(2)

WL-TR-91-2004

AD-A242 870

CGRAPH - A FORTRAN CALLABLE GRAPHICS LIBRARY

Steven L. Puterbaugh

Technology Branch
Turbine Engine Division

July 1990

INTERIM REPORT FOR PERIOD 1 JAN 1988 - 1 MAY 1990

Approved for Public Release; Distribution Unlimited

Aero Propulsion and Power Directorate
Wright Laboratory
Air Force Systems Command
Wright Patterson Air Force Base, Ohio 45433-6563

91-15980

91 1120 011

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility nor any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise, in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report is releasable to the National Technical Information Service (NTIS). At NTIS it will be available to the general public including foreign nations.

Marvin A. Stibich

MARVIN A. STIBICH, TAM
Compressor Research Group
Technology Branch

Francis R. Ostdiek

FRANCIS R. OSTDIEK, Chief
Technology Branch
Turbine Engine Division

FOR THE COMMANDER

THOMAS J. SIMS, Director
Turbine Engine Division
Aero Propulsion and Power Directorate

If your mailing address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify WL/POTX, Wright-Patterson AFB, OH 45433-6563 to help maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION: Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for Public Release; Distribution Unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S): WL-TR-91-2004	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION: Aero Propulsion and Power Directorate; 6b. OFFICE SYMBOL (If applicable): WL/POTX	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code)	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION; 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
61102F	2307	S1	27

11. TITLE (Include Security Classification)

Cgraph - A FORTRAN Callable Graphics Library

12. PERSONAL AUTHOR(S)

S. L. Puterbaugh

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Interim Report	FROM JAN88 TO MAY90	1990 July	47

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	
01	01		Computer Graphics
01	03		

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

A description of and a user's manual for a FORTRAN callable graphics library is presented. The routines are written in FORTRAN 77 and developed on an IBM AT compatible microcomputer and a DEC MicroVAX minicomputer. The routines are organized so that many output devices may be supported.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☑ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code)	22c. OFFICE SYMBOL
Steven L. Puterbaugh	(513) 255-4738	WL/POTX

DD Form 1473, JUN 86 — Previous editions are obsolete.

PREFACE

This interim report was prepared by Steven L. Puterbaugh of the Technology Branch, Turbine Engine Division, Aero Propulsion and Power Directorate, Wright Laboratory, Air Force Systems Command, Wright-Patterson AFB, Ohio. The work was accomplished between 1 January 1988 and 1 May 1990.

The report represents results from a portion of the effort of the Compressor Research Group, supervised by Dr. Arthur J. Wennerstrom, and was conducted under Work Unit 27, Task S1, of Project 2307, "Turbomachinery Fluid Mechanics."

This report describes a FORTRAN callable graphics library which was developed in-house for use in preparing technical reports, interactive graphics, and miscellaneous illustrations. The motivation for this work came from the need to port graphics and 'graphics access' programs to various machines for use with various output devices. This was thought to be the most inexpensive, flexible approach.

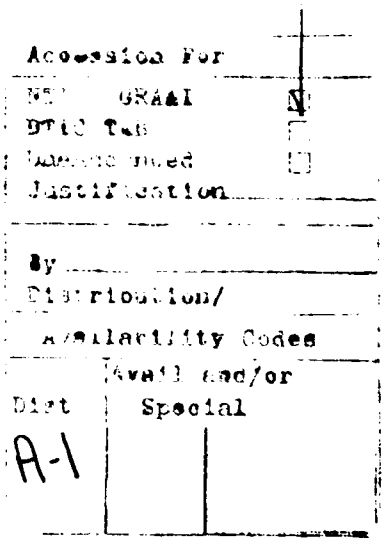

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

SECTION I

INTRODUCTION

This report describes a library of FORTRAN graphics subroutines and provides a guide for their use. The reader is expected to be a competent FORTRAN programmer.

The "Cgraph" library contains a set of subroutines which give the user the ability to create graphics output. The specific type of output may vary from simple line plots with labeled axes to contour plots to three-dimensional perspective plots (Figures 1-3). Any graphics output device may be used as long as the proper device dependent routine, or 'driver', is used. Specific information about the use and creation of drivers is included in subsequent sections.

The Cgraph subroutines are designed to require a minimum amount of information from the user (or user's program). A small number of subroutines may be accessed to create a rather elaborate plot.

The user must first determine the type of graphics output required, then write a FORTRAN program which contains calls to Cgraph routines. The source file would then be compiled and linked with the Cgraph library. The resulting executable module may then be used to create the desired output.

The motivation behind this work arose from the need for a very flexible, inexpensive graphics package. Since the software was developed in-house, the source is available for porting the package, or any portion of the package, to any machine (without need for a software license agreement) which contains a FORTRAN compiler. Computers which have run the Cgraph package include IBM PC/XT/AT and compatibles (MS-DOS FORTRAN), Digital MicroVAX III (VAX FORTRAN), and MODCOMP Classic II/15 (MODCOMP FR5

FORTRAN). The expense of the package development was minimal, the initial development requiring less than three man-weeks.

The Cgraph package is currently used by the group in many applications. These include plotting Phase I and II Data Reduction results for reports, compressor design system results, Data Acquisition System real-time performance map, and other miscellaneous small programs.

SECTION II

APPROACH

The approach to develop the Cgraph package was to implement a multi-level/modular design. Three 'levels' of routines were used which included 'user', 'intermediate', and 'low' levels. The names imply their uses, the user level routines are those which are normally accessed by the user's program. The typical user need not be aware of the existence of the other two levels. The intermediate level routines are those which are accessed by the user level routines. Their functions include computations, assigning values, and error checking among others. The low level routines, or 'device drivers', are those which interact directly with the output device. Therefore, these are the only routines which are device dependent, a different set of low level routines being needed for each device. The user and intermediate level routines access the device driver through a single routine using a predefined command convention.

This approach allowed a 'building block' procedure to be used in development. All graphics output was built upon 'primitive' functions peculiar to a given output device. Examples of these would be pen up, pen down, and pen movement commands for a pen plotter, or pixel on/off commands for a crt device. The low level routines use these command primitives to provide a 'standard' set of output capabilities to the higher level routines. In this way the higher level routines could be output device independent.

The user incorporates the Cgraph capabilities into his program during the link-edit step. The Cgraph library and the low level (or 'device driver') library corresponding to the desired output device must be linked with the user's object module to create the executable module.

SECTION III

USER'S MANUAL

1. FEATURES

Cgraph offers the user a wide range of capabilities. It is possible to create extremely simple plots consisting of several lines up to very complex labeled contour plots. The system was designed to be easy to use by providing a full set of parameter defaults which correspond to each output device along with a straight forward subroutine calling sequence.

Features include curve plotting with labeled axes (optional autoscaling), scalable characters (internally generated by Cgraph when needed), curve smoothing, three-dimensional plotting, and contour plotting.

2. USING THE SYSTEM

A. TYPICAL CALLING SEQUENCE

This subsection outlines the subroutines which are called during a typical plotting task. Detailed descriptions of all subroutines normally required by the user may be found in SECTION IV, SUBROUTINE DESCRIPTION/SECTION 1 'USER LEVEL DEVICE INDEPENDENT ROUTINES'.

Figure 4 illustrates some of the terms which will be used to identify specific items on the plotted page. The page can be optionally oriented in a landscape (longest page dimension horizontal) or a portrait (longest page dimension vertical) layout. The x axis is always coincident with the horizontal direction and the y axis with the vertical. The subplot area is that part of the page which will contain the user's plot, with the exception of title or axis labeling. It defaults to 50% of the page dimension in either direction, centered on the page.

The subplot area may be relocated by redefining the subplot origin, the lower left hand corner of the subplot area. Note that a location on the page is defined in terms of an (x,y) coordinate, in inches, relative to the 'page origin' (lower left hand corner of the page).

Access to the Cgraph library should begin with a call to the initialization routine PLTRDY. Two exceptions are OUTPUT and PICTUR which must be called before PLTRDY. The routine PLTEND is called at the completion of a plot. The remaining Cgraph subroutines may be called in any order but care must be taken so that information required by Cgraph is available when needed. For example, subroutine SCALE, which defines the plot scale factors, must be called before the curve drawing routine CURVE is called. The user is expected to keep up with his program's plotting needs.

Even though Cgraph provides default values for many parameters, the user will almost always want to redefine several of them. The following sequence of subroutine calls will very often be used.

PLTRDY	Initialize the plotting system
PAGE	Define the page and subplot limits
SCALE	Define the plotting scale factors
XLABL	Create the labeled x-axis
YLABL	Create the labeled y-axis
CURVE	Draw the curve
PLTEND	Finalize the plotting system

B. CURVE DRAWING

As discussed above, curves are drawn by supplying subroutine CURVE with (x,y) coordinates (in user coordinates), the number of coordinates, and a curve marker plotting option value. CURVE connects each point with a straight line unless the spline smoothing option has been requested by a previous call to subroutine SPLINE. If this option is invoked, a curve is drawn through each point using a cubic spline interpolation scheme. The curve

marker plotting option value determines how curve markers will be plotted. A specific curve marker is selected by setting the absolute value of this parameter corresponding to one of the curve markers in Figure 5. A value of 0 disables curve markers, the curve drawn with no markers. A value greater than zero directs CURVE to draw the markers along with the curve, a value of less than zero directs CURVE to draw the markers only.

An additional curve drawing capability exists, the ability to draw dashed curves. If subroutine DASH is called with the parameter set to .TRUE., the subsequent calls to CURVE will produce a curve with dashed lines. The length of the solid and blanked sections of the curve may be redefined by the user by calling subroutine DSHLIN.

C. PRINTED TEXT

Cgraph provides the user with the ability to plot character strings on the page. This is useful for creating a symbol legend, labeling curves, etc. Subroutine PRINT is used to plot character strings. The arguments include the coordinates of the reference point, relative to the page origin in inches, the character string and number of characters, and the display mode. The display mode defines how the string will be plotted relative to the specified coordinate. A value of one causes the string to be positioned to the upper left of the reference point, a fairly standard positioning scheme. Eight other display modes are available, as discussed under the subroutine description.

The height of the characters may be set by calling subroutine HEIGHT with the desired height in inches before calling PRINT. A bold character attribute may be selected by calling subroutine BOLD with a .TRUE. value parameter. Subsequent calls to PRINT will produce character strings using bold type.

A title may be plotted by calling subroutine TITLE. The required parameters are the

title character string, number of characters, and height of characters. The title is centered over the subplot area and centered heightwise between the top of the page and the top of the subplot area.

D. PLOT ENHANCEMENT

Cgraph provides the user with many ways to enhance his plot. To begin with, the plot axes may be labeled with ticks, tick labels, and titles. This is done through calls to XLABL, for the x axis, and YLABL, for the y axis. If real value tick labels are to be used, Cgraph automatically determines how many places to the right side of the decimal are to be plotted with a maximum of three. Exponential notation is not supported. Integer value tick labels may be requested by calling subroutine INTLAB before calling XLABL or YLABL.

The subplot area may be framed by a thick line by calling FRAME. A grid may be overlaid on the subplot area by calling GRID. Note that both of these routines must be called before calls to XLABL or YLABL and that both XLABL and YLABL must be called for GRID to execute.

The default page orientation is 'landscape' mode. The page orientation may be redefined by calling subroutine PICTUR with the appropriate argument. Please note that calls to PICTUR 'reload' the default plotting parameters. Therefore, PICTUR should be called immediately after PLTRDY.

It is possible to place several plots on a single page. The user would first define the subplot area (required for a single plot) with a call to PAGE. The subplot repositioning routines SPOABS or SPOREL may then be called to place the subplot area as desired by the user. Subsequent calls to CGRAPH routines, for example XLABL and YLABL, would then plot on the newly defined and positioned subplot area as shown in Figure 6.

E. THREE-DIMENSIONAL PLOTTING

Cgraph provides the user with three-dimensional (perspective view) plotting capability. The subroutines SCAL3D, VIEW3D, and CURV3D are used for three-dimensional plotting. Figure 7 illustrates the parameters defined by these routines. The routines must be called in the order SCAL3D, VIEW3D, and CURV3D. The subplot area defined by subroutine PAGE is still relevant.

An imaginary plane called the 'projection plane' , Figure 7, is located in space between the viewpoint and the figure to be plotted. An edge of the figure to be plotted, for example, is defined by a set of coordinates in three dimensions. Lines which connect these points and the viewpoint intersect the projection plane at various locations. These intersections are then connected to create the three-dimensional projection of the edge on the projection plane. The two-dimensional image on the projection plane is then the three-dimensional perspective plot produced by Cgraph.

F. CONTOUR PLOTTING

Cgraph provides two fairly simple contour plotting subroutines, CONTUR and CNTUR2. In each case, a mesh is required to be supplied with I number of x values and J number of y values and 'surface function' values defined at each of the intersections. CONTUR requires that one set of x values and one set of y values be defined, i.e., only horizontal and vertical lines (relative to the subplot area) can be used to define the mesh. This approach limits the user to plot over a rectangular area only. CNTUR2 allows the user to define each (x,y) coordinate of the mesh intersection, Figure 8.

3. INTERMEDIATE LEVEL ROUTINE ACCESS

Occasionally it is convenient for the user to access the intermediate level subroutines. An example would be the creation of a symbol legend for a plot. The intermediate level routines are all prefixed with 'QQ' so that their names will be unique and the required routines will be included during the link edit step. No special access information is required apart from the argument information identified under SECTION IV SUBROUTINE DESCRIPTION, INTERMEDIATE LEVEL DEVICE INDEPENDENT ROUTINES.

4. CURRENTLY SUPPORTED OUTPUT DEVICES

As of this writing, device drivers have been written for 7 different output devices. They include the Color Graphics Adapter (CGA) card, Enhanced Graphics Adapter (EGA) card, Hercules card, Western Graphtec MP 2300 xy plotter, Tektronix 4014 graphics terminal, Tektronix 4114 graphics terminal, DEC LN03/Plus laser printer, and PostScript printers. Specific information required for developing a device driver is found under SECTION V DEVICE DEPENDENT DRIVER DEVELOPMENT.

5. INSTALLATION AND IMPLEMENTATION

This section will describe the methods used to install and implement the Cgraph software on two different types of systems, a personal computer running MS-DOS and a Digital MicroVAX running Ultrix (Digital's version of UNIX). The overall approach is to upload the source, compile each subroutine, and build the library.

The user and intermediate level routines reside in a library named 'Cgraph' while the device driver routine(s) reside in separate libraries, each corresponding to a different output device. The user must link with the Cgraph library and the device driver library associated

with the required output device. For example, if a personal computer with an EGA display is being used for on-screen graphics, the user would link his object module with the Cgraph library and the EGA device driver library. If a Digital LN03 is connected to the computer, then he would link with the Cgraph library and the LN03 device driver library. The user's object code would not need to be changed even though two different output devices could be accessed (not simultaneously).

Examples of the MS-DOS (personal computer) command streams are as follows (note that ????? denotes a file name):

Compile the subroutine, creating an object module:

```
C:\SWDEV\FORTRAN\FOR1 ?????;
C:\SWDEV\FORTRAN\PAS2
```

Add the object module to the Cgraph library:

```
C:LIB CGRAPH +?????;
```

Compile the user's program and link with the Cgraph system (using the LN03 laser printer as the output device):

```
C:\SWDEV\FORTRAN\FOR1 ?????;
C:\SWDEV\FORTRAN\PAS2
SET LIB=C:\SWDEV\FORTRAN
C:\SWDEV\UTIL\LINK ?????,?????,nul.map,8087+FORTRAN+CGRAPH+LN03
```

Examples of the Ultrix (MicroVAX) shell scripts are as follows:

Compile the subroutine, creating an object module:

```
set tpath = $(0)
set tfile = $tpath:t$$
if ( $1 == 'all' )then
 ls -l *.f | awk '{print $8}' > $tfile
 set filelist = 'awk -F. '{print $1}' $tfile'
 foreach file ($filelist)
  echo compiling $file
  objgen $file
 end
 rm $tfile
```

```
else
 objgen $1
endif
```

Add the object module to the Cgraph library:

```
set tpath = ${0}
set tfile = /tmp/$tpath:t$$
set library = ../cgraph_lib
cd ../object
if ( $1 == 'all' )then
 if (-e $library) rm $library
 ls -l *.o | awk '{print $8}' > $tfile
 set filelist = `awk -F. '{print $1}' $tfile`
 foreach file ($filelist)
  ar q $library $file.o
 end
 rm $tfile
else
 ar r $library $1.o
endif
cp $library $library.a
ranlib $library.a
cd ../source
```

Compile the user's program and link with the Cgraph system (using the LN03 laser printer as the output device):

```
if ( ! -e $1.f ) then
  echo ftnxy cannot find $1.f
  goto end
endif
set lib = /usr/crgsw/graphics/cgraph_lib.a
set drvpath = /usr/crgsw/graphics
set drvlib = $drvpath/ln03_lib.a
if ( $#argv == 2 ) then
  if ( $2 == 'ln03' ) then
    set drvlib = $drvpath/ln03_lib.a
  else if ( $2 == 'tk4114' ) then
    set drvlib = $drvpath/tk4114_lib.a
  else if ( $2 == 'tk4014' ) then
    set drvlib = $drvpath/tk4014_lib.a
  else
    echo unrecognized graphics library specification
    echo
    echo recognized libraries are:
    echo   ln03 ..... digital ln03 laser printer
```

```
    echo   tk4114 ..... tektronix 4114 graphics terminal
    echo   tk4014 ..... tektronix 4014 graphics terminal
    goto end
  endif
endif
fort -o $1 $1.f $lib $drvlib
end:
```

SECTION IV

SUBROUTINE DESCRIPTION

1. USER LEVEL DEVICE INDEPENDENT ROUTINES

This section gives detailed descriptions of the uses of each user level subroutine and its parameters. Each subsection contains the description for a single routine beginning with the FORTRAN calling convention, then the purpose and use of the routine, and finishing with a description of its parameters.

AUTOSC(XMIN,XMAX,XINC,IERR)

Given a minimum and maximum of a range of values, AUTOSC computes rounded min and max values which bound the range and a well behaved increment value useable for axis definition. XINC will be chosen such that four to nine 'tick' marks could be drawn. Note that XMIN and XMAX will be redefined upon exit from AUTOSC.

ARGUMENT	TYPE	DESCRIPTION
XMIN	Real	Minimum value of range of interest on input, rounded value on output
XMAX	Real	Maximum value of range of interest on input, rounded value on output
XINC	Real	Computed 'best' increment
IERR	Integer	Error indicator, if 1 on output, an increment could not be determined, if 0 on output, then no error occurred

BOLD(DOIT)

Enable/disable bold character plotting for subsequent calls to subroutine PRINT.

ARGUMENT	TYPE	DESCRIPTION
DOIT	Logical	A value of .TRUE. turns on the bold character feature, a value of .FALSE. turns it off

CNTUR2(XM,YM,MESH,HGTS,UNUSED,P,Q,R)

Create a contour plot over a mesh of unevenly spaced x and y coordinates.

ARGUMENT	TYPE	DESCRIPTION
XM	Real	X coordinate of mesh point - p X q array
YM	Real	Y coordinate of mesh point - p X q array
MESH	Real	Value of three-dimensional function at each (X,Y) coordinate - p X q array
HGTS	Real	Values of r contour levels to be plotted - r length array
UNUSED	Logical	Work space - p X q array
P	Integer	Number of X-direction mesh lines
Q	Integer	Number of Y-direction mesh lines
R	Integer	Number of contour levels to plot

CONTUR(XM,YM,MESH,HGTS,UNUSED,P,Q,R)

Create a contour plot over a mesh of equally spaced x and y coordinates. Same arguments at CNTUR2.

CURV3D(X,Y,Z,NXYZ)

Draw a curve in three dimensions. CURV3D must be called after subroutines SCAL3D and VIEW3D.

ARGUMENT	TYPE	DESCRIPTION
X	Real	X coordinate of a point to be plotted - nxyz length array
Y	Real	Y coordinate of a point to be plotted - nxyz length array
Z	Real	Z coordinate of a point to be plotted - nxyz length array
NXYZ	Integer	Number of coordinate sets to be plotted

CURVE(X,Y,NPT,JMARK)

Draw a curve in two dimensions with optional curve markers.

ARGUMENT	TYPE	DESCRIPTION
X	Real	X coordinate of a point to be plotted - npt length array
Y	Real	Y coordinate of a point to be plotted - npt length array
NPT	Integer	Number of coordinate pairs to be plotted
JMARK	Integer	Curve marker indicator: JMARK < 0 - Curve will be drawn using curve marker number |JMARK| only (not connected by lines) JMARK = 0 - Curve will be drawn using lines only (no curve markers) JMARK > 0 - Curve will be drawn using both connecting lines and curve marker |JMARK|

DASH(DOIT)

Enable/disable dashed line plotting for subsequent calls to subroutine CURVE.

ARGUMENT	TYPE	DESCRIPTION
DOIT	Logical	A value of .TRUE. turns on the dashed line plotting feature, a value of .FALSE. turns if off

DSHLIN(SOLID,BLANK)

Specify the length of the solid and blank sections of a dashed line for subsequent dashed line plotting.

ARGUMENT	TYPE	DESCRIPTION
SOLID	Real	Length of the solid section of line (inches), default is 0.125 inches
BLANK	Real	Length of the blank section of line (inches), default is 0.063 inches

FRAME

Draw a thick line 'frame' around the subplot area. No arguments are used.

GETSP(X,Y)

Report the current sub plot origin coordinates.

ARGUMENT	TYPE	DESCRIPTION
X	Real	X coordinate of subplot origin
Y	Real	Y coordinate of subplot origin

GRID

Request a grid be plotted over the subplot area. GRID must be called before XLABL and YLABL. Both XLABL and YLABL must be called before the grid is actually plotted. No arguments are used.

HEIGHT(HITE)

Define the height of characters and curve markers in subsequent calls to PRINT and CURVE.

ARGUMENT	TYPE	DESCRIPTION
HITE	Real	Character or curve marker height in inches

INTLAB(DOIT)

Request axis labeling using integer values during subsequent calls to XLABL and/or YLABL.

ARGUMENT	TYPE	DESCRIPTION
DOIT	Logical	A value of .TRUE. turns on the integer labeling feature, a value of .FALSE. turns it off

MINMAX(XARR,NARR,XMIN,XMAX)

Determine the minimum and maximum values in an array. This capability is often handy in plotting situations.

ARGUMENT	TYPE	DESCRIPTION
XARR	Real	Set of values to be searched - narr length array
NARR	Integer	Number of values in the XARR array
XMIN	Real	The smallest value in XARR
XMAX	Real	The largest value in XARR

NEWPEN(NPEN)

Request subsequent plotting be done with a new color. This feature is obviously limited to output devices with color capability.

ARGUMENT	TYPE	DESCRIPTION
NPEN	Integer	Color identifier. If the output device is a multipen x-y plotter, NPEN is the pen bin number. If the output device is a crt device the following color assignment has been adopted:

0 ..... Black
1 ..... Blue
2 ..... Green
3 ..... Cyan
4 ..... Red
5 ..... Magenta
6 ..... Brown
7 ..... Light Grey
8 ..... Dark Grey
9 ..... Light Blue
10 ..... Light Green
11 ..... Light Cyan
12 ..... Light Red
13 ..... Light Magenta
14 ..... Yellow
15 ..... Bright White

OUTPUT(DEV,NCHAR)

Specify the destination of the plotting information, e.g. a port id for MS-DOS such as lpt3 or com2 or a filename for any system. Must be called before PLTRDY.

ARGUMENT	TYPE	DESCRIPTION
DEV	Char	Name of destination
NCHAR	Integer	Number of characters in DEV

PAGE(XGP,YGP,XSP,YSP)

Define the global plot (page) size and the subplot size.

ARGUMENT	TYPE	DESCRIPTION
XGP	Real	Page width (horizontal direction) in inches
YGP	Real	Page length (vertical direction) in inches
XSP	Real	Subplot width (horizontal direction) in inches
YSP	Real	Subplot length (vertical direction) in inches

PICTUR(TYPE)

Define the page orientation, either landscape (longest page dimension horizontal) or portrait (longest page dimension vertical). Must be called before PLTRDY.

ARGUMENT	TYPE	DESCRIPTION
TYPE	Char	Orientation specification, 'landscap' for landscape, 'portrait' for portrait (landscape is the default).

PLTEND

Finalize the plotting session. This gives the termination command sequence to the output device. No arguments are used.

PLTRDY

Initialize the plotting session. This initializes the output device and sets default parameters.

PRINT(X,Y,CHAR,NCHAR,MODE)

Plot a string of characters at a location on the page defined by the (X,Y) coordinate and the MODE identifier. Note that the coordinates are referenced to the lower left hand corner of the page. The MODE identifies the display mode of the string, i.e., where the string is located relative to the specified coordinate as described below.

ARGUMENT	TYPE	DESCRIPTION
X	Real	X coordinate of reference position
Y	Real	Y coordinate of reference position
CHAR	Char	Character string to be plotted
NCHAR	Integer	Number of characters in CHAR
MODE	Integer	Relative position indicator. Depending on the value of mode, (X,Y) is located at the beginning, middle, or end of the length of the string or at the bottom, middle, or top of the height of the string as shown in Figure 9.

SCAL3D(U,V,DRATX)

Define the scale factors for three-dimensional plotting. SCAL3D sets the proportions and position of the projection plane in space. SCAL3D must be called before VIEW3D.

ARGUMENT	TYPE	DESCRIPTION
U	Real	Horizontal dimension of projection plane (in inches)
V	Real	Vertical dimension of projection plane (in inches)
DRATX	Real	Ratio of the distance from viewpoint to projection plane to the distance from viewpoint to the origin.

SCALE(XMIN,XMAX,YMIN,YMAX)

Define the scale factors (inches/user units) for two-dimensional plotting.

ARGUMENT	TYPE	DESCRIPTION
XMIN	Real	X axis value at the origin in user units
XMAX	Real	X axis value maximum in user units
YMIN	Real	Y axis value at the origin in user units
YMAX	Real	Y axis value maximum in user units

SETERR(I1,I2,FILNAM)

Define error handler options including error indication and error log file name. The defaults for these values are: INDERR=1, IESHOW=1, FILNAM=PLERRS.DAT.

ARGUMENT	TYPE	DESCRIPTION
INDERR	Integer	Immediate error notification option. A value of 0 dictates that the user not be notified immediately of non-fatal errors, 1 dictates that non-fatal errors be logged to FILNAM, and 2 indicates that non-fatal errors be logged to the terminal device. IESHOW (Int): Final error notification option. A value of 0 dictates that the user not be notified at plotting completion of the occurrence of non-fatal errors. A value of 1 dictates that an error message be written to the screen if any non-fatal errors occurred during the previous plotting session.
FILNAM	Char	Error log file name (must be character*20)

SPLINE

Request that spline smoothing be used during subsequent calls to subroutine CURVE. Note that due to speed requirements and internal memory limitations, curves with greater than 100 points are not smoothed. No arguments are used.

SPOABS(X,Y)

Reposition the subplot origin to coordinate (X,Y) relative to the lower left hand corner of the page. For relative repositioning see SPOREL.

ARGUMENT	TYPE	DESCRIPTION
X	Real	X coordinate of new subplot origin (inches)
Y	Real	Y coordinate of new subplot origin (inches)

SPOREL(DX,DY)

Reposition the subplot origin by specifying a delta from the current location. For absolute reposition, see SPOABS.

ARGUMENT	TYPE	DESCRIPTION
DX	Real	Delta value to be added to the current subplot origin X coordinate. New coordinate will be X+DX.
DY	Real	Delta value to be added to the current subplot origin Y coordinate. New coordinate will be Y+DY.

TITLE(CHAR,NCHAR,HITE)

Create a plot title positioned above the subplot area.

ARGUMENT	TYPE	DESCRIPTION
CHAR	Char	Title character string
NCHAR	Integer	Number of characters in CHAR
HITE	Real	Character height of the title (inches)

VIEW3D(CVX,CVY,CVZ)

Define the viewpoint in space. The viewpoint is the position from which the viewer observes the object to be plotted. VIEW3D must be called after SCAL3D and before CURV3D.

ARGUMENT	TYPE	DESCRIPTION
CVX	Real	X coordinate of viewpoint relative to the origin of the object to be plotted. Varying CVX effectively rotates the object left and right.
CVY	Real	Y coordinate of viewpoint relative to the origin of the object to be plotted. Varying CVY effectively rotates the object up and down.
CVZ	Real	Z coordinate of viewpoint relative to the origin of the object to be plotted. Varying CVZ affects the distortion (apparent closeness) of the object.

XLABL(XLCHAR,NXLCH,XINC)

Create a labeled X axis.

ARGUMENT	TYPE	DESCRIPTION
XLCHAR	Char	Axis title character string
NXLCH	Integer	Number of characters in XLCHAR
XINC	Real	Axis tick interval

YLABL(YLCHAR,NYLCH,YINC)

Create a labeled Y axis.

ARGUMENT	TYPE	DESCRIPTION
YLCHAR	Char	Axis title character string
NYLCH	Integer	Number of characters in YLCHAR
YINC	Real	Axis tick interval

2. INTERMEDIATE LEVEL DEVICE INDEPENDENT ROUTINES

CCDRAW

Draw each segment of the current contour. Segments are defined as contour sections between labels. The segment information is retrieved from common memory. No arguments are used.

CCLABL(N,H)

At the approximate location of point n (sequential identifier on this contour), the position and angle of the label for contour level h is computed and plotted. Also the contour line is blanked where it interferes with the label.

ARGUMENT	TYPE	DESCRIPTION
N	Integer	Point identifier on this contour
H	Real	Contour level

CCLCAT(N,X,Y)

PURPOSE: Determine appropriate positions for labels on the current contour.

ARGUMENT	TYPE	DESCRIPTION
N	Integer	Point identifier on this contour
X	Real	X coordinate of label
Y	Real	Y coordinate of label

CCPRS2(IF,JF,IAF,JAF,XM,YM,MESH,P,Q,UNUSED)

Follow a contour of a given height through the mesh using a method of B.R. Heap.

ARGUMENT	TYPE	DESCRIPTION
IF	Integer	Starting point mesh ID in X direction
JF	Integer	Starting point mesh ID in Y direction
IAF	Integer	Delta (in mesh ID numbers) to look for first meshline 'crossing' in X direction
JAF	Integer	Delta (in mesh ID numbers) to look for first meshline 'crossing' in Y direction
XM	Real	X coordinates of mesh - p X q array
YM	Real	Y coordinates of mesh - p X q array
MESH	Real	Functional values at each mesh point - p X q array
P	Integer	Number of mesh points in X direction
Q	Integer	Number of mesh points in Y direction
UNUSED	Logical	Workspace

CCPRSU(IF,JF,IAF,XM,YM,MESH,P,Q,UNUSED)

Follow a contour of a given height through the mesh using a method of B.R. Heap. This routine is used for evenly spaced meshes. See CCPRS2 for argument description.

CCRCRD(X,Y,POLAR,FIRST)

Record coordinates on a contour line for processing label placements.

ARGUMENT	TYPE	DESCRIPTION
X	Real	X coordinate of current point on contour
Y	Real	Y coordinate of current point on contour
POLAR	Logical	Polar coordinate plotting option
FIRST	Logical	Buffer initialization indicator

CCSEGX(NF,XF,YF,NB,XB,YB)

Determine start and end point information for a given curve segment.

ARGUMENT	TYPE	DESCRIPTION
NF	Integer	Start point ID on this segment
XF	Real	Start point X coordinate on this segment
YF	Real	Start point Y coordinate on this segment
NB	Integer	End point ID on this segment
XB	Real	End point ID on this segment
YB	Real	End point ID on this segment

CCSTRT(NSTART,NN,NINC)

Determine the starting point, number of points, and increment between points for the labels on this contour.

ARGUMENT	TYPE	DESCRIPTION
NSTART	Integer	Label existence flag, a value of 1 indicates the existence of at least one label on this contour, a value of 0 indicates no labels
NN	Integer	Number of labels on this contour
NINC	Integer	Increment between point ID's for each label location

QQCMRK(X,Y,NPT,IMARK)

Plot a marker at each point on the curve.

ARGUMENT	TYPE	DESCRIPTION
X	Real	X coordinate of each point on the curve - npt length array
Y	Real	Y coordinate of each point on the curve - npt length array
NPT	Integer	Number of points in the curve
IMARK	Integer	Curve marker ID number

QQDCRV(X1,Y1,X2,Y2,I)

Draw a segment of a curve from (X1,Y1) to (X2,Y2) using dashed lines. This routine is called on a point-by-point basis by subroutine CURVE.

ARGUMENT	TYPE	DESCRIPTION
X1	Real	X coordinate of starting point
Y1	Real	Y coordinate of starting point
X2	Real	X coordinate of ending point
Y2	Real	Y coordinate of ending point
I	Integer	Point ID of current point

QQDFLT

Assign default values for several plotting parameters. No arguments are used.

QQDRAW(X,Y)

Draw a line from the current point to (X,Y).

ARGUMENT	TYPE	DESCRIPTION
X	Real	X coordinate of destination point
Y	Real	Y coordinate of destination point

QQERRH(ESTRNG,ISEV)

Process errors encountered during the plotting session.

ARGUMENT	TYPE	DESCRIPTION
ESTRNG	Char	80 character string describing the error
ISEV	Integer	Error severity indicator, a value of 1 indicates a fatal error, 0 indicates a warning error

QQFRAM

Draw a frame (thick line) around the subplot area. No arguments are used.

QQGRID(ID,NTICK,XYINC)

Draw a grid over the subplot area. Note that the grid is actually drawn the second time that QQGRID is called.

ARGUMENT	TYPE	DESCRIPTION
ID	Integer	X-Y caller indicator, a value of 1 indicates that XLABL is the caller, 2 indicates that YLABL is the caller
NTICK	Integer	Number of grid lines per tick
XYINC	Integer	Number of user units between ticks

QQICNV(IX,CHAR,NCHAR)

Convert an integer value to a character string.

ARGUMENT	TYPE	DESCRIPTION
IX	Integer	Integer value to be converted
CHAR	Char	Converted character string
NCHAR	Integer	Number of characters in CHAR

QQMARK(IMARK)

Draw a curve marker at the current pen position

ARGUMENT	TYPE	DESCRIPTION
IMARK	Integer	Curve marker ID number

QQMMLT(A,JA,IA,B,JB,IB,C,JC,IC)

Multiply matrices in the form A x B = C.

ARGUMENT	TYPE	DESCRIPTION
A	Real	'A' matrix - ja X ia array
JA	Integer	Number of rows in matrix 'A'
IA	Integer	Number of columns in matrix 'A'
B	Real	'B' matrix - jb X ib array
JB	Integer	Number of rows in matrix 'B'
IB	Integer	Number of columns in matrix 'B'
C	Real	'C' matrix - jc X ic array
JC	Integer	Number of rows in matrix 'C'
IC	Integer	Number of columns in matrix 'C'

QQMOVE(X,Y)

Change current pen position to (X,Y) relative to the lower left hand corner of the page.

ARGUMENT	TYPE	DESCRIPTION
X	Real	X coordinate of destination (inches)
Y	Real	Y coordinate of destination (inches)

QQRCNV(XIN,NDEC,CHAR,NCHAR)

Convert a real value to a character string.

ARGUMENT	TYPE	DESCRIPTION
XIN	Real	Real value to be converted
NDEC	Integer	Number of decimal places to be retained
CHAR	Char	Converted character string
NCHAR	Integer	Number of characters in CHAR

QQROT8(ANG)

Request that all subsequent character strings be plotted at an angle of ANG degrees relative to horizontal.

ARGUMENT	TYPE	DESCRIPTION
ANG	Real	Angle orientation (degrees)

QQSF(XINC,NDEC)

Determine the number of places to the right of the decimal to be retained by a subsequent character string conversion.

ARGUMENT	TYPE	DESCRIPTION
XINC	Real	Real value to be analyzed
NDEC	Integer	Number of decimal places to be retained

QQSPLN(XDATA,YDATA,NDATA,XIN,YOUT,SLOPE,NXY,NTYPE,NWOT)

Spline interpolate a value given the curve coordinates and the independent value of the point to be interpolated.

ARGUMENT	TYPE	DESCRIPTION
XDATA	Real	X coordinates of curve - ndata length array
YDATA	Real	Y coordinates of curve - ndata length array
NDATA	Integer	Number of (X,Y) coordinate pairs
XIN	Real	Value of independent variable - nxy length array
YOUT	Real	Value of dependent variable (interpolated value) - nxy length array
SLOPE	Real	Slope of curve at XIN (optional) - nxy length array
NXY	Integer	Number of interpolations to be done
NTYPE	Integer	Type of interpolation
NWOT	Integer	Slope calculation indicator

QQUDRW(X,Y)

Draw a line from the current point the point (X,Y) specified in user units.

ARGUMENT	TYPE	DESCRIPTION
X	Real	X coordinate of destination (user units)
Y	Real	Y coordinate of destination (user units)

QQUMOV(X,Y)

Change current pen position to (X,Y) specified in user units.

ARGUMENT	TYPE	DESCRIPTION
X	Real	X coordinate of destination (user units)
Y	Real	Y coordinate of destination (user units)

QQXPC(X)

Convert from user units to page units in the x direction.

ARGUMENT	TYPE	DESCRIPTION
X	Real	Value to be converted

QQXUC(X)

Convert from page units to user units in the x direction.

ARGUMENT	TYPE	DESCRIPTION
X	Real	Value to be converted

QQYPC(Y)

Convert from user units to page units in the y direction.

ARGUMENT	TYPE	DESCRIPTION
Y	Real	Value to be converted

QQYUC(Y)

Convert from page units to user units in the y direction.

ARGUMENT	TYPE	DESCRIPTION
Y	Real	Value to be converted

SECTION V

DEVICE DEPENDENT DRIVER DEVELOPMENT

1. DEVELOPMENT STRATEGY

The overall strategy involved in developing a device dependent driver should be to complement the Cgraph requirements with the 'on board' capabilities of the output device. For example, if the output device can generate its own character set, then the internal character generation feature of Cgraph (which is actually contained in the driver) need not be included. In other words, let the device do as much as it is capable to reduce the communications requirement.

Driver development begins with identifying the most fundamental output capability of the device, for example, pixel control for a crt device or vector control for a Tektronix device. More elaborate capabilities that Cgraph may require may then be 'built' from these 'primitives'.

The most advantageous approach to developing a new driver is to use an existing driver as a template and modify as required. The overall layout of each of the drivers developed to date are very similar and it is expected that this should remain so for most output devices.

2. DEVICE DEPENDENT DRIVER INTERFACE

All calls to the driver by the user and intermediate level device independent subroutines are made in the form:

QQDRVR(command,xx,yy,charstrg).

This is the device dependent driver interface. 'command' is a four character string which is one of a standard set of commands, or directives, recognized by the device driver. 'xx' and

'yy' are values which are passed to the driver, their functions differ depending on the command. 'charstrg' is a character string which is to be plotted as required. QQDRVR is the driver main subroutine, the same name no matter what the output device may be. Other subroutines may be accessed by QQDRVR, a naming convention prefixing 'ZZ' to the subroutine names has been adopted.

The driver commands are categorized as either information requests or action requests. The syntax of the commands are four characters (capital letters) with information requests always ending with a question mark. The information requests return information through xx and yy. A description of the driver commands and functions follows:

DRAW: 'Draw line' Directive. Draw a line from the current 'pen position' (defined during a previous call) to (xx,yy). The current pen position then becomes (xx,yy).

FINI: 'Plot termination' Directive. Complete the plotting process by transmitting a termination command sequence, close an output file, or halt execution for on-screen viewing depending on the output device.

HITE: 'Set character height' Directive. Set the current character height to xx inches.

INIT: 'Initialize plotting device' Directive. Initialize the plotting process by transmitting an initialization command sequence or open an output file depending on the output device.

MARK: 'Plot curve marker' Directive. Plot the curve marker denoted by xx at the current pen position.

MOVE: 'Change pen position' Directive. Physically move the pen to (xx,yy) if the output device is a plotter, update the current pen position to (xx,yy) if not.

NPEN: 'Change pen color' Directive. Begin plotting with a new color if the output device supports color. If the output device is a plotter, then the pen in bin number xx is loaded, if not, then the color associated with xx is used.

PICT: 'Change page orientation' Directive. Landscape orientation is invoked if xx equals 0.0 and portrait orientation is invoked if xx equals 1.0.

PGL?: 'Request page limits' Directive. The page limits consistent with the current page orientation is returned in xx and yy.

PMOD: 'Print mode' Directive. Subsequent calls for printing character strings shall be

made according to the mode identified in xx. This defines how the character string will be positioned relative to the location specified in the PRNT directive. Recognized values range from 1-9, see Figure 9 for details.

PRNT: 'Print character string' Directive. The character string contained in charstrg (end delimited by a '$') is to be plotted at (xx,yy) according to the currently active print mode.

ROT8: 'Define character string angle orientation' Directive. Subsequent character and marker plotting shall be done at an angle of xx degrees relative to horizontal.

SECTION VI

CONCLUSIONS

A FORTRAN callable plotting library, Cgraph, has been developed. It is adequate for many report and quick plot requirements. Though not as powerful as many commercially available packages, the fact that the FORTRAN source is available for modification and use in locally developed software makes it a very worthwhile package.

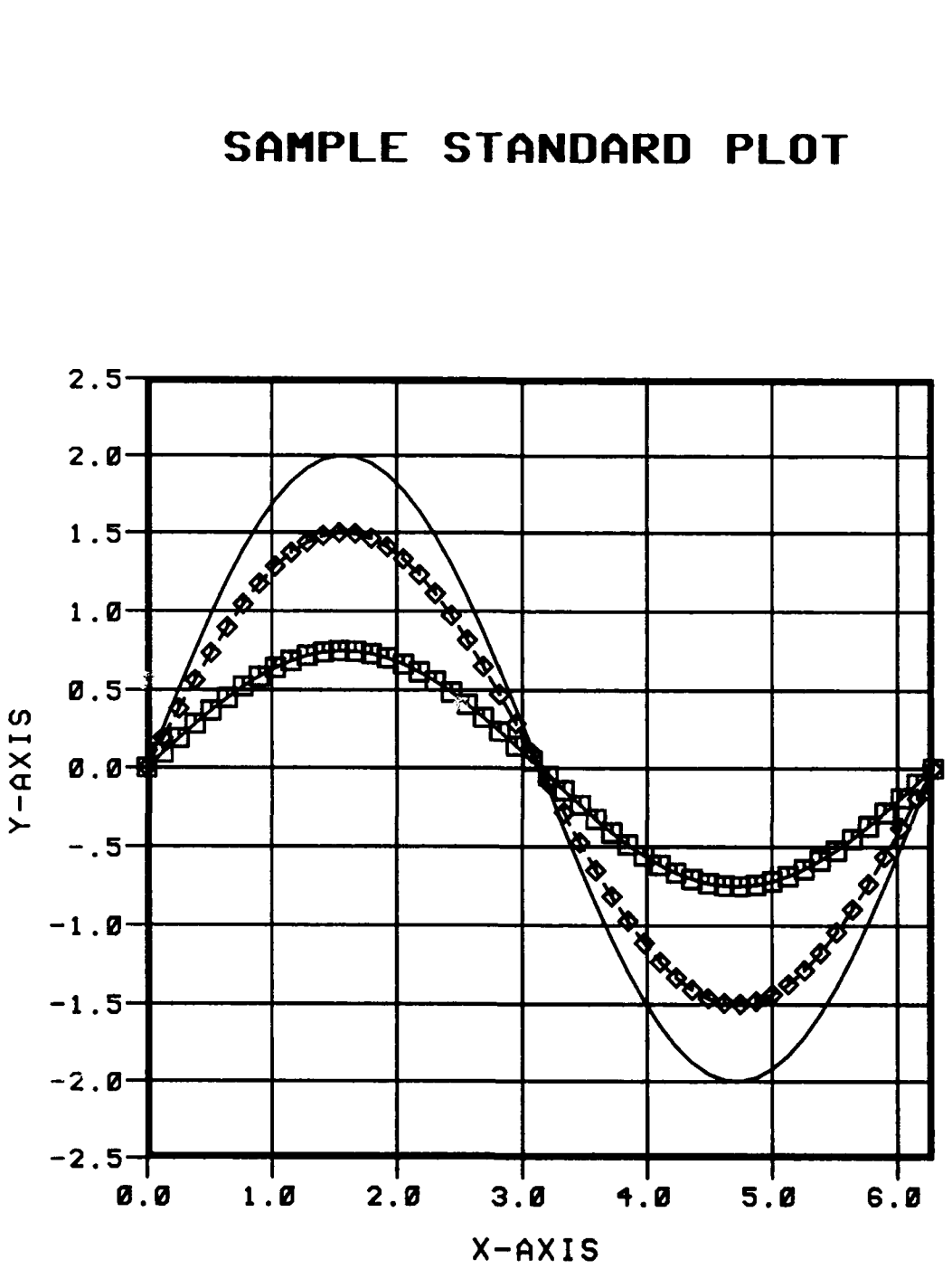

Figure 1. Standard plot example.

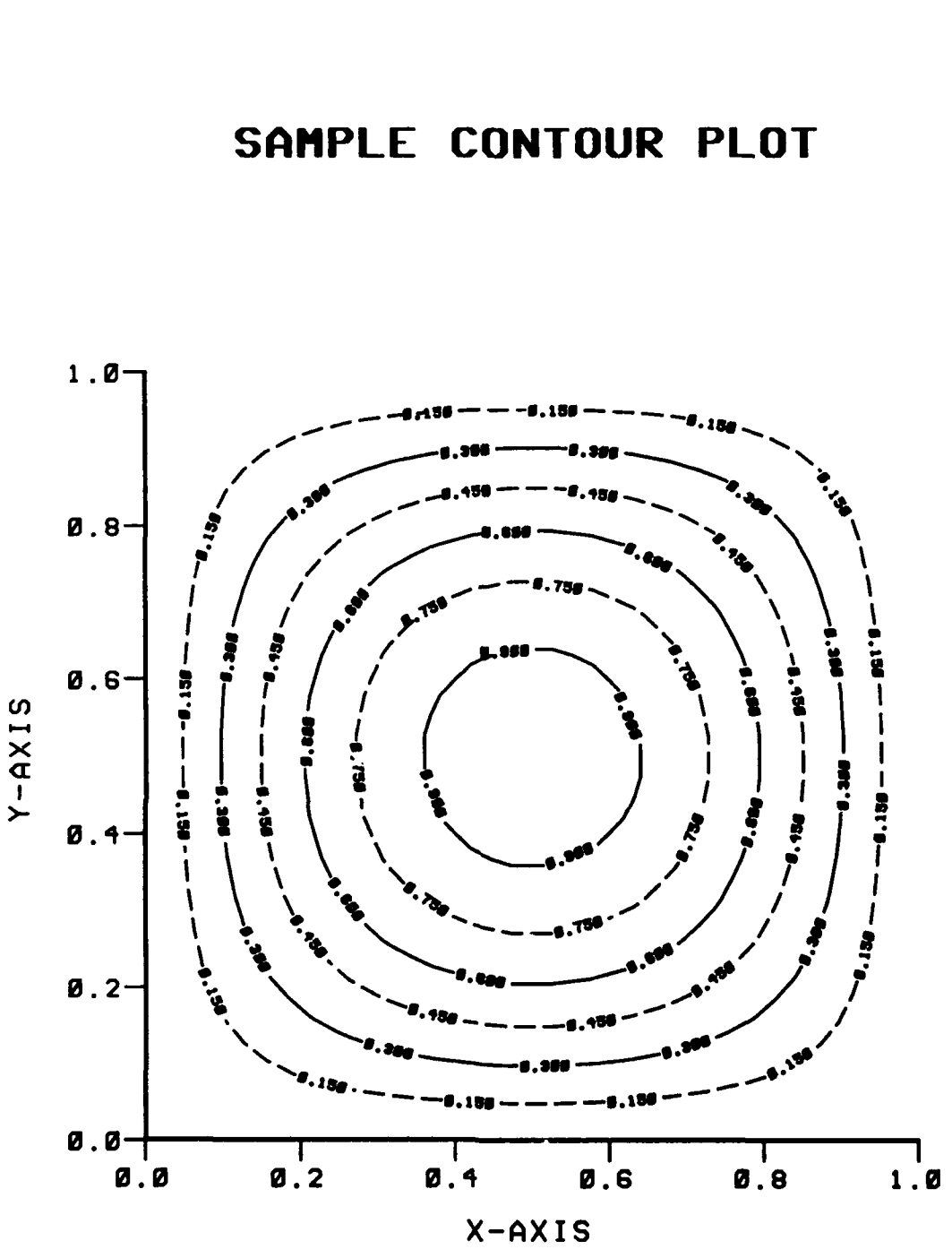

Figure 2. Contour plot example.

SAMPLE 3-D PLOT

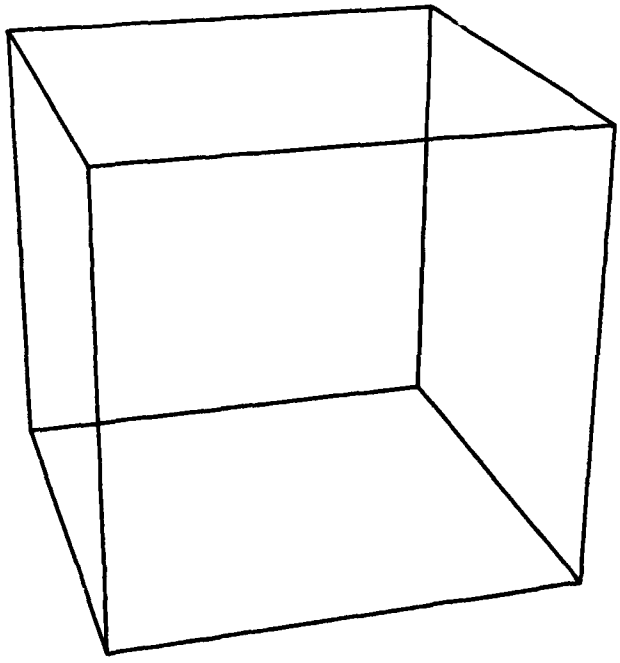

Figure 3. Perspective plot example.

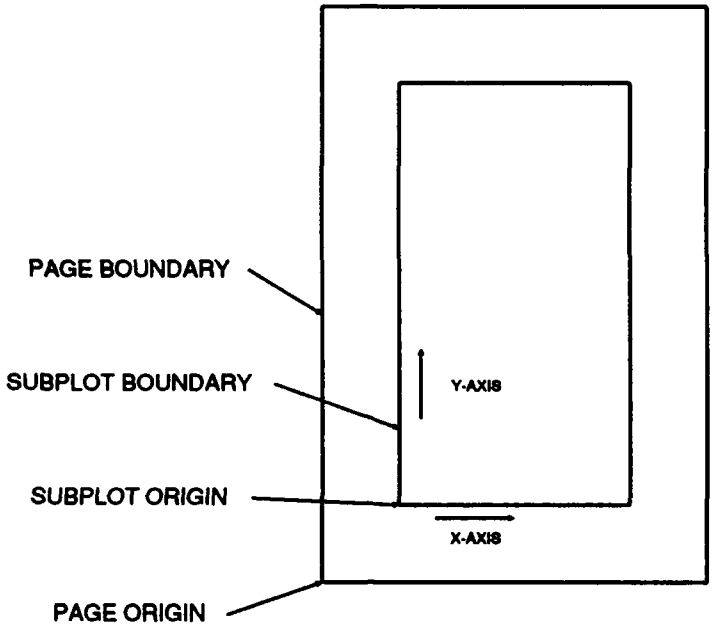

Figure 4. Plot page layout

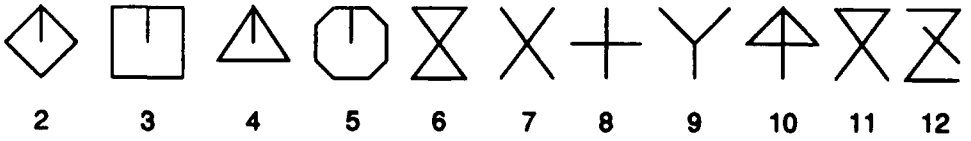

Figure 5. Curver markers

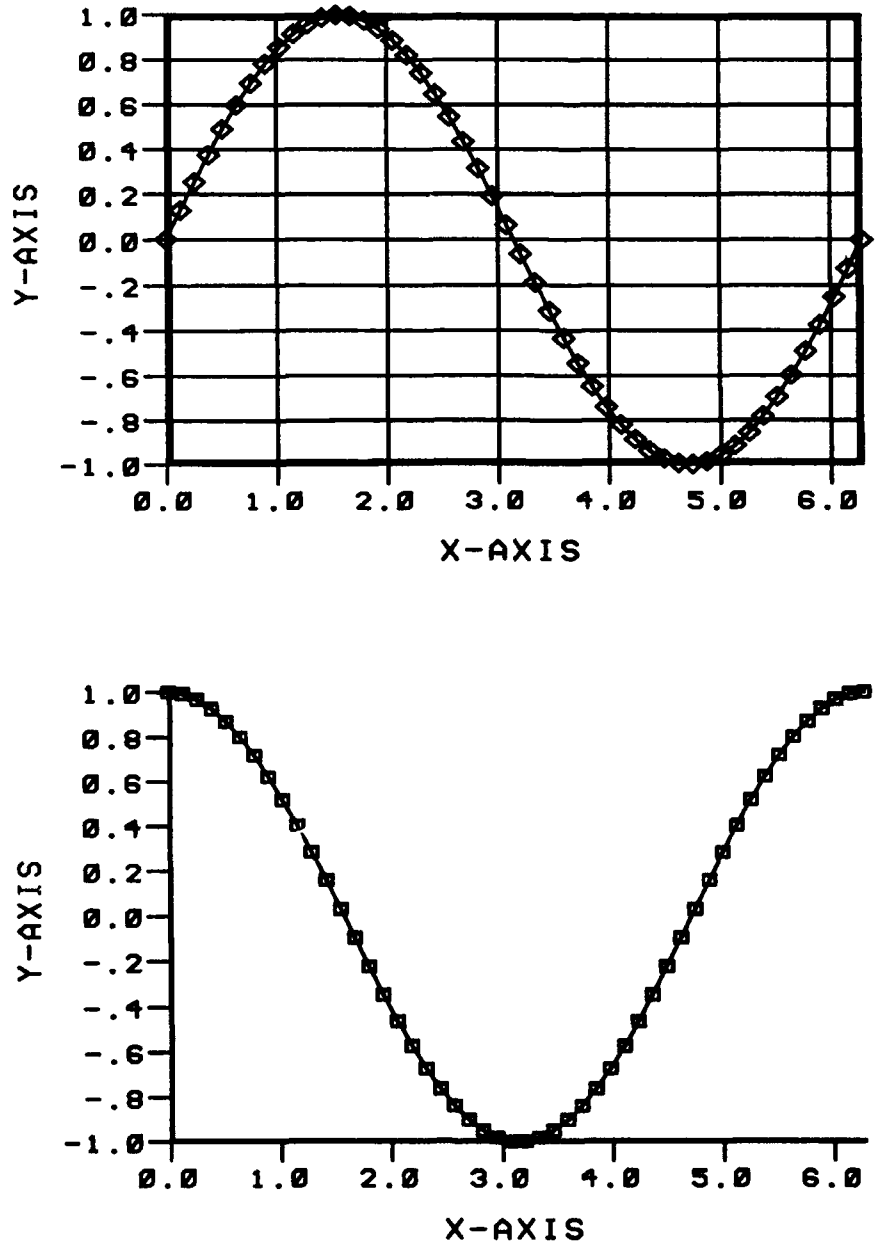

Figure 6. Multiple Plot Per Page Example

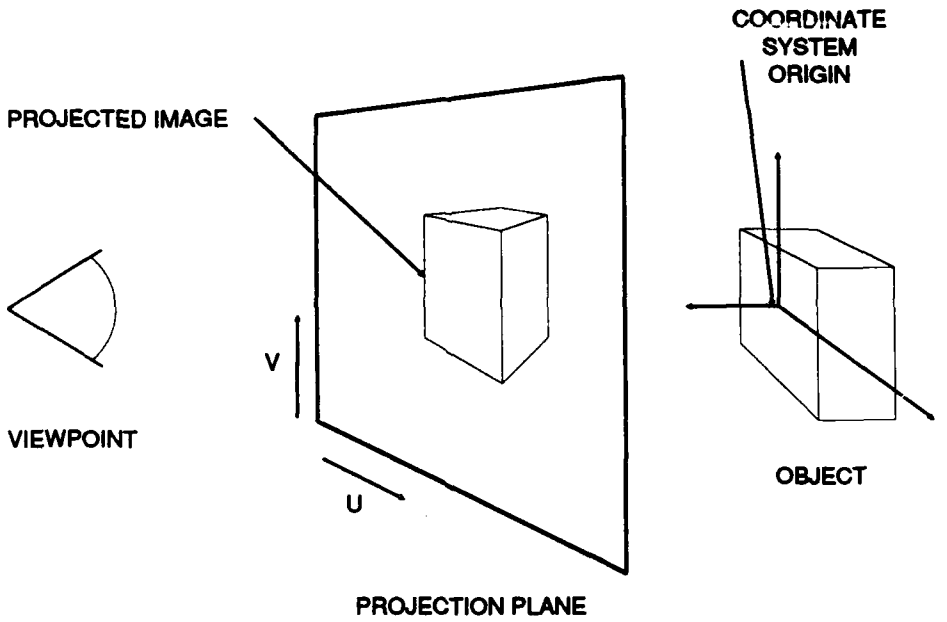

Figure 7. 3-D Plotting Parameters

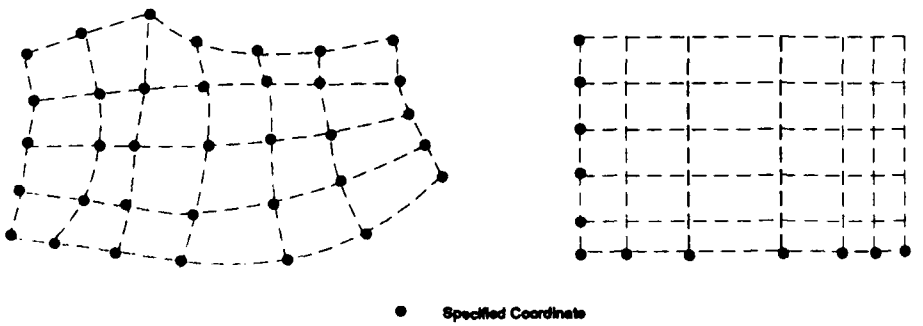

Figure 8. Meshes for Contour Plots

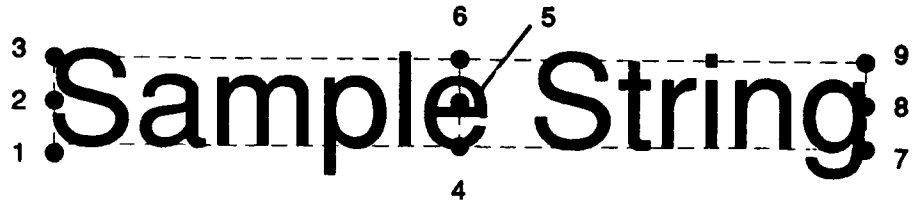

Figure 9. Print Mode Illustration.

APPENDIX A.

EXAMPLE PROGRAM FRAGMENTS

Fragments of programs used to create Figures 1-3 and 6.

```
c
c  Figure 1 of Cgraph report - standard plot example
c
      call pltrdy
      call page(11.0,8.5,5.0,5.0)
      call scale(0.0,6.28,-2.5,2.5)
      call frame
      call grid
      call xlabl('X-AXIS',6,1.0)
      call ylabl('Y-AXIS',6,0.5)
      call bold(.true.)
      call title('SAMPLE STANDARD PLOT',20,0.2)
      call bold(.false.)
      call curve(x,y,50,0)
      do 110 i=1,50
        y(i)=0.75*y(i)
110   continue
      call height(0.2)
      call dash(.true.)
      call curve(x,y,50,2)
      call dash(.false.)
      do 120 i=1,50
        y(i)=0.5*y(i)
120   continue
      call spline
      call curve(x,y,50,3)
      call pltend


c
c  Figure 2 of Cgraph report - contour plot example
c
c  determine the contour levels
      zmin=z(1,1)
      zmax=z(1,1)
      do 120 i=1,20
        do 120 j=1,20
          zmin=amin1(zmin,z(i,j))
          zmax=amax1(zmax,z(i,j))
120   continue
      call autosc(zmin,zmax,zinc,ierr)
      ninc=nint((zmax-zmin)/zinc)+1
      do 130 n=1,ninc
        h(n)=float(n-1)*zinc
130   continue
```

```
c  do the plotting
      call pltrdy
      call page(11.0,8.5,5.0,5.0)
      call scale(0.0,1.0,0.0,1.0)
      call xlabl('X-AXIS',6,0.2)
      call ylabl('Y-AXIS',6,0.2)
      call bold(.true.)
      call title('SAMPLE CONTOUR PLOT',19,0.2)
      call bold(.false.)
      call contur(x,y,z,h,wspace,20,20,ninc)
      call pltend


c
c  Figure 3 of Cgraph report - 3-d plot example
c
      call pltrdy
      call page(11.0,8.5,5.0,5.0)
      call bold(.true.)
      call title('SAMPLE 3-D PLOT',15,0.2)
      call bold(.false.)
      call scal3d(3.0,3.0,0.8)
      call view3d(7.0,3.0,2.0)
      do 100 i=1,4
        xi(i)=x(i,1)
        yi(i)=y(i,1)
        zi(i)=z(i,1)
100   continue
      xi(5)=x(1,1)
      yi(5)=y(1,1)
      zi(5)=z(1,1)
      call curv3d(xi,yi,zi,5)
      do 110 i=1,4
        xi(i)=x(i,2)
        yi(i)=y(i,2)
        zi(i)=z(i,2)
110   continue
      xi(5)=x(1,2)
      yi(5)=y(1,2)
      zi(5)=z(1,2)
      call curv3d(xi,yi,zi,5)
      do 120 i=1,4
        xi(1)=x(i,1)
        yi(1)=y(i,1)
        zi(1)=z(i,1)
        xi(2)=x(5-i,2)
        yi(2)=y(5-i,2)
        zi(2)=z(5-i,2)
        call curv3d(xi,yi,zi,2)
120   continue
      call pltend
```

```
c
c  Cgraph Report Figure 6
c
      call pltrdy
      call pictur('portrait')
      call page(8.5,11.0,4.0,3.0)
      call scale(0.0,6.28,-1.0,1.0)
      call spoabs(2.25,6.0)
      call bold(.true.)
      call title('MULTIPLE PLOT EXAMPLE',21,0.25)
      call bold(.false.)
      call grid
      call frame
      call xlabl('X-AXIS',6,1.0)
      call ylabl('Y-AXIS',6,0.2)
      call height(0.2)
      call curve(x,y1,50,2)
      call spoabs(2.25,1.5)
      call xlabl('X-AXIS',6,1.0)
      call ylabl('Y-AXIS',6,0.2)
      call curve(x,y2,50,3)
      call pltend
```


☆U.S. Government Printing Office:1991—648-127/62020