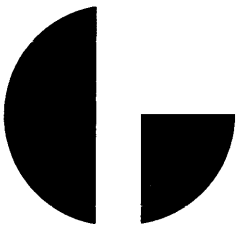


DATA GENERAL
CORPORATION
Southboro,
Massachusetts 01772
(617) 485-9100

PROGRAM

DATAPLOT

TAPE

Library: 099-000021

ABSTRACT

DATAPLOT is used for programming digital incremental plotters and consists of a group of relocatable binary subroutines comprising a library tape. The subroutines are callable from FORTRAN and are used with NOVA computers to drive the Data General incremental plotting systems.

093-000060-01

This revision of the Dataplot manual, 093-000060-01, replaces revision 093-000060-00. The only change corrects an error on page 6. The current coordinates of the pen when the value of IPEN is 0 are (0, 0).

TABLE OF CONTENTS

INTRODUCTION

DATAPLOT is a group of relocatable binary subroutines, released in the form of a library tape, 099-000021 which can be used to drive the Data General incremental plotting systems on NOVA computers.

The DATAPLOT subroutines are written in FORTRAN IV and Assembly Language. They can be called by any FORTRAN language program, using the FORTRAN CALL statement. When used in conjunction with Data General digital plotters, the subroutines facilitate the production of charts, maps, drawings, special curves, and alphabetic characters or symbols.

DATAPLOT CHARACTERISTICS

A plot is generated by the operation of the three independent axes X, Y, and Z. The chart (X axis) moves bi-directionally under the pen beam; the pen carriage (Y axis) moves bi-directionally over the width of the paper's surface; and the pen (Z axis) is raised and lowered.

The plotter is driven in eight basic directions as illustrated in Figure 1. Any other motion is achieved by driving the plotter in an appropriate sequence of moves involving two or more of the basic directions.

Accuracy and precision of lines and characters depend upon the size of the pen point and the plotter's basic increment size.

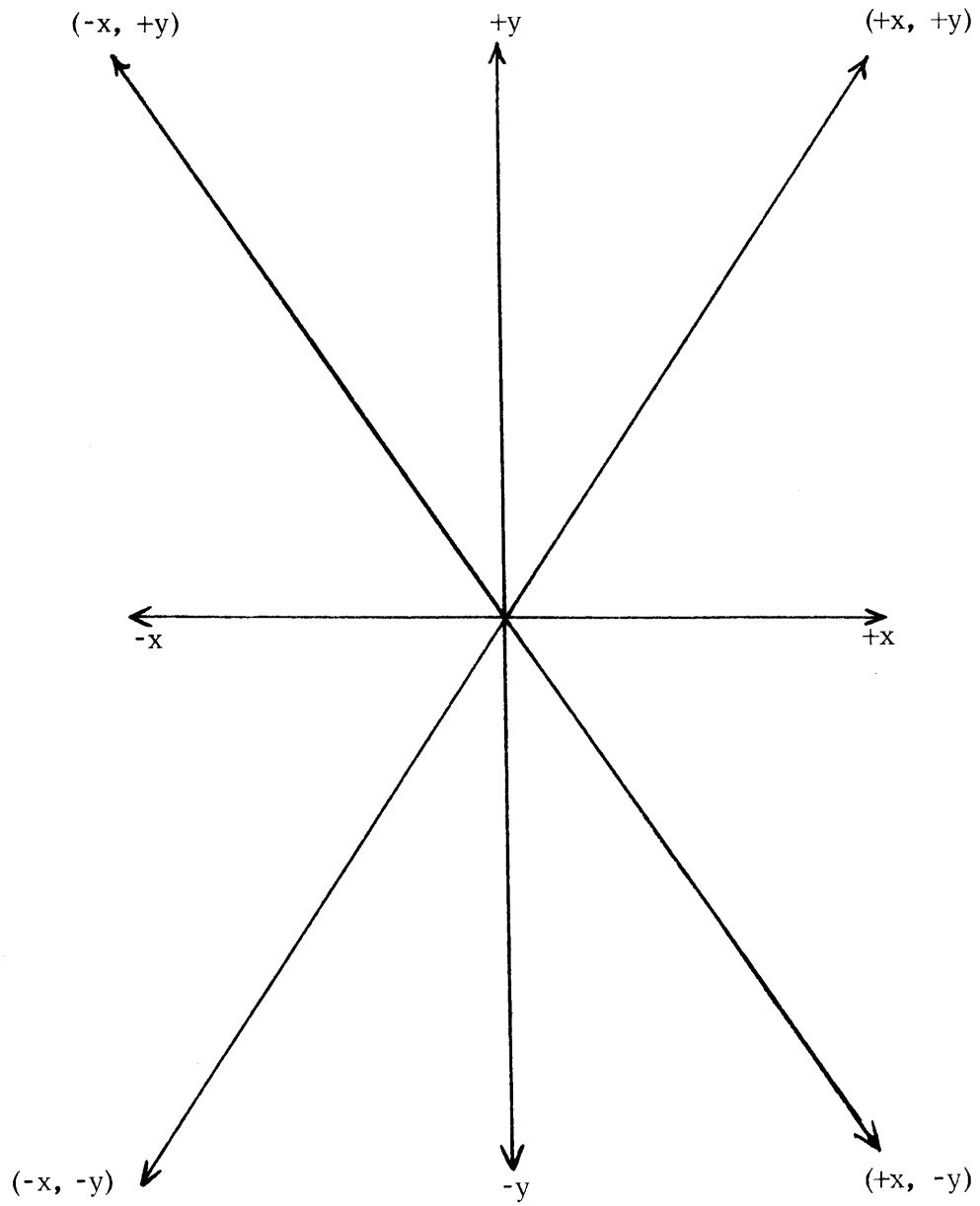


Figure 1

DATAPLOT SUBROUTINES

There are twelve primary plotter subroutines.

Name	Purpose
INITAL	Sets up certain constants that are a function of the plotter to be used.
RSTR	Restores the plotter to a position on a new page and positions the pen at the bottom of the page.
PLOT	Drives the pen from its present position to a new position with the pen raised or lowered. It is also used to redefine the present position of the pen or to locate the present position of the pen.
LINE	Drives the pen through an array of points for point-to-point plotting.
WHERE	Returns the current pen position coordinates to the calling program.
MARKER	Draws an event marker at the present pen position.
SCALE	Sets and stores scaling information for the AXIS and LINE routines.
PENUP	Raises the pen.
PENDN	Lowers the pen.
SYMBOL	Positions the pen and forms the characters to be plotted. (Available characters are 0-9, A-Z, and various special characters.)
NUMBER	Positions the pen and forms the numerals to be plotted based on a real input variable.
AXIS	Draws an axis with an identification label and "tic" marks at one-inch intervals.

DATAPLOT SUBROUTINES (Continued)

INITAL Subroutine

The INITAL subroutine initializes the plotting routines for the creation of one plot or a group of plots by a given program. INITAL must be called only once and before any other call to the plotting subroutines is made.

A call to INITAL also drives the pen down toward the furthest (Ø, -Y) point on the current page. The pen is then positioned one half inch above this point, where (X, Y) is defined as (Ø, Ø).

The call to INITAL has the following format:

```
CALL INITAL (IU, IS, W, P)
```

where:

IU - is the logical output device number (unit number 6 for DGC FORTRAN)

IS - is resolution of the plotter in increments per inch.

W - is the plotter width in inches (Y axis).

P - is the page length in inches (X axis).

Example:

```
CALL INITAL (6, 100, 11.0, 8.5)
```

RSTR Subroutine

The RSTR subroutine restores the plotter to a position on a new page when a new plot is begun and positions the pen at the bottom of the page.

To use the subroutine correctly, the paper should be aligned initially so that a page boundary is directly beneath the pen point. This is useful in multiple plots to insure that each of the plots falls on a different sheet (based on the page

RSTR Subroutine (Continued)

length specified to INITAL).

The call to RSTR has the following format:

```
CALL RSTR
```

When the subroutine is called, the paper is indexed to the next unwritten page, the pen is positioned one half inch above the extreme (Ø, -Y) coordinate and this pen position is redefined as as (Ø, Ø).

PLOT Subroutine

PLOT is the subroutine most commonly used by the programmer. PLOT is used to drive the pen from its present position to a new position, as specified by (X, Y) cartesian coordinates. The coordinates are given in inches.

In addition, PLOT performs several auxiliary functions that are determined by the value of the third argument of the call.

The call to PLOT has the following format:

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

Where:

X - is the floating point X coordinate to which the pen is to be moved (or the coordinate of the present pen position depending upon the value of IPEN). X is the independent variable.

Y - is the floating point Y coordinate to which the pen is to be moved (or the coordinate of the present pen position). Y is the dependent variable.

IPEN - is a fixed point argument whose sign and magnitude determine the functions performed by the subroutine as shown below.

PLOT Subroutine (Continued)

Value of IPEN	Action Taken by PLOT
0	No plotter movement. The present position of the pen is redefined as (0, 0).
1	Plotter moves to (X, Y) with the present condition of the pen (up or down).
2	The pen is dropped and the plotter moves to (X, Y).
3	The pen is raised and the plotter moves to (X, Y).
4	There is no plotter movement; the present pen coordinates are returned in (X, Y).
-1 -2 -3	The action of the pen and plotter is the same as for the same positive value, but after (X, Y) is reached, X and Y are redefined as (Ø, Ø).

Examples:

```
CALL PLOT (1.2,3.0,-3)

CALL PLOT (X1,Y1,2)
```

LINE Subroutine

LINE is used for point-to-point plotting of graphs, functions, charts, etc. The values of X and Y are stored in single subscripted arrays passed to the subroutine. The subroutine can also draw one of five special characters or any left justified character at points specified by the programmer in the call.

The call to LINE has the following format:

```
CALL LINE (X,Y,N,KODE,ISPACE)
```

DATAPLOT SUBROUTINES (Continued)

LINE Subroutine (Continued)

Where:

X - is a single subscripted array of independent variables.

Y - is a single subscripted array of dependent variables.

N - is the number of data points.

KODE - determines the character that will mark selected points in the plot and whether or not lines are drawn from point to point. The effect of values of KODE are described below.

ISPACE - specifies the interval at which the evenly spaced markers (the character given by KODE) will be drawn. For example if ISPACE is 5, symbols will be drawn at the first data point, fifth, tenth, fifteenth, etc. The symbols are drawn centered around the data points.

The value of KODE may be a left-justified character, such as "Z" or "A", which is used to mark the points on the plot given by the argument ISPACE. If the value of KODE is Ø, no character marks the data points.

If the value of KODE is 1, 2, 3, 4 or 5, special characters are used to mark data points. The values 1 through 5 may be signed, where:

\+ - indicates lines will be drawn connecting data points, and

\- - indicates that data points are not connected by lines.

The special characters are as follows:

Value of KODE	Character on PLOT
±1	+
±2	X
±3	*
±4	#
±5	!

DATAPLOT SUBROUTINES (Continued)

LINE Subroutine (Continued)

Examples:

```
CALL LINE (X,Y,NTOP,-1,5)
CALL LINE (X,Y,NTOP,Ø,1)
CALL LINE (X,Y,NTOP,"Q",2)
```

WHERE Subroutine

WHERE returns the current pen position coordinates to the two arguments of the calling sequence, permitting user-written subroutines to know the current pen position for optimizing pen movement.

The call to WHERE has the format:

```
CALL WHERE (X,Y)
```

Where:

X and Y are floating point coordinates to be returned for the present pen position.

Example:

```
CALL WHERE (X,Y)
```

MARKER Subroutine

MARKER is used to write any one of the five special symbols defined for the LINE subroutine or any left justified character. MARKER symbols are centered The call to marker has the following format:

```
CALL MARKER (IMK)
```

Where:

IMK evaluates to a left justified character or to one of the values 1 through 5 producing the special characters shown below:

IMK=1 + drawn on plot

IMK=2 X drawn on plot

IMK=3 * drawn on plot

IMK=4 # drawn on plot

IMK=5 ! drawn on plot

Examples:

```
CALL MARKER (1)          ← + drawn on plot
CALL MARKER ("A")        ← A drawn on plot
```

PENUP Subroutine

PENUP causes the pen to be lifted. The call to PENUP has the format:

```
CALL PENUP
```

PENDN Subroutine

PENDN causes the pen to be lowered. The call to PENDN has the format:

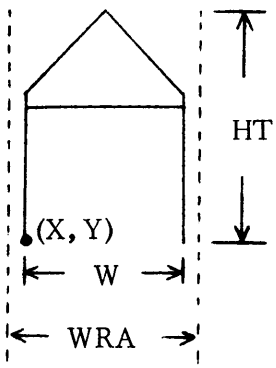
```
CALL PENDN
```

SYMBOL Subroutine

SYMBOL is used to generate characters of the alphabet so that plots can be annotated. To use the routine, the user must specify the characters to be written, the starting location of the characters (X, Y), the height of the characters, and the angle at which the characters are to be drawn. With this input data, SYMBOL supplies the specific data to the plotter that is needed to draw the characters.

Each character is drawn within a defined region, called a "reference area". The dimensions of the characters are based upon the height of the character, given by an input argument. The reference point for the reference area is at the lower lefthand corner of the reference area for drawing of all characters except when a single character is to be drawn. This occurs on calls to LINE and MARKER and on a call to SYMBOL when the last argument is -1 or Ø. When a single character is to be drawn, the reference point is at the center.

The reference point (X, Y) is the starting position of the pen when drawing a character. The width of the reference area and the width of the character are calculated from the height (argument HT) as follows:



Height = HT
Width = 5/7 HT
Width of Reference Area = 6/7 HT

SYMBOL Subroutine (Continued)

The call to SYMBOL has the following format:

```
CALL SYMBOL (X, Y, HT, IHOL, THETA, N)
```

Where:

X and Y — specify the coordinates of the reference point.

HT — specifies the height of the character. The minimum height is .Ø7 inch and the argument must be .Ø7 or an integer multiple of that value.

IHOL — is a single subscripted array containing N characters. Characters are stored two per word in IHOL.

THETA — specifies the angle in degrees at which characters are to be drawn. Although THETA may be given as any angular value, characters are only drawn at 90° angles according to the table shown below.

N — specifies the number of characters to be drawn. If N = Ø or -1, a single character left-justified in IHOL will be drawn.

Value of THETA	Angle	Example
Ø. ≤ AMOD(THETA, 36Ø) < 45.	Ø (+X, +Y)	ABC
45. ≤ AMOD(THETA, 36Ø) < 135.	9Ø (-X, +Y)	ABC
135. ≤ AMOD(THETA, 36Ø) < 225.	18Ø (-X, -Y)	ABC
225. ≤ AMOD(THETA, 36Ø) < 36Ø.	27Ø (+X, -Y)	ABC

Example:

```
CALL SYMBOL(Ø., 8.5, .21, MAB, Ø., 12)
```

NUMBER Subroutine

NUMBER is similar to SYMBOL, except that it is used to specify numbers rather than characters. NUMBER formats a floating point number and draws the resulting alphanumeric characters. It establishes the starting location and angle, and then draws the required floating point number. The call to NUMBER has the following format:

```
CALL NUMBER (X, Y, HT, FLT, THETA, N)
```

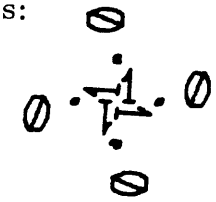
Where:

X and Y — specify the coordinates of the reference point (as in SYMBOL).

HT — is a signed floating point number. Its magnitude specifies the height of the number in inches. The minimum value of HT is .Ø7 and the argument must be an integer multiple of .Ø7. As in SYMBOL, the width of field and the number within the field are determined from the height. The sign of HT determines whether the output of characters begins or ends at (X, Y). If HT is positive, output begins at (X, Y); if negative, output terminates at (X, Y).

FLT — is the floating point number that is to be drawn.

THETA — specifies the angle in degrees at which numbers are to be drawn. Numbers can be drawn at Ø, 9Ø, 18Ø, 27Ø degrees according to the value specified for THETA, as shown in the writeup on SYMBOL. An example of 1.Ø drawn at the four possible angles is:



N — is an optionally signed integer. Its magnitude gives the number of digits after the decimal point if N is positive. If N=0, output terminates at the decimal point. If N= -1, the decimal point is suppressed.

Example:

```
CALL NUMBER (-.2, YY, -.14, YFLT, Ø.Ø, 1)
```

AXIS Subroutine

AXIS is used to generate the axes for a graph. It will draw an axis with a linear scale, place tic marks on the axis at one-inch intervals, and label the axis with letters and numbers Ø.14 inches high. There are normally two calls for this routine per graph, one for the X axis and one for the Y axis. The call to AXIS has the following format:

```
CALL AXIS(X, Y, LBL, NC, S, THETA, SMIN, DS, NN)
```

Where:

X and Y — are the coordinates in inches of the starting point of the axis.

LBL — is a single subscripted array containing the identification label for the axis and is approximately centered and written with letters Ø.14 in. high.

NC — is a signed integer argument whose magnitude specifies the number of characters in LBL. If NC is positive, the scale, tic marks, and label are drawn on the counterclockwise side of the axis; if NC is negative, they are drawn on the clockwise side of the axis.

S — specifies the length in inches of the axis. If signed, the absolute value of S is taken.

THETA — is the angle in degrees at which the axis is drawn, where:

THETA=Ø. implies a horizontal axis, reading -X to +X.
THETA=9Ø. implies a vertical axis, reading -Y to +Y.
THETA=18Ø. implies a horizontal axis, reading +X to -X.
THETA=27Ø. implies a vertical axis, reading +Y to -Y.

SMIN — is the minimum value shown on the axis.

DS — is the scale increment or the increment value per inch on the scale. The tic marks are labeled at one-inch intervals with values calculated as:

SMIN, SMIN+DS, SMIN+2DS,...

NN — specifies the number of digits after the decimal point on values shown on the axis. A magnitude must be given for NN.

AXIS Subroutine (Continued)

Example:

```
CALL AXIS (Ø.Ø, 2.Ø, LAB, -1, 6.Ø, Ø.Ø, XMIN, DX, 1)
```

SCALE Subroutine

SCALE is used to determine a convenient linear scale based on plot dimensions. If data to be plotted does not have a zero base scale or cannot be conveniently arranged to fit in a zero base scale, the subroutine will reset the scaling as necessary. The call to SCALE has the following format:

```
CALL SCALE (X, N, S, XMIN, DX)
```

Where:

X — is a single subscripted array for which the scale values (XMIN and DX) are to be determined.

N — is the number of points of X to be scaled.

S — is a floating point argument specifying the length in inches of the axis against which the data is plotted. The first N X's are plotted within a scale length S.

XMIN — is computed by SCALE from the first three arguments and is the smallest value to be plotted on the axis.

DX — is computed by SCALE from the first three arguments and takes on one of the following values in order to produce a convenient scale:

$$DX = K*10^{m}$$

where:

K — is 1, 2, 4, 5, or 8.

10^{m} — is the order of magnitude of (XMAX-XMIN)/S, i.e., for the first N X's, X= (X-XMIN)/DX.

Example:

```
CALL SCALE (IPTS, 1ØØ, 8.Ø, XMIN, DX)
```

LOADING DATAPLOT

The DATAPLOT library tape is loaded immediately after the FORTRAN main program and any FORTRAN subprograms. Loading proceeds as follows:

1. Load the FORTRAN main program relocatable binary.

2. Load any FORTRAN subprograms called by the main program.

3. Load DATAPLOT library tape 099-000021.

4. Load the FORTRAN library tapes appropriate to the particular NOVA configuration.

SAMPLE PROGRAM

Following is a sample FORTRAN main program which calls many of the DATAPLOT subroutines. The program generates the function, SIN(X)/X from .01 to 10.1/

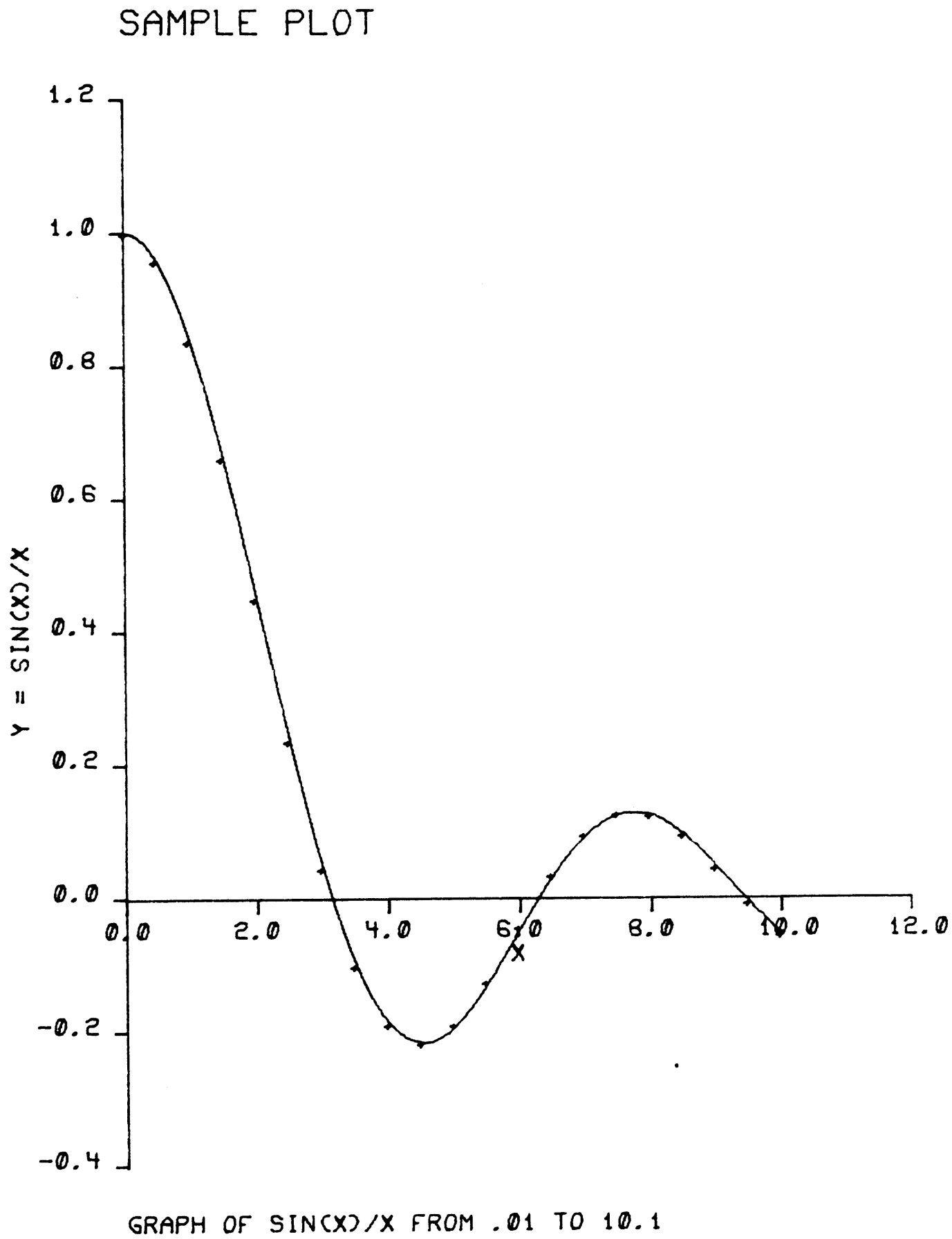
```
      COMMON /TEMPZ/KAB,LAB,MAB,NAB,KAAD,LABEL,ISYM,ISPC
      DIMENSION Y(102),X(103),KAB(7),LAB(2),MAB(7),NAB(18),KAAD(2)
     DIMENSION LABEL(5)
     DIMENSION ISYM(20)
     DIMENSION ISPC(20)
      DATA KAB/4HY = ,4HSIN(,4HX)/X/
      DATA LAB/1HX/
      DATA MAB/4HSAMP,4HLE P,4HLOT /
      DATA NAB/4HGRAP,4HH OF,4H SIN,4H(X)/,4HX FR,4HOM .,4H01 T,
     1 4HO 10,4H.1  /
      DATA KAAD/4HTST1/
       X2=SQRT(27.0)
       Y2=X2
       X2=3.0
       X1=6.0
       Y1=0.0
       X0=0.0
       Y0=0.0
       CALL INITAL(6,100,11.,8.5)

        DELX=0.1
         XMIN=0.01
        XMAX=10.01
        NTOP=(XMAX-XMIN)/DELX+0.5
        NTOP=NTOP+1
        X(1)=XMIN
        DO 30 I=1,NTOP
        Y(I)=SIN(X(I))/X(I)
        X(I+1)=X(I)+DELX
   30   CONTINUE
      DO 50 I=1,NTOP
      X(I) = X(I)/2.
      Y(I) = Y(I)*5.0+2.0
50    CONTINUE
```

SAMPLE PROGRAM (Continued)

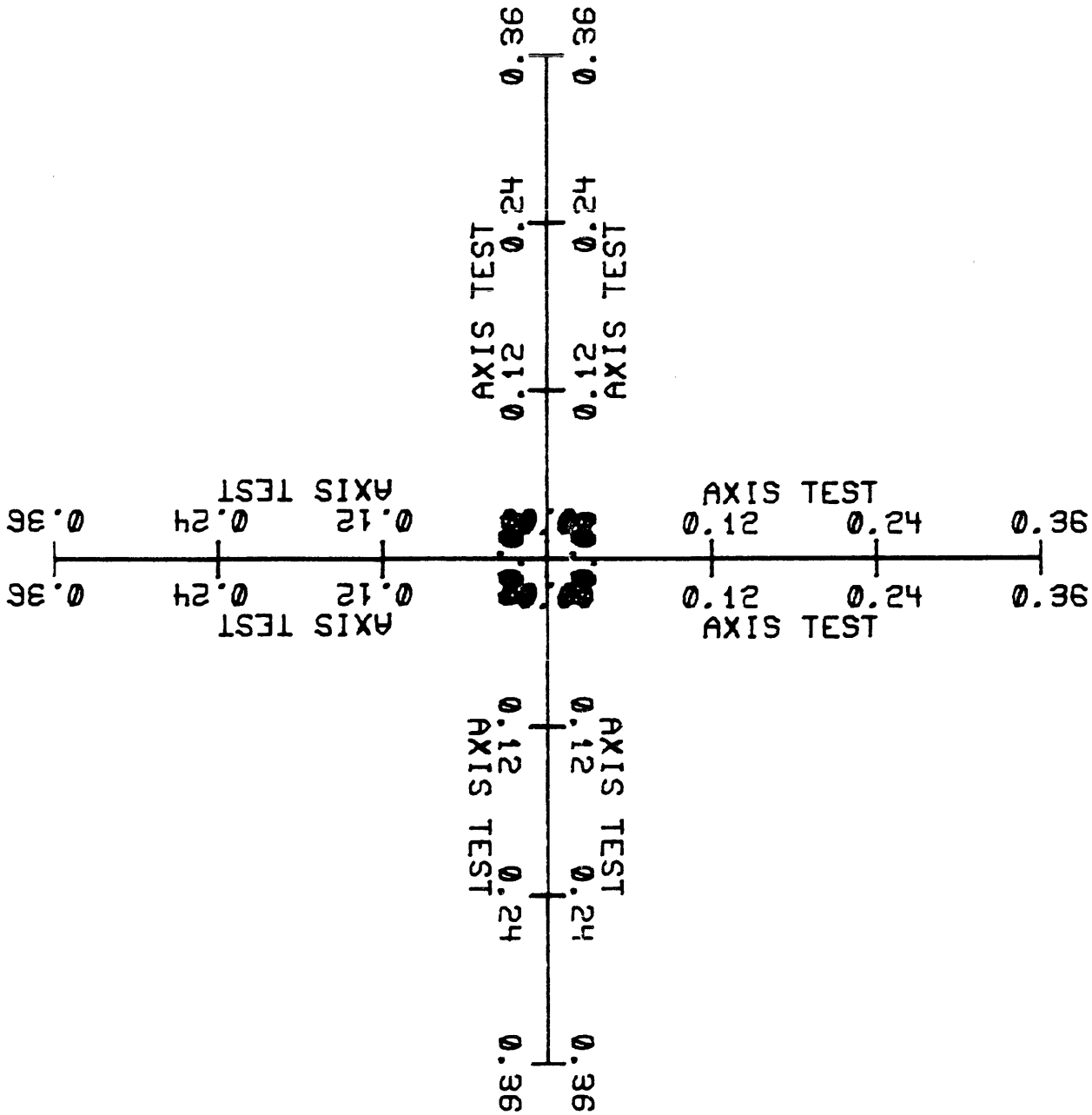
```
      DY=0.2
      DX=2.0
      XMIN=0.0
      YMIN=-1.4
      CALL PLOT(1.5,1.0,3)
      CALL PLOT(0.0,0.0,0)
      CALL PLOT(0.0,0.0,3)
      CALL PLOT(0.0,8.0,2)
      YY=2.0
      YFLT=YMIN
      DO 40 I=1,9
      CALL NUMBER(-.2,YY,-.14,YFLT,0.0,1)
      CALL PLOT(-.1,YY,3)
      CALL PLOT(0.0,YY,2)
      YY=YY+1.0
      YFLT=YFLT+DY
   40 CONTINUE
      CALL SYMBOL(-0.75,3.25,.14,KAB,90.0,12)
      CALL AXIS(0.0,2.0,LAB,-1,6.0,0.0,XMIN,DX,1)
      CALL SYMBOL(0.,3.5,.21,NAB,0.,12)
      CALL LINE(X,Y,NTOP,-1,5)
      CALL LINE(X,Y,NTOP,0,1)
      CALL SYMBOL(0.,-.5,.14,NAB,0.,35)
      CALL PST
      END
```

SAMPLE PROGRAM (Continued)

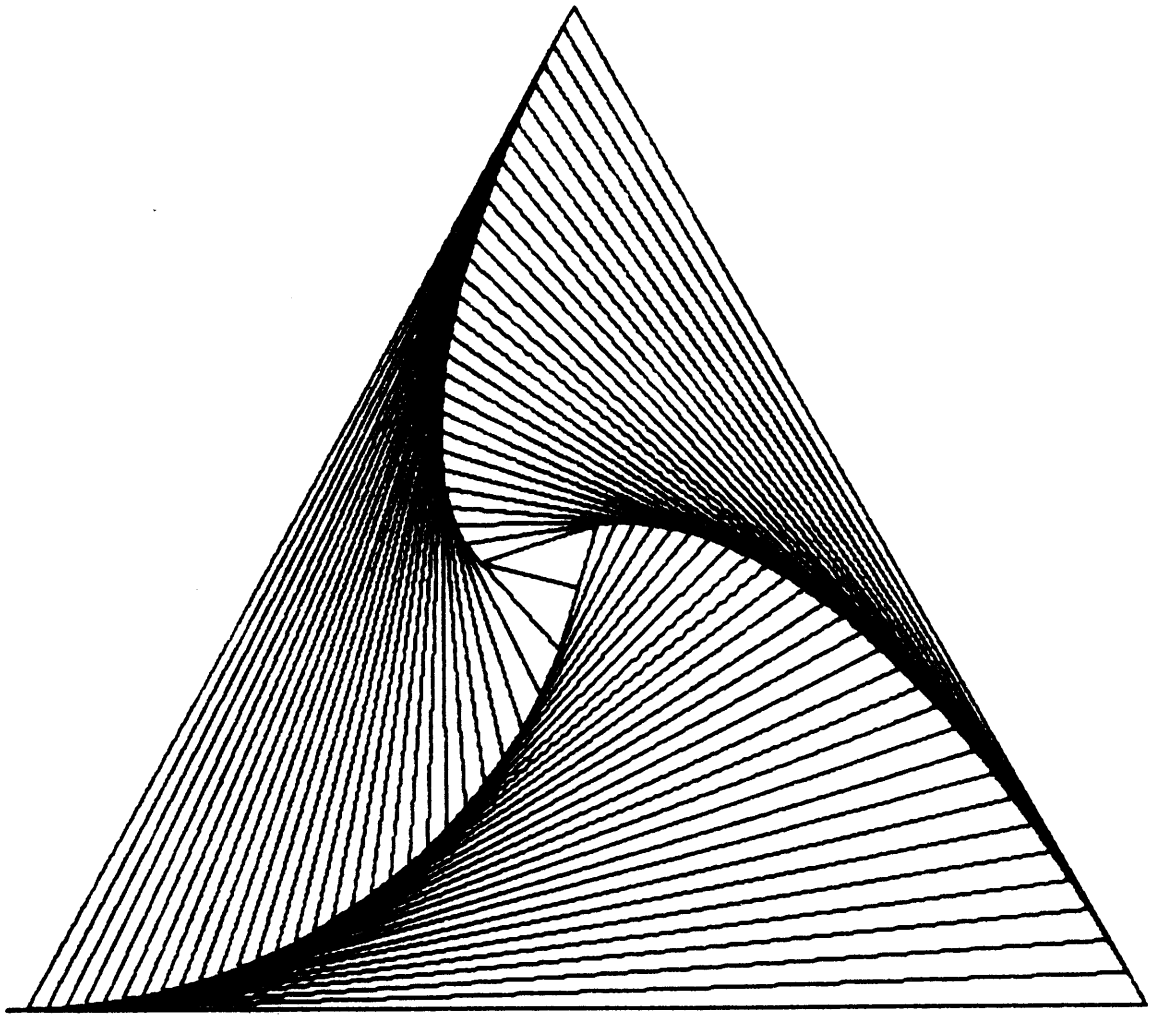


SAMPLE DATAPLOT OUTPUTS

Following are some sample outputs of DATAPLOT, showing some of the features of the program.



SAMPLE DATAPLOT OUTPUTS (Continued)



Following is a sample of all alphanumerics and symbols that can be output using DATAPLOT.

ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

[\]↑←!"#$%&'()*+,-./:;<=>@

cut along dotted line

DATA GENERAL CORPORATION
PROGRAMMING DOCUMENTATION
REMARKS FORM

DOCUMENT TITLE _____

DOCUMENT NUMBER (lower righthand corner of title page) _____

TAPE NUMBER (if applicable) _____

Specific Comments. List specific comments. Reference page numbers when applicable. Label each comment as an addition, deletion, change or error if applicable.

General Comments and Suggestions for Improvement of the Publication.

FROM: Name: _____ Date: _____
Title: _____
Company: _____
Address: _____
_____

FOLD DOWN FIRST FOLD DOWN

FIRST
CLASS
PERMIT
No. 26
Southboro
Mass. 01772

BUSINESS REPLY MAIL

No Postage Necessary If Mailed In The United States

Postage will be paid by:

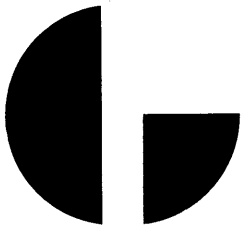
Data General Corporation

Southboro, Massachusetts 01772

ATTENTION: Programming Documentation

FOLD UP SECOND FOLD UP

STAPLE



DATA GENERAL
CORPORATION
Southboro,
Massachusetts 01772
(617) 485-9100