X-692-71-378
PREPRINT

NASA TM X- 65708

PLOTTER ROUTINES FOR THE IBM 1800 COMPUTER

T. P. CARLETON

SEPTEMBER 1971

GSFC — GODDARD SPACE FLIGHT CENTER —
GREENBELT, MARYLAND

PLOTTER ROUTINES FOR THE IBM 1800 COMPUTER

by

T. P. Carleton

Laboratory for Extraterrestrial Physics
NASA - Goddard Space Flight Center
Greenbelt, Maryland

CONTENTS

Introduction

This document describes subroutines which have been written to facilitate use of the IBM 1627 plotter associated with an IBM 1800 computer. These subroutines have been implemented under the Multiprogramming Executive Operating System (MPX) and all information pertains to the MPX versions.

The programs in the first section are designed to permit graphic presentation of data with a minimum of programming effort. Most standard types of plots can be produced by calling one of the programs in this section. Options are included for linear or logarithmic tickmarks or grids, multi-color plotting, the tracing of several curves or sets of data points on the same base, and histogram plotting.

The subroutines of the second section provide the basic tools for those who wish to design their own special purpose plots. Included are routines for controlling pen movement in terms of inches or user defined units, writing alphabetic or numeric text, plotting special symbols, changing the scale of plots, and isometric presentation of three dimensional data.

The third section describes special subroutines used by the programs of the first two sections. These are included primarily for informational purposes since they will be of relatively little utility to the average user.

The fourth section discusses certain other available plotting routines.

Three appendices contain tables giving the core storage requirements of each routine, a list of the subroutines required by each, and listings of the routines themselves.

The table of contents has been designed to give the user a quick overview of what is available by including a brief description of the purpose of each routine.

Notation

The standard FORTRAN conventions for variable names are used throughout this document. Variables whose names begin with the letters I through N are one-word integers, and those names begin with the letters A through H or O through Z are single precision (two-word) floating point numbers.

In the FORTRAN calling statements for the various subroutines, the parameter lists may contain either variable names or appropriate numerical values. The following two calling sequences, for example, are equivalent.

```
(1)     X=10.0
        Y=3.32
        I=1
        CALL MOVE(X,Y,I)

(2)     CALL MOVE(10.0,3.32,1)
```

I. One-call plot routines

The routines of this section have been designed to construct the most commonly used types of plots with a minimum of programming effort.

HISTO CALL HISTO(Y,N,XSIZ,YSIZ,YL,YU,IBAR)

Subroutine HISTO plots N values of floating point array Y in the form of a histogram inside a rectangular frame of XSIZ by YSIZ inches. The N bars are of equal width. YL and YU are respectively the user assigned lower and upper bounds of the data. Bars for values which exceed these limits will only be plotted to the appropriate boundary, and a message noting the error will be printed on the typewriter (cf. PLERR.)

If YL<0<YU, a horizontal line at Y = 0 will be drawn across the plotting area. The bars of the histogram will be drawn as if extending between Y = 0 and Y = Y_i, whether Y = 0 lies within the plotting area or not. (see figure 1 (a).) If IBAR = 1 the individual bars will be separated by lines extending towards Y = 0 (figure 1 (b)), if IBAR = 0 these lines will be omitted (figure 1(c).)

At the end of plotting the origin is reset at a point even with the bottom boundary of the last plot and two inches to the right of its right edge. This permits successive calls to HISTO to be plotted on separate areas.

QUICK CALL QUICK(X,Y,N,XS,YS,XL,XU,YL,YU,ISYM,SS,LINE,IR,IP)

Subroutine QUICK plots N points whose coordinates, in user's units, are stored in floating point arrays X and Y. The plot is contained in a rectangular area of XS by YS inches. XL and XU are respectively the user-assigned lower and upper boundaries of the X data, and YL and YU those of the Y data. X or Y values which exceed the assigned limits will be plotted on the appropriate boundary, and an error message will be printed on the typewriter (see PLERR.) If ISYM is a number between one and eight the corresponding symbol (see subroutine SYMBL) will be plotted at each point. If ISYM is zero no symbol will be drawn. SS is the size of the symbol, if any, in inches and must be an integral multiple of 0.04" and must not be less than 0.04". If LINE equals zero a line will be traced between consecutive points, if equal to one no line will be drawn.

If IR equals zero, indicating an initial plot, a rectangular frame of XS by YS inches will be drawn before plotting the data, while a value of one will cause the data alone to be plotted over the area. If either the X or Y axes fall within the plotting area (i.e., if XL < 0 < XU or YL < 0 < XU,) they will be drawn. For example the X axis (Y = 0) has been plotted in figure 2. If tickmarks or full line divisions

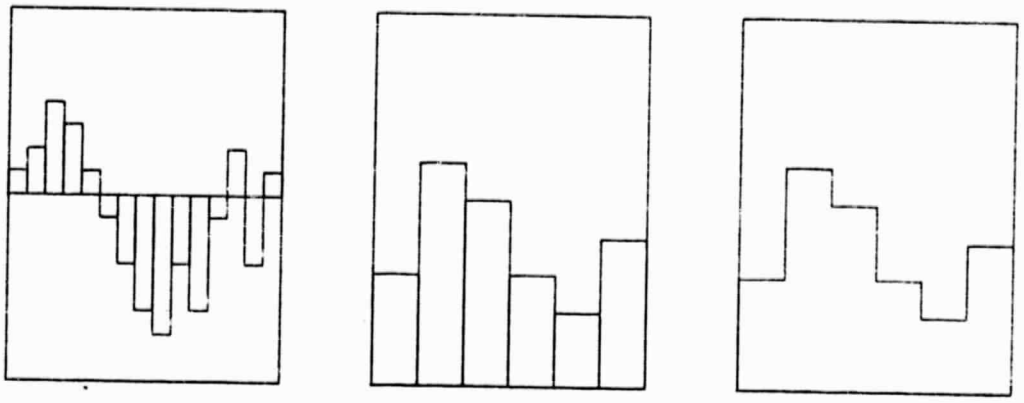


Figure 1. HISTO plots

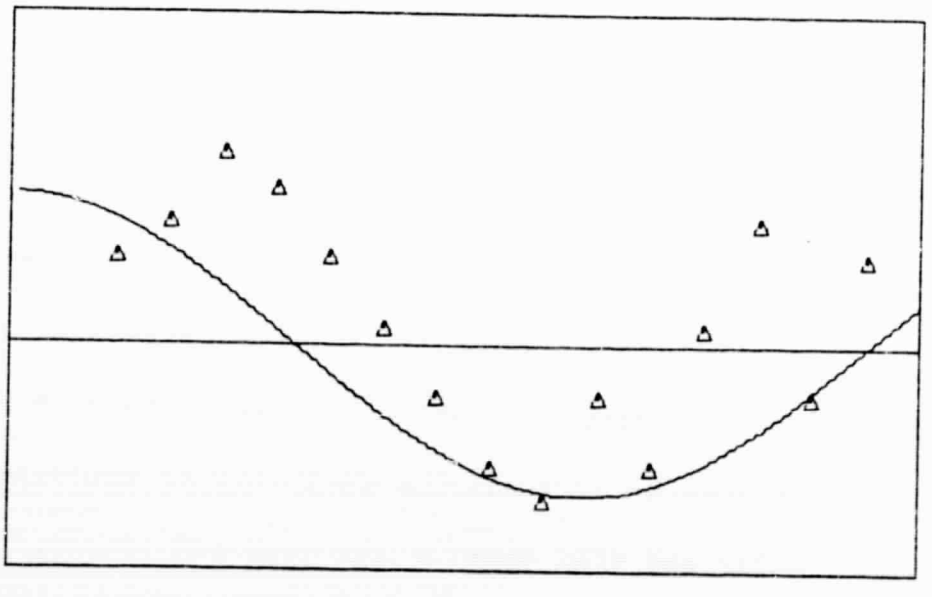


Figure 2. QUICK plot.

are desired subroutine GRID may be used, followed by one or more calls to QUICK with IR = 0.

A value of one for IP will cause a pause before plotting for paper adjustment or pen change. Console START must be pressed to continue execution. If IP is zero no pause will be made.

If two or more separate plots are desired subroutine MOVE may be used to reposition the pen. The next call to QUICK will then reestablish the origin at the new pen location.

The plot shown in figure 2 was produced by the following two calls to QUICK, where X1 and Y1 contain the coordinates of the 100 points determining the solid line, X2 and Y2 the 15 points marked with a triangular symbol, and XL1, XU1, etc. define the ranges of the corresponding data.

```
CALL QUICK (X1,Y1,100,5.0,3.0,XL1,XU1,YL1,YU1,0,0.0,0,0,0)
CALL QUICK (X2,Y2,15,5.0,3.0,XL2,XU2,YL2,YU2,5,0.08,1,1,0)
```

GRID CALL GRID (XSIZE,YSIZE,IXTYP,IYTYP,XINC,YINC)

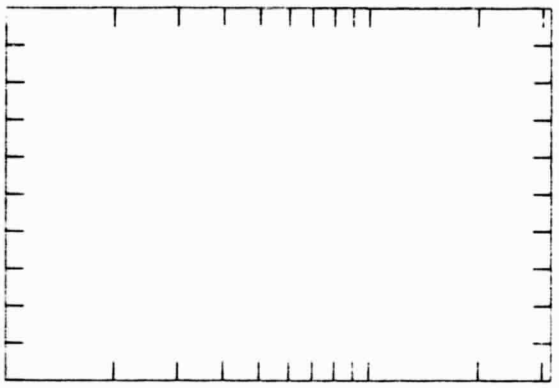
GRID draws a rectangle of XSIZE by YSIZE inches, subdivided by tickmarks or full-line divisions as explained in Table 1. IXTYP and XINC specify the type and interval of division along the X axis, while IYTYP and YINC specify type and interval on the Y axis.

IXTYP (IYTYP)	XINC (YINC)
1 Tickmarks inside rectangle	Interval in inches between equally spaced tickmarks or lines.
2 Tickmarks outside rectangle	
3 Full lines across rectangle	
4 Tickmarks inside rectangle	Length in inches of one complete logarithmic cycle (1 to 10) of tickmarks or lines.
5 Tickmarks outside rectangle	
6 Full lines across rectangle	

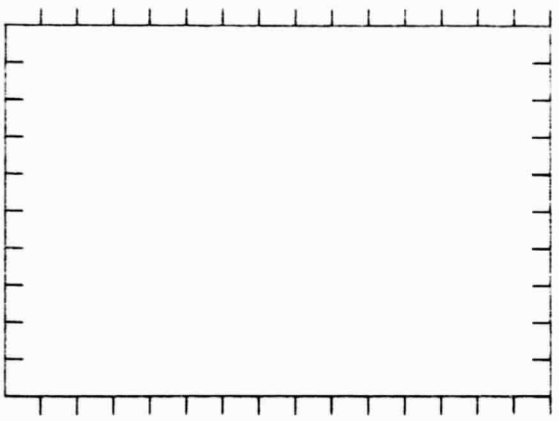
Table I.

All types of divisions can be used on either axis, and the two axes are independent. The program listed below produced figure 3, (a) to (f) in succession. These are only six of many possible combinations.

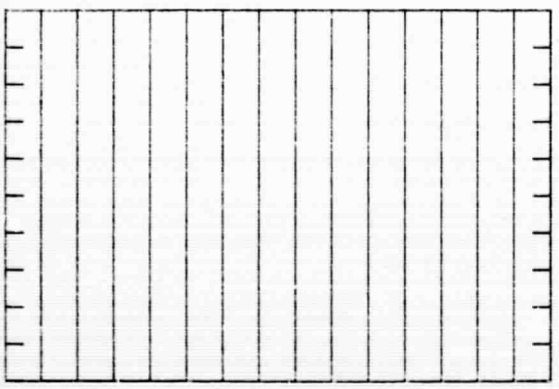
```
// JOB    27 JUL 71 23.596 HRS
// FOR TEST    27 JUL 71 23.596 HRS
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
*NONPROCESS PROGRAM
      CALL GRID(2.0,3.0,1,4,0.2,2.0)
      CALL MOVE(3.0,0.0,-1)
      CALL GRID(2.0,3.0,1,2,0.2,0.2)
      CALL MOVE(3.0,0.0,-1)
      CALL GRID(2.0,3.0,1,3,0.2,0.2)
      CALL MOVE(-6.0,4.0,-1)
      CALL GRID(2.0,3.0,3,3,0.2,0.4)
      CALL MOVE(3.0,0.0,-1)
      CALL GRID(2.0,3.0,3,6,0.2,2.0)
      CALL MOVE(3.0,0.0,-1)
      CALL GRID(2.0,3.0,6,6,2.0,1.0)
      CALL EXIT
      END
```



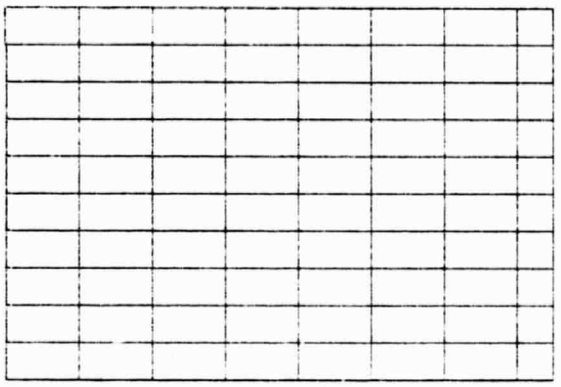
(a)



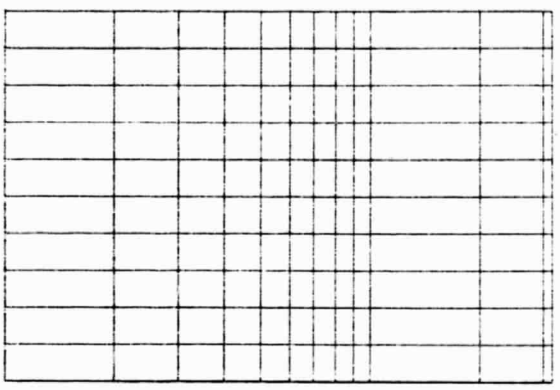
(b)



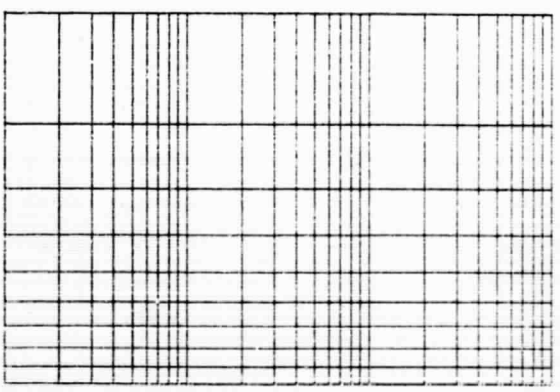
(c)



(d)



(e)



(f)

Figure 3. GRID routine plots.

II. General Purpose Subroutines

The subroutines described in this section are designed to facilitate the creation of special purpose plots. It should be noted that the routines SCAL and FACTR affect the operation of the other subroutines of this section.

SCAL CALL SCAL (XL,XU,XSIZE,YL,YU,YSIZE)

Subroutine SCAL must be called before the routines PLOT or PLOTA are used. This subroutine sets the origin at the current pen location and defines a plotting area in terms of user units. The location of the origin may later be changed if desired by use of the subroutine MOVE. Such a change will not affect the relation of user units to inches.

XL - Value of X at origin in user units.

XU - Values of X at XSIZE inches from the origin in user's units.

XSIZE - Extent of the basic plotting area along the X axis in inches. (Note: This definition and that of YSIZE do not prevent plotting outside of the basic area defined; i.e., no comparison of the data with limits is made in the routines of this section.)

XL - Value of Y at origin in user's units.

YU - Value of Y at YSIZE inches from the origin in user's units.

YSIZE - Extent of the basic plotting area along the Y axis in inches.

Example Given a set of X data with values ranging between 200 and 400, and a corresponding set of Y values ranging between -0.5 and 0.5 to be plotted in an 8" x 10" area, the initializing call would be:

```
CALL SCALE (200.0,400.0,8.0,-0.5,0.5,10.0)
```

FACTR CALL FACTR(XFAC,YFAC)

After a call to FACTR all pen movements in the X direction will be multiplied by the scaling factor XFAC, and all Y movements by YFAC. A call to FACTR affects all subroutines of this and the preceding section; i.e., HISTO, QUICK, GRID, PLOT, PLOTA, MOVE, GRID, SYMBL, ALPHA, DECD, TRAN, CIRCL, and PLOT3. Where successive calls are used, XFAC and YFAC are always relative to the original, or true, scale, rather than being cumulative. A call of CALL FACTR(1.0,1.0) will always reestablish the original scale.

PLOT CALL PLOT(X,Y,I)

Subroutine PLOT moves the pen along a straight line to the point whose coordinates are X and Y in user's units. The pen will be down (i.e., a line will be drawn) if I = 0 and will be raised if I = 1.

PLOTA CALL PLOTA(X,Y,N)

X and Y are Fortran arrays of coordinates expressed in user's units. Subroutine PLOTA traces a segmented line, starting at the current pen location and moving to each of the N points (X(I),Y(I),I=1,N) in succession.

MOVE CALL MOVE(X,Y,I)

Subroutine MOVE moves the pen to a point (X,Y) inches from the origin. If I = 0 the pen will be lowered (i.e., a line drawn), if I = 1 the pen will be raised, and if I = -1 the pen will be raised and the origin reestablished at the new pen location.

TGRID CALL TGRID(XSIZE,YSIZE,XINC,YINC)

TGRID draws a rectangle of XSIZE by YSIZE inches, with tickmarks at successive intervals of XINC inches from the origin along the boundaries parallel to the X axis, and at intervals of YINC inches along those parallel to the Y axis. A rectangle or frame without tickmarks may be drawn if desired by setting XINC equal to XSIZE, and YINC to YSIZE.

SYMBL CALL SYMBL(I,S)

Subroutine SYMBL causes a symbol selected by the value of I to be plotted, centered at the current pen location. S is the diameter of the symbol in inches. S must be an integral multiple of 0.04", otherwise loss of origin may

occur. The symbols corresponding to various values of I are shown in figure 4. If I is less than 1 or greater than 8 a return will be made to the calling routine without a symbol being plotted, and an error message will be printed on the typewriter (see PLERR).

I	1	2	3	4	5	6	7	8
symbol	X	+	□	◇	△	▽	◁	▷

Figure 4. SYMBL output.

ALPHA CALL ALPHA(N,IA,HGHT,ANGLE)

Subroutine ALPHA plots a string of alphanumeric or special characters with user specified size and angle to the X axis. N is the number of characters to be plotted. IA is the name of a FORTRAN integer array containing the code numbers corresponding to these characters (see table II.) HGHT is the character height in inches. The minimum value of HGHT is 0.07" and best results are obtained if HGHT is an integral multiple of this figure. ANGLE is the clockwise angle between the line of characters and the plotter X axis expressed in radians.

Character plotting commences at the current pen position when the call to ALPHA is made, and the pen is returned to this position at the end of plotting. As an example, the coding shown in figure 5 would cause the message "TIME 1"

to be plotted in 0.21" letters at an angle of 45 deg. to the X axis.

character	code	character	code	character	code
A	1	Q	17	5	32
B	2	R	18	6	33
C	3	S	19	7	34
D	4	T	20	8	35
E	5	U	21	9	36
F	6	V	22	blank	37
G	7	W	23	,	48
H	8	X	24	.	40
I	9	Y	25	+	39
J	10	Z	26	-	45
K	11	∅	27	*	41
L	12	1	28	/	46
M	13	2	29	=	51
N	14	3	30	(	38
O	15	4	31	)	42
P	16				

Table II. ALPHA arguments

If an illegal character code (i.e., less than zero or greater than 51,) is found in the array, an error message will be printed (see PLERR) and that character ignored.

```
DIMENSION IA(6)
DATA IA/20,9,13,5,37,28/
  .
  .
CALL ALPHA (6,IA,0.21,0.7854)
```

Figure 5. Call to ALPHA

DECD CALL DECD(F,HGHT,ANGLE)

Subroutine DECD causes the integral portion of the floating point number F to be plotted. This integral portion may be no larger than 6 digits. If it contains less than six significant digits, leading zeroes will be added. HGHT is the character height in inches and should be an integral multiple of 0.07". ANGLE is the clockwise angle between the number and the X axis, expressed in radians.

TRAN CALL TRAN(N,IA,HGHT,ANGLE)

Subroutine TRAN causes the EBCDIC contents (i.e., "A" format) of the Fortran integer array IA to be plotted. This subroutine makes it unnecessary for the programmer to code his message numerically as described under subroutine ALPHA. N is the number of characters, and must not exceed 64. HGHT is the character height in inches and should be an integral multiple of 0.07". ANGLE is the clockwise angle between the message and the plotter X axis, expressed in radians.

CIRCL CALL CIRCL(X,Y,R)

CIRCL draws a circle whose radius in inches is R, centered at the point whose coordinates in inches are X and Y.

SCAL3 CALL SCAL3(XL,XU,XSIZ,YL,YU,YSIZ,ZL,ZU,ZSIZ,XANG,ZANG)

Subroutine SCAL3 performs the same function for PLOT3 which SCAL does for PLOT, and must be called before PLOT3 is used. It establishes the origin at the current pen location, and defines scaling factors which permit the isometric representation of a 3-dimensional point to be located by giving its three coordinates in user units. Figure 6 shows the relationship of the various parameters to the coordinate system.

XL,YL,ZL - Values of X,Y,Z at origin in user units.

XU,YU,ZU - Values of X,Y,Z at XSIZ,YSIZ,ZSIZ inches from origin in user units.

XSIZ,YSIZ,ZSIZ - Extent of the basic plotting area along the X,Y,Z axes in inches. (Note: These definitions do not prevent plotting outside of the basic area defined; i.e., no comparison of the data with any limits is made in these routines.)

XANG, ZANG - Angles in degrees which the plotted representations of the X and Z axes make with the plotter X axis (see figure 6.)

The location of the origin may be changed by a new call to SCAL3 after the pen has been moved to the desired location.

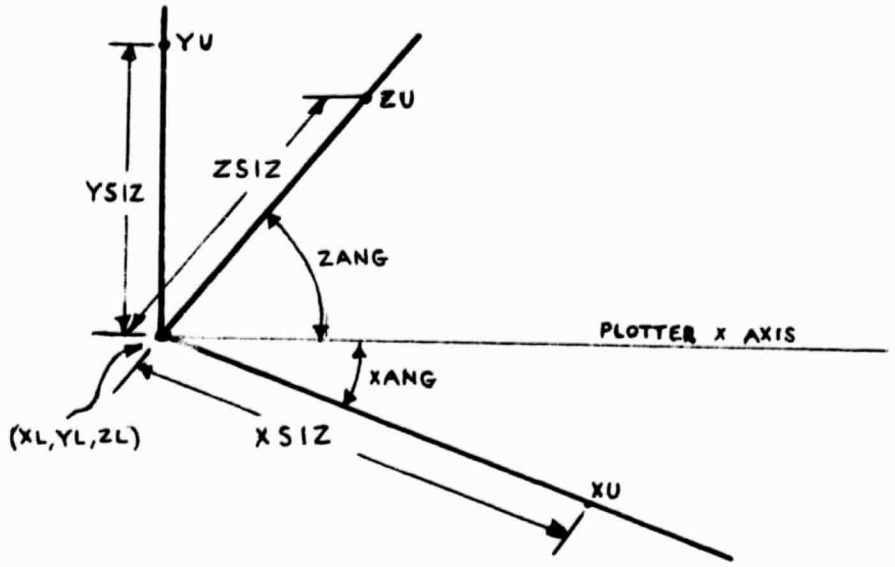


Figure 6. SCAL3 parameters

Note that this group of 3-dimensional plot routines does not share the same origin or coordinate system as the other plot routines, and they cannot be used together without extreme care.

PLOT3 CALL PLOT3(X,Y,Z,I)

Subroutine PLOT3 moves the pen to the isometric representation of the 3-dimensional point (X,Y,Z) in the coordinate system initialized by SCAL3 (see figure 7.) If I = 0 the pen will be down (i.e., a line will be drawn) and if I = 1 the pen will be raised.

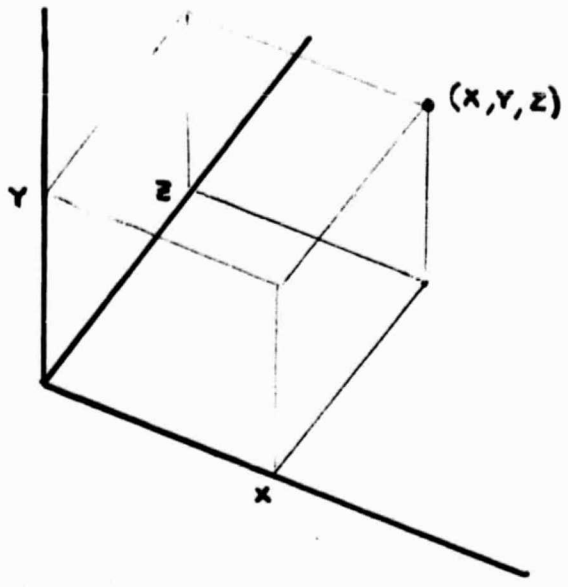


Figure 7. Isometric representation of point (X,Y,Z)

III. System subroutines

The subroutines described in this section are used by those of the previous sections in performing their functions. They are included mainly for the sake of completeness since the majority of programming problems can be solved more simply and flexibly with the previously described routines.

PLERR CALL PLERR(N)

Subroutine PLERR causes an error warning message to be printed on the typewriter. The message format is shown in figure 8. The integer N is printed in hexadecimal form as the first of the group of six numbers (0063 or 99_{10} in figure 8) and indicates the source of the message. The

following numbers indicate errors in certain routines:

1 - Illegal parameter in call to SYMBL. Call is ignored.

2 - Data out of specified limits in HISTO. Data bar will be plotted only to the boundary.

3 - Data out of specified limits in QUICK. Data point will be plotted on boundary.

7 - Illegal character in call to ALPHA. The individual character will be ignored.

```
  00.421
PLOT   ERROR NO. ******  0063  0000  0000  0000  0000  0000  //
```

Figure 8. PLERR message

PIKUP — CALL PIKUP(XF,YF)

A call to PIKUP sets XF and YF equal respectively to the current X and Y expansion factors (cf. FACTR.)

BRING/BITS

```
CALL  BRING      CALL  BITS
DC     addr      DC    n
DC     bit                    1 ≤ n ≤ 16
```

The two entry subroutine BRING/BITS allows the retrieval of data which has been packed in contiguous groups of one to sixteen bits. Calls may only be made from assembly language routines.

The subroutine is initialized by a call to BRING, where 'addr' is the absolute word address and 'bit' the bit number

at which the data retrieval process is to begin. The first call to BITS, followed by a positive data constant less than or equal to sixteen, will bring that number of bits into the accumulator, starting at the specified address. Each successive call fetches the next sequential group of bits, whose count is determined by the calling parameter. Word boundaries are ignored.

This routine is used by both ALPHA and SYMBL. In both of these a pen command is stored as two four bit groups which determine the direction and magnitude of the pen movement, followed by a single bit indicating whether the pen is to be up or down. These nine bit command groups are packed contiguously in data arrays at the end of each routine.

PLOTX

```
        CALL PLOTX
        DC   LIST

LIST    DC   *-*      completion code
        DC   /000n
```

PLOTX transmits a command determined by the value of n, the fourth hexadecimal digit of the second control parameter. This command may put the plotter on or off-line, raise or lower the pen, or cause the pen to be moved one plotter increment (0.01") along the X or Y axes or diagonally in a combination X and Y movement. Table III lists the commands corresponding to each value. For further information on this routine see reference 1.

Value	Plotter command
0	Pen down
1	Line segment +Y
2	" " +X,+Y
3	" " +X
4	" " +X,-Y
5	" " -Y
6	" " -X,-Y
7	" " -X
8	" " -X,+Y
9	Pen up
A	Take plotter off line
B	Put plotter on line
C-F	Not used

Table III. PLOTX commands

LINE/SKIP CALL LINE(IX,IY)

CALL SKIP(IX,IY)

A call to subroutine LINE causes the plotter to draw the closest approximation of a straight line from the current pen location to a given point. IX and IY are the integer X and Y coordinates of this point in hundredths of an inch, with the starting point defined as the origin. A call to

SKIP causes the plotter to move to the specified point with the pen raised. The straight line approximation is determined by the following algorithm.

Straight line plotting algorithm.

The plotter moves vertically and horizontally in steps of 0.01". There are eight possible plotter movements; vertical up or down, horizontal right or left, and the four diagonal combinations of horizontal and vertical movement (see figure 9.)

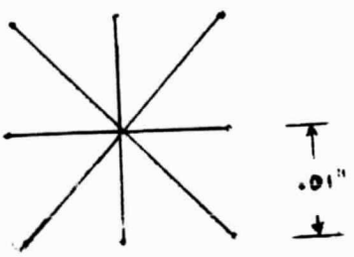


Figure 9. Plotter incremental movements.

It is assumed that the closest approximation to a directed line segment can be formed using the two plotter movements which define the octant in which the segment lies. A line segment to be plotted will be defined as a vector with two integer components; the number respectively of horizontal and vertical 0.01" steps necessary to reach the end point of the segment.

Suppose that P is a line segment with components m and n, and that $n \geq m$. Then the closest approximation to P

is defined by a series of n points. The coordinates of the ith point will be i in the direction of the larger component, and

$$\left[\frac{2im + n}{2n}\right]$$

in the direction of the smaller, where the square brackets represent integer truncation of the quotient. The second coordinate is the value of the integer nearest to i times the quotient m/n. Figure 10 shows the example P = (3,5).

i	$X_i = \left[\frac{2i(3) + 5}{10}\right]$	$Y_i = i$
1	[11/10] = 1	1
2	[17/10] = 1	2
3	[23/10] = 2	3
4	[29/10] = 2	4
5	[35/10] = 3	5

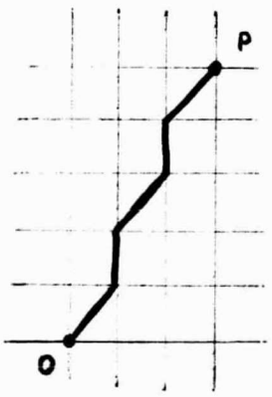


Figure 10. Approximation of the line P = (3,5)

Each of the i successive points has its coordinates computed in this manner. The pen movement for the ith step is then the difference between the coordinates of that point and

those of the previous point:

$$(X_i - X_{i-1}, Y_i - Y_{i-1}).$$

The successive commands for the example of figure would be:

(1,1) (0,1) (1,1) (0,1) (1,1).

IV. Other Plotting routines

CCOMP

The program CCOMP produces a IBM 1637 plot from a Cal Comp 77C plotter tape written on the IBM 360/75 J computer. The plot is the nearest approximation to that which would be drawn on the 770 consistent with the difference in resolution of the two plotters. Ordinarily very little perceptible distortion occurs.

The calling sequences for this program is given below. The fourth card contains the numbers of the first and last CalComp files to be plotted, in 2I6 format. The CalComp file numbers are three digits integers, and 999 is generally used to end the tape. A detailed description of this program will be given in a document now in preparation.

```
// JOB
// XEQ CCOMP
*  CCEND          four cards
     1    999
```

TGRAF and PGRAF

TGRAF and PGRAF produce X-Y plots on the 1053 typewriter and 1443 printer respectively. Their operation is described in detail in reference 2.

Acknowledgements

The author wishes to thank Mr. William Mish for advice and for the contribution of several of the routines contained in this document.

References

1. "IBM 1800 Multiprogramming Executive Operating System Subroutine Library", IBM Systems Reference Library, GC26 - 3724 - 2
2. Carleton, T.P., TGRAF and PGRAF Typewriter and Line Printer Plotting Routines for the IBM 1800 Computer, NASA-GSFC X-692-70-72.

APPENDIX A. SUBROUTINE STORAGE REQUIREMENTS

Name	words
HISTO	252
QUICK	294
GRID	484
SCAL/FACTR/PIKUP/ MOVE/PLOT	204
PLOTA	58
TGRID	280
SYMBL	180
ALPHA	398
DECD	122
TRAN	192
CIRCL	242
SCAL3/PLOT3	252
PLERR	42
BRING/BITS	52
PLOTX	277
LINE/SKIP	132
PGRAF	392
TGRAF	390

APPENDIX B. REQUIRED SUBROUTINES

Name	Subroutines used
HISTO	SUBIN, SCAL, FLD, LDFAC, SUBSC, FADD, PLERR, FMPY, FDIV, SNR, LINE, FARC, PLOTX
QUICK	SUBIN, SCAL, PAUSE, FLD, LDFAC, SUBSC, SYMBL, PLERR, FMPY, FDIV, SNR, FADD, LINE, FARC, BRING, PLOTX
GRID	SUBIN, FLD, SUBSC, FMPY, LDFAC, MOVE, FARC, FDIV, SNR, FADD, LINE, PLOTX
SCAL/FACTR/PIKUP/ MOVE/PLOT	FMPY, FDIV, SNR, FADD, LINE, FLD, FARC, PLOTX
PLOTA	SUBIN, SUBSC, FMPY, FDIV, SNR, FADD, LINE, FLD, FARC, PLOTX
TGRID	SUBIN, MOVE, FLD, LDFSC, FMPY, FADD, LINE, FDIV, SNR, FARC, PLOTX
SYMBL	PIKUP, FLD, FDIV, FMPY, BRING, FADD, LINE, PLERR, SNR, FARC, PLOTX
ALPHA	FSIN, FMPY, FDIV, PIKUP, BRING, SNR, FADD, LINE, PLERR, XMDS, FARC, FLD, PLOTX
DECD	SUBIN, FABS, FLD, FAXI, FDIV, FMPY, SUBSC, ISTOX, LDFAC, ALPHA, FARC, FSIN, PIKUP, BRING, SNR, FADD, LINE, PLERR, XMDS, PLOTX
TRAN	ALPHA, FSIN, FMPY, FDIV, PIKUP, BRING, SNR, FADD, LINE, PLERR, XMDS, FARC, FLD, PLOTX
CIRCL	SUBIN, FLD, FADD, MOVE, SUBSC, FMPY, FARC, FDIV, SNR, LINE, PLOTX
PLERR	none
BRING/BITS	none
PLOTX	none
LINE/SKIP	PLOTX

Name	Subroutines used
SCAL3/PLOT3	FADD, FMPY, DFIV, SNR, FSIN, FLD, PIKUP, LINE, FARC, XMDS, PLOTX
PGRAF	EBPRT, PRT, IPCNN
TGRAF	EBPRT, PRT, ITCNN

APPENDIX C

Subroutine Listings

```
      SUBROUTINE QUICK(X,Y,N,XSIZ,YSIZ,XL,XU,YL,YU,ISYM,SSIZ,LINE,IR,IP)
      DIMENSION X(1),Y(1)
      IF(IP)16,16,17
   17 PAUSE
   16 CALL SCAL(XL,XU,XSIZ,YL,YU,YSIZ)
      IF(IR)12,19,12
   19 TT=XSIZ
      IF(TT-5.0)40,40,41
   41 TT=5.0
   40 CALL MOVE(TT,0.0,0)
      CALL MOVE(0.0,0.0,1)
      CALL MOVE(XSIZ, 0.0,0)
      CALL MOVE(XSIZ,YSIZ,0)
      CALL MOVE( 0.0,YSIZ,0)
      CALL MOVE( 0.0, 0.0,0)
      IF(XL)6,8,8
    6 IF(XU)8,8,7
    7 CALL PLOT(0.0,YL,1)
      CALL PLOT(0.0,YU,0)
    8 IF(YL)9,12,12
    9 IF(YU)12,12,10
   10 CALL PLOT(XL,0.0,1)
      CALL PLOT(XU,0.0,0)
   12 IE=0
      DO 13 I=1,N
      L=LINE
      IF(I-1)30,30,31
   30 L=1
   31 XP=X(I)
      YP=Y(I)
      IF(XP-XU)21,21,20
   20 XP=XU
      GO TO 23
   21 IF(XL-XP)24,24,22
   22 XP=XL
   23 IE=1
   24 IF(YP-YU)26,26,25
   25 YP=YU
      GO TO 28
   26 IF(YL-YP)29,29,27
   27 YP=YL
   28 IE=1
   29 CALL PLOT(XP,YP,L)
      IF(ISYM)13,13,14
   14 CALL SYMBL(ISYM,SSIZ)
   13 CONTINUE
      IF(IE)33,33,32
   32 CALL PLERR(3)
   33 CALL MOVE(0.0,0.0,1)
      RETURN
      END
```

```
      SUBROUTINE GRID(XLIM,YLIM,IXTYP,IYTYP,XINC,YINC)
      DIMENSION FLOG(9),IPEN(3),TICK(3)
      DATA IPEN/0,0,1/,TICK/0.1,-0.1,0.0/
      DATA FLOG/0.301,0.477,0.602,0.699,0.778,0.845,0.903,0.954,1.0/
C
C       SUBR. EGRID  8/18/70    PROGRAMMER - T. CARLETON
C       THIS SUBROUTINE WILL  DRAW A RECTANGLE OF SIZE XLIM (IN.) BY YLIM (IN.).
C       THIS WILL BE DIVIDED IN THE X AND Y DIRECTIONS AS SPECIFIED BY THE INTE-
C       GER PARAMETERS IXTYP AND IYTYP RESPECTIVELY ACCORDING TO THE FOLLOWING
C       TABLE.
C            1 - TICKMARKS INSIDE RECTANGLE DRAWN AT INTERVALS OF XINC(YINC)
C                INCHES STARTING AT ORIGIN,
C            2 - TICKMARKS OUTSIDE RECTANGLE AT SAME INTERVAL AS ABOVE.
C            3 - FULL LINES DRAWN ACROSS RECTANGLE AT SAME INTERVAL AS ABOVE.
C            4 - TICKMARKS INSIDE RECTANGLE WITH LOGARITHMIC CYCLE OF XINC(YINC)
C                INCHES STARTING AT ORIGIN.
C            5 - OUTSIDE TICKMARKS WITH LOGARITHMIC DIVISION AS ABOVE.
C            6 - FULL LINES WITH LOGARITHMIC DIVISIONS AS ABOVE.
C
      I2=(IXTYP-1)/3
      I1=IXTYP-3*I2
      I4=(IYTYP-1)/3
      I3=IYTYP-3*I4
      TICK(3)=YLIM
      XT=TICK(I1)
      TICK(3)=XLIM
      YT=TICK(I3)
    8 L1=0
      L2=0
   10 L1=L1+1
      IF(I2)14,14,11
   11 IF(L1-10)13,12,12
   12 L1=1
      L2=L2+1
   13 X=XINC*(L2+FLOG(L1))
      GO TO 15
   14 X=L1*XINC
   15 IF(X-XLIM)16,17,17
   16 CALL MOVE(X,0.0,0)
      CALL MOVE(X,XT,0)
      CALL MOVE(X,0.0,IPEN(I1))
      GO TO 10
   17 CALL MOVE(XLIM,0.0,0)
      L3=0
      L4=0
   20 L3=L3+1
      IF(I4)24,24,21
   21 IF(L3-10)23,22,22
   22 L3=1
      L4=L4+1
   23 Y=YINC*(L4+FLOG(L3))
      GO TO 25
   24 Y=L3*YINC
   25 IF(Y-YLIM)26,27,27
   26 CALL MOVE(XLIM,Y,0)
      CALL MOVE(XLIM-YT,Y,0)
      CALL MOVE(XLIM,Y,IPEN(I3))
      GO TO 20
   27 CALL MOVE(XLIM,YLIM,0)
      IF(I1-3)30,38,38
   30 L1=L1-1
      IF(I2)35,35,31
   31 IF(L1)32,32,34
   32 L2=L2-1
      IF(L2)38,33,33
   33 L1=9
   34 X=XINC*(L2+FLOG(L1))
      GO TO 37
   35 IF(L1)38,38,36
   36 X=L1*XINC
   37 CALL MOVE(X,YLIM,0)
      CALL MOVE(X,YLIM-XT,0)
      CALL MOVE(X,YLIM,0)
      GO TO 30
   38 CALL MOVE(0.0,YLIM,0)
      IF(I3-3)40,48,48
   40 L3=L3-1
      IF(I4)45,45,41
   41 IF(L3)42,42,44
```

```
42 L4=L4-1
   IF(L4)48,43,43
43 L3=9
44 Y=YINC*(L4+FLOG(L3))
   GO TO 47
45 IF(L3)48,48,46
46 Y=L3*YINC
47 CALL MOVE(0.0,Y,0)
   CALL MOVE(YT,Y,0)
   CALL MOVE(0.0,Y,0)
   GO TO 40
48 CALL MOVE(0.0,0.0,0)
   RETURN
   END
```

```
   SUBROUTINE HISTO(Y,N,XSIZ,YSIZ,YL,YU,IBAR)
   DIMENSION Y(1)
   FN=N
   CALL SCAL(0.,FN,XSIZ,YL,YU,YSIZ)
   TT=XSIZ
   IF(TT-5.0)30,30,31
31 TT=5.0
30 CALL MOVE(TT,0.0,0)
   CALL MOVE(0.0,0.0,1)
   CALL MOVE(XSIZ, 0.0,0)
   CALL MOVE(XSIZ,YSIZ,0)
   CALL MOVE( 0.0,YSIZ,0)
   CALL MOVE( 0.0, 0.0,0)
   IF(YL)1,3,3
 1 IF(YU)4,4,2
 2 Z=0.0
   CALL PLOT(0.,0.,1)
   CALL PLOT(FN,0.,0)
   GO TO 5
 3 Z=YL
   GO TO 5
 4 Z=YU
 5 CALL PLOT(0.,Z,1)
   IE=0
   XP=0.
   DO 6 I=1,N
   YP=Y(I)
   IF(YP-YU)8,8,7
 7 YP=YU
   IE=1
   GO TO 10
 8 IF(YL-YP)10,10,9
 9 YP=YL
   IE=1
10 CALL PLOT(XP,YP,0)
   XP=I
   CALL PLOT(XP,YP,0)
   IF(IBAR)6,6,15
15 CALL PLOT(XP,Z,0)
 6 CONTINUE
   CALL MOVE(XSIZ+2.,0.0,-1)
   IF(IE)12,12,11
11 CALL PLERR(2)
12 RETURN
   END
```

```
        ENT     SCAL
        ENT     PLOT
        ENT     PIKUP
        ENT     FACTR
        ENT     MOVE
SCAL    DC      0
        BSI     SAVE
        LDX   1 -8              LOOP CONTROL.
LOOP1   LD    2 0               GETT ADDR. OF XL (OR YL)
        MDX   2 +1
        STO     AD2             STORE ADDRESS
        STO     AD4             STORE ADDRESS
        BSI     LOAD            GET XU (OR YU)
        LIBF    FSUB
AD2     DC      0               SUBTRACT XL (OR YL)
        LIBF    FSTO
        DC      TEMP            STORE DIFFERENCE.
        BSI     LOAD            GET X (OR Y) DIMENSION
        LIBF    FMPY
        DC      F100            MULTIPLY BY 100
        LIBF    FDIV
        DC      TEMP            DIVIDE BY DIFFERENCE
        LIBF    FSTOX
        DC      J               STORE INCHES/UNIT (X OR Y)
        LIBF    FMPY
AD4     DC      0               MULTIPLY BY XL (OR YL)
        LIBF    SNR             CHANGE SIGN
        LIBF    FSTOX
        DC      J+2             STORE ORIGIN CORRECTION
        MDX   1 +4              INCREMENT & LOOP.
        MDX     LOOP1
        SLA     16              RESET ORIGIN BY ZEROING X
        STO     J               & Y PEN POSITION
        STO     J+1             ACCUMULATORS
        MDX     EXIT
PLOT    DC      0
        BSI     SAVE
        BSI     LOAD            GET X COORDINATE
        LIBF    FMPY
        DC      C               MULTIPLY BY INCHES/UNIT
        LIBF    FADD
        DC      C+2             ADJUST FOR ORIGIN VALUE
        LIBF    FMPY
        DC      FACTX           MULT. BY X EXPANSION FACTOR
        LDX   L1 J
        BSI     PROC1
        BSI     LOAD            GET Y COORDINATE
        LIBF    FMPY
        DC      C+4            MULTIPLY BY INCHES/UNIT (Y)
        LIBF    FADD
        DC      C+6             ADJUST FOR ORIGIN
        LIBF    FMPY
        DC      FACTY           MULT. BY Y EXP. FACTOR
        MDX   1 +1
        BSI     PROC1
        MDX   2 +1
        LD    I2 -1             GET PEN UP OR DOWN COMMD.
        BSC   L PROC2,Z         BRANCH IF NONZERO
        CALL    LINE
        DC      JX              DRAW LINE TO POINT
        DC      JY
        MDX     EXIT
PROC2   CALL    SKIP
        DC      JX              MOVE TO PT. WITH PEN UP
        DC      JY
        MDX     EXIT
PROC1   DC      0           U----------------------------
        LIBF    FADD        1
        DC      FHALF       1   THIS SECTION ROUNDS OFF
        LIBF    IFIX        1
        S     1 0           1   THE FLOAT. PT. INCREMENT
        STO   1 +2          1
        A     1 0           1   COUNT & STORES AS PEN COM.
        STO   1 0           1
        BSC   I PROC1       1----------------------------
        BSS   E 0
SAVAQ   BSS     2
C       BSS     8
J       BSS     2
```

```
JX      DC      0
JY      DC      0
F100    DEC     100.0
FHALF   DEC     0.5
FACTX   DEC     1.0
FACTY   DEC     1.0
TEMP    DC      0
SAVE    DC      0           S--------------------------
        STX   1 XR1+1       S
        STX   2 XR1+3       S
        STX   3 XR1+5       S  THIS SECTION SAVES STATUS
        STD     SAVAQ       S
        STS     STAT        S  AND LOADS THE ADDRESS OF
        LDX   I2 SAVE       S
        MDX   2 -2          S  THE FIRST CALLING PARAMETER
        STX   2 AD98+1      S
AD98    LDX   I2 0          S  INTO INDEX REG. 2
        LDX   I3 103        S
        BSC   I  SAVE       S--------------------------
EXIT    STX   2 AD99+1      E--------------------------
XR1     LDX   L1 0          E
        LDX   L2 0          E THIS SECTION RESTORES STA-
        LDX   L3 0          E  TUS & RETURNS TO CALLING
        LDD     SAVAQ       E  PROGRAM.
STAT    LDS     0           E
AD99    BSC   L  0          E--------------------------
LOAD    DC      0           L--------------------------
        LD    2 0           L
        STO     AD1         L  FLOAT. PT. NO. IS LOADED
        LIBF    FLD         L  & INDEX REG. 2 INCREMENTED
AD1     DC      0           L
        MDX   2 +1          L
        BSC   I  LOAD       L--------------------------
PIKUP   DC      0           P--------------------------
        BSI     SAVE        P
        LDD     FACTX       P  THIS SECTION RETURNS THE
        STD   I2 0          P  X AND Y EXPANSION FACTORS
        LDD     FACTY       P  TO THE CALLING PROGRAM.
        STD   I2 1          P
        MDX   2 +2          P
        MDX     EXIT        P--------------------------
MOVE    DC      0
        BSI     SAVE
        LDX   1 -2             LOOP CONTROL
LOOP2   BSI     LOAD           GET COORDINATE
        LD    L1 AD80+2     X--------------------------
        STO     AD81        X  MULTIPLY BY X OR Y
        LIBF    FMPY        X  EXPANSION FACTOR
AD81    DC      0           X--------------------------
        LIBF    FMPY
        DC      F100           MULTIPLY BY 100
        LIBF    FADD
        DC      FHALF          ROUND OFF
        LIBF    IFIX
        S     L1 J+2           SUBTRACT CURRENT PEN LOC.
        STO   L1 JX+2          STORE DIFF. AS PEN COMMAND
        MDX   1 +1
        MDX     LOOP2          LOOP BACK
        LD    I2 0             GET PEN UP OR DOWN COMMD
        MDX   2 +1
        BSC   L  PROC3,Z       BRANCH IF NONZERO
        CALL    LINE
        DC      JX              DRAW LINE TO PT.
        DC      JY
        MDX     PROC4
PROC3   CALL    SKIP
        DC      JX             MOVE TO PT. WITH PEN UP
        DC      JY
        BSC   L  PROC4,-
        SLA     16             RESET ORIGIN BY ZEROING
        STO     J             X AND Y PEN POSITION
        STO     J+1            ACCUMULATORS.
        MDX     EXIT
PROC4   LD      J           U--------------------------
        A       JX          UU
        STO     J           U   THIS SECTION UPDATES THE
        LD      J+1         U   X & Y PEN LOC. ACCUMULATORS
        A       JX+1        U
        STO     J+1         U--------------------------
```

```
         MDX      EXIT
FACTR    DC       0          F---------------------------
         BSI      SAVE       F
         LDD   I2 0          F   THIS SECTION RESETS THE
         STD      FACTX      F
         LDD   I2 1          F   X & Y EXPANSION FACTORS.
         STD      FACTY      F
         MDX    2 +2         F
         MDX      EXIT       F---------------------------
AD80     DC       FACTX
         DC       FACTY
         END
```

```
      SUBROUTINE TGRID(XL,YL,XI,YI)
      CALL PIKUP(XF,YF)
      XLIM=XL*XF
      YLIM=YL*YF
      XINC=XI*XF
      YINC=YI*YF
      NX=XLIM/XINC
      IF(NX)5,5,11
   11 DO 1 I=1,NX
      X=I*XINC
      CALL MOVE(X,0.0,0)
      CALL MOVE(X,0.1,0)
    1 CALL MOVE(X,0.0,0)
    5 CALL MOVE(XLIM,0.0,0)
      NY=YLIM/YINC
      IF(NY)6,6,12
   12 DO 2 I=1,NY
      Y=I*YINC
      CALL MOVE(XLIM,Y,0)
      CALL MOVE(XLIM-0.1,Y,0)
    2 CALL MOVE(XLIM,Y,0)
    6 CALL MOVE(XLIM,YLIM,0)
      IF(NX)7,7,13
   13 DO 3 I=1,NX
      X=(NX+1-I)*XINC
      CALL MOVE(X,YLIM,0)
      CALL MOVE(X,YLIM-0.1,0)
    3 CALL MOVE(X,YLIM,0)
    7 CALL MOVE(0.0,YLIM,0)
      IF(NY)8,8,14
   14 DO 4 I=1,NY
      Y=(NY+1-I)*YINC
      CALL MOVE(0.0,Y,0)
      CALL MOVE(0.1,Y,0)
    4 CALL MOVE(0.0,Y,0)
    8 CALL MOVE(0.0,0.0,0)
      RETURN
      END
```

```
         ENT       SYMBL
SYMBL    DC        0
         STX    1  XR1+1
         LDX    I1 SYMBL
         STX    3  XR3+1
         LDX    I3 103
         STD       AQ
         STS       RESTO
         CALL      PIKUP
         DC        XF
         DC        YF
         LDD    I1 1
         STD    L  F1          LETTER SIZE
         LIBF      FLD
         DC        F1
         LIBF      FDIV
         DC        F04
         LIBF      FSTO
         DC        F1
         LIBF      FMPY
         DC        XF
         LIBF      FSTO
         DC        XF
         LIBF      FLD
         DC        F1
         LIBF      FMPY
         DC        YF
         LIBF      FSTO
         DC        YF
         LD     I1 0
         BSC    L  ERROR,+
         CMP       C8
         MDX       ERROR
         NOP
         S         C1          TEMPORARY
         STO       T1
         SLA       1
         A         T1
         SLA       2
         SRT       4
         A         P1
         STO       Q
         SRA       16
         SLT       4
         STO       R
         CALL      BRING
Q        DC        0
R        DC        0
L2       CALL      BITS
         DC        8
         A         P2
         STO       S
         CALL      BITS
         DC        4
         STO       T
         CALL      BRING
S        DC        0
T        DC        0
L3       CALL      BITS
         DC        4
         BSC    L  EXIT,+-
         S         C8
         STO       X
         CALL      BITS
         DC        4
         S         C8
         LDX    1  -4
         MDX       *+1
L4       LD        X
         LIBF      FLOAT
         LIBF      FMPYX
         DC        YF+4
         LIBF      IFIX
         BSC    L  NEGA,Z+
         LIBF      FADD
         DC        FHALF
         MDX       L5
NEGA     LIBF      FSUB
         DC        FHALF
```

```
L5      LIBF      IFIX
        STO   L1  Y+4
        MDX    1  +2
        MDX       L4
        CALL      BITS
        DC        1
        BSC   L   SKIP,+-
        CALL      LINE
        DC        X
        DC        Y
        MDX       L3
        MDX       EXIT
ERROR   CALL      PLERR
        DC        C1
EXIT    LDD       AQ
        MDX   L   SYMBL,+2
RESTO   LDS       0
XR1     LDX   L1  0
XR3     LDX   L3  0
        BSC   I   SYMBL
AQ      DEC       0
SKIP    CALL      SKIP
        DC        X
        DC        Y
        MDX       L3
C1      DC        1
C8      DC        8
YF      DEC       0             CARD 1
XF      DEC       0             CARD2
F1      DEC       0             CARD3
F04     DEC       .0395
FHALF   DEC       0.5           CARD 8
Y       DC        0             CARD9
T1      DC        0
X       DC        0             CARD 11
P1      DC        ADD1
P2      DC        ADD2
ADD1    BSS       0
        DC        /0000         S01
        DC        /3A07          02
        DC        /40AE         S03
        DC        /0E81         S04
        DC        /1914         S05
        DC        /A17B         S06
ADD2    BSS       0
        DC        /AAA2         T01
        DC        /6AAD         T02
        DC        /5C4B         T03
        DC        /5422         T04
        DC        /B098         T05
        DC        /AD45         T06
        DC        /2351         T07
        DC        /0AAC         T08
        DC        /2523         T09
        DC        /19C8         T10
        DC        /B302         T11
        DC        /2B4D         T12
        DC        /66B5         T13
        DC        /6AB0         T14
        DC        /C08A         T15
        DC        /D252         T16
        DC        /3598         T17
        DC        /6043         T18
        DC        /5B39         T19
        DC        /164C         T20
        DC        /501A         T21
        DC        /3958         T22
        DC        /4A56         T23
        DC        /A015         T24
        DC        /146C         T25
        DC        /671A         T26
        DC        /0000         T27
        END
```

```
        ENT     ALPHA
AQ      DEC     0
C46     DC      51          TOTAL NO. OF CHARACTERS
EXIT    LDD     AQ
RESTO   LDS     0
XR1     LDX  L1 0
XR2     LDX  L2 0
XR3     LDX  L3 0
        BSC  I  ALPHA
ALPHA   DC      0
        STD     AQ
        STS     RESTO
        STX   1 XR1+1
        STX   2 XR2+1
        STX   3 XR3+1
        LDX  I3 103         TRANSFER VECTOR ADDRESS
        LDX  I1 ALPHA
        LD   I1 0
        STO  L  LC
        LD    1 1
        STO     L1+1
        LDD  I1 +2
        STD  L  F3          LETTER SIZE
        LDD  I1 +3
        STD  L  F2          LETTER ANGLE
        CALL    FSIN
        DC      F2
        LIBF    FMPY
        DC      F3
        LIBF    FDIV
        DC      F07
        LIBF    FSTO
        DC      F1
        CALL    FCOS
        DC      F2
        LIBF    FMPY
        DC      F3
        LIBF    FDIV
        DC      F07
        LIBF    FSTO
        DC      F2
        MDX  L  ALPHA,+4
        CALL    PIKUP
        DC      XF
        DC      YF
        SLA     16
        STO     XTOT
        STO     YTOT
L1      LD   L  0
        MDX  L  L1+1,-1
        CMP     C46
        MDX     ERROR
        NOP
        BSC  L  ERROR,+
        S       C1          TEMPORARY
        STO     T1
        SLA     1
        A       T1
        SLA     2
        SRT     4
        A    L  P1
        STO     Q
        SRA     16
        SLT     4
        STO     R
        CALL    BRING
Q       DC      0
R       DC      0
L2      CALL    BITS
        DC      8
        A       P2
        STO     S
        CALL    BITS
        DC      4
        STO     T
        CALL    BRING
S       DC      0
T       DC      0
L3      CALL    BITS
```

```
         DC        4
         BSC   L   L2,+-
         S         C1
         BSC   L   NEXT,+-
         S         C7
         STO       X
         CALL      BITS
         DC        4
         S         C8
         STO       Y
         LIBF      FLOAT
         LIBF      FMPY
         DC        F2
         LIBF      FSTO
         DC        F3
         LD        Y
         LIBF      FLOAT
         LIBF      FMPY
         DC        F1
         LIBF      SNR
         LIBF      FSTO
         DC        F4
         LDX   1   -4
L4       LD        X
         LIBF      FLOAT
         LIBF      FMPYX
         DC        F1+4
         LIBF      FADDX
         DC        F3+4
         LIBF      FMPYX
         DC        YF+4
         LIBF      IFIX
         BSC   L   NEGA,Z+
         LIBF      FADD
         DC        FHALF
         MDX       L5
NEGA     LIBF      FSUB
         DC        FHALF
L5       LIBF      IFIX
         STO   L1  Y+4
         MDX   1   +2
         MDX       L4
         LD        XTOT
         S         X
         STO       XTOT
         LD        YTOT
         S         Y
         STO       YTOT
         CALL      BITS
         DC        1
         BSC   L   SKIP,+-
         CALL      LINE
         DC        X
         DC        Y
         MDX       L3
SKIP     CALL      SKIP
         DC        X
         DC        Y
         MDX       L3
XTOT     DC        0
YTOT     DC        0
C1       DC        1
C7       DC        7
C8       DC        8
T1       DC        0
ERROR    CALL      PLERR
         DC        C7
NEXT     MDX   L   LC,-1
         MDX       L1
         CALL      SKIP
         DC        XTOT
         DC        YTOT
         BSC   L   EXIT
YF       DEC       0          CARD 1
XF       DEC       0          CARD2
F1       DEC       0          CARD3
F2       DEC       0          CARD4
F3       DEC       0          CARD5
F4       DEC       0          CARD 6
```

```
F07     DEC     0.069
FHALF   DEC     0.5             CARD 8
Y       DC      0               CARD9
LC      DC      0               CARD 10
X       DC      0               CARD 11
P1      DC      ADD1
P2      DC      ADD2
ADD1    DC      /0000           A1
        DC      /430B           A2
        DC      /3108           A3
        DC      /14B1           A4
        DC      /5D18           A5
        DC      /E1B9
        DC      /1DD1           A7
        DC      /F323           A08
        DC      /6270           A09
        DC      /2982           A10
        DC      /C930           A11
        DC      /6343           A12
        DC      /2EB3           A13
        DC      /2838           A14
        DC      /63E7           A15
        DC      /4184           A16
        DC      /3A46           A17
        DC      /248D           A18
        DC      /4B54           A19
        DC      /EF51           A20
        DC      /A53C           A21
        DC      /5645           A22
        DC      /C564           A23
        DC      /7678           A24
        DC      /6AC6           A25
        DC      /F271           A26
        DC      /4751           A27
        DC      /13E8           A28
        DC      /C57B           A029
        DC      /2844           A030
        DC      /7EC8           A031
        DC      /F692           A032
        DC      /7927           A033
        DC      /7CD8           A034
        DC      /7E92           A035
        DC      /7829           A036
        DC      /9279           A037
        DC      /2789           A038
        DC      /4000           A039
ADD2    DC      /8DD5           B1
        DC      /69B0           B02
        DC      /78BA           B03
        DC      /4794           B04
        DC      /31FB           B05
        DC      /8CBE           B06
        DC      /1EEF           B07
        DC      /58DC           B08
        DC      /65F0           B09
        DC      /D77A           B10
        DC      /C7A0           B11
        DC      /3927           B12
        DC      /7B45           B13
        DC      /E71B           B14
        DC      /99D4           B15
        DC      /65F4           B16
        DC      /418F           B17
        DC      /D469           B18
        DC      /B0B6           B19
        DC      /6B47           B20
        DC      /A039           B21
        DC      /048C           B22
        DC      /7F22           B23
        DC      /8AB8           B24
        DC      /DA06           B25
        DC      /AD51           B26
        DC      /8680           B27
        DC      /5E47           B28
        DC      /E179           B29
        DC      /184C           B30
        DC      /7E84           B31
        DC      /3148           B32
        DC      /7CBE           B33
```

```
DC      /A333     B34
DC      /8ED0     B35
DC      /863F     B36
DC      /9044     B37
DC      /CCAD     B38
DC      /3501     B39
DC      /8FC0     B40
DC      /F235     B41
DC      /018F     B42
DC      /D9EF     B43
DC      /703A     B44
DC      /80C7     B45
DC      /F060     B46
DC      /3BB9     B47
DC      /06AB     B48
DC      /3339     B49
DC      /0700     B50
DC      /BCC8     B51
DC      /35D9     B52
DC      /51FB     B53
DC      /8CBE     B54
DC      /1AEF     B55
DC      /58F2     B56
DC      /8625     B57
DC      /2FA8     B58
DC      /CCF2     B59
DC      /AF36     B60
DC      /8BCE     B61
DC      /2601     B62
DC      /C11E     B63
DC      /C8B4     B64
DC      /6079     B65
DC      /018F     B66
DC      /4140     B67
DC      /8163     B68
DC      /D43A     B69
DC      /FD08     B70
DC      /63D4     B71
DC      /3AF8     B72
DC      /221E     B73
DC      /7D21     B74
DC      /83A8     B75
DC      /0C7A     B76
DC      /9757     B77
DC      /65C2     B78
DC      /7203     B79
DC      /1EC8     B80
DC      /A0C0     B81
DC      /A0B3     B083
DC      /C830     B084
DC      /306A     B085
DC      /E4CE     B086
DC      /0770     B087
DC      /18E4     B088
DC      /CEA3     B089
DC      /2F86     B090
DC      /BBDA     B091
DC      /2EF8     B092
DC      /6814     B093
DC      /1472     B094
DC      /6751     B095
DC      /97C3     B096
DC      /DDED     B097
DC      /1A8C     B098
DC      /BE1A     B099
DC      /EF68     B100
DC      /BCF9     B101
DC      /C370     B102
DC      /8FAA     B103
DC      /7234     B104
DC      /A1CF     B105
DC      /2461     B106
DC      /3330     B107
DC      /604C     B108
DC      /E3CD     B109
DC      /A2EF     B110
DC      /83C5     B111
DC      /41A4     B112
DC      /E3D9     B113
```

```
DC        /104D     B114
DC        /0894     B115
DC        /5666     B116
DC        /F389     B117
DC        /82E7     B118
DC        /661D     B119
DC        /ED17     B120
DC        /9C56     B121
DC        /6751     B122
DC        /97C1     B123
DC        /C187     B124
DC        /EA91     B125
DC        /9624     B126
DC        /6323     B127
DC        /4819     B128
DC        /E525     B129
DC        /A159     B130
DC        /5603     B131
DC        /EB4B     B132
DC        /A66D     B133
DC        /0A84     B134
DC        /4DE3     B135
DC        /0F98     B136
DC        /E407     B137
DC        /3F42     B138
DC        /18B6     B139
DC        /4529     B140
DC        /91A3     B141
DC        /0D41     B142
DC        /EB17     B143
DC        /9AE0     B144
DC        /86A1     B145
DC        /358B     B146
DC        /BD70     B147
DC        /4220     B148
END
```

```
      SUBROUTINE PLOTA(XA,YA,N)
      DIMENSION XA(1),YA(1)
      DO 1 I=1,N
    1 CALL PLOT(XA(I),YA(I),0)
      RETURN
      END
```

```
         ENT       TRAN                                                00010
TRAN     DC        0                                                   00020
         STD   L   TEMP          SAVE STATUS                         00030
         STS       SAVE                                              00040
         STX    1  SVXR1+1                                           00050
         STX    2  SVXR1+3                                           00060
         STX    3  SVXR1+5                                           00070
         SLT       32                                                00080
         LDX   I1  TRAN          PICK UP ADD(ADD) OF N               00090
         LD    I1  0             LOAD N                              00100
         SRA       1                                                 00110
         SLA       1                                                 00120
         SLA       9                                                 00130
         SRA       9                                                 00140
         BSC   L   ABORT,+                                           00150
         STO   L   WDCNT         STORE N IN WDCNT                    00160
         MDX   L   WDCNT,-1                                          00170
         LDX   I2  WDCNT         STORE N IN XR2                      00180
         MDX   L   TRAN,+1                                           00190
         NOP                                                         00200
         LD    I   TRAN                                              00210
         STO       ADDZ+1                                            00220
ADDZ     LDX   L1  0                                                 00230
ADD      LD     1  0             LOAD FIRST EBCDIC CHARA             00240
         LDX    3  +2                                                00250
         RTE       8                                                 00260
         STO       EBCD                                              00270
REPT     LD        EBCD                                              00280
         S         D240                                              00290
         BSC   L   NEX1,-                                            00300
         LD        EBCD                                              00310
         S         D225                                              00320
         BSC   L   NEX2,-Z                                           00330
         LD        EBCD                                              00340
         S         D208                                              00350
         BSC   L   NEX3,-Z                                           00360
         LD        EBCD                                              00370
         S         D192                                              00380
         BSC   L   NEXT,-Z                                           00390
         LD        EBCD                                              00391
         SLA       10                                                00392
         SRA       10                                                00393
         RTE       2                                                 00394
         SRA       2                                                 00395
         SLT       2                                                 00396
         A         D37                                               00397
         MDX       NEXT                                              00398
NEX1     A         D9                                                00400
NEX2     A         D9                                                00410
NEX3     A         D9                                                00420
NEXT     NOP                                                         00430
ADD2     STO   L2  IA                                                00440
         MDX       RTRN                                              00450
         NOP                                                         00460
RTRN     SLA       16                                                00470
         SLT       8                                                 00480
         STO       EBCD                                              00490
         MDX   L   ADD2+1,-1                                         00500
         MDX    3  -1                                                00510
         MDX       REPT                                              00520
         MDX   L   ADD2+1,+2                                         00530
         MDX    1  -1                                                00540
         MDX    2  -2                                                00550
         MDX       ADD                                               00560
         LDX   I1  TRAN                                              00570
         MDX    1  +1                                                00580
         NOP                                                         00590
         LD     1  0                                                 00600
         STO       HGHT                                              00610
         MDX    1  +1                                                00620
         NOP                                                         00630
         LD     1  0                                                 00640
         STO       ANGLE                                             00650
         LD        HGHT-1                                            00660
         A         WDCNT                                             00670
         STO       HGHT-1                                            00680
         MDX   L   WDCNT,+1                                          00690
         LDX   I3  103           TRANS VECTOR ADDRESS                00700
         CALL      ALPHA                                             00710
```

```
         DC        WDCNT                                          00720
         DC        IA                                             00730
HGHT     DC        0                                              00740
ANGLE    DC        0                                              00750
ABORT    MDX   L   TRAN,+3                                        00760
SVXR1    LDX   L1  0                                              00770
         LDX   L2  0                                              00780
         LDX   L3  0                                              00790
SAVE     LDS       0                                              00800
         LDD       TEMP                                           00810
         BSC   I   TRAN                                           00820
TEMP     BSS   E   2                                              00830
EBCD     DC        0                                              00840
D208     DC        208                                            00850
D225     DC        225                                            00860
D240     DC        240                                            00870
D192     DC        192                                            00880
D9       DC        9                                              00890
D37      DC        37                                             00891
WDCNT    DC        0                                              00900
XR3      DC        0                                              00910
RETIN    DC        0                                              00920
IA       BSS       64                                             00930
         END                                                      00940
```

```
      SUBROUTINE DECD(AA,HGHT,ANGLE)
      DIMENSION          IA(6),ITBL(10)
      DATA ITBL/27,28,29,30,31,32,33,34,35,36/
      DATA N/6/
      A=ABS(AA)
      DO3J=1,N
      B=10.0**J
      FINT=IFIX(A/B)
      C=(A-FINT*B)/(10.0**(J-1))
      IC=IFIX(C)+1
      J9=N+1-J
      IA(J9)=ITBL(IC)
3     CONTINUE
      IF(AA)5,6,6
5     IA(1)=45
6     CONTINUE
      CALL ALPHA(N,IA,HGHT,ANGLE)
      RETURN
      END
```

```
      SUBROUTINE CIRCL(X,Y,R)
      DIMENSION SS(19)
      DATA SS/0.0,0.0872,0.1736,0.2588,0.342,0.4226,0.5,0.5736,0.6428,
     10.7071,0.766,0.8192,0.866,0.9063,0.9397,0.9659,0.9848,0.9962,1.0/
      CALL MOVE(X,Y+R,1)
      DO 1 I=1,18
      J=19-I
    1 CALL MOVE(X+SS( I+1)*R,Y+SS(    J)*R,0)
      DO 2 I=1,18
      J=19-I
    2 CALL MOVE(X+SS(    J)*R,Y-SS( I+1)*R,0)
      DO 3 I=1,18
      J=19-I
    3 CALL MOVE(X-SS( I+1)*R,Y-SS(    J)*R,0)
      DO 4 I=1,18
      J=19-I
    4 CALL MOVE(X-SS(    J)*R,Y+SS( I+1)*R,0)
      RETURN
      END
```

```
        ENT       PLERR
PLERR   DC        0
        STD       AQ
        STS       RESTO
        LD    I   PLERR
        STO       ADDR+1
ADDR    LD    L   0
        STO       HEXNO
        CALL      EACPT
        DC        MSG1
        DC        MSG2
        DC        HEXNO
        DMES      //'E
        XIO   L   46
        LDD       AQ
RESTO   LDS       0
        MDX   L   PLERR,+1
        BSC   I   PLERR
MSG1    DMES      'RPLOT 'E
MSG2    DMES      ERROR NO. 'E
HEXNO   BSS       6
AQ      DEC       0
        END
```

```
        ENT       LINE
        ENT       SKIP
LINE    DC        0
        BSI       BEGIN
        LDX   I1  LINE
        BSC   L   START,+-
        CALL      PLOTX
        DC        LIST1
        SLA       16
        MDX       START
SKIP    DC        0
        BSI       BEGIN
        LDX   I1  SKIP
        BSC   L   START,Z-
        CALL      PLOTX
        DC        LIST2
        LD        C1
START   STO       IND
        STX    2  XR2+1
        STS       RESTO
        SLT       32
        LDX    2  0
        LD     I1 0
        BSC   L   A1,-
        MDX    2  +8
        SLA       16
        S      I1 0           ABSOLUTE VALUE
        CMP       MAXN
        MDX       A1
        MDX       A1
        LD        MAXP
A1      STO       X
        LD     I1 1
        BSC   L   A2,-
        MDX    2  +4
        SLA       16
        S      I1 1           ABSOLUTE VALUE
        CMP       MAXN
        MDX       A2
        MDX       A2
        LD        MAXP
A2      STO       Y
        MDX    1  +2
        STX    1  BACK+1
        CMP       X              Y GREATER THAN X
        MDX       A3              IF SO, GO TO A3
        MDX       A6
        BSC   L   XR1,+-
A6      SRT       15             M2=2Y
        STD       M2
        LD        X
        SLA       1              N2=2X
        STO       N2
        SRT       17             T1=N
        STD       T1
        LD        COMM1
        MDX       A4
A3      SLA       1              N2=2Y
        STO       N2
        SRT       17             T1=N
        STD       T1
        LD        X
        SRT       15             M2=2X
        STD       M2
        LD        COMM2
A4      SRA    2  0              SHIFT FOR CORRECT VERT/HORIZ COMMAND
        AND       MASK
        STO       T3
        LD        COMM3
        SRA    2  0              SHIFT FOR CORRECT DIAGONAL COMMAND
        AND       MASK
        STO       T3+1
        SLA       16             T2=0
        STO       T2
        LDX    I1 T1+1           INITIALIZE STEP COUNT
LOOP    LDD       T1
        AD        M2             CALC 2IM+N
        STD       T1
        D         N2             (2IM+N)/2N
```

```
         STO        X          STORE RESULT
         S          T2
         A          ADDR       GET COMMAND ADDRESS
         STO        A5+1
A5       LD    L    0
         STO        COMMD
         CALL       PLOTX
         DC         LIST3
         LD         X
         STO        T2
         MDX     1  -1
         MDX        LOOP
XR1      LDX   L1   0
XR2      LDX   L2   0
         LDD        AQ
RESTO    LDS        0
BACK     BSC   L    0
BEGIN    DC         0
         STX     1  XR1+1
         STD        AQ
         LD         IND
         BSC   I    BEGIN
IND      DC         -1
C1       DC         1
X        DC         0
Y        DC         0
N2       DC         0
COMM1    DC         /7733      HORIZONTAL COMMANDS
COMM2    DC         /5151      VERTICAL   COMMANDS
COMM3    DC         /6842      DIAGONAL   COMMANDS
MASK     DC         /000F
T2       DC         0
ADDR     DC         T3
MAXN     DC         /8000
MAXP     DC         /3FFF
         BSS   E    0
AQ       DEC        0
T1       DEC        0
M2       DEC        0
T3       DEC        0
LIST1    DC         0
         DC         0          PEN DOWN COMMAND
LIST2    DC         0
         DC         9          PEN UP   COMMAND
LIST3    DC         0
COMMD    DC         0
         END
```

```
         ENT       BRING
         ENT       BITS
BRING    DC        0
         STX    1  XR1+1
         LDX    I1 BRING
         LD     1  0
         STO       A1+1
         LD     I1 0
         STO       T1+1
         LD        C16
         STO       COUNT
         MDX    1  +1
         MDX       A2
BITS     DC        0
         STX    1  XR1+1
         LDX    I1 BITS
A2       LD     1  0
         STO       NUMBR
         MDX    1  +1
         STX    1  BACK+1
         BSC    L  XR1,+-
         SLA       16
         STO       T1
LOOP     LDD       T1
         SLT       1
         STD       T1
         MDX    L  COUNT,-1
         MDX       NEXT
         MDX    L  A1+1,+1
A1       LD     L  0
         STO       T1+1
         LD        C16
         STO       COUNT
NEXT     MDX    L  NUMBR,-1
         MDX       LOOP
         LD        T1
XR1      LDX    L1 0
BACK     BSC    L  0
C16      DC        16
T2       DC        0
T1       DEC       0
COUNT    DC        0
NUMBR    DC        0
         END
```

```
        ENT     SCAL3
        ENT     PLOT3
SCAL3   DC      0
        BSI     SAVE
        LDX   1 -12             LOOP CONTROL.
LOOP1   LD    2 0               GETT ADDR. OF XL (YL,ZL)
        MDX   2 +1
        STO     AD2             STORE ADDRESS
        STO     AD4             STORE ADDRESS
        BSI     LOAD            GET XU (OR YU,ZU)
        LIBF    FSUB
AD2     DC      0               SUBTRACT XL (OR YL,ZL)
        LIBF    FSTO
        DC      TEMP            STORE DIFFERENCE.
        BSI     LOAD            GET X (OR Y,Z) DIMENSION
        LIBF    FMPY
        DC      F100            MULTIPLY BY 100
        LIBF    FDIV
        DC      TEMP            DIVIDE BY DIFFERENCE
        LIBF    FSTOX
        DC      C+12            STORE INCHES/UNIT (X,Y,Z)
        LIBF    FMPY
AD4     DC      0               MULTIPLY BY XL (OR YL,ZL)
        LIBF    SNR             CHANGE SIGN
        LIBF    FSTOX
        DC      C+14            STORE ORIGIN CORRECTION
        MDX   1 +4              INCREMENT & LOOP.
        MDX     LOOP1
        BSI     P1
        LIBF    FSTO
        DC      TEMP
        LIBF    FMPY
        DC      C+2             MULT. BY X ORIGIN CORR.
        LIBF    FADDX
        DC      C               ADD TO Y ORIGIN CORR.
        LIBF    FSTOX
        DC      C               NEW Y ORIGIN CORRECTION
        LIBF    FLD
        DC      TEMP
        LIBF    FMPY
        DC      C               MULT. BY X UNITS/INCH
        LIBF    FSTO
        DC      C+14            Y AXIS CORR. FOR X DATA
        CALL    FCOS
        DC      ANGLE           COSINE OF X ANGLE
        LIBF    FSTO
        DC      TEMP
        LIBF    FMPY
        DC      C+2             MULT. BY X ORIGIN CORR.
        LIBF    FSTO
        DC      C+2             NEW X ORIGIN CORRECTION
        LIBF    FLD
        DC      TEMP
        LIBF    FMPY
        DC      C               MULT. BY X UNITS/INCH
        LIBF    FSTO
        DC      C+12            X AXIS CORR. FOR X DATA
        BSI     P1
        BSI     P3
        LIBF    FSTO
        DC      C               Y AXIS CORR. FOR Z DATA
        CALL    FCOS
        DC      ANGLE           COSINE OF Z ANGLE
        LDX   1 2
        BSI     P3
        LIBF    FSTO
        DC      C+8             X AXIS CORR. FOR Z DATA
        SLA     16              RESET ORIGIN BY ZEROING X
        STO     J               & Y PEN POSITION
        STO     J+1             ACCUMULATORS
        MDX     EXIT
P1      DC      0
        BSI     LOAD            GET ANGLE OF X (Z) AXIS
        LIBF    FMPY
        DC      RAD             CONVERT TO RADIANS
        LIBF    FSTO
        DC      ANGLE           STORE ANGLE
        CALL    FSIN
        DC      ANGLE           GET SINE OF ANGLE
```

```
         LDX    1  6
         BSC    I  P1
P3       DC        0
         LIBF      FSTO
         DC        TEMP
         LIBF      FMPY
         DC        C+10          MULT. BY Z ORIGIN CORR.
         LIBF      FADDX
         DC        C             ADD TO X (Y) ORIGIN CORR.
         LIBF      FSTOX
         DC        C             NEW X (Y) ORIGIN CORRECTION
         LIBF      FLD
         DC        TEMP
         LIBF      FMPY
         DC        C+8         MULT. BY Z UNITS/INCH
         BSC    I  P3
SAVE     DC        0          S----------------------------
         STX     1 XR1+1      S
         STX     2 XR1+3      S
         STX     3 XR1+5      S  THIS SECTION SAVES STATUS
         STD       SAVAQ      S
         STS       STAT       S  AND LOADS THE ADDRESS OF
         LDX    I2 SAVE       S
         MDX     2 -2         S  THE FIRST CALLING PARAMETER
         STX     2 AD98+1     S
AD98     LDX    I2 0          S  INTO INDEX REG. 2
         LDX    I3 103        S
         BSC    I  SAVE       S----------------------------
EXIT     STX     2 AD99+1     E----------------------------
XR1      LDX    L1 0          E
         LDX    L2 0          E THIS SECTION RESTORES STA-
         LDX    L3 0          E  TUS & RETURNS TO CALLING
         LDD       SAVAQ      E  PROGRAM.
STAT     LDS       0          E
AD99     BSC    L  0          E----------------------------
LOAD     DC        0          L----------------------------
         LD      2 0          L
         STO       AD1        L  FLOAT. PT. NO. IS LOADED
         LIBF      FLD        L  & INDEX REG. 2 INCREMENTED
AD1      DC        0          L
         MDX     2 +1         L
         BSC    I  LOAD       L----------------------------
         BSS    E  0
SAVAQ    BSS       2
C        BSS       16
J        BSS       2
JX       DC        0
JY       DC        0
PLOT3    DC        0
         BSI       SAVE
         CALL      PIKUP
         DC        FACTX
         DC        FACTY
         BSI       LOAD          GET X COORDINATE
         LIBF      FSTO
         DC        TEMP2
         LIBF      FMPY
         DC        C+12
         LIBF      FSTO
         DC        TEMP
         LIBF      FLD
         DC        TEMP2
         LIBF      FMPY
         DC        C+14
         LIBF      FSTO
         DC        TEMP2
         BSI       LOAD          GET Y COORDINATE
         LIBF      FMPY
         DC        C+4         MULTIPLY BY INCHES/UNIT (Y)
         LIBF      FADD
         DC        TEMP2
         LIBF      FSTO
         DC        TEMP2
         BSI       LOAD
         LIBF      FSTO
         DC        Z
         LIBF      FMPY
         DC        C+8
         LIBF      FADD
```

```
        DC      TEMP
        LIBF    FADD
        DC      C+2             ADJUST FOR ORIGIN VALUE
        LIBF    FMPY
        DC      FACTX           MULT. BY X EXPANSION FACTOR
        LDX  L1 J
        BSI     PROC1
        LIBF    FLD
        DC      Z
        LIBF    FMPY
        DC      C
        LIBF    FADD
        DC      TEMP+2
        LIBF    FADD
        DC      C+6             ADJUST FOR ORIGIN
        LIBF    FMPY
        DC      FACTY           MULT. BY Y EXP. FACTOR
        MDX   1 +1
        BSI     PROC1
        MDX   2 +1
        LD    I2 -1             GET PEN UP OR DOWN COMMD.
        BSC   L PROC2,Z         BRANCH IF NONZERO
        CALL    LINE
        DC      JX              DRAW LINE TO POINT
        DC      JY
        MDX     EXIT
PROC2   CALL    SKIP
        DC      JX              MOVE TO PT. WITH PEN UP
        DC      JY
        MDX     EXIT
PROC1   DC      0           U----------------------------
        LIBF    FADD        1
        DC      FHALF       1   THIS SECTION ROUNDS OFF
        LIBF    IFIX        1
        S     1 0           1   THE FLOAT. PT. INCREMENT
        STO   1 +2          1
        A     1 0           1   COUNT & STORES AS PEN COM.
        STO   1 0           1
        BSC   I PROC1       1----------------------------
F100    DEC     100.0
FHALF   DEC     0.5
FACTX   DEC     1.0
FACTY   DEC     1.0
RAD     DEC     0.017453        CONVERSION FACTOR RADIANS
ANGLE   DEC     0
TEMP    DEC     0
TEMP2   DEC     0
Z       DEC     0
        END
```