(4)=1
        NFLAG(5)=1
        NFLAG(6)=1
        CALL CCLIMITS (-.505,10.505,-.505,10.505)
        CALL CCLINSCL (XRAY,600,1,10.,TY)
        CALL CCLINSCL(YRAY,600,1,10.,TX)
        CALL CCLINE (YRAY,XRAY,NC,NFLAG,TX,TY,10.,10.)
        DO 2K=1,600
 2      XRAY(K)=EXPF(-YRAY(K)/12.)
        NC(3)=-5
        NC(4)=2
        CALL CCLINE (YRAY,XRAY,NC,NFLAG,TX,TY,10.,10.)
        DO 3 K=1,600
 3      XRAY(K)=SINF(YRAY(K))
        NC(3)=5
        NC(4)=0
        CALL CCLINE (YRAY,XRAY,NC,NFLAG,TX,TY,10.,10.)
        CALL CCENDPLT
        END
```

Figure 5a. Fortran Program for Figure 5b

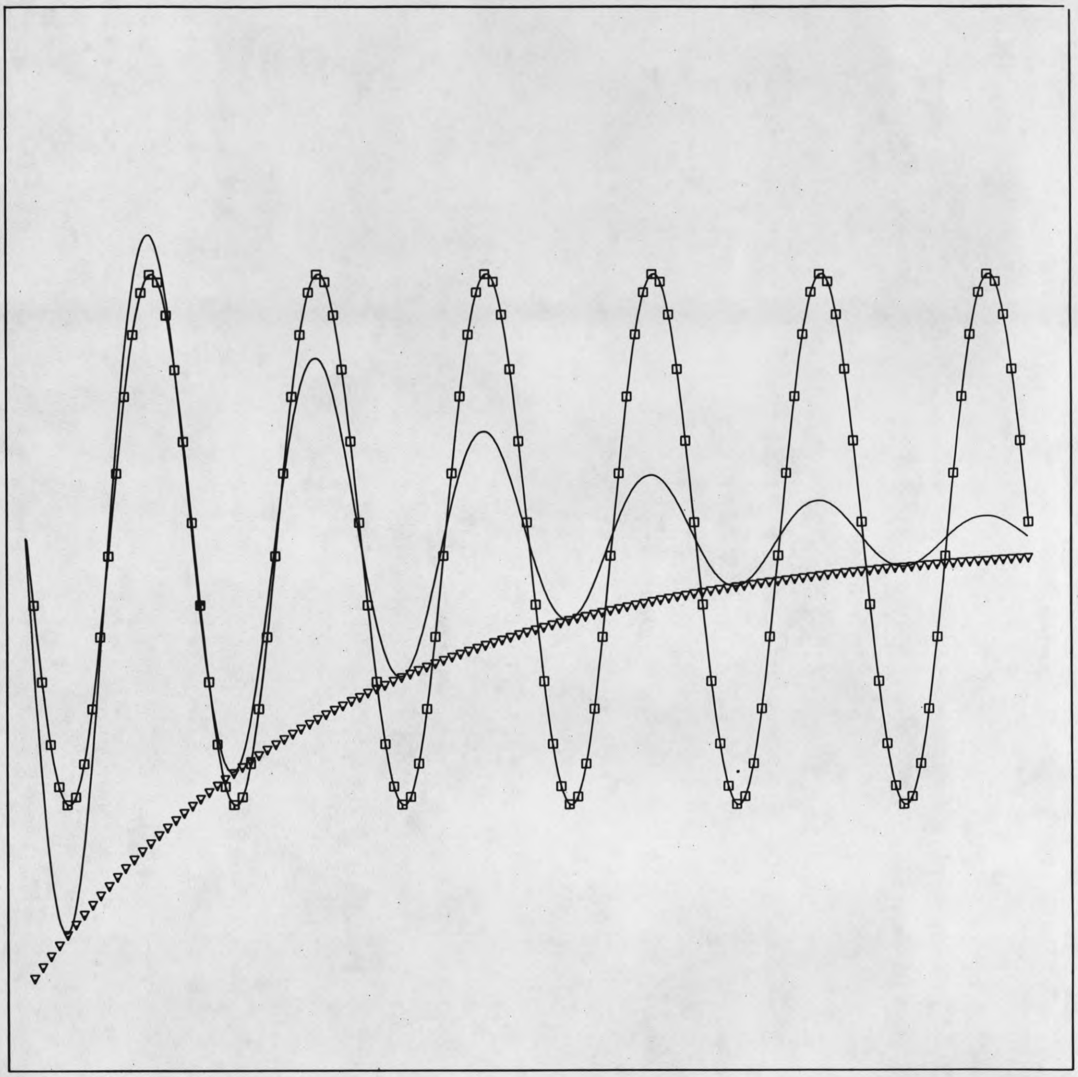


Figure 5b - Three Modes of CCLINE Curves

on CCLINE. NC(5) and NC(6) are described as follows:

```
NC(5)   +  no x axis grid drawn
        -  x axis grid drawn
NC(6)   +  no y axis grid drawn
        -  y axis grid drawn
```

NFLAG is a seven (7) word array used to describe the format of the graph and the coordinate arrays. NFLAG(1) - NFLAG(6) are described in the section on CCLINE. NFLAG(7) is an additional word used to select production of a new graph or to overlay data on a preceding one. Entry to CCGRAPH with NFLAG(7) = + will result in the scaling of X and Y, drawing of axes and/or grids, and the curve described by X and Y. Subsequent entries to CCGRAPH with NFLAG(7) = - will overlay new curves using the same axes, grids, and scale factors. A call to CCGRAPH with NFLAG(7) = - without a previous call with NFLAG(7) = + will result in an error message typeout on the console typewriter and the offending call will be abandoned. Each call to CCGRAPH exits with NFLAG(7) = - to facilitate loops involving overlay calls to CCGRAPH.

BCDX, BCDY and BCDQ are the labels to be used on the x-axis, y-axis, and over the completed graph. Each is a BCD character string usually specified as a Hollerith array. BCDX and BCDY are placed on the axes when they are drawn. BCDQ is placed either at the top of the graph when the x-axis is at the bottom, or at the bottom when the x-axis is at the top. The letters of BCDX and BCDY will be 0.15 units high and the letters of BCDQ will be 0.25 units high.

N is a four (4) word array containing counts of characters in the labels and controlling axis drawing and date placement. If N(1), N(2) and N(3) are positive integers, they are the numbers of characters in BCDX, BCDY and BCDQ, respectively. If any one of N(1), N(2) or N(3) is negative or zero, the associated axis and/or label will not be drawn. Thus, the user may leave off one or both axes which will be replaced with a simple straight line border.

N(4) controls the placement of the current date on the completed graph. N(4) = + will cause the date to be omitted. N(4) = - will place the date below the x-axis base line if the x grid is drawn and above the x-axis base line if no x grid is drawn.

XSIZE and YSIZE are the dimensions of the base lines of the axes. Lettering, tic marks, etc. protrude approximately 0.6 units in all four directions. CCGRAPH results are subject to scaling and offset factors in CCFACTOR and CCOFFSET. CCGRAPH does not modify the limits of the "user limited plotter area." Thus, the user is responsible for control of the environment in which CCGRAPH must operate. The user also is responsible for separation of sequential graphs on the plotter paper. This may easily be accomplished by calling CCPLOT as follows:

```
PLOT,XSIZE+1.5,0.0,-3                                          (Fortran)

CALL      CCPLOT,(loc.of.XSIZE+1.5),=0.0,=-3                   (Illar)
```

<u>Warning</u>: Make sure the call to CCPLOT does not cause a boundary violation.

In Figure 6b, the complete graph for the same data used in Figure 5b is shown. The FORTRAN program used to generate the graph is shown in Figure 6a.

3.3.9 CCIRCLE

Beginning with this description of a circle drawing routine, we will be describing routines which provide drafting functions for the calcomp user. CCIRCLE may be used to draw all or part of a circle.

The calling sequence for CCIRCLE is as follows:

```
CSL FORTRAN OF NOV. 1966, DATE 5/26/67
 --FORTRAN
         PROGRAM REPORT
         DIMENSION XRAY(600),YRAY(600),NC(6),NFLAG(7),N(4)
         PI=3.1415
         Q=-2.*PI
         DO 1 L=1,600
         XRAY(L)=EXPF(-Q/12.)*SINF(Q)
         YRAY(L)=Q
 1       Q=Q+12.*PI/600.
         CALL CCFACTOR (2.,2.)
         NC(1)=600
         NC(2)=600
         NC(4)=0
         NC(5)=-1
         NC(6)=-1
         NC(3)=0
         NFLAG(1)=1
         NFLAG(2)=1
         NFLAG(3)=1
         NFLAG(4)=1
         NFLAG(5)=1
         NFLAG(6)=1
         NFLAG(7)=1
         N(1)=1
         N(2)=24
         N(3)=30
         N(4)=-1
         CALL CCLIMITS (-1.,21.,-1.,21.)
         CALL CCGRAPH (YRAY,XRAY,NC,NFLAG,10.,10.,
        11H ,24HF(T) = E**-T/12.*SINE(T)
        230HEXPONENTIALLY DAMPED SINE WAVE,N)
         DO 2 K=1,600
         XRAY-(K)=EXPF(-XRAY(K)/12.)
 2       CONTINUE
         CALL CCSYMBOL (5.-6.*.15-.3/.7*.15,-.5,.15,3H-2*,0.,3)
         CALL CCSYMBOL (5.1-4.*.6/.7*.15-.3/.7*.15,-.45,.15,117B,0.,-1)
         CALL CCSYMBOL (5.-2.*6./7.*.15-3./7.*.15,-.45,.15,15B,0.,-1)
         CALL CCSYMBOL (5.-3/.7*.15,-.5,.15,1HT,0.,1)
         CALL CCSYMBOL (5.+.6/.7*.15+.3/.7*.15,-.45,.15,15B,0.,-1)
         CALL CCSYMBOL (5.+3.*6./7.*.15+.3/.7*.15,-.5,.15,3H10*,0.,3)
         CALL CCSYMBOL (5.1+6.*6./7.*.15+3/.7*.15,-.45,.15,117B,0.,-1)
         NC(3)=-5
         NC(4)=2
         CALL CCGRAPH (YRAY,XRAY,NC,NFLAG)
         DO 3 K=1,600
 3       XRAY(K)=SINF(YRAY(K))
         NC(3)=5
         NC(4)=0
         CALL CCGRAPH (XRAY,XRAY,NC,NFLAG)
         CALL CCENDPLT
         END
--END
```

Figure 6a. Fortran Program for Figure 6b

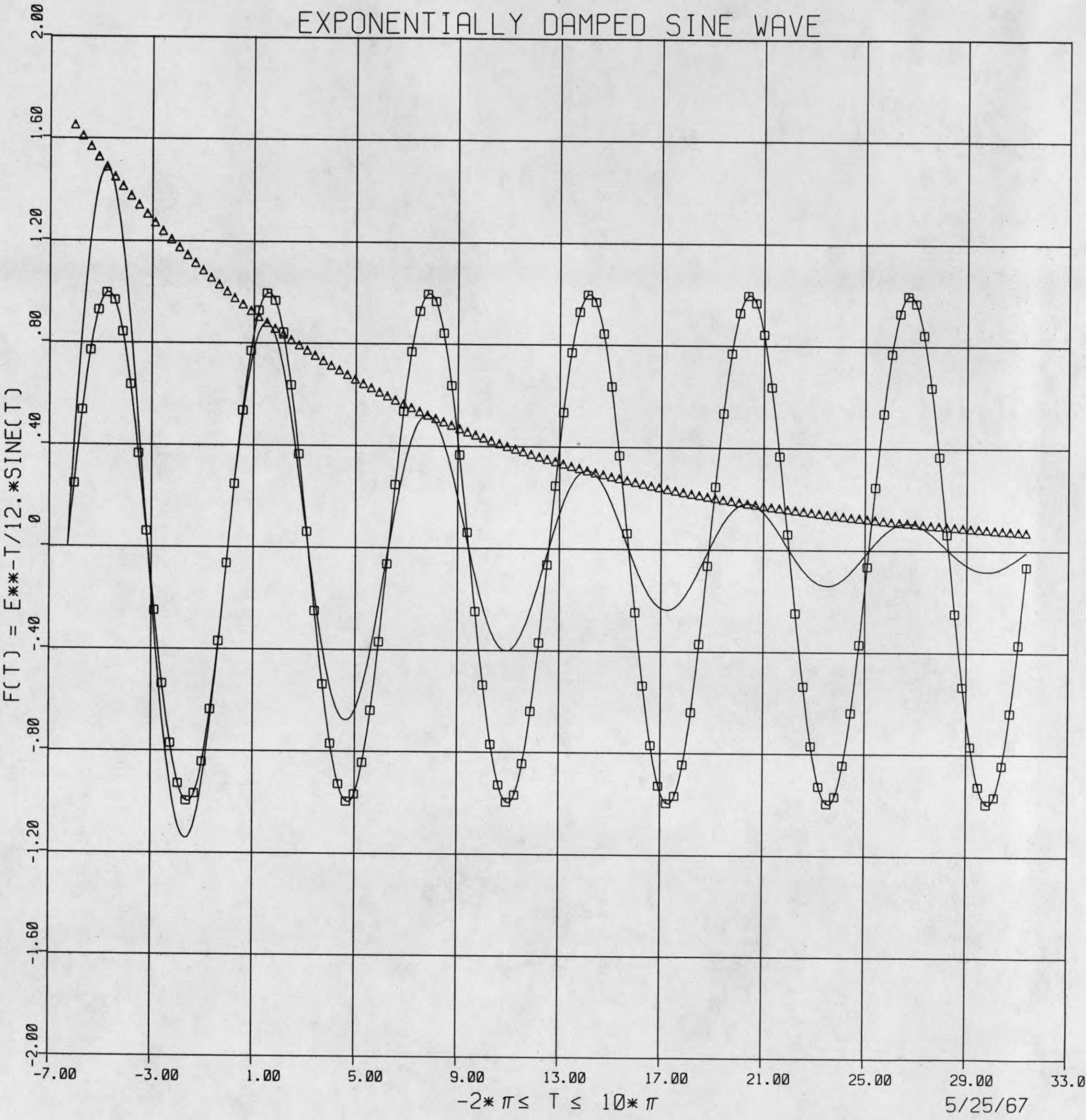


Figure 6b - Example of CCGRAPH

```
CALL      CCIRCLE(X,Y,R,SANGLE,EANGLE)                    (Fortran)

CALL      CCIRCLE,X,Y,R,SANGLE,EANGLE                     (Illar)
```

X and Y are the coordinates of the center of the circle expressed in floating point units of linear length. R is the radius of the circle in floating point units of linear length.

SANGLE and EANGLE are the angles of the starting and ending points of the arc or circle, respectively. They are expressed in floating point degrees.

The circle or arc will be drawn using an increment of one (1) degree for radii of less than 1.0 units, an increment of 0.5 degree for radii of 1.0 to 5.0 units, and an increment of 0.1 degree for all radii over 5.0 units in length. Figure 7 shows the terminology used.

A special entry, CCENDCR, allows the user to obtain the coordinates of the ends of the arc drawn. The calling sequence is as follows:

```
CALL      CCENDCR(POINTS)                                 (Fortran)

CALL      CCENDCR,POINTS                                  (Illar)
```

POINTS is a four (4) word array in which will be returned the coordinates in the following order:

Loc.	Coordinate
POINT(Ø)	starting point x value
POINT(1)	starting point y value
POINT(2)	ending point x value
POINT(3)	ending point y value

3.3.10 CCSPIRAL

CCSPIRAL draws partial or complete spiral figures under complete specification of the user.

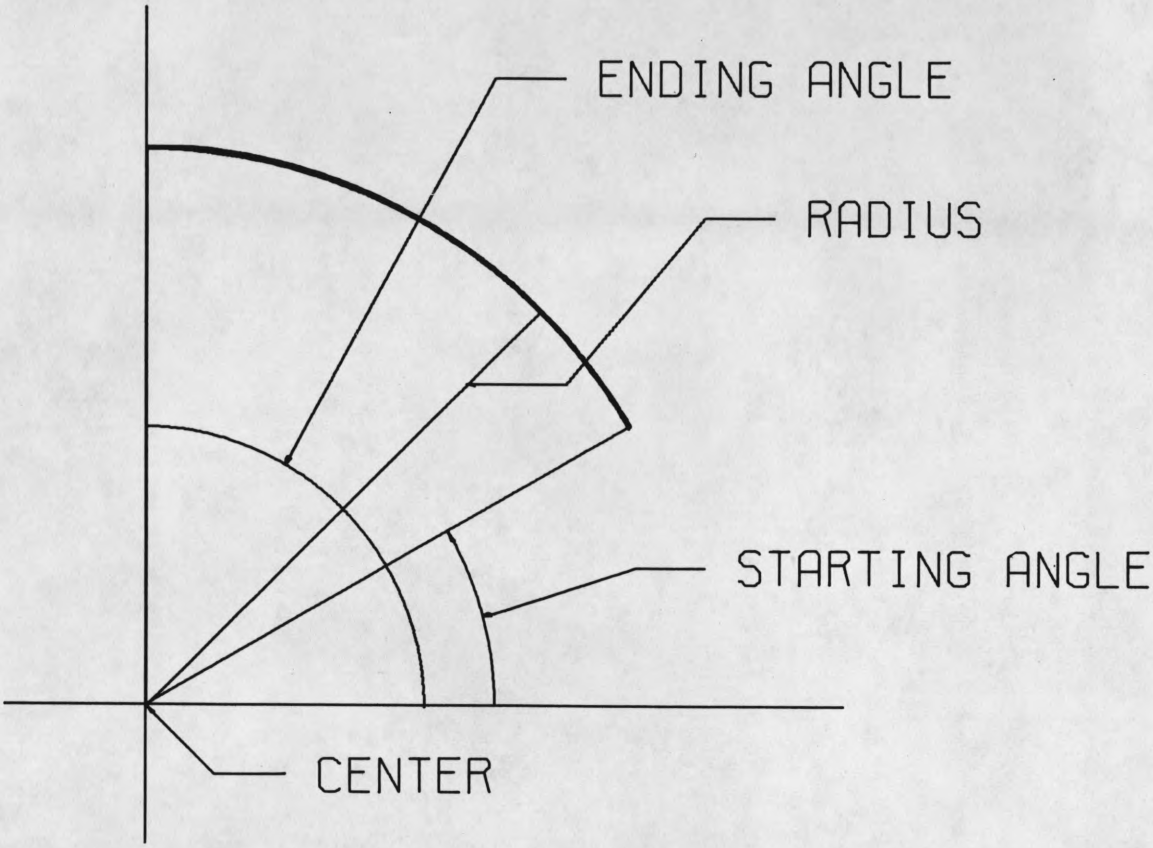
The calling sequence for CCSPIRAL is as follows:

```
CALL      CCSPIRAL(X,Y,RØ,R1,SANGLE,EANGLE)               (Fortran)

CALL      CCSPIRAL,X,Y,RØ,R1,SANGLE,EANGLE                (Illar)
```

X and Y are the coordinates of the center of the spiral expressed in floating point units of linear length. RØ is the starting radius and R1 is the ending radius. Both are expressed in floating points units of linear length.

SANGLE and EANGLE are the starting and ending angles respectively. Both are expressed in floating point degrees. No restrictions are placed on the size or order of either the radii or the angles. Note that making the radii equal causes CCSPIRAL to duplicate the efforts of CCIRCLE.

In determining the angle increment, the larger of the two radii, RØ or R1, is used as a criterion. Increments used are 1.0 degrees for radii under 1.0 units, 0.5 degrees for radii between 1.0 and 5.0 units, and 0.1 degree for all radii over 5.0 units of linear length. Figure 8 illustrates the terminology used.



CALIFORNIA COMPUTER PRODUCTS, INC.   ANAHEIM, CALIFORNIA   CHART NO. 400

Figure 7 - Nomenclature for CCIRCLE

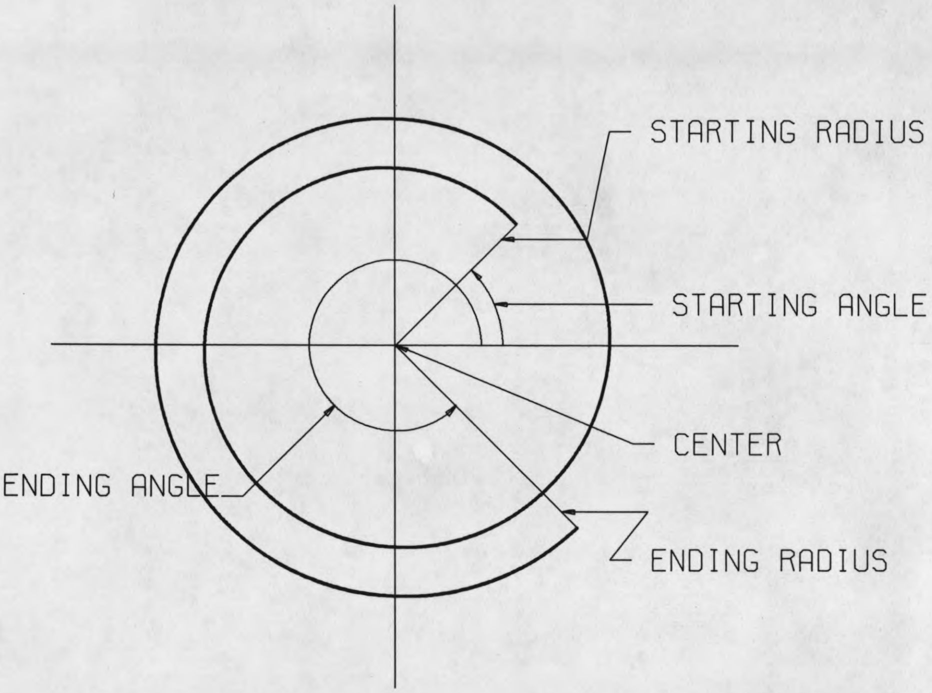


Figure 8 - Nomenclature for CCSPIRAL

A special entry point, CCENDSP, allows the user to obtain the coordinates of the ends of the spiral drawn. The calling sequence and argument are the same as were described for CCIRCLE.

3.3.11 CCPARLEL

CCPARLEL draws parallelograms under complete specifications by the user.

The calling sequence for CCPARLEL is as follows:

```
CALL      CCPARLEL(X,Y,A,B,THETA,PHI)                              (Fortran)

CALL      CCPARLEL,X,Y,A,B,THETA,PHI                               (Illar)
```

X and Y are the coordinates of the base point of the parallelogram. Both are expressed in floating point units of linear length. A and B are the lengths of the major and minor sides, respectively, of the parallelogram. Both are expressed in floating point units of linear length.

The orientation of the entire parallelogram is specified by THETA, the angle between the major side and the long edge of the plotters paper. The angle between the major and minor sides at the base point is specified by PHI. Both angles are expressed in floating point degrees. Figure 9 illustrates the terminology used.

3.3.12 CCELIPSE

CCELIPSE draws partial or complete ellipses under complete specification by the user.

The calling sequence for CCELIPSE is as follows:

```
CALL      CCELIPSE(X,Y,AMA,AMI,THETA,SANGLE,EANGLE)                (Fortran)

CALL      CCELIPSE,X,Y,AMA,AMI,THETA,SANGLE,EANGLE                 (Illar)
```

X and Y are the coordinates of the center point of the ellipse. Both are expressed in floating point units of linear length. AMA and AMI are the lengths of the semi-major and semi-minor axes, respectively. Both are expressed in floating point units of linear length. Note that the meaning of semi-major axis indicates the axis by which the angle of the entire ellipse is measured. Figure 10 shows the terminology in use and how each definition applies.

THETA is the angle of the ellipse and is measured between the semi-major axis and the long edge of the plotter paper. With reference to the semi-major axis, SANGLE and EANGLE are the starting and ending angles of the elliptic arc, respectively. All three angles are expressed in floating point degrees.

A special entry, CCENDEL, allows the user to obtain the coordinates of the starting and ending points of the elliptic arc. The calling sequence and argument are described in the description of CCIRCLE.

3.3.13 CCDASHLN

CCDASHLN draws straight dashed lines in the manner of CCPLOT. The calling sequence for CCDASHLN is as follows:

```
CALL      CCDASHLN(X,Y)                                            (Fortran)

CALL      CCDASHLN,X,Y                                             (Illar)
```

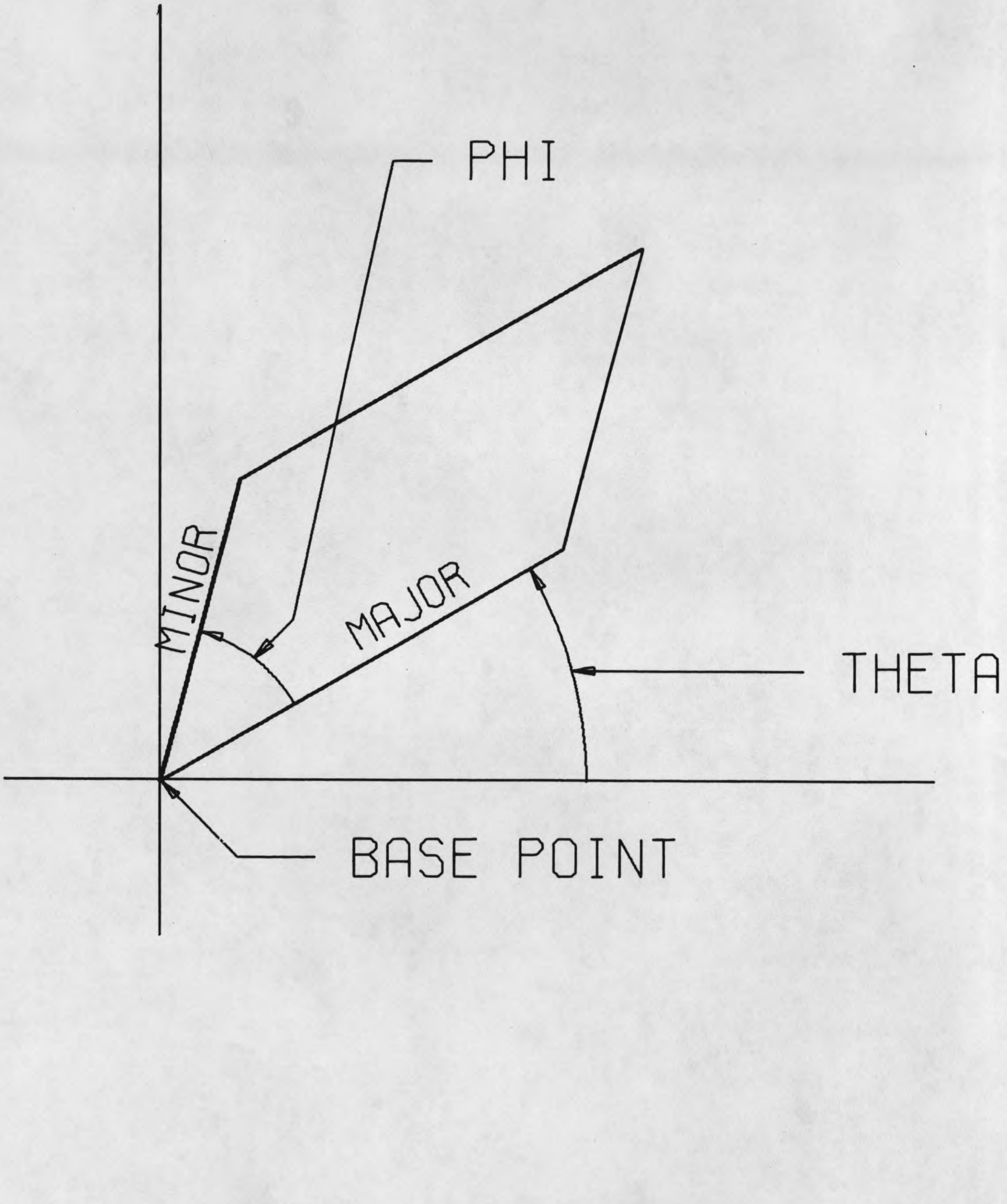


Figure 9 - Nomenclature for CCPARLEL

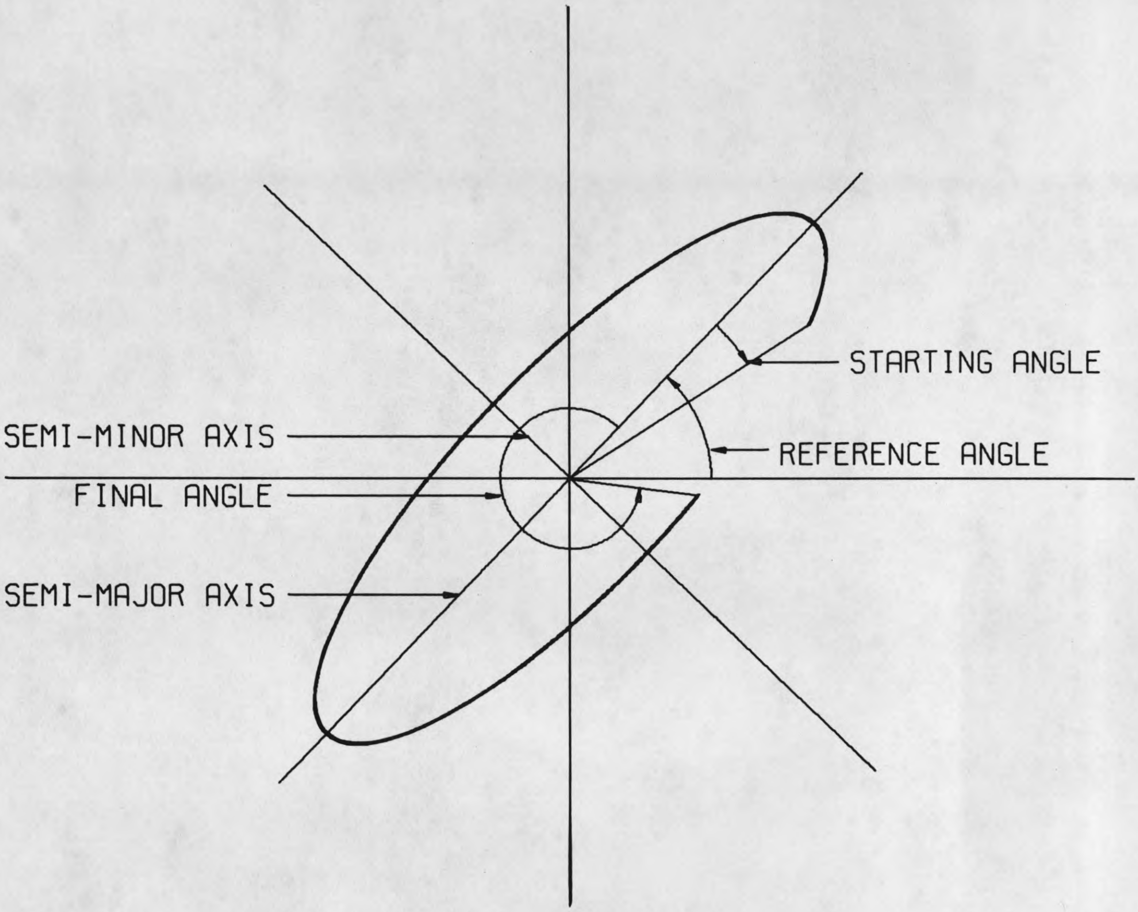


Figure 10 - Nomenclature for CCELIPSE

There is no pen position indicator argument because duplication of effort with CCPLOT would result.

The dashed line is plotted from the current position of the "user" pen to (X,Y), the coordinates of the end point. Both parameters are specified in floating point units of linear length.

Lines under 0.25 units in length will be plotted as solid lines. Lines whose lengths lie between 0.25 and 0.625 units of length will be plotted in five (5) equal length segments: line-space-line-space-line. All other lines will start and end with dash marks 0.1875 units long with the remainder of the line consisting of dash marks 0.1875 units long and space marks 0.0625 units long or longer fitted to fill up the required distance. Figure 11 shows several lines drawn with CCDASHLN.

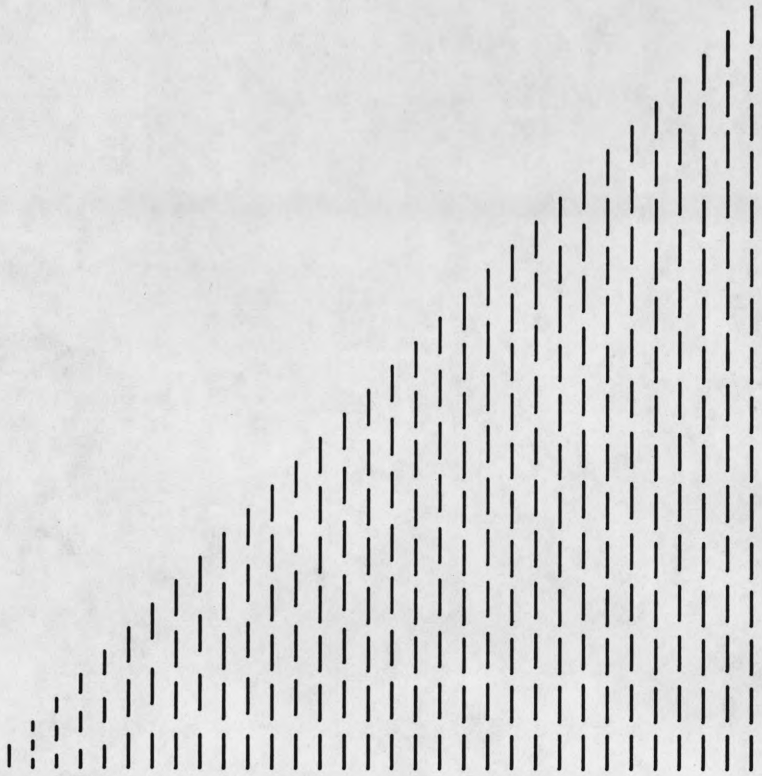


Figure 11 - Examples of the Use of CCDASHLN

4.0 Error Messages and Pitfalls

The number of error messages given out in CCSYSTEM is relatively small. This is due to the general nature of the system which prevents it from making very many decisions on form and syntax.

Associated with the "user limited plotting area" is the indication of boundary violations as shown on page 5. The movements of the "user" pen continue but the actual plotter pen movement is constrained to lie within the declared boundaries.

A call to CCPLOT with a negative pen position indicator which violates a boundary will cause the value of the x coordinate of the "user" pen position to be set to zero. Thus, the plotter pen will be out of position with respect to the "user" pen and erroneous results will occur. This problem is most likely to occur when the user is moving the pen beyond the current plotting area to separate successive plots with a block address.

The user must remember that practically all of the routines in CCSYSTEM produce results that are scaled and offset. Thus, a move to (6.0,6.0) with scale factors of 2.0 and 2.0 will violate a boundary placed at y = 10.0".

Users taking advantage of the user supplied symbol feature in CCSYMBOL are solely responsible for making sure that the keywords and symbol strings are in the proper format.

When using CCPBCDOT, all errors resulting from improper formats or datawords result in console typewriter messages which are standard for all BCD output routines.

All routines in CCSYSTEM access many subroutines of the ILLAR system which in turn have their own error messages for violations. These error returns will generally occur in very rare cases, usually due to machine failures or because of overwriting by a defective user program. To facilitate debugging in these cases, Appendix IV contains a listing of all subroutines called by each of the routines in CCSYSTEM.

All letter indices for CCSYMBOL in the special symbol mode are treated modulo 128. The value of NC(4) for CCLINE and CCGRAPH are treated modulo 32.

An error message will result from a call to CCGRAPH with NFLAG(7) = - (overlay indication) if no previous call has been made to CCGRAPH with NFLAG(7) = + (new graph indication). The offending call to CCGRAPH will be ignored. Remember that every call to CCGRAPH will return NFLAG(7) = -.

An error message will occur if any call to CCABSPLT carries a pen position indicator value <u>not</u> equal to 1, 2, 3. If the user wishes to continue, the offending call will be ignored. Note that this message will occur if the pen position indicator in a call to CCPLOT is not equal to 1,2,3 -1,-2, or -3.

APPENDIX I

In this appendix, we shall list, for reference purposes, all calling sequences to routines in CCSYSTEM. Only brief descriptions of the arguments will be given. Both sample FORTRAN and ILLAR calls will be given as examples.

CCABSPLT   (Section 3.1.1)

```
CALL      CCABSPLT(X,Y,IC)                                     (Fortran)
CALL      CCABSPLT,X,Y,IC                                      (Illar)

     X         - x coordinate
     Y         - y coordinate
     IC        - pen position indicator

CALL      CCBLKADD(IFLAG)                                      (Fortran)
CALL      CCBLKADD,IFLAG                                       (Illar)

     IFLAG     - flag to govern data record writeout

CALL      CCENDPLT                                             (Fortran)
CALL      CCENDPLT                                             (Illar)
```

CCPLOT   (Section 3.1.2)

```
PLOT,X,Y,IPEN                                                  (Fortran)
CALL      CCPLOT,X,Y,IPEN                                      (Illar)

     X         - x coordinate
     Y         - y coordinate
     IPEN      - pen position, block address indicator

CALL      CCWHERE(X,Y)                                         (Fortran)
CALL      CCWHERE,X,Y                                          (Illar)

     X         - x coordinate
     Y         - y coordinate

CALL      CCLIMITS(XMIN,XMAX,YMIN,YMAX)                        (Fortran)
CALL      CCLIMITS,XMIN,XMAX,YMIN,YMAX                         (Illar)

     XMIN      - minimum x boundary
     XMAX      - maximum x boundary
     YMIN      - minimum y boundary
     YMAX      - maximum y boundary

CALL      CCOFFSET(XOFFSET,YOFFSET)                            (Fortran)
CALL      CCOFFSET,XOFFSET,YOFFSET                             (Illar)

     XOFFSET   - x offset distance
     YOFFSET   - y offset distance

CALL      CCFACTOR(XFACTOR,YFACTOR)                            (Fortran)
CALL      CCFACTOR,XFACTOR,YFACTOR                             (Illar)

     XFACTOR   - x scale factor                                (Fortran)
     YFACTOR   - y scale factor                                (Illar)
```

```
CALL      CCFCTRS(XOFFSET,YOFFSET,XFACTOR,YFACTOR)                  (Fortran)
CALL      CCFCTRS,XOFFSET,YOFFSET,XFACTOR,YFACTOR                   (Illar)

     XOFFSET  - current value of x offset
     YOFFSET  - current value of y offset
     XFACTOR  - current value of x scale factor
     YFACTOR  - current value of y scale factor
```

CCSYMBOL   (Section 3.1.3)

Normal symbol string plotting

```
CALL      CCSYMBOL(X,Y,HEIGHT,BCD,ANGLE,N)                          (Fortran)
CALL      CCSYMBOL,X,Y,HEIGHT,BCD,ANGLE,N                           (Illar)

     X        - x coordinate of base point
     Y        - y coordinate of base point
     HEIGHT   - height of symbols
     BCD      - symbol string
     ANGLE    - angle between base line and x-axis
     N        - number of characters in string
```

Special symbol plotting

```
CALL      CCSYMBOL(X,Y,HEIGHT,BCD,ANGLE,N)                          (Fortran)
CALL      CCSYMBOL,X,Y,HEIGHT,BCD,ANGLE,N                           (Illar)

     X        - x coordinate of base point
     Y        - y coordinate of base point
     HEIGHT   - height of symbol
     BCD      - index of special symbol
     ANGLE    - angle between base line and x-axis
     N        - pen position during travel to base point
```

User supplied symbol plotting

```
CALL      CCSYMBOL(X,Y,HEIGHT,BCD,ANGLE,N)                          (Fortran)
CALL      CCSYMBOL,X,Y,HEIGHT,BCD,ANGLE,N                           (Illar)

     X        - x coordinate of base point
     Y        - y coordinate of base point
     HEIGHT   - height of symbol
     BCD      - address of user symbol keyword
     ANGLE    - angle between base line and x-axis
     N        - pen position during travel to base point
```

```
CALL      CCSOFSET(IXOFSET,IYOFSET)                                 (Fortran)
CALL      CCSOFSET(IXOFSET,IYOFSET)                                 (Illar)

     IXOFSET  - x offset for symbol plotting
     IYOFSET  - y offset for symbol plotting
```

CCPBCDOT   (Section 3.2.1)

Initial call in ILLAR

```
CALL      CCPBCDOT,FORMAT,X,Y,HEIGHT,ANGLE                          (Illar)
```

```
     FORMAT   - address of format string
     X        - x coordinate of base point
     Y        - y coordinate of base point
     HEIGHT   - height of symbols
     ANGLE    - angle between base line and x-axis
```

Data delivery

```
CALL      WRDBCDOT,DATAWORD                                        (Illar)

     DATAWORD - word to be converted
```

Completion

```
CALL      ENDBCDOT                                                 (Illar)
```

Complete call for FORTRAN

```
PLOTLINE #(X,Y,HEIGHT,ANGLE)/list                                  (Fortran)

     #        - format statement number
     X        - x coordinate of base point
     Y        - y coordinate of base point
     HEIGHT   - height of symbols
     ANGLE    - angle between base line and x-axis
     list     - string of datawords to be converted
```

<u>SCPTOPLT</u> (Section 3.2.3)

```
CALL      SCPTOPLT(BUFFER,NUM,ENDX,ENDY)                           (Fortran)
CALL      SCPTOPLT,BUFFER,NUM,ENDY,ENDY                            (Illar)

     BUFFER   - base address of CRT display command buffer
     NUM      - contains number of words in BUFFER
     ENDX     - x coordinate of final pen location
     ENDY     - y coordinate of final pen location
```

<u>CCLINSCL</u> (Section 3.3.1)

```
CALL      CCLINSCL(ARRAY,NUM,NFLAG,S,T)                            (Fortran)
CALL      CCLINSCL,ARRAY,NUM,NFLAG,S,T                             (Illar)

     ARRAY    - list to be scaled
     NUM      - contains number of elements in ARRAY
     NFLAG    - type of data format
     S        - length of data axis
     T        - 2-word "scale factor array"
```

<u>CCLOGSCL</u> (Section 3.3.2)

```
CALL      CCLOGSCL(ARRAY,NUM,NFLAG,T)                              (Fortran)
CALL      CCLOGSCL,ARRAY,NUM,NFLAG,T                               (Illar)

     ARRAY    - list to be scaled
     NUM      - contains number of elements in ARRAY
     NFLAG    - type of data format
     T        - 2-word "scale factor array"
```

CCLINAX (Section 3.3.3)

```
CALL     CCLINAX(X,Y,BCD,NC,SIZE,ANGLE,T)                      (Fortran)
CALL     CCLINAX,X,Y,BCD,NC,SIZE,ANGLE,T                       (Illar)

     X        - x coordinate of base point of base line
     Y        - y coordinate of base point of base line
     BCD      - symbol string for labeling
     NC       - number of symbols in string and position of axis indicator
     SIZE     - length of base line
     ANGLE    - angle between base line and x axis
     T        - 2-word "scale factor array"
```

CCLOGAX (Section 3.3.4)

```
CALL     CCLOGAX(X,Y,BCD,NC,SIZE,ANGLE,T)                      (Fortran)
CALL     CCLOGAX,X,Y,BCD,NC,SIZE,ANGLE,T                       (Illar)

     X        - x coordinate of base point of base line
     Y        - y coordinate of base point of base line
     BCD      - symbol string for labeling
     NC       - number of symbols in string and position of axis indicator
     SIZE     - length of base line
     ANGLE    - angle between base line and x-axis
     T        - 2-word "scale factor array"
```

CCLINGRD (Section 3.3.5)

```
CALL     CCLINGRD(X,Y,XSIZE,YSIZE,ANGLE)                       (Fortran)
CALL     CCLINGRD,X,Y,XSIZE,YSIZE,ANGLE                        (Illar)

     X        - x coordinate of base point of grid
     Y        - y coordinate of base point of grid
     XSIZE    - length of base side and position of grid indicator
     YSIZE    - length of perpendicular side of grid
     ANGLE    - angle between base side and x-axis
```

CCLOGGRD (Section 3.3.6)

```
CALL     CCLOGGRD(X,Y,XSIZE,YSIZE,ANGLE,T)                     (Fortran)
CALL     CCLOGGRD,X,Y,XSIZE,YSIZE,ANGLE,T                      (Illar)

     X        - x coordinate of base point of grid
     Y        - y coordinate of base point of grid
     XSIZE    - length of base side and position of grid indicator
     YSIZE    - length of perpendicular side of grid
     ANGLE    - angle between base side and x-axis
     T        - 2-word "scale factor array"
```

CCLINE (Section 3.3.7)

```
CALL     CCLINE(X,Y,NC,NFLAG,TX,TY,XSIZE,YSIZE)                (Fortran)
CALL     CCLINE,X,Y,NC,NFLAG,TX,TY,XSIZE,YSIZE                 (Illar)
```

X - array in x direction
Y - array in y direction
NC - 4-word array containing counts of arrays and type of curve indicators
NFLAG - 6-word array containing type of data formats and type of axes formats
TX - 2-word "scale factor array" for X
TY - 2-word "scale factor array" for Y
XSIZE - length of x-axis base line
YSIZE - length of y-axis base line

<u>CCGRAPH</u> (Section 3.3.8)

```
CALL    CCGRAPH(X,Y,NC,NFLAG,XSIZE,YSIZE,BCDX,BCDY,BCDQ,N)        (Fortran)
CALL    CCGRAPH,X,Y,NC,NFLAG,XSIZE,YSIZE,BCDX,BCDY,BCDQ,N          (Illar)
```

X - array in x direction
Y - array in y direction
NC - 6-word array containing array counts, type of curve indicators, and flags to draw grids
NFLAG - 7-word array containing type of data formats, type of axes formats, and overlay indicator
XSIZE - length of x-axis base line
YSIZE - length of y-axis base line
BCDX - symbol string for labeling x-axis
BCDY - symbol string for labeling y-axis
BCDQ - symbol string for labeling completed graph
N - 4-word array containing counts of symbol strings and date placement indicator

<u>CCIRCLE</u> (Section 3.3.9)

```
CALL    CCIRCLE(X,Y,R,SANGLE,EANGLE)        (Fortran)
CALL    CCIRCLE,X,Y,R,SANGLE,EANGLE          (Illar)
```

X - x coordinate of center
Y - y coordinate of center
R - radius
SANGLE - starting angle with respect to the x-axis
EANGLE - ending angle with respect to the x-axis

```
CALL    CCENDCR(POINTS)        (Fortran)
CALL    CCENDCR,POINTS          (Illar)
```

POINTS - 4-word array to hold coordinates of starting and ending points of circular arc

<u>CCSPIRAL</u> (Section 3.3.10)

```
CALL    CCSPIRAL(X,Y,RØ,R1,SANGLE,EANGLE)        (Fortran)
CALL    CCSPIRAL,X,Y,RØ,R1,SANGLE,EANGLE          (Illar)
```

X - x coordinate of center
Y - y coordinate of center
RØ - starting radius
R1 - ending radius
SANGLE - starting angle with respect to the x-axis
EANGLE - ending angle with respect to the x-axis

```
CALL     CCENDSP(POINTS)                                       (Fortran)
CALL     CCENDSP,POINTS                                        (Illar)
```

POINTS - 4-word array to hold coordinates of starting and ending points of spiral

<u>CCPARLEL</u> (Section 3.3.11)

```
CALL     CCPARLEL(X,Y,A,B,THETA,PHI)                           (Fortran)
CALL     CCPARLEL,X,Y,A,B,THETA,PHI                            (Illar)
```

X - x coordinate of the base point
Y - y coordinate of the base point
A - length of major side
B - length of minor side
THETA - angle between major side and x-axis
PHI - angle between minor and major sides measured at the base point.

<u>CCELIPSE</u> (Section 3.3.12)

```
CALL     CCELIPSE(X,Y,AMA,AMI,THETA,SANGLE,EANGLE)             (Fortran)
CALL     CCELIPSE,X,Y,AMA,AMI,THETA,SANGLE,EANGLE              (Illar)
```

X - x coordinate of center
Y - y coordinate of center
AMA - length of semi-major axis
AMI - length of semi-minor axis
THETA - angle between semi-major axis and x-axis
SANGLE - starting angle with respect to the semi-major axis
EANGLE - ending angle with respect to the semi-major axis

```
CALL     CCENDEL(POINTS)                                       (Fortran)
CALL     CCENDEL,POINTS                                        (Illar)
```

POINTS - 4-word array to hold coordinates of starting and ending points of elliptic arc

<u>CCDASHLN</u> (Section 3.3.13)

```
CALL     CCDASHLN(X,Y)                                         (Fortran)
CALL     CCDASHLN,X,Y                                          (Illar)
```

X - x coordinate of destination point
Y - y coordinate of distination point

APPENDIX II

Table I displays all of the sixty-four (64) standard and sixty-four (64) special symbols which may be plotted by CCSYMBOL. Table II lists the symbols by name and gives their suggested base points for those users who wish to offset them. A good example of this is any one of the arrow heads. They were used in the production of Figures 7, 8, 9, and 10.

TABLE I

Ø	:	16		32	−	48	+	64	α	80	ρ	96	Δ	112	≢
1	1	17	/	33	J	49	A	65	β	81	σ	97	%	113	⊕
2	2	18	S	34	K	50	B	66	γ	82	τ	98	\	114	≅
3	3	19	T	35	L	51	C	67	Δ	83	υ	99	⧗	115	⊥
4	4	2Ø	U	36	M	52	D	68	∈	84	φ	1ØØ	&	116	⌙
5	5	21	V	37	N	53	E	69	δ	85	χ	1Ø1	#	117	
6	6	22	W	38	O	54	F	7Ø	η	86	ψ	1Ø2	∫	118	□
7	7	23	X	39	P	55	G	71	θ	87	ω	1Ø3	±	119	○
8	8	24	Y	4Ø	Q	56	H	72	ι	88	°	1Ø4	÷	12Ø	◇
9	9	25	Z	41	R	57	I	73	κ	89	//	1Ø5	∴	121	|
1Ø	Ø	26	]	42	∨	58	<	74	λ	9Ø	∇	1Ø6	⍋	122	▷
11	=	27	,	43	$	59	.	75	μ	91	'	1Ø7	⌹	123	▵
12	≠	28	(	44	*	6Ø	)	76	ν	92	☆	1Ø8	▯	124	◁
13	≤	29	→	45	↑	61	≥	77	ξ	93	◫	1Ø9	∼	125	√
14	!	3Ø	≡	46	↓	62	?	78	ο	94	⏻	11Ø	“	126	¯
15	[	31	∧	47	>	63	;	79	π	95	⟠	111	”	127	_

TABLE II

All indicies are given in octal integer form.

Standard Character Set (suggested base points all at 0,0)

Ø	colon	40	minus
1	number 1	41	letter J
2	number 2	42	letter K
3	number 3	43	letter L
4	number 4	44	letter M
5	number 5	45	letter N
6	number 6	46	letter O
7	number 7	47	letter P
10	number 8	50	letter Q
11	number 9	51	letter R
12	number Ø	52	or mark
13	equality	53	dollar sign
14	unequality	54	asterisk
15	less than or equal to	55	upward directed arrow
16	exclamation mark	56	downward directed arrow
17	left bracket	57	greater than
20	space	60	plus
21	slash mark	61	letter A
22	letter S	62	letter B
23	letter T	63	letter C
24	letter U	64	letter D
25	letter V	65	letter E
26	letter W	66	letter F
27	letter X	67	letter G
30	letter Y	70	letter H
31	letter Z	71	letter I
32	right bracket	72	less than
33	comma	73	period
34	left parenthesis	74	right parenthesis
35	right directed arrow	75	greater than or equal to
36	equivalence	76	question mark
37	and mark	77	semi-colon

TABLE II (cont.)

		Suggested			Suggested
Special Character Set		-base point-	Special Character Set		-base point-
100	greek letter alpha	0,0	140	Centered delta	2,3
101	greek letter beta	0,0	141	Per cent	0,0
102	greek letter gamma	0,0	142	Backward slash	0,0
103	greek letter delta	0,0	143	Hour glass	0,0
104	greek letter epsilon	0,0	144	Ampersand	0,0
105	greek letter zeta	0,0	145	Number sign	0,0
106	greek letter eta	0,0	146	Integral	0,0
107	greek letter Theta	0,0	147	Plus or minus	0,0
110	greek letter Iota	0,0	150	Division	0,0
111	greek letter Kappa	0,0	151	Therefore	0,0
112	greek letter Lambda	0,0	152	Center return delta	2,3
113	greek letter Mu	0,0	153	Center return MINE	2,3
114	greek letter Nu	0,0	154	letter sized box	0,0
115	greek letter Xi	0,0	155	approximately	0,0
116	greek letter Omicron	0,0	156	left hand quotation	0,0
117	greek letter Pi	0,0	157	right hand quotation	0,0
120	greek letter Rho	0,0	160	non-equivalence	0,0
121	greek letter Sigma	0,0	161	exclusive - or	0,0
122	greek letter Tau	0,0	162	approximately equal	0,0
123	greek letter Upsilon	0,0	163	perpendicular mark	0,0
124	greek letter Phi	0,0	164	vector slash	0,0
125	greek letter Chi	0,0	165	dot	2,3
126	greek letter Psi	0,0	166	centered square	2,3
127	greek letter Omega	0,0	167	centered octagon	2,3
130	degree	0,0	170	centered diamond	2,3
131	parallel marks	0,0	171	tie mark	2,7
132	downward directed arrowhead	2,0	172	right directed arrowhead	4,2
			173	upward directed arrowhead	2,7
133	apostrophe	0,0	174	left directed arrowhead	0,2
134	Star	0,0	175	square root	0,0
135	Center return square	2,3	176	overline (extend square root)	0,0
136	Center return octagon	2,3			
137	Center return diamond	2,3	177	underline	0,0

APPENDIX III

Table III lists the thirty-two (32) curve plotting symbols used by CCLINE. The set of symbols chosen is the same set employed several years back by California Computer Products in their software.

Because the symbol set for CCSYSTEM was redesigned, some small discrepancies cannot be avoided.

TABLE III

Symbols Used by CCLINE

CCLINE Index	Name	CCSYMBOL Index	Base Point
0	square c	135b	2,3
1	octagon c	136b	2,3
2	delta c	152b	2,3
3	plus	60b	2,3
4	plus (45 degree angle)	60b	2,3
5	diamond c	137b	2,3
6	up arrow	55b	2,5
7	hour glass	143b	2,3
8	z	31b	2,3
9	y	30b	2,4
10	mine	153b	3,3
11	asterisk	54b	2,3
12	hour glass	143b	2,3
13	tic mark	171b	2,3
14	star	134b	2,3
15	colon	0b	2,3
16	one	1b	2,3
17	two	2b	2,3
18	three	3b	2,3
19	four	4b	2,3
20	five	5b	2,3
21	six	6b	2,3
22	seven	7b	2,3
23	eight	10b	2,3
24	nine	11b	2,3
25	zero	12b	2,3
26	equal	13b	2,3
27	not equal	14b	2,3
28	less than equal	15b	2,3
29	exclamation	16b	2,3
30	left bracket	17b	2,3
31	space	20b	2,3

Appendix IV

This section contains a listing of all subroutines called by each of the routines in CCSYSTEM.

CCABSPLT (Section 3.1.1)

XFIXF LOCK TYPEMSGE FLOATF
TAPBCDOT WRDBCDOT ENDBCDOT FASTWRIT
FRSTWRIT ERMSGARG IBJOB

CCPLOT (Section 3.1.2)

CCOUTBND CCABSPLT CCBLKADD IBJOB

CCSYMBOL (Section 3.1.3)

COSF SINF CCPLOT FLOATF

CCPBCDOT (Section 3.2.1)

INTBCDOT CCSYMBOL

CALCOMP (Section 3.2.2)

CCENDPLT CCPLOT CCOFSET CCSYMBOL
CCPBCDOT WRDBCDOT ENDBCDOT CCWHERE
TYPEMSGE IBJOB

SCPTOPLT (Section 3.2.3)

CCPLOT FLOATF CCSYMBOL CCWHERE
CCSOFSET

CCLINSCL (Section 3.3.1)

MAXMIN INTF LOGF FIXFLOAT
EXPRI FLOATFIX EXPRR

CCLOGSCL (Section 3.3.2)

MAXMIN LOGF FIXFLOAT FLOATFIX

CCLINAX (Section 3.3.3)

XABSF FIXFLOAT COSF SINF
FLOATFIX CCPLOT CCSOFSET CCSYMBOL
CCPBCDOT WRDBCDOT ENDBCDOT CCWHERE

CCLOGAX (Section 3.3.4)

XABSF FLOATFIX FIXFLOAT COSF
SINF SQRTF CCSOFSET CCPLOT
CCSYMBOL CCPBCDOT WRDBCDOT ENDBCDOT

CCLINGRD (Section 3.3.5)

INTF ABSF COSF SINF
CCPLOT

CCLOGGRD (Section 3.3.6)

ABSF COSF SINF CCPLOT

CCLINE (Section 3.3.7)

XABSF INTF CCWHERE FLOATFIX
LOGF CCPLOT CCSYMBOL CCSOFSET

CCGRAPH (Section 3.3.8)

DATE FLOATFIX CCSOFSET XABSF
INTF CCLINSCL CCLOGSCL CCPLOT
CCLOGAX CCSYMBOL CCLINAX CCLOGGRD
CCLINGRD CCLINE TAPBCDOT ENDBCDOT
IBJOB

CCIRCLE (Section 3.3.9)

COSF SINF CCPLOT

CCSPIRAL (Section 3.3.10)

COSF SINF CCPLOT

CCPARLEL (Section 3.3.11)

CCPLOT COSF SINF

CCELIPSE (Section 3.3.12)

EXPRI SQRTF COSF SINF
CCPLOT ERROREXT

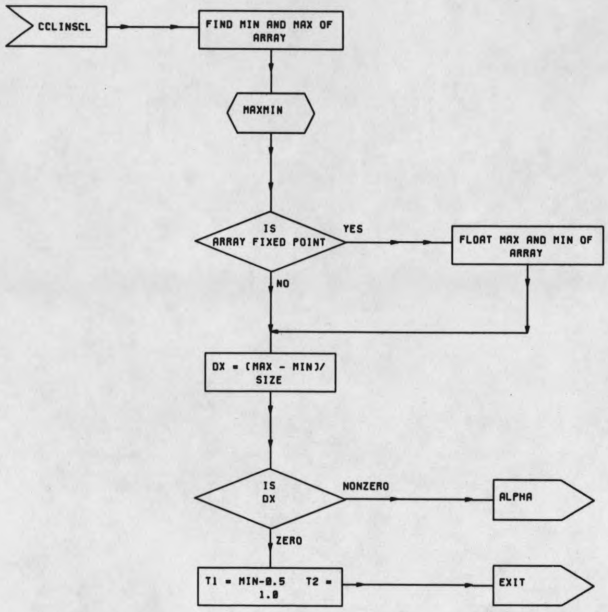
CCDASHLN (Section 3.3.13)

CCWHERE SQRTF INTF XFIXF
CCPLOT

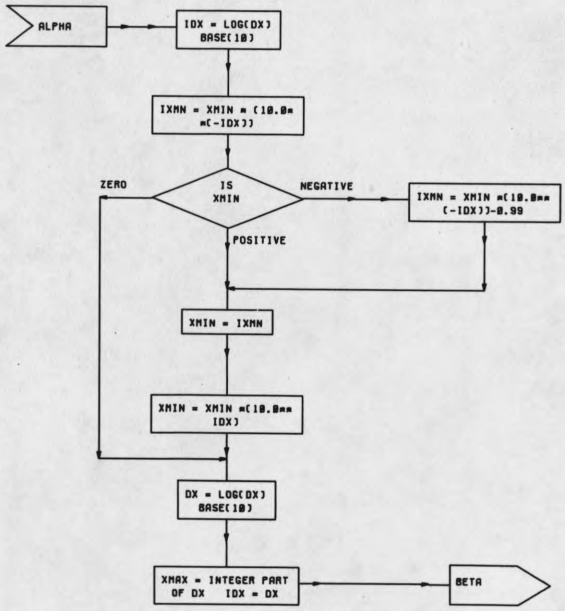
Appendix V

Contained in this section are flow charts for all programs in Group 3 except CCDASHLN.

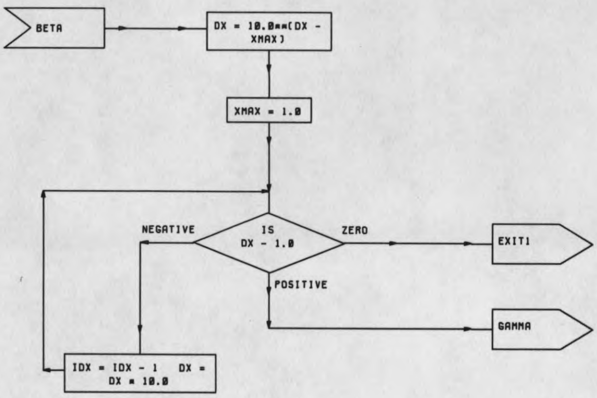
CCLINSCL - INITIAL MIN, MAX SEARCH --CCSYSTEM-- 8/25/67 PAGE 1



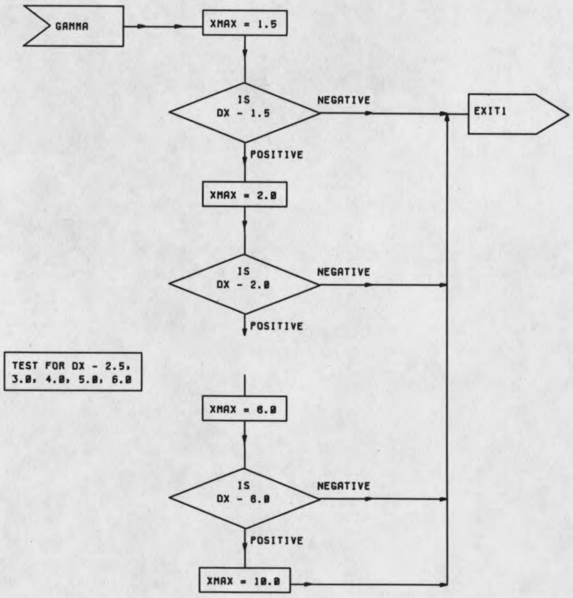
CCLINSCL - ESTABLISH TRAIL VALUES TO COMPUTE STEP SIZE --CCSYSTEM-- 8/25/67 PAGE 2


ALPHA
IDX = LOG(DX) BASE(10)
IXMN = XMIN * (10.0* *(-IDX))
ZERO
IS XMIN
NEGATIVE
IXMN = XMIN *(10.0** (-IDX))-0.99
POSITIVE
XMIN = IXMN
XMIN = XMIN *(10.0** IDX)
DX = LOG(DX) BASE(10)
XMAX = INTEGER PART OF DX IDX = DX
BETA

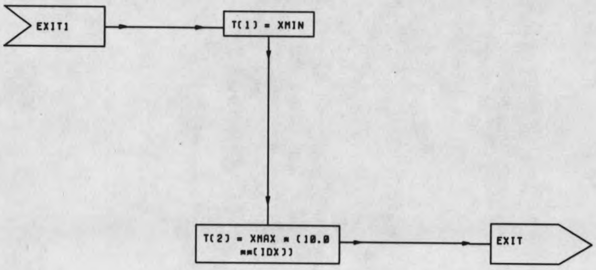
CCLINSCL - DECIDE ON STEP SIZE COMPUTATION --CCSYSTEM-- 8/25/67 PAGE 3



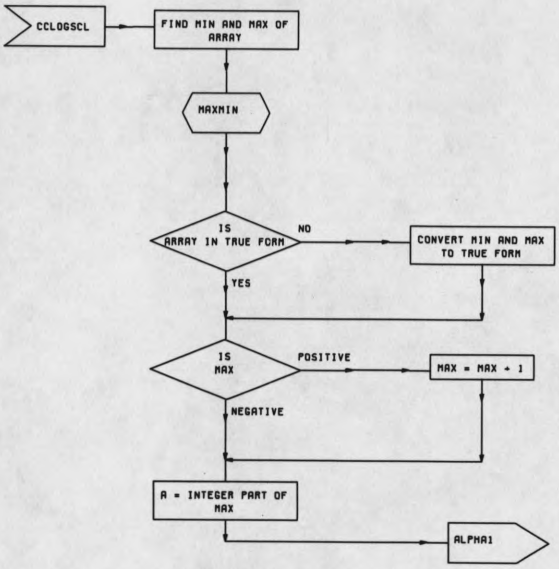
CCLINSCL - SELECT SCALE RANGE FOR STEP SCALE --CCSYSTEM-- 8/25/67 PAGE 4



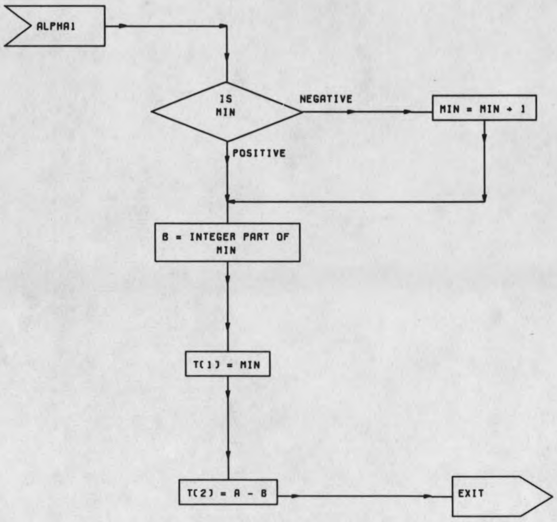
CCLINSCL - COMPUTE FINAL SCALE FACTOR ARRAY --CCSYSTEM-- 8/25/67 PAGE 5



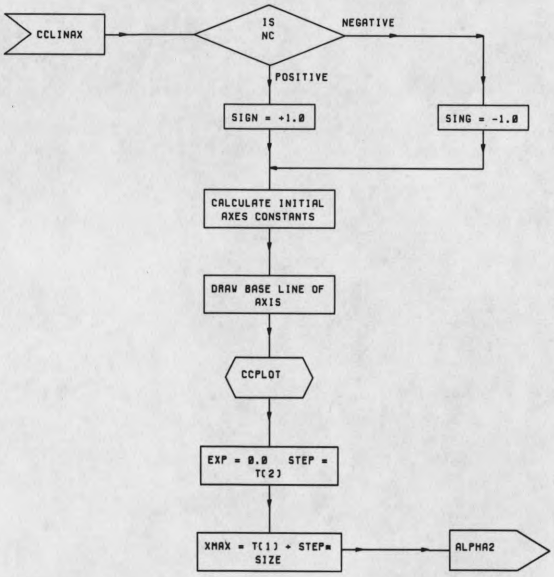
CCLOGSCL - FIND MIN, MAX AND UPPER END OF COVER --CCSYSTEM-- 8/25/67 PAGE 6



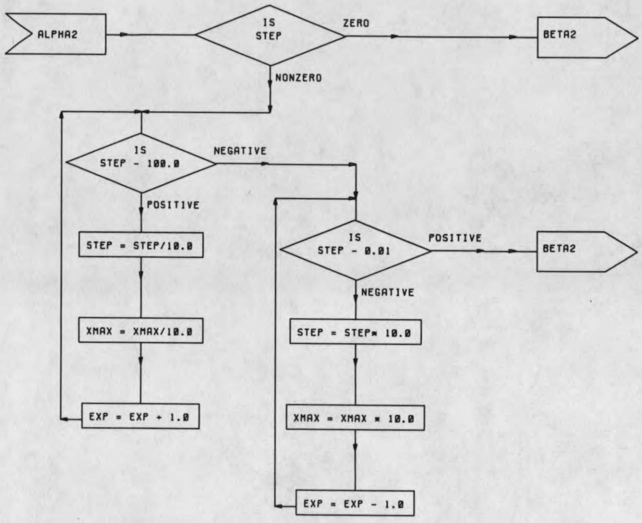
CCLOGSCL - FIND LOWER END, GET NUMBER OF COVERING CYCLES --CCSYSTEM-- 6/25/67 PAGE 7



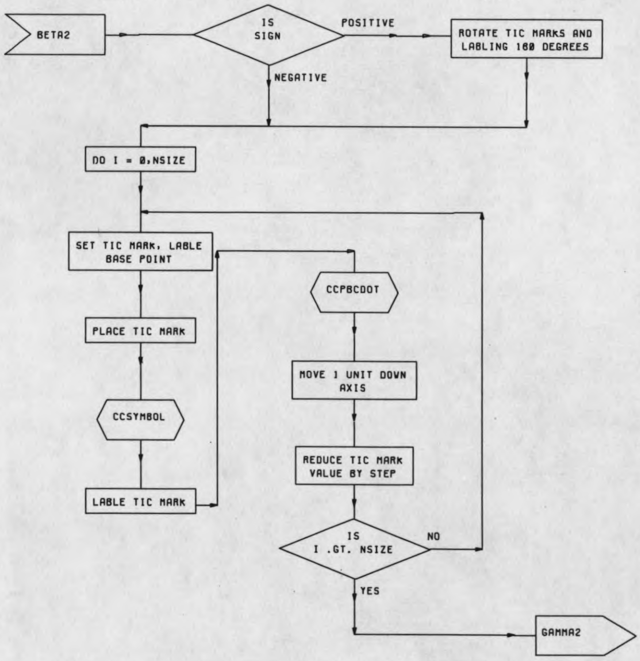
CCLINAX - INITIALIZATION, DRAW BASELINE --CCSYSTEM-- 6/25/67 PAGE 8



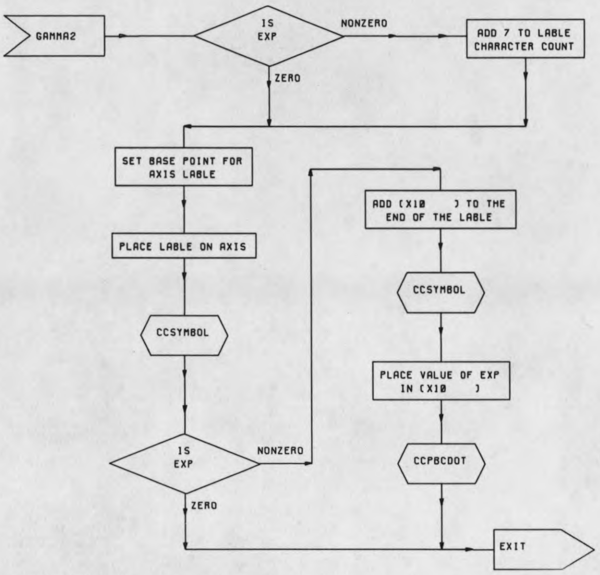
CCLINAX - ADJUST BASE 10 RANGE OF AXIS --CCSYSTEM-- 8/25/67 PAGE 9



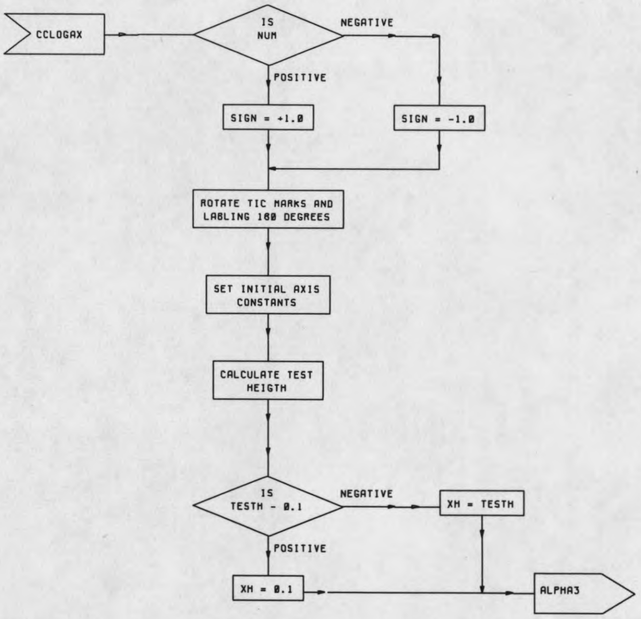
CCLINAX - DRAW AND LABLE TIC MARKS --CCSYSTEM-- 8/25/67 PAGE 10



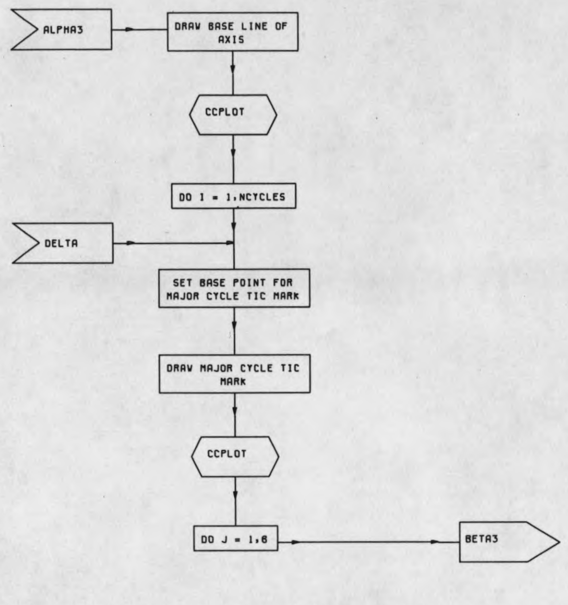
CCLINAX - LABLE THE AXIS --CCSYSTEM-- 8/25/67 PAGE 11



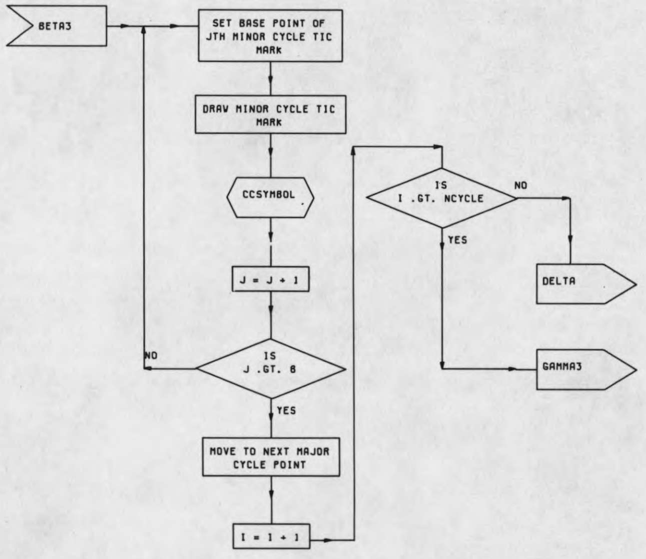
CCLOGAX - INITIAL SET UP OF CONSTANTS --CCSYSTEM-- 8/25/67 PAGE 12



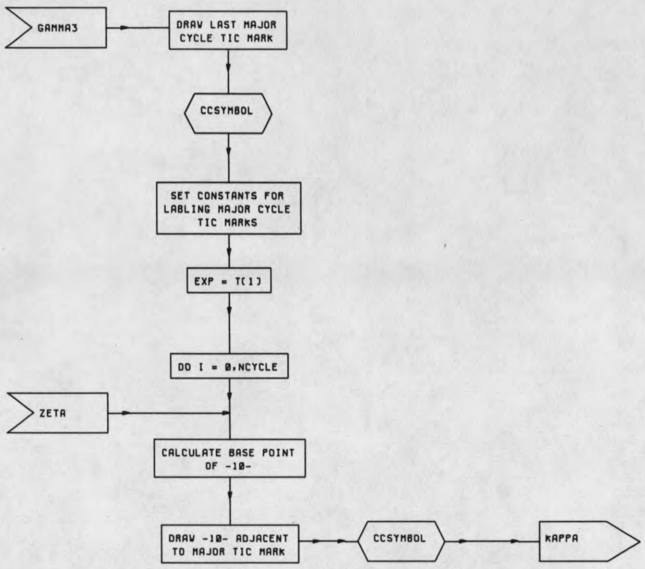
CCLOGAX - DRAW BASE LINE AND MAJOR TIC MARKS --CCSYSTEM-- 8/25/67 PAGE 13



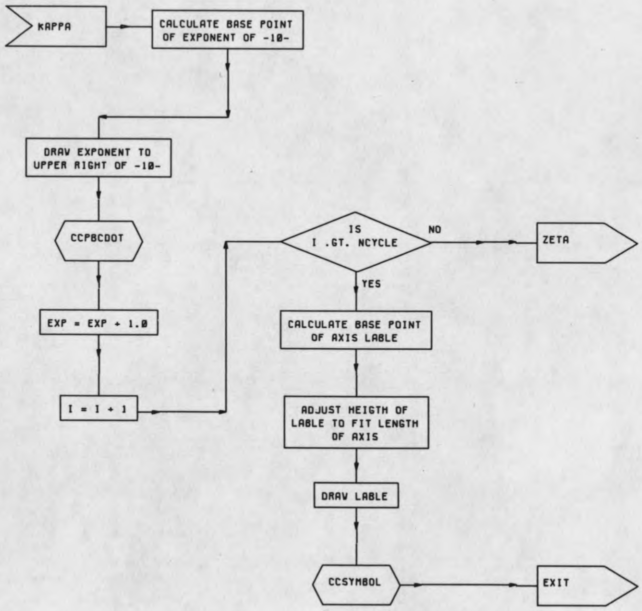
CCLOGAX - DRAW MINOR CYCLE TIC MARKS --CCSYSTEM-- 8/28/67 PAGE 14


BETA3
SET BASE POINT OF JTH MINOR CYCLE TIC MARK
DRAW MINOR CYCLE TIC MARK
CCSYMBOL
J = J + 1
NO
IS J .GT. 8
YES
MOVE TO NEXT MAJOR CYCLE POINT
I = I + 1
IS I .GT. NCYCLE
NO
YES
DELTA
GAMMA3

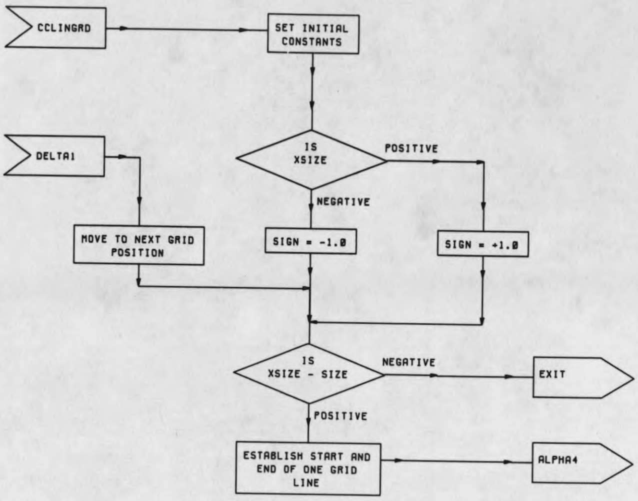
CCLOGAX - BEGIN LABLING MAJOR CYCLE TIC MARKS --CCSYSTEM-- 8/28/67 PAGE 15


GAMMA3
DRAW LAST MAJOR CYCLE TIC MARK
CCSYMBOL
SET CONSTANTS FOR LABLING MAJOR CYCLE TIC MARKS
EXP = T(1)
DO I = 0,NCYCLE
ZETA
CALCULATE BASE POINT OF -10-
DRAW -10- ADJACENT TO MAJOR TIC MARK
CCSYMBOL
KAPPA

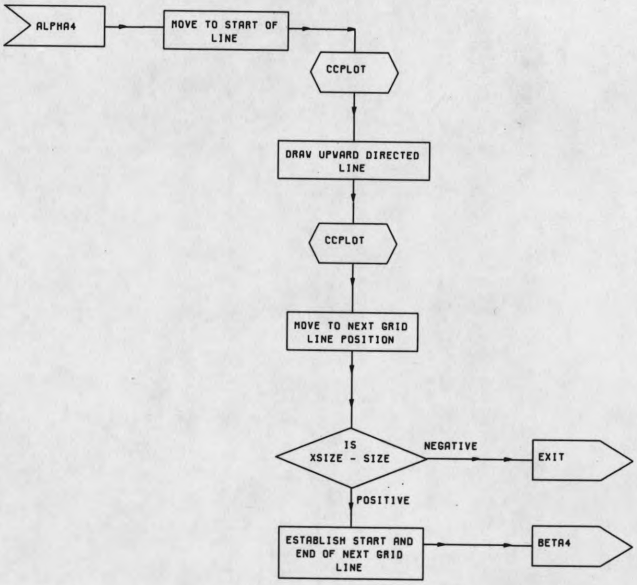
CCLOGAX - COMPLETE MAJOR CYCLE LABLES, LABLE AXIS --CCSYSTEM-- 8/28/67 PAGE 16


KAPPA
CALCULATE BASE POINT OF EXPONENT OF -10-
DRAW EXPONENT TO UPPER RIGHT OF -10-
CCPBCDOT
EXP = EXP + 1.0
I = I + 1
IS I .GT. NCYCLE
NO
ZETA
YES
CALCULATE BASE POINT OF AXIS LABLE
ADJUST HEIGTH OF LABLE TO FIT LENGTH OF AXIS
DRAW LABLE
CCSYMBOL
EXIT

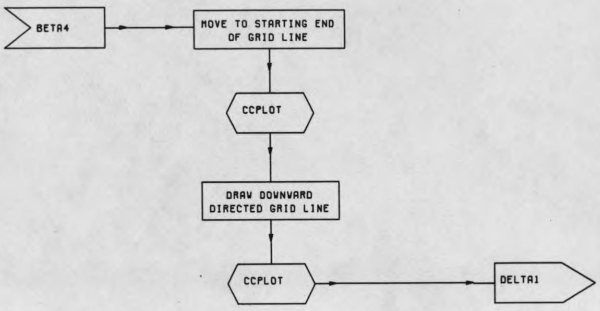
CCLINGRD - INIT., START DRAWING OF UPWARD DIRECTED LINE --CCSYSTEM-- 8/28/67 PAGE 17



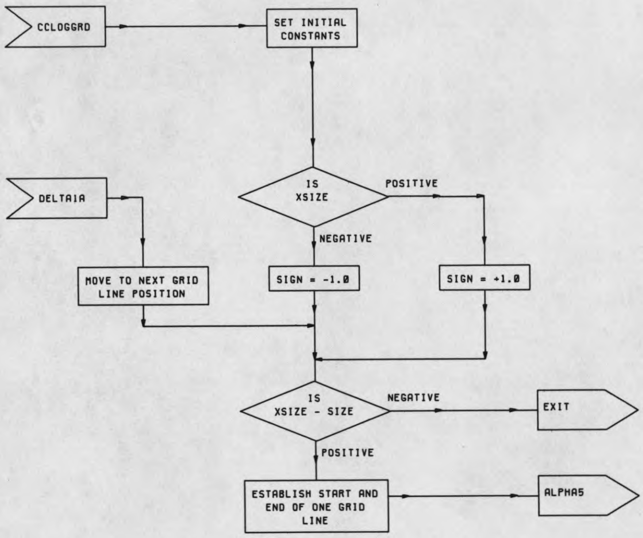
CCLINGRD - DRAW UPWARD LINE - START DOWNWARD LINE --CCSYSTEM-- 8/28/67 PAGE 18


ALPHA4
MOVE TO START OF LINE
CCPLOT
DRAW UPWARD DIRECTED LINE
CCPLOT
MOVE TO NEXT GRID LINE POSITION
IS XSIZE - SIZE
NEGATIVE
EXIT
POSITIVE
ESTABLISH START AND END OF NEXT GRID LINE
BETA4

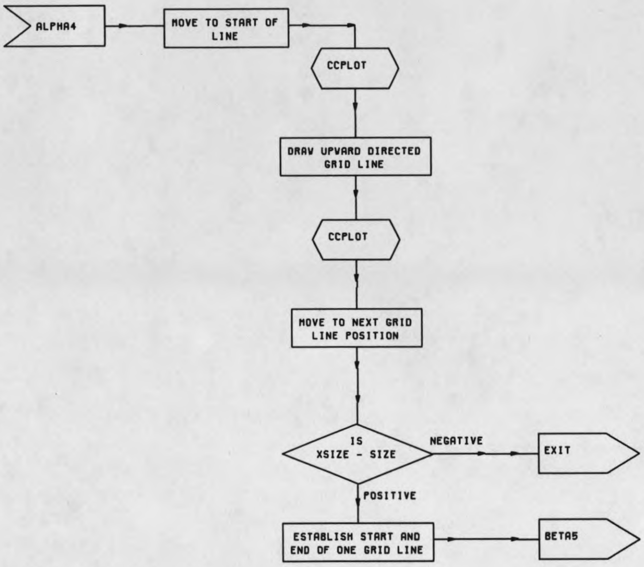
CCLINGRD - DRAW DOWNWARD DIRECTED LINE --CCSYSTEM-- 8/28/67 PAGE 19


BETA4
MOVE TO STARTING END OF GRID LINE
CCPLOT
DRAW DOWNWARD DIRECTED GRID LINE
CCPLOT
DELTA1

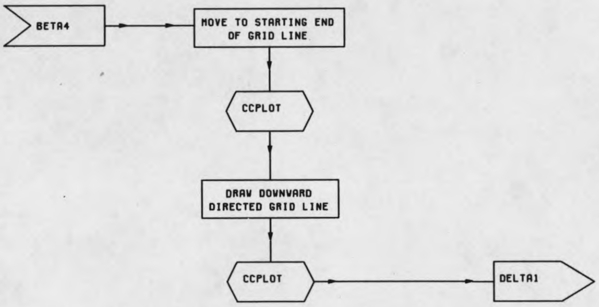
CCLOGGRD - INIT., START DRAWING UPWARD DIRECTED LINE --CCSYSTEM-- 8/28/67 PAGE 20



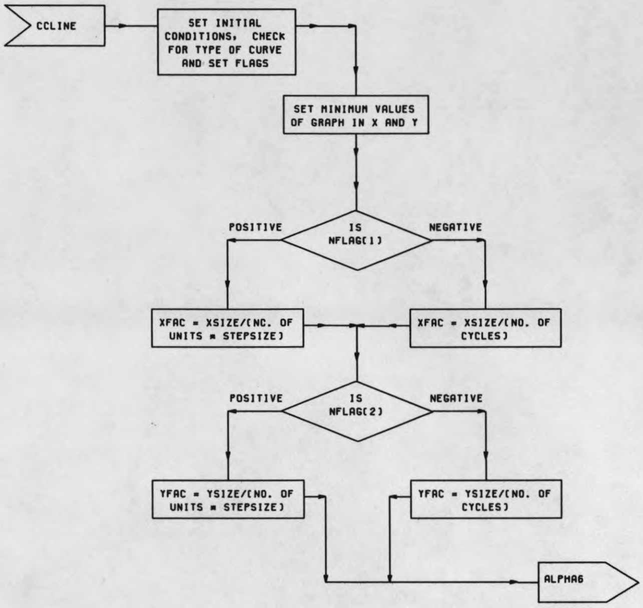
CCLOGGRD - DRAW UPWARD LINE, START DOWNWARD DIRECTED LINE --CCSYSTEM-- 8/28/67 PAGE 21


ALPHA4
MOVE TO START OF LINE
CCPLOT
DRAW UPWARD DIRECTED GRID LINE
CCPLOT
MOVE TO NEXT GRID LINE POSITION
IS XSIZE - SIZE
NEGATIVE
EXIT
POSITIVE
ESTABLISH START AND END OF ONE GRID LINE
BETA5

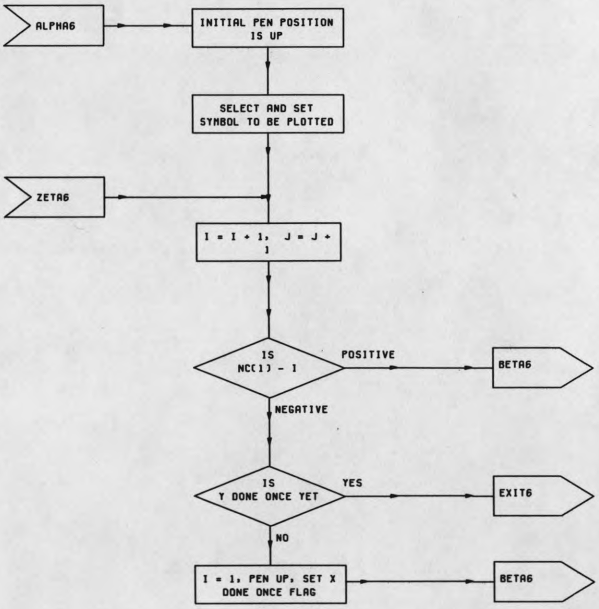
CCLOGGRD - DRAW DOWNWARD DIRECTED GRID LINE --CCSYSTEM-- 8/28/67 PAGE 22


BETA4
MOVE TO STARTING END OF GRID LINE
CCPLOT
DRAW DOWNWARD DIRECTED GRID LINE
CCPLOT
DELTA1

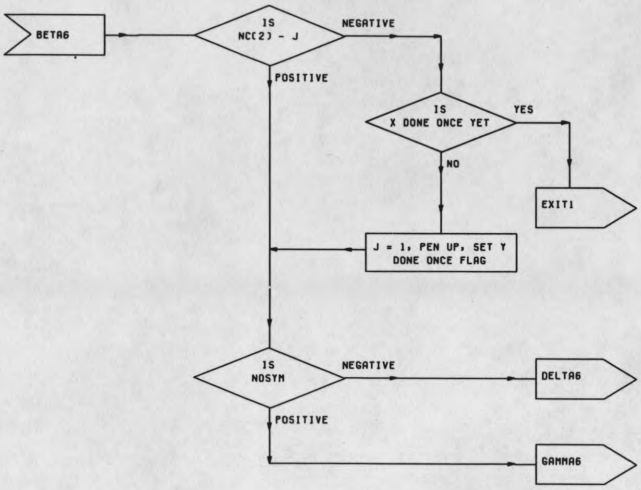
CCLINE - SET INITIAL CONSTANTS --CCSYSTEM-- 8/28/67 PAGE 23


CCLINE
SET INITIAL CONDITIONS, CHECK FOR TYPE OF CURVE AND SET FLAGS
SET MINIMUM VALUES OF GRAPH IN X AND Y
POSITIVE
IS NFLAG(1)
NEGATIVE
XFAC = XSIZE/(NO. OF UNITS * STEPSIZE)
XFAC = XSIZE/(NO. OF CYCLES)
POSITIVE
IS NFLAG(2)
NEGATIVE
YFAC = YSIZE/(NO. OF UNITS * STEPSIZE)
YFAC = YSIZE/(NO. OF CYCLES)
ALPHA6

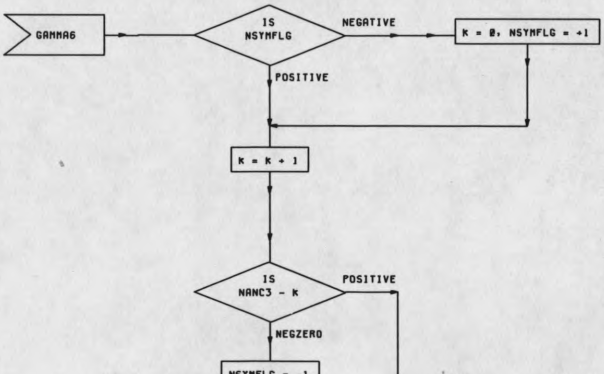
CCLINE - CHECK X DONE, Y DONE, OR RESET X --CCSYSTEM-- 8/28/67 PAGE 24



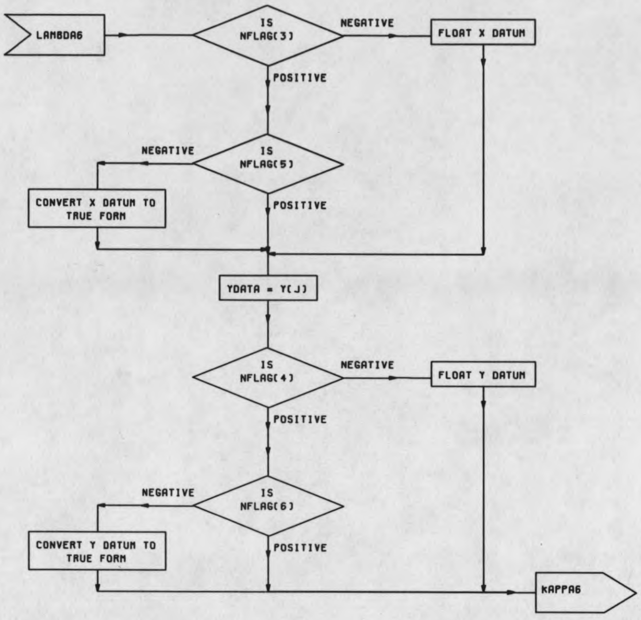
CCLINE - CHECK Y DONE, X DONE OR RESET Y --CCSYSTEM-- 8/28/67 PAGE 25



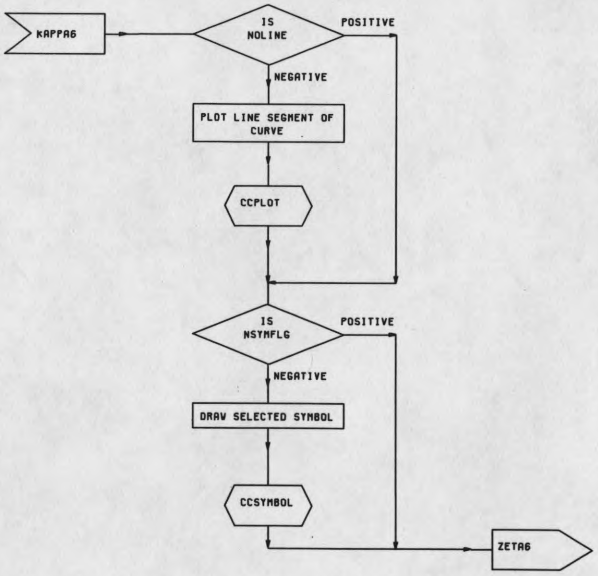
CCLINE - SET FLAGS AND CHECK COUNTERS FOR SYMBOL PLOTTING --CCSYSTEM-- 8/28/67 PAGE 26



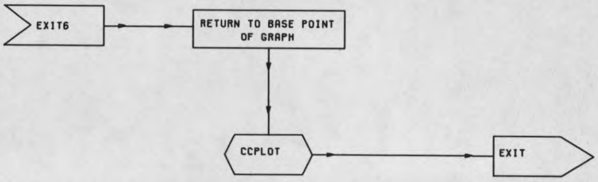
CCLINE - CHECK DATA AND GET TRUE FORMS --CCSYSTEM-- 8/28/67 PAGE 27



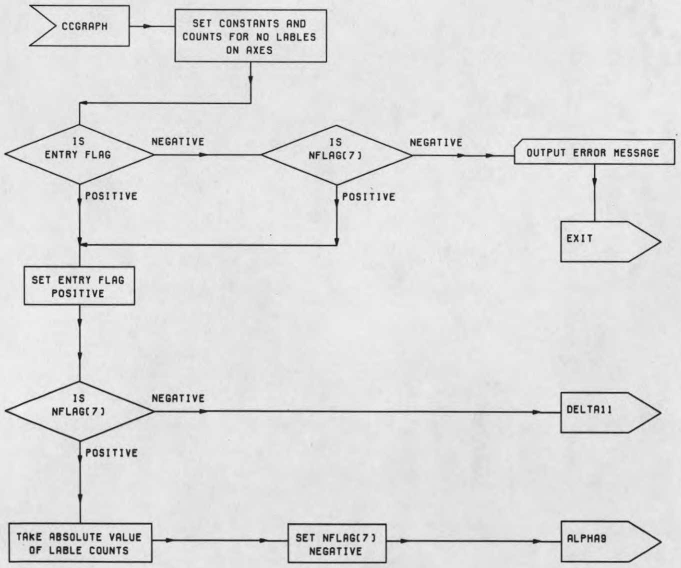
CCLINE - DRAW CURVE AND/OR SYMBOLS --CCSYSTEM-- 8/28/67 PAGE 28



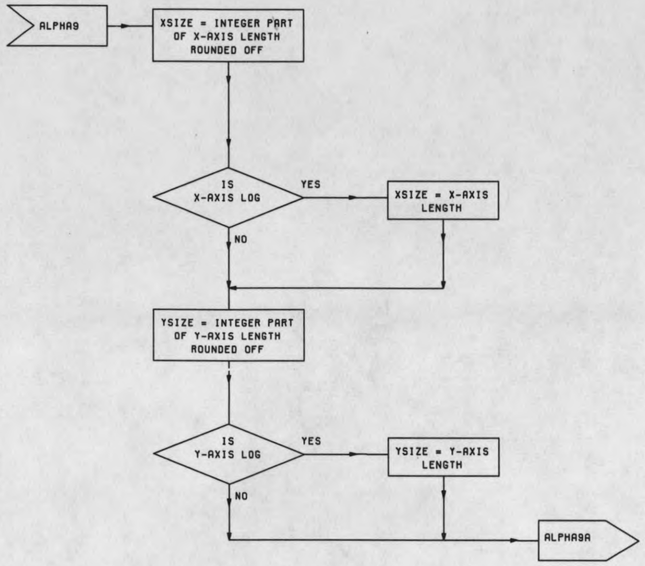
CCLINE - RETURN TO BASE POINT OF GRAPH AND EXIT --CCSYSTEM-- 8/28/67 PAGE 29



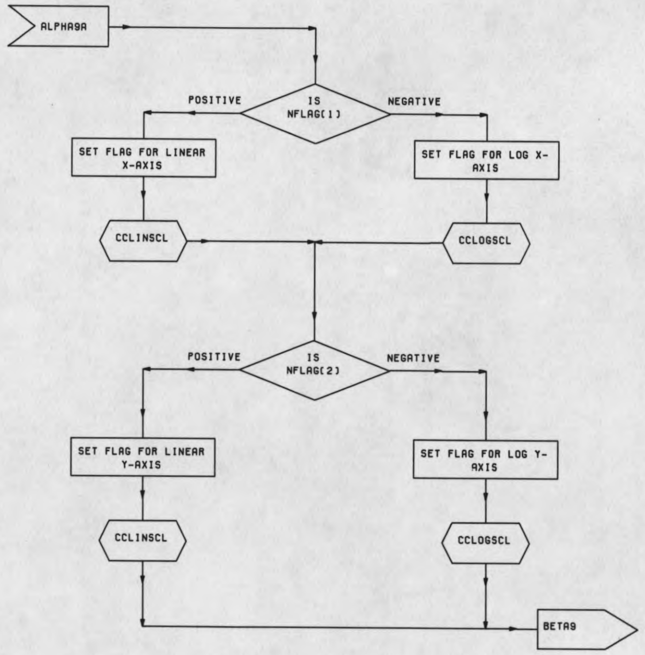
CCGRAPH - CHECK OVERLAY, SET LABLE COUNTS AND FLAGS --CCSYSTEM-- 8/29/67 PAGE 30



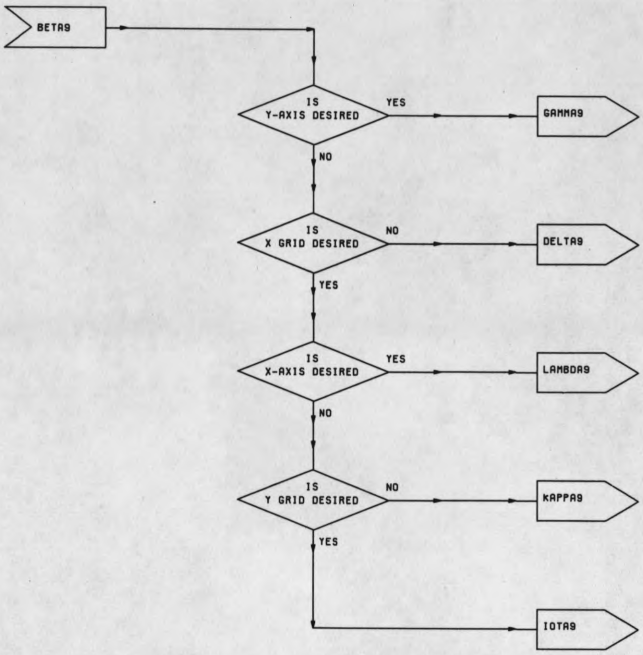
CCGRAPH - SET LENGTHS OF AXES --CCSYSTEM-- 8/29/67 PAGE 31


ALPHA9
XSIZE = INTEGER PART OF X-AXIS LENGTH ROUNDED OFF
IS X-AXIS LOG
YES
XSIZE = X-AXIS LENGTH
NO
YSIZE = INTEGER PART OF Y-AXIS LENGTH ROUNDED OFF
IS Y-AXIS LOG
YES
YSIZE = Y-AXIS LENGTH
NO
ALPHA9A

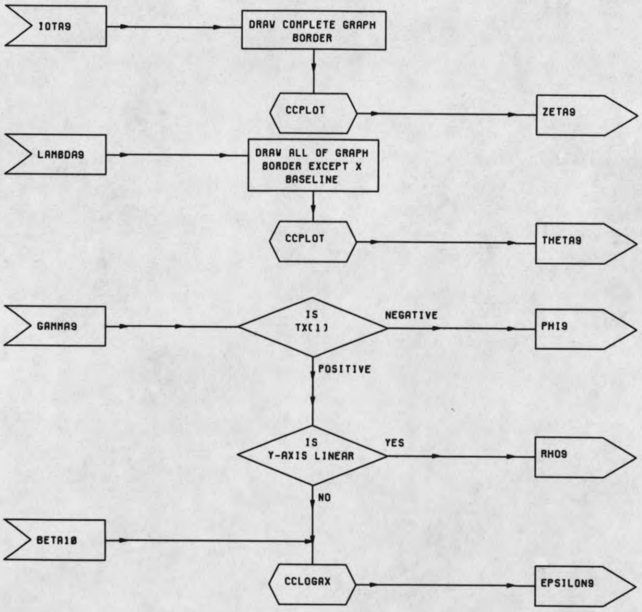
CCGRAPH - SCALE X AND Y AXES --CCSYSTEM-- 8/29/67 PAGE 32



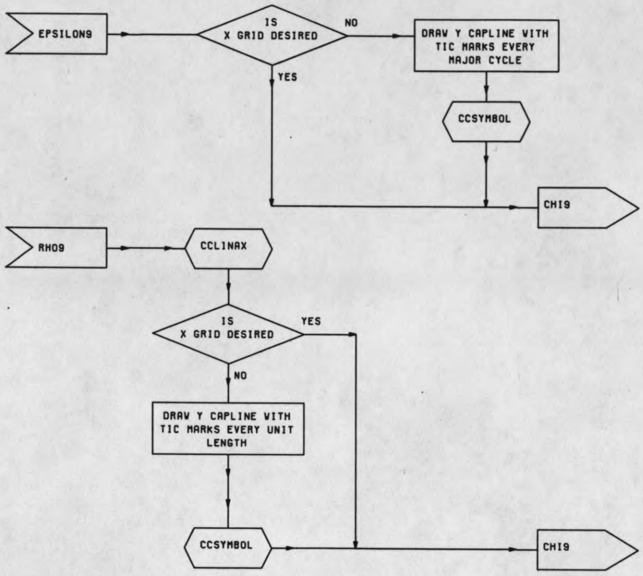
CCGRAPH - MAKE PATH DECISIONS VITH REGARD TO AXIS DRAVINGS --CCSYSTEM-- 8/29/67 PAGE 33



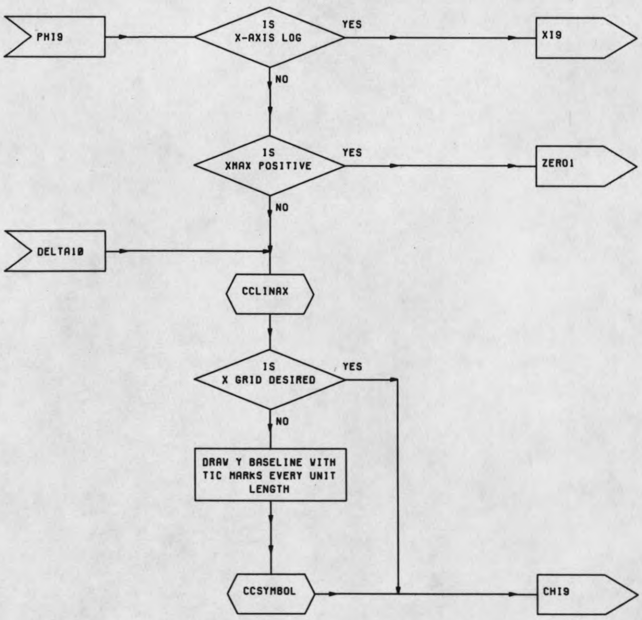
CCGRAPH - DRAV SOME BORDERS, DECIDE ON X-AXIS TYPE --CCSYSTEM-- 8/29/67 PAGE 34



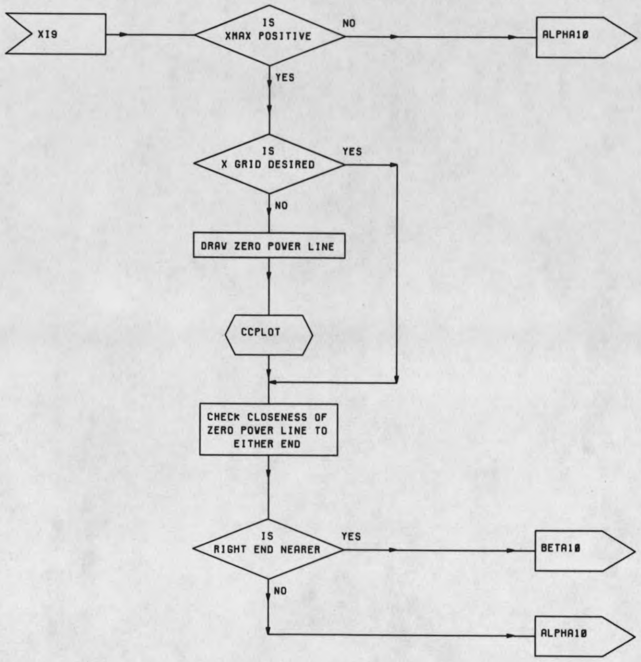
CCGRAPH - DRAW Y AXES, Y CAPLINES, AND TIC MARKS --CCSYSTEM-- 8/29/67 PAGE 35


EPSILON9
IS X GRID DESIRED
NO
YES
DRAW Y CAPLINE WITH TIC MARKS EVERY MAJOR CYCLE
CCSYMBOL
CHI9
RHO9
CCLINAX
IS X GRID DESIRED
YES
NO
DRAW Y CAPLINE WITH TIC MARKS EVERY UNIT LENGTH
CCSYMBOL
CHI9

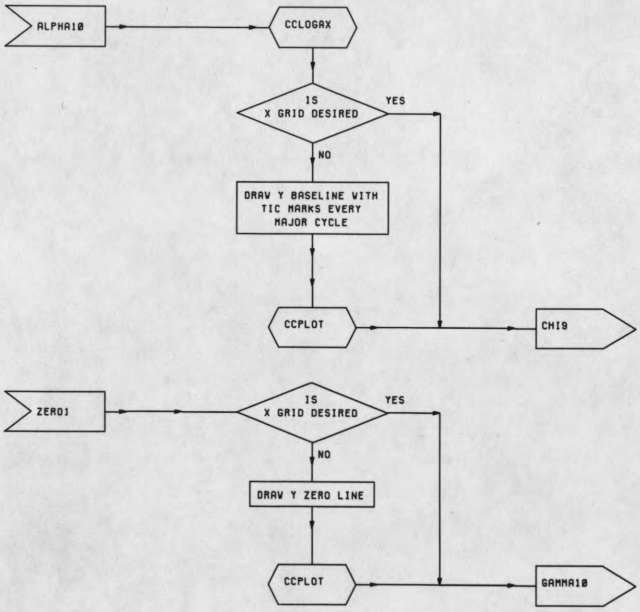
CCGRAPH - DRAW LINEAR Y-AXIS, Y BASELINE AND TIC MARKS --CCSYSTEM-- 8/29/67 PAGE 36



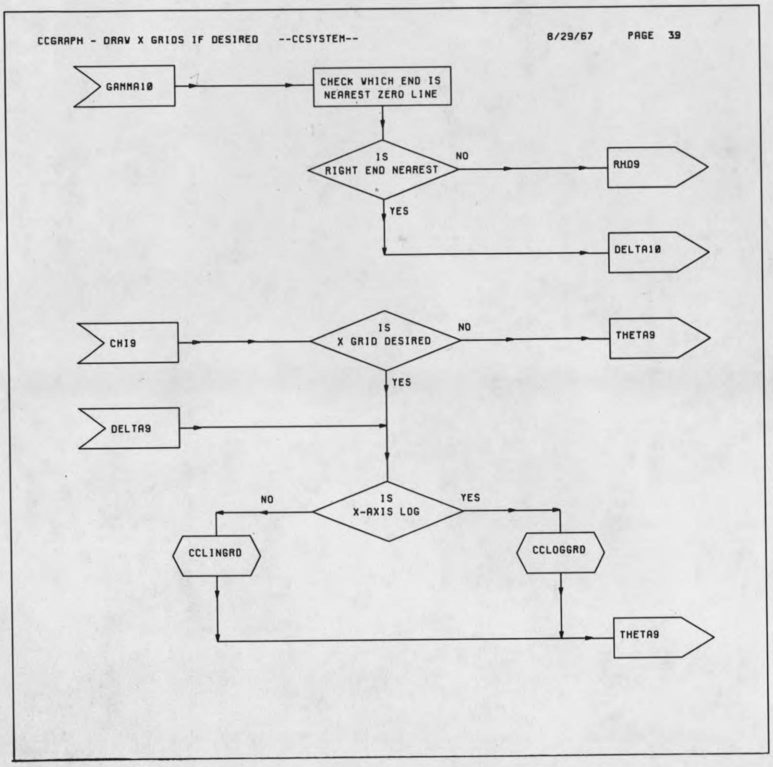
CCGRAPH - CHECK AND DRAV X ZERO POVER LINE ON LOG GRAPH --CCSYSTEM-- 8/29/67 PAGE 37



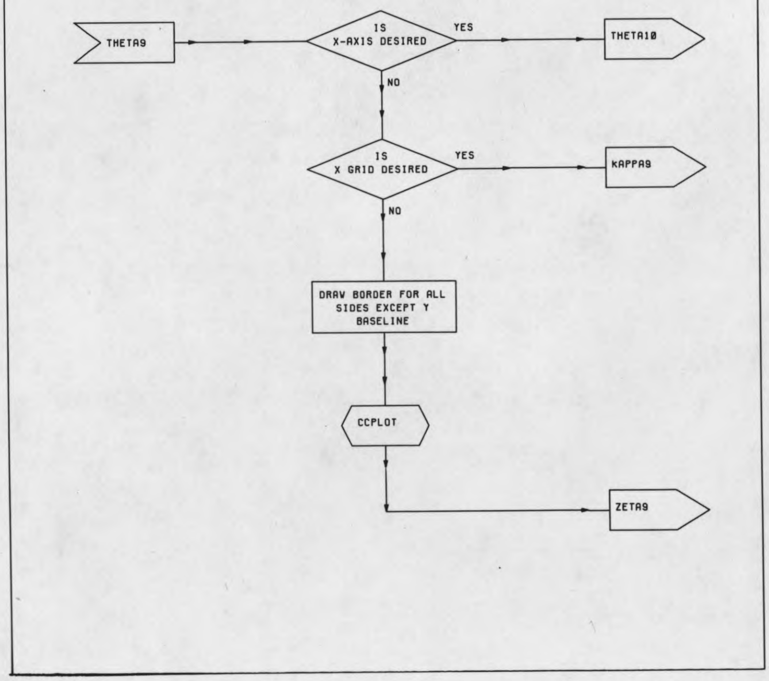
CCGRAPH - DRAV Y BASELINE, Y ZERO LINE ON LINEAR GRAPH --CCSYSTEM-- 8/29/67 PAGE 38



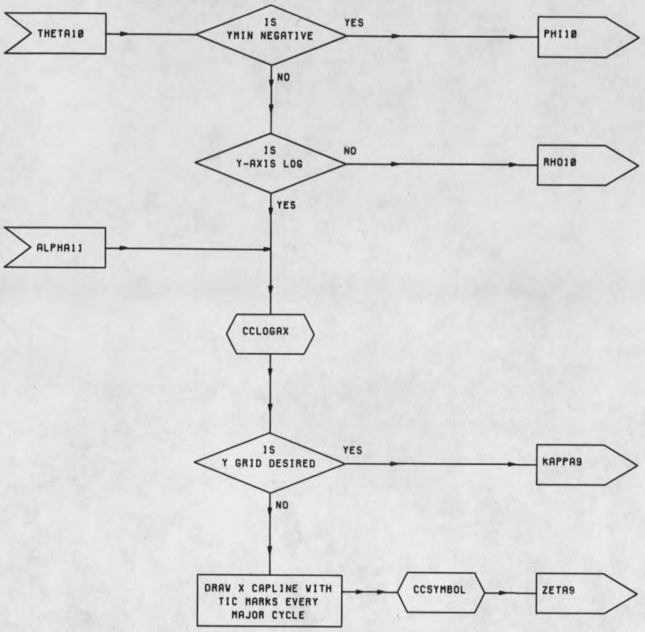
CCGRAPH - DRAW X GRIDS IF DESIRED --CCSYSTEM-- 8/29/67 PAGE 39



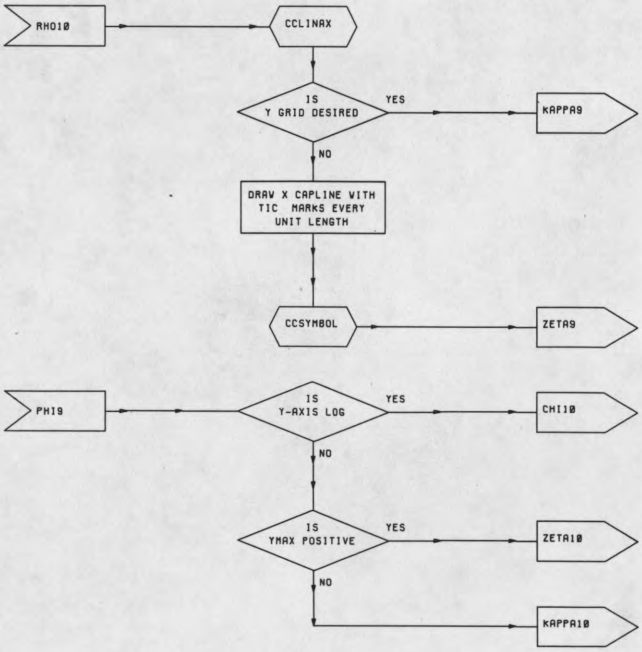
CCGRAPH - DRAW BORDER AND DECIDE ON X GRID FOR NO X-AXIS --CCSYSTEM-- 8/29/67 PAGE 40



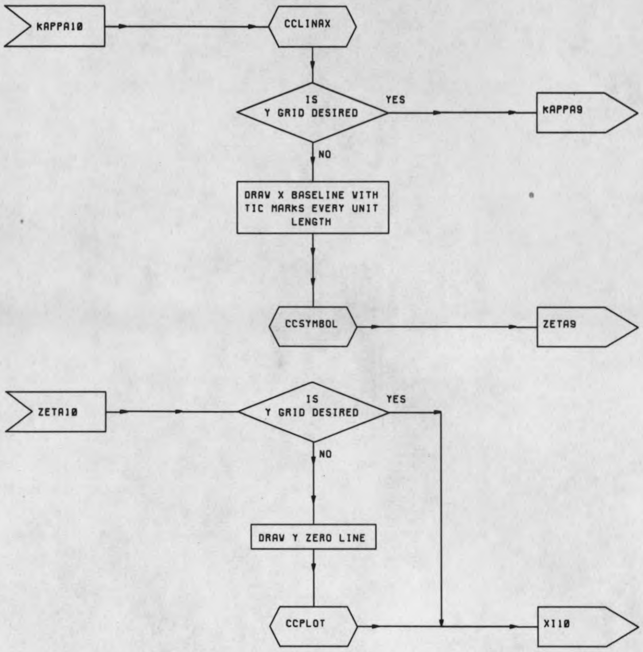
CCGRAPH - DRAW Y-AXIS AND X CAPLINE WITH TIC MARKS --CCSYSTEM-- 8/29/67 PAGE 41



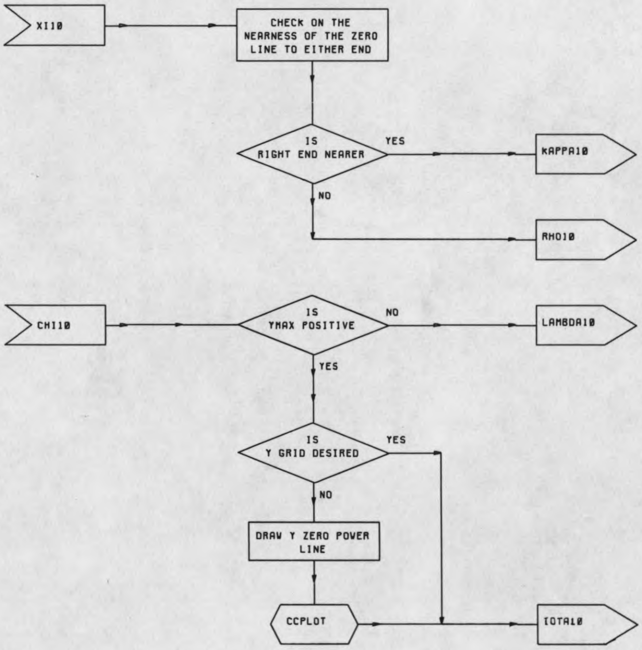
CCGRAPH - DRAW X CAPLINE, CHECK ON X-AXIS PLACEMENT --CCSYSTEM-- 8/29/67 PAGE 42



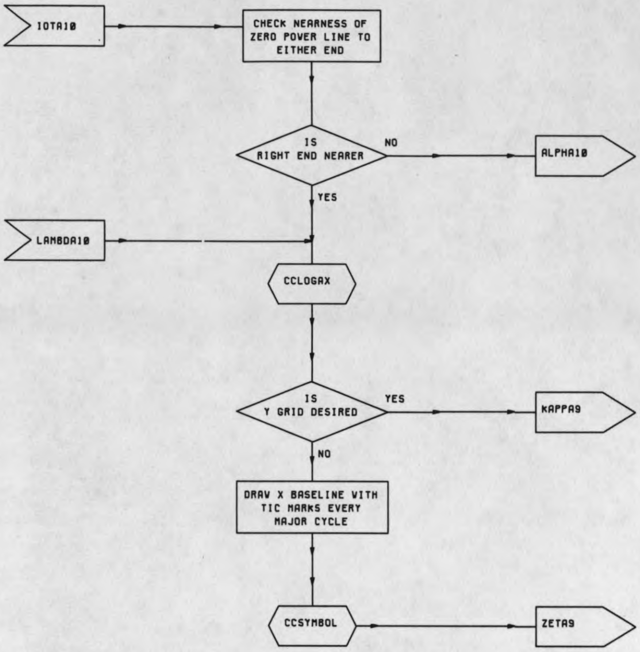
CCGRAPH - DRAV X BASELINE, Y ZERO LINE ON LINEAR GRAPH --CCSYSTEM-- 8/29/67 PAGE 43



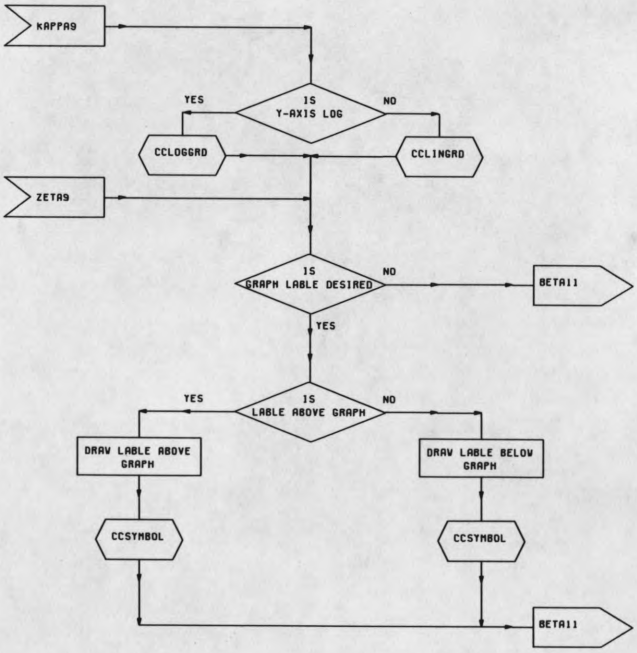
CCGRAPH - DRAV Y ZERO POWER LINE ON LOG GRAPH --CCSYSTEM-- 8/29/67 PAGE 44



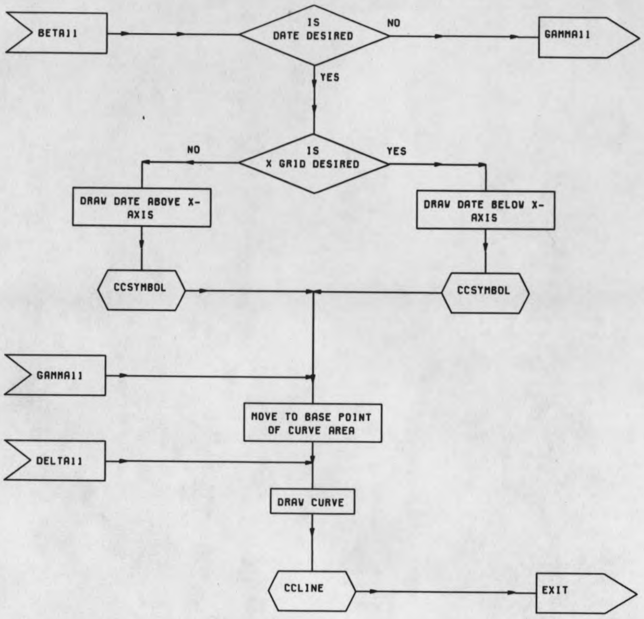
CCGRAPH - DRAV X BASELINE AND TIC MARKS FOR Y-AXIS LOG --CCSYSTEM-- 9/5/67 PAGE 45


IOTA10
CHECK NEARNESS OF ZERO POVER LINE TO EITHER END
IS RIGHT END NEARER
NO
ALPHA10
YES
LAMBDA10
CCLOGAX
IS Y GRID DESIRED
YES
KAPPA9
NO
DRAV X BASELINE VITH TIC MARKS EVERY MAJOR CYCLE
CCSYMBOL
ZETA9

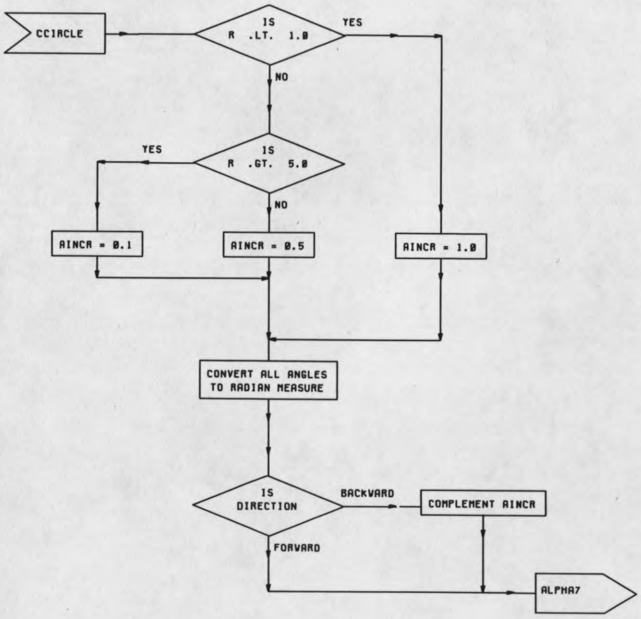
CCGRAPH - DRAV Y GRIDS, TAKE CARE OF GRAPH LABLE --CCSYSTEM-- 8/29/67 PAGE 46



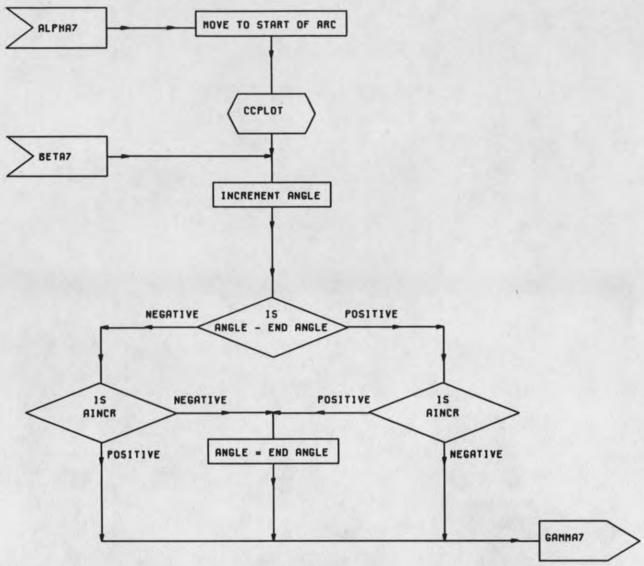
CCGRAPH - PLACE DATE AND DRAW CURVE --CCSYSTEM-- 8/29/67 PAGE 47



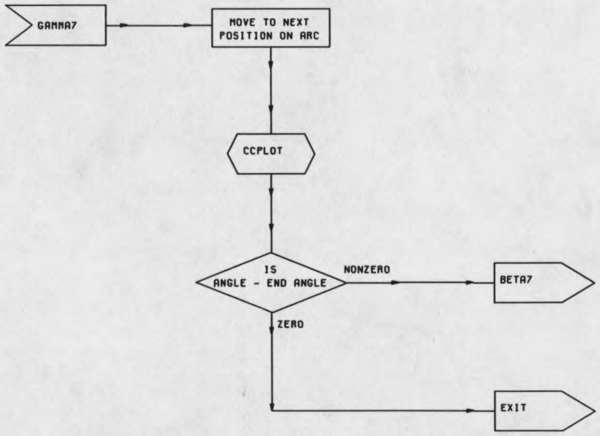
CCIRCLE - SET ANGLE INCREMENT AND SIGN --CCSYSTEM-- 8/29/67 PAGE 48



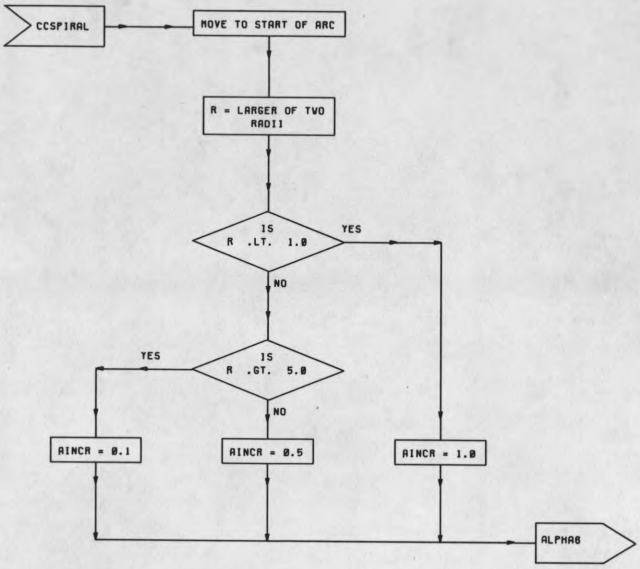
CCIRCLE - CHECK ON PROGRESS OF ARC DRAVING AND SET END TEST --CCSYSTEM- 8/29/67 PAGE 49



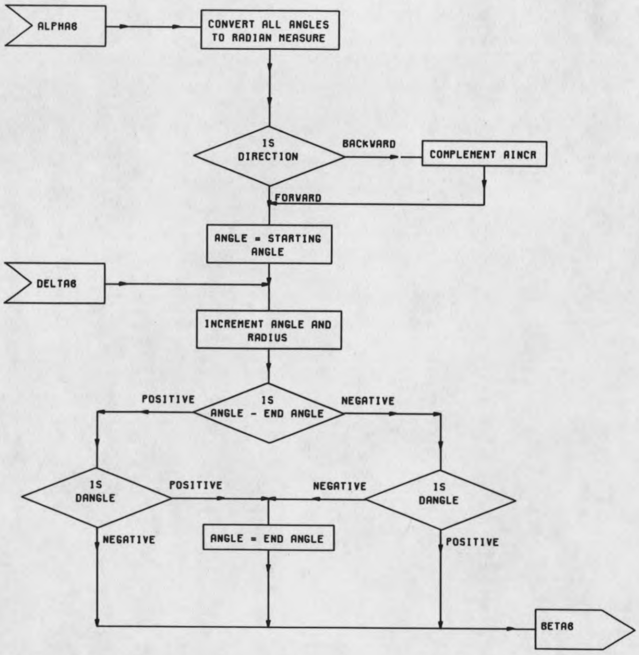
CCIRCLE - DRAV ARC SEGMENT AND TEST FOR END --CCSYSTEM-- 8/29/67 PAGE 50



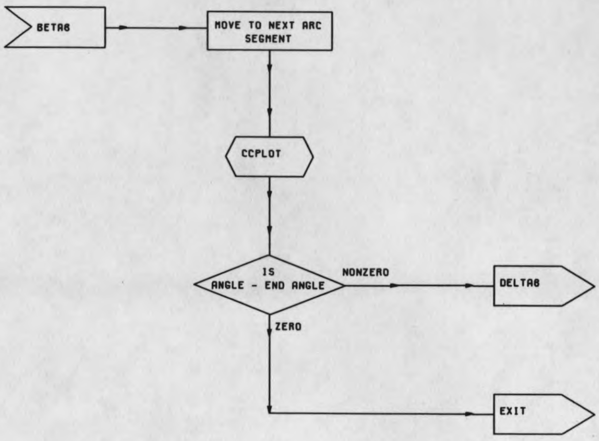
CCSPIRAL - SET UP PROPER ANGLE INCREMENT --CCSYSTEM-- 8/29/67 PAGE 51



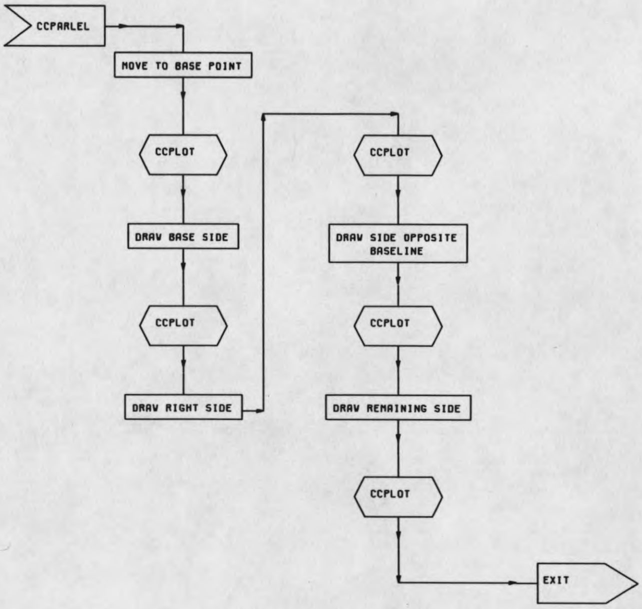
CCSPIRAL - MOVE TO NEXT SEGMENT AND CHECK END TEST --CCSYSTEM-- 8/29/67 PAGE 52



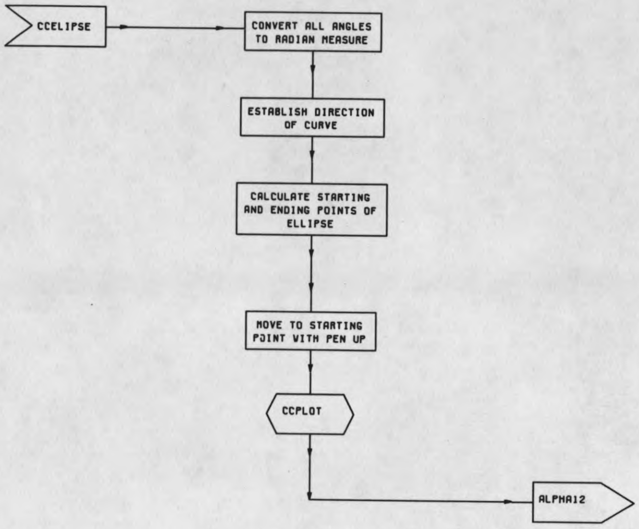
CCSPIRAL - DRAV SEGMENT AND CHECK END TEST --CCSYSTEM-- 8/29/67 PAGE 53



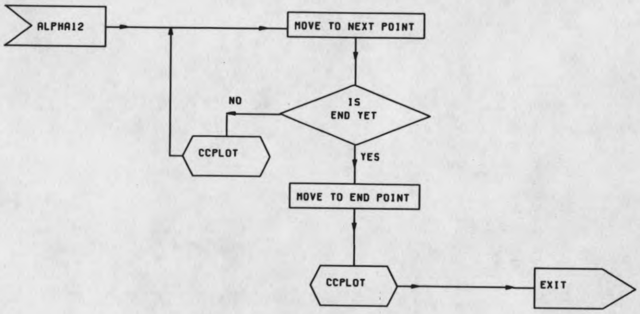
CCPARLEL - DRAV COMPLETE PARALLELOGRAM --CCSYSTEM-- 8/29/67 PAGE 54



CCELIPSE - SET INITIAL CONSTANTS TO DRAW ELLIPSE --CCSYSTEM-- 8/29/67 PAGE 55


CCELIPSE
CONVERT ALL ANGLES TO RADIAN MEASURE
ESTABLISH DIRECTION OF CURVE
CALCULATE STARTING AND ENDING POINTS OF ELLIPSE
MOVE TO STARTING POINT WITH PEN UP
CCPLOT
ALPHA12

CCELIPSE - DRAW PARTIAL OR FULL ELLIPSE --CCSYSTEM-- 8/29/67 PAGE 56



DISTRIBUTION LIST AS OF APRIL 1, 1967

1 Dr. Edward M. Reilley
Asst. Director (Research)
Ofc. of Defense Res. & Engrg.
Department of Defense
Washington, D. C. 20301

1 Office of Deputy Director
(Research and Information Rm. 3D1037)
Department of Defense
The Pentagon
Washington, D. C. 20301

1 Director
Advanced Research Projects Agency
Department of Defense
Washington, D. C. 20301

1 Director for Materials Sciences
Advanced Research Projects Agency
Department of Defense
Washington, D. C. 20301

1 Headquarters
Defense Communications Agency (333)
The Pentagon
Washington, D. C. 20305

50 Defense Documentation Center
Attn: TISIA
Cameron Station, Bldg. 5
Alexandria, Virginia 22314

1 Director
National Security Agency
Attn: TDL
Fort George G. Meade, Maryland 20755

1 Weapons Systems Evaluation Group
Attn: Col. Daniel W. McElwee
Department of Defense
Washington, D. C. 20305

1 National Security Agency
Attn: R4-James Tippet
Office of Research
Fort George G. Meade, Maryland 20755

1 Central Intelligence Agency
Attn: OCR/DD Publications
Washington, D. C. 20505

1 Colonel Kee
AFRSTE
Hqs. USAF
Room 1D-429, The Pentagon
Washington, D. C. 20330

1 Colonel A. Swan
Aerospace Medical Division
Brooks Air Force Base, Texas 78235

1 AUL3T-9663
Maxwell AFB, Alabama 36112

1 AFFTC (FTBPP-2)
Technical Library
Edwards AFB, California 93523

1 Space Systems Division
Air Force Systems Command
Los Angeles Air Force Station
Los Angeles, California 90045
Attn: SSSD

1 Major Charles Waespy
Technical Division
Deputy for Technology
Space Systems Division, AFSC
Los Angeles, California 90045

1 SSD(SSTRT/Lt. Starbuck)
AFUPO
Los Angeles, California 90045

1 Det. #6, OAR (LOOAR)
Air Force Unit Post Office
Los Angeles, California 90045

1 Systems Engineering Group (RTD)
Technical Information Reference Branch
Attn: SEPIR
Directorate of Engineering Standards
& Technical Information
Wright-Patterson AFB, Ohio 45433

1 ARL (ARIY)
Wright-Patterson AFB, Ohio 45433

1 Dr. H. V. Noble
Air Force Avionics Laboratory
Wright-Patterson AFB, Ohio 45433

1 Mr. Peter Murray
Air Force Avionics Laboratory
Wright-Patterson AFB, Ohio 45433

1 AFAL (AVTE/R.D. Larson)
Wright-Patterson AFB, Ohio 45433

2 Commanding General
Attn: STEWS-WS-VT
White Sands Missile Range,
New Mexico 88002

1 RADC (EMLAL-I)
Griffiss AFB, New York 13442
Attn: Documents Library

1 Academy Library (DFSLB)
U. S. Air Force Academy
Colorado Springs, Colorado 80912

1 Lt. Col. Bernard S. Morgan
Frank J. Seiler Research Laboratory
U. S. Air Force Academy
Colorado Springs, Colorado 80912

1 APGC (PGBPS-12)
Elgin AFB, Florida 32542

1 Commanding Officer
Human Engineering Laboratories
Aberdeen Proving Ground, Maryland 21005

1 Director
U. S. Army Engineer Geodesy, Intelligence
and Mapping
Research and Development Agency
Fort Belvior, Virginia 22060

1 Commandant
U. S. Army Command and General Staff College
Attn: Secretary
Fort Leavenworth, Kansas 66270

1 Dr. H. Robl
Deputy Chief Scientist
U. S. Army Research Office (Durham)
Box CM, Duke Station
Durham, North Carolina 27706

1 Commanding Officer
U. S. Army Research Office (Durham)
Attn: CRD-AA-IP (Richard O. Ulsh)
Box CM, Duke Station
Durham, North Carolina 27706

1 Librarian
U. S. Army Military Academy
West Point, New York 10996

1 The Walter Reed Institute of Research
Walter Reed Medical Center
Washington, D. C. 20012

1 Commanding Officer
U. S. Army Electronics R&D Activity
Fort Huachuca, Arizona 85163

1 Commanding Officer
U. S. Army Engineer R&D Laboratory
Attn: STINFO Branch
Fort Belvoir, Virginia 22060

1 Commanding Officer
U. S. Army Electronics R&D Activity
White Sands Missile Range, New Mexico 88002

1 Dr. S. Benedict Levin, Director
Institute for Exploratory Research
U. S. Army Electronics Command
Fort Monmouth, New Jersey 07703

1 Director
Institute for Exploratory Research
U. S. Army Electronics Command
Attn: Mr. Robert O. Parker, Executive
Secretary, JSTAC (AMSEL-XL-D)
Fort Monmouth, New Jersey 07703

1 Commanding General
U. S. Army Electronics Command
Fort Monmouth, New Jersey 07703

Attn: AMSEL-SC
RD-D
RD-G
RD-GF
RD-MAT
XL-D
XL-E
XL-C
XL-S
HL-D
HL-CT-R
HL-CT-P
HL-CT-L
HL-CT-O
HL-CT-I
HL-CT-A
NL-D
NL-A
NL-P
NL-R
NL-S
KL-D
KL-E
KL-S
KL-T
VL-D
WL-D

1 Chief of Naval Research
Department of the Navy
Washington, D. C. 20360
Attn: Code 427

3 Chief of Naval Research
Department of the Navy
Washington, D. C. 20360
Attn: Code 437

2 Naval Electronics Systems Command
ELEX 03
Falls Church, Virginia 22046

1 Naval Ship Systems Command
SHIP 031
Washington, D. C. 20360

1 Naval Ship Systems Command
SHIP 035
Washington, D. C. 20360

2 Naval Ordnance Systems Command
ORD 32
Washington, D. C. 20360

2 Naval Air Systems Command
AIR 03
Washington, D. C. 20360

2 Commanding Officer
Office of Naval Research Branch Office
Box 39, Navy No. 100 F.P.O.
New York, New York 09510

1 AFETR Technical Library
(ETV, MU-135)
Patrick AFB, Florida 32925

1 AFETR (ETLLG-I)
STINFO Officer (For Library)
Patrick AFB, Florida 32925

1 Dr. L. M. Hollingsworth
AFCRL (CRN)
L. G. Hanscom Field
Bedford, Massachusetts 01731

1 AFCRL (CRMXLR)
AFCRL Research Library, Stop 29
L. G. Hanscom Field
Bedford, Massachusetts 01731

1 Colonel Robert E. Fontana
Department of Electrical Engineering
Air Force Institute of Technology
Wright-Patterson AFB, Ohio 45433

1 Colonel A. D. Blue
RTD (RTTL)
Bolling Air Force Base, D. C. 20332

1 Dr. I. R. Mirman
AFSC (SCT)
Andrews AFB, Maryland 20331

1 Colonel J. D. Warthman
AFSC (SCTR)
Andrews AFB, Maryland 20331

1 Lt. Col. J. L. Reeves
AFSC (SCBB)
Andrews AFB, Maryland 20331

2 ESD (ESTI)
L. G. Hanscom Field
Bedford, Massachusetts 01731

1 AEDC (ARO, INC)
Attn: Library/Documents
Arnold AFS, Tennessee 37389

2 European Office of Aerospace Research
Shell Building
47 Rue Cantersteen
Brussels, Belgium

5 Lt. Col. Robert B. Kalisch
Chief, Electronics Division
Directorate of Engineering Sciences
Air Force Office of Scientific Research
Arlington, Virginia 22209

1 U. S. Army Research Office
Attn: Physical Sciences Division
3045 Columbia Pike
Arlington, Virginia 22204

1 Research Plans Office
U. S. Army Research Office
3045 Columbia Pike
Arlington, Virginia 22204

1 Commanding General
U. S. Army Materiel Command
Attn: AMCRD-RS-DE-E
Washington, D. C. 20315

1 Commanding General
U. S. Army Strategic Communications Command
Washington, D. C. 20315

1 Commanding Officer
U. S. Army Materials Research Agency
Watertown Arsenal
Watertown, Massachusetts 02172

1 Commanding Officer
U. S. Army Ballistics Research Laboratory
Attn: V. W. Richards
Aberdeen Proving Ground
Aberdeen, Maryland 21005

1 Commandant
U. S. Army Air Defense School
Attn: Missile Sciences Division, C&S Dept.
P.O. Box 9390
Fort Bliss, Texas 79916

1 Redstone Scientific Information Center
Attn: Chief, Document Section
Redstone Arsenal, Alabama 35809

1 Commanding General
Frankford Arsenal
Attn: SMUFA-1310 (Dr. Sidney Ross)
Philadelphia, Pennsylvania 19137

1 U. S. Army Munitions Command
Attn: Technical Information Branch
Picatinney Arsenal
Dover, New Jersey 07801

1 Commanding Officer
Harry Diamond Laboratories
Attn: Dr. Berthold Altman (AMXDO-TI)
Connecticut Avenue and Van Ness Street, N.W.
Washington, D. C. 20438

1 Commanding Officer
U. S. Army Security Agency
Arlington Hall
Arlington, Virginia 22212

1 Commanding Officer
U. S. Army Limited War Laboratory
Attn: Technical Director
Aberdeen Proving Ground
Aberdeen, Maryland 21005

1 Commanding Officer
Office of Naval Research Branch Office
219 South Dearborn Street
Chicago, Illinois 60604

1 Commanding Officer
Office of Naval Research Branch Office
1030 East Green Street
Pasadena, California 91101

1 Commanding Officer
Office of Naval Research Branch Office
207 West 24th Street
New York, New York 10011

1 Commanding Officer
Office of Naval Research Branch Office
495 Summer Street
Boston, Massachusetts 02210

8 Director, Naval Research Laboratory
Technical Information Officer
Washington, D. C. 20390
Attn: Code 2000

1 Commander
Naval Air Development and Material Center
Johnsville, Pennsylvania 18974

2 Librarian
U. S. Naval Electronics Laboratory
San Diego, California 95152

1 Commanding Officer and Director
U. S. Naval Underwater Sound Laboratory
Fort Trumbull
New London, Connecticut 06840

1 Librarian
U. S. Navy Post Graduate School
Monterey, California 93940

1 Commander
U. S. Naval Air Missile Test Center
Point Mugu, California 95468

1 Director
U. S. Naval Observatory
Washington, D. C. 20390

2 Chief of Naval Operations
OP-07
Washington, D. C. 20350

1 Director, U. S. Naval Security Group
Attn: G43
3801 Nebraska Avenue
Washington, D. C. 20016

2 Commanding Officer
Naval Ordnance Laboratory
White Oak, Maryland 21162

1 Commanding Officer
Naval Ordnance Laboratory
Corona, California 91720

1 Commanding Officer
Naval Ordnance Test Station
China Lake, California 93555

1 Commanding Officer
Naval Avionics Facility
Indianapolis, Indiana 46218

1 Commanding Officer
Naval Training Device Center
Orlando, Florida 32813

1 U. S. Naval Weapons Laboratory
Dahlgren, Virginia 22448

1 Weapons Systems Test Division
Naval Air Test Center
Patuxtent River, Maryland 20670
Attn: Library

1 Head, Technical Division
U. S. Naval Counter Intelligence Support Center
Fairmont Building
4420 North Fairfax Drive
Arlington, Virginia 22203

1 Mr. Charles F. Yost
Special Asst. to the Director of Research
National Aeronautics and Space Administration
Washington, D. C. 20546

1 Dr. H. Harrison, Code RRE
Chief, Electrophysics Branch
National Aeronautics and Space Administration
Washington, D. C. 20546

1 Goddard Space Flight Center
National Aeronautics and Space Administration
Attn: Library C3/TDL
Green Belt, Maryland 20771

1 NASA Lewis Research Center
Attn: Library
21000 Brookpark Road
Cleveland, Ohio 44135

1 National Science Foundation
Attn: Dr. John R. Lehmann
Division of Engineering
1800 G Street, N.W.
Washington, D. C. 20550

1 U. S. Atomic Energy Commission
Division of Technical Information Extension
P. O. Box 62
Oak Ridge, Tennessee 37831

1 Los Alamos Scientific Laboratory
Attn: Reports Library
P. O. Box 1663
Los Alamos, New Mexico 87544

2 NASA Scientific & Technical Information Facility
Attn: Acquisitions Branch (S/AK/DL)
P. O. Box 33
College Park, Maryland 20740

1 Director
Research Laboratory of Electronics
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

1 Polytechnic Institute of Brooklyn
55 Johnson Street
Brooklyn, New York 11201
Attn: Mr. Jerome Fox
Research Coordinator

1 Director
Columbia Radiation Laboratory
Columbia University
538 West 120th Street
New York, New York 10027

1 Director
Coordinated Science Laboratory
University of Illinois
Urbana, Illinois 61801

1 Director
Stanford Electronics Laboratories
Stanford University
Stanford, California 94305

1 Director
Electronics Research Laboratory
University of California
Berkeley, California 94720

1 Director
Electronic Sciences Laboratory
University of Southern California
Los Angeles, California 90007

1 Professor A. A. Dougal, Director
Laboratories for Electronics and Related Sciences Research
University of Texas
Austin, Texas 78712

1 Division of Engineering and Applied Physics
210 Pierce Hall
Harvard University
Cambridge, Massachusetts 02138

1 Aerospace Corporation
P. O. Box 95085
Los Angeles, California 90045
Attn: Library Acquisitions Group

1 Professor Nicholas George
California Institute of Technology
Pasadena, California 91109

1 Aeronautics Library
Graduate Aeronautical Laboratories
California Institute of Technology
1201 East California Boulevard
Pasadena, California 91109

1 Director, USAF Project RAND
Via: Air Force Liaison Office
The RAND Corporation
1700 Main Street
Santa Monica, California 90406
Attn: Library

1 The Johns Hopkins University
Applied Physics Laboratory
8621 Georgia Avenue
Silver Spring, Maryland 20910
Attn: Boris W. Kuvshinoff
Document Librarian

1 Hunt Library
Carnegie Institute of Technology
Schenley Park
Pittsburgh, Pennsylvania 15213

1 Dr. Leo Young
Stanford Research Institute
Menlo Park, California 94025

1 Mr. Henry L. Bachmann
Assistant Chief Engineer
Wheeler Laboratories
122 Cuttermill Road
Great Neck, New York 11021

1 School of Engineering Sciences
Arizona State University
Tempe, Arizona 85281

1 University of California at Los Angeles
Department of Engineering
Los Angeles, California 90024

1 California Institute of Technology
Pasadena, California 91109
Attn: Documents Library

1 University of California
Santa Barbara, California 93106
Attn: Library

1 Carnegie Institute of Technology
Electrical Engineering Department
Pittsburgh, Pennsylvania 15213

1 University of Michigan
Electrical Engineering Department
Ann Arbor, Michigan 48104

1 New York University
College of Engineering
New York, New York 10019

1 Syracuse University
Department of Electrical Engineering
Syracuse, New York 13210

1 Yale University
Engineering Department
New Haven, Connecticut 06520

1 Airborne Instruments Laboratory
Deerpark, New York 11729

1 Bendix Pacific Division
11600 Sherman Way
North Hollywood, California 91605

1 General Electric Company
Research Laboratories
Schenectady, New York 12301

1 Lockheed Aircraft Corporation
P. O. Box 504
Sunnyvale, California 94088

1 Raytheon Company
Bedford, Massachusetts 01730
Attn: Librarian

1 Dr. G. J. Murphy
The Technological Institute
Northwestern University
Evanston, Illinois 60201

1 Dr. John C. Hancock, Director
Electronic Systems Research Laboratory
Purdue University
Lafayette, Indiana 47907

1 Director
Microwave Laboratory
Stanford University
Stanford, California 94305

1 Emil Schafer, Head
Electronics Properties Info Center
Hughes Aircraft Company
Culver City, California 90230

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* University of Illinois Coordinated Science Laboratory Urbana, Illinois 61801	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

CCSYSTEM Calcomp Plotter Computer Programming System

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

5. AUTHOR(S) *(First name, middle initial, last name)*

Bouknight, Jack

6. REPORT DATE September, 1967	7a. TOTAL NO. OF PAGES 50	7b. NO. OF REFS 0

8a. CONTRACT OR GRANT NO. DAAB-07-67-C-0199 b. PROJECT NO.	9a. ORIGINATOR'S REPORT NUMBER(S) R-366
c. d.	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT

DISTRIBUTION OF THIS REPORT IS UNLIMITED.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Joint Services Electronics Program thru U.S. Army Electronics Command Fort Monmouth, New Jersey 07703

13. ABSTRACT

A complete description is given of a programming package written at CSL for the CDC 1604 computer system allowing usage of the California Computer Products (CALCOMP) digital incremental plotter system for a graphical output medium.

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Incremental Plotter						
Graphical Output						
Software						