PRENTICE COMPUTER CENTRE

University of Queensland

Technical Manual No. 11
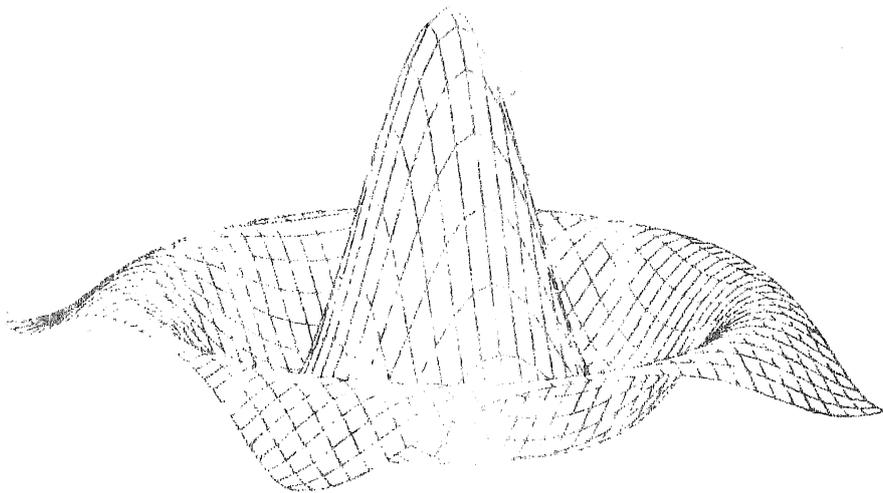
Digital Plotting Manual

PRENTICE COMPUTER CENTRE

University of Queensland

Technical Manual No. 11
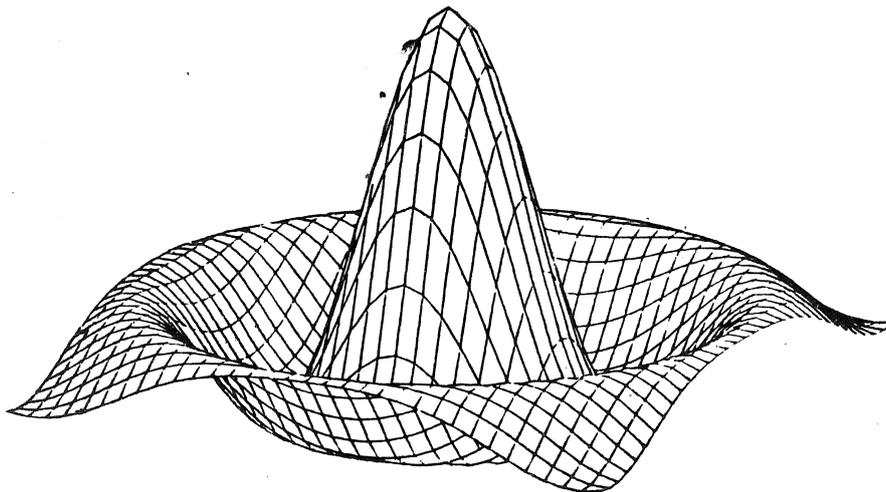
Digital Plotting Manual

PRENTICE COMPUTER CENTRE

University of Queensland

Technical Manual No. 11

Digital Plotting Manual

CHAPTER 1

DIGITAL PLOTTING

1.1 INTRODUCTION

This manual concerns the creation of graphical output generated by
the plotting module library CALF77 (and utility programs which use
CALF77) available on the computers at the Prentice Computer Centre,
University of Queensland.   By "graphical output", we mean any
graph, table, illustration or image which is produced on a screen
or on paper, by a computer. There are many examples of typical
"graphical output" in this manual.

In this manual, you will find discussions on the following aspects
of computer graphics generation, which should assist you in
producing your own "tailor-made" computer images:

    * plotting devices which are available at the Prentice
       Computer Centre,

    * personal plotting devices which you may own and wish to
       access from the Centre's computing facilities,

    * the CALF77 graphics subroutines, with which programmers may
       develop graphics programs,

    * procedures to follow for running your programs, on each
       mainframe computer system at the Prentice Computer Centre,

    * utility programs which can be used for conversion or
       transformation of plot files,

    * easy methods for generation of presentable graphs and
       tables.

In this manual, you will NOT find discussions on plotting packages such as

* TCS (PLOT 10 Terminal Control System from Tektronix)

* TEK GKS (PLOT 10 Graphical Kernel System from Tektronix)

* <PLOT79> (from University of Utah)

* SAS/GRAPH (from SAS Institute Inc.)

These packages are all documented in their own manuals, some of which are available online on some of the central mainframes. However, comments concerning queuing of program output for plotting should in most cases be equally applicable to these plotting packages.

This manual is aimed mainly at programmers who use FORTRAN or Pascal, but it will also be of some help to others who have no programming experience, but who wish to use the utility programs described in chapter 2.

The manual is intended to be quite comprehensive - that is, you should have to go no further than these pages in order to have your plotting requirements fulfilled. Of course, if you have any queries or comments, or if any particular problem arises which is not covered here, please contact the Prentice Computer Centre.

1.2   GRAPHICS CONVENTIONS

The most commonly used types of graphical output devices are plotters and display devices (cathode-ray tube, or CRT, devices). Even though these two types may appear to be quite different, they can be treated functionally in much the same way. Throughout this manual, we often use the terms "graphics" and "plotting" interchangeably, even when it is apparent that no plotter is involved.

1.2.1   The Pen

Computer graphics are produced by the movement of a "pen" across the plotting area. This "pen" may not be an actual pen at all - in CRT devices it is an electron beam, but we still refer to it as a pen.

In every type of plotting device, it must be possible to change the
colour of the pen, even if it is only from "visible" (a line will
appear if the pen is moved) to "invisible" (no change will be seen
after a pen movement).

1.2.2 The Origin

A line is produced in a plotter by the movement of a pen across the
surface of the chart paper, and in a graphics terminal by the
movement of an electron beam across the phosphor-coated screen. In
either case, the paper or screen is seen as a section of the
cartesian plane, with the origin of the X and Y axes located
initially in the bottom, left-hand corner.

1.2.3 The Coordinate Plane

In different types of graphics devices (be they plotters or CRT
devices), the area available for plotting will vary in size, but
will in every case be bounded. In order to prevent overwriting of
the plotter page, "hard-clip" limits are imposed on the position of
the pen and an error condition will occur if these bounds are
crossed. X and Y bounds equivalent to the paper or screen size are
imposed.

1.2.4 The Window

A "window" is a subset of the co-ordinate plane within which
plotted images will appear. At window boundaries all plotted lines
are "clipped", or cut off, so that they don't appear outside of the
window.

While a plotting window may be specified by the user, there will
always be a default, or implied, window corresponding to the
"hard-clip" limits of the graphics device.

1.2.5 More Conventions

Following is a list of the graphics conventions which have been
adopted in the CalComp standard.

(a)   The origin is initially at the bottom left of the plotter
       paper.

(b)   All coordinates are expressed in millimetres (floating
       point) relative to the current origin.

(c)   All coordinates relative to the initial origin must be in
       the range:

$$0.0 <= X <= XMAX$$
$$0.0 <= Y <= YMAX$$

       where XMAX and YMAX are the plotter bounds which are
       determined by the paper dimensions, or by the user.

(d)   The user coordinates are converted by the plotting
       routines to an integral number of plotter steps (the size
       of which depend upon the plotter).   To minimize the
       accumulation of rounding errors as far as possible, the
       plotter routines reconvert these values to floating point
       millimetres and the current stored position of the pen is
       updated using these reconverted increments.

(e)   All angles are expressed in degrees, (floating point) with
       the positive sense in a counter-clockwise direction from
       the positive X axis.

1.3 PLOTTING HARDWARE

1.3.1 PLOTTERS AVAILABLE AT THE PRENTICE COMPUTER CENTRE

There are several plotting devices at the Prentice Computer Centre
which are available to users for the production of graphical
output. With the exception of the Tektronix equipment (interactive
DECsystem-1099 and VAXen use only), each device can be accessed
from any of the mainframe computer systems, in much the same way as
the line printing devices are accessed.   That is, the user or
user's program creates a file of plot commands which is then
submitted, using the "PLOT" command, to a plot queue. The queue is
processed sequentially and the user's request is eventually
fulfilled.

Of course, the set of devices owned by the Prentice Computer Centre
is quite dynamic and is likely to be extended or reduced at times
in the future. The devices which were available at the time of
publication of this manual are described in the following sections.

1.3.1.1 Hewlett-Packard 7585 - The HP7585 is an accurate,
high-speed plotting device, which will accept paper in sizes up
to A0 (available plotting area of 810mm by 1134mm). Paper is
available in several types and sizes, and the plotter's carousel
will hold up to 8 pens, which may be selected from an array of
many types and colours. See Appendix A for further details of
stationery for this plotter.

This plotter is known as "PLT3" or "HP7585".

1.3.1.2 Hewlett-Packard 7475 - The HP7475 is a smaller version
of the HP7585, described above. It will accept paper of up to A3
size and has a carousel which holds 6 pens.

There are two HP7475 devices which can be accessed from the
computers at the Prentice Computer Centre - one (known as "PLT4"
or "HP7475") is located within the Centre itself and the other
(known as "PLT5") is located within Computing Services at
Griffith University.

1.3.1.3 Tektronix Terminals - The Prentice Computer Centre has a
Tektronix 4014 terminal and an attached Tektronix 4611 plotter
which is available for hardcopies of screen images. For access
to it, users should enquire at the Help Desk at the Prentice
Computer Centre.

1.3.2 PERSONAL GRAPHICS DEVICES

It is possible to use other devices - graphics terminals or
plotters of your own - with the plotting software at the Prentice
Computer Centre. Graphics terminals can be connected to the
mainframe systems in the normal way, while personal plotters can be
connected in series with a normal terminal.

Descriptions of some of these devices follow.

1.3.2.1 Tektronix terminals - Tektronix graphics terminals fall
into one of two categories. The 4010, 4014, and 4015 models
consist of a "Direct View Storage Tube" (or DVST), on which
images and text can be displayed, but not selectively erased;
while the 4105 and 4107 are "raster-scan" devices.

1.3.2.2 DEC terminals - Digital Equipment Corporation (DEC)
manufactures a variety of terminals which have graphics
capabilities. Among these are the GIGI and VT240 series
terminals, which differ from DVST devices (above) in that they
are "Raster-Scan" devices and can be selectively erased (simply
by "over-plotting" with a pen of a suitable colour, such as
black).

1.3.2.3 Remote Plotters - The HP7585 and HP7475 plotters, which
are described above, and the Tektronix 4662 plotter can all be
accessed remotely from the computers at the Prentice Computer
Centre by connecting them, in series, to a normal terminal.
Plotting commands can then be sent to the plotter, simply by
doing normal writes to the terminal, and in addition,
communication can be switched between the plotter and the
terminal, using applicable commands. For more information, see
discussion of the TERMON and TERMOF subroutines in Chapter 3.

In the case of the Hewlett-Packard devices, connection must be
done with a special part (a 17455A cable) purchased from the
manufacturer, but for the Tektronix, an extra RS-232 cable is all
that is required. For more details on the connection of these
plotters, see the applicable manuals.

These graphics devices may be used interactively from the
DECsystem-1099 or the VAXen, either under program control, or by
typing out the plot file. On the IBM 3083, plot files may be
viewed (with the exception of Tektronix 4010, 4014 and 4015
terminals) by TPRINTing the plot files.

1.4 PLOTTING SOFTWARE

The software which you use to obtain plotted output from a computer
will fall into two obvious categories - these are, programs which
you write yourself to fulfill your own specific need; and programs
which are written by others to fulfill more general requirements.
In this manual, both categories are covered.

1.4.1 USER PROGRAMS

To aid programmers in the production of their own graphics application programs, there is a large library of useful graphics subroutines which is available on a number of mainframes and minicomputers. Instructions for accessing this library on various computers are given in Chapter 2, and the subroutines themselves are described in Chapters 3 and 4.

1.4.2 UTILITY PROGRAMS

For those who have not the time, inclination or patience to undertake the creation of their own graphics programs, there are a small number of programs available which allow you to create with ease graphical computer output. These are described in Chapter 2.

You may find that although easy to use, such programs are not flexible enough to provide exactly the plot which you want, and in this case, custom programming may be necessary.

1.5 PLOT FILES

Since the information stored in a plot file is dependent on the plotter for which it was produced, it is necessary that a plot file be sent to the correct device for plotting.

The spooling programs will queue to the correct plotter any files which have been incorrectly queued, although this may not happen until a considerable time after the PLOT command has been issued. To minimize the elapsed time between issuing the PLOT command and the completion of plotting, all plot files should be queued to the plotter for which they were created. The exact commands to use, on each of the mainframe computers, will be explained in the following chapter.

When entering files in the plotter queues on the DECsystem-1099, the QUEUE program will insert as part of the queue entry a maximum time limit to be applied when plotting the files. If the user fails to specify a limit, then a limit will be calculated from the size of the plot file, but in some cases this limit can be wildly inaccurate. A small file which plots long straight lines may require a time limit many times greater than the limit calculated from its size and a large plot file in which pen movements are small may finish plotting much sooner than expected.

[This page intentionally left blank.]

CHAPTER 2

GENERAL INFORMATION

2.1 INTRODUCTION

This chapter contains some general instructions and advice to the
reader concerning the use of the Prentice Computer Centre's
graphics facilities. It contains general instructions in the use
of various graphics-oriented utilities and application programs and
gives outlines of compiling, linking, and execution procedures for
creating your own graphics programs on the various hosts (main
computers) at the Centre.   Finally, a   section   on   general
considerations and precautions should help you with any problems or
"trouble spots".

2.2 USING CALCOMP SUBROUTINES

The CalComp subroutines, contained in the CALF77 library, are
available to users on the UQKL10, UQVAX and UQVM hosts, as well as
to those who use PDP-11 RSX systems.   The following sections
describe how to access the routines on each of these systems.

2.2.1 On The DECsystem-1099 (Host UQKL10) ...

On the Prentice Computer Centre's DECsystem-1099 (commonly known as
the KL, the KL-10 or the DEC-10), there are three versions of
FORTRAN which can be used, but it is important to use only
FORTRAN-77 to compile a program which calls the CalComp routines,
since those routines are themselves written in FORTRAN-77.

The easiest way to ensure that FORTRAN-77 is used to compile your
FORTRAN program is to issue the following monitor command:

    .PATH /NEW

This command tells the system always to take the latest versions of
any program (for instance, the FORTRAN compiler) which you request.

To run a main program which calls any of the CalComp subroutines it
is necessary to give one of the following monitor command sequences
(Where "prog" is the name of your FORTRAN application).

    .EXECUTE prog,PLO:CALF77/SEARCH

       or

    .COMPILE prog
    .LOAD prog,PLO:CALF77/SEARCH,...
    .SAVE prog
    .RUN prog

The reference to the file PLO:CALF77 in the LOAD command indicates
to the system that our program contains references to subroutines
contained in that library file.

2.2.2 On The VAX 11/780 (Host UQVAX) ...

Unlike the DEC-10, which was discussed on the previous section, the
VAX-11 has a consistent subroutine calling standard for all of its
languages. This allows you to call the CALF77 routines from
languages apart from FORTRAN-77. At the time of printing of this
manual, the languages from which CALF77 subroutines could be called
were FORTRAN and Pascal.

FORTRAN programs:

On the VAX there is only one version of FORTRAN, which is run when
you issue either of the VAX DCL commands, "COMPILE" or "FORTRAN".
It is the correct version to use with the CALF77 library.

The graphics subroutine library is contained in the file
CALF77.OLB, on an area called SYS$LIBRARY. The following commands
should be given to compile the program, access the library and link
in the subroutines.

```
$ FORTRAN prog
$ LINK prog,SYS$LIBRARY:CALF77/LIB
```

This will create an executable image file, called "prog.EXE", which
can be run in the normal way, as follows.

```
$ RUN prog
```

Pascal programs:

In VAX Pascal, there is a mechanism for reading a library of
external const, var, type, procedure and function definitions, for
inclusion in your code.
You must insert the following line in your program, just before the
"Program ..." line, and thereafter you may use normal Pascal
procedure calls to access the routines.

```
INHERIT [('SYS$SHARE:CALF77')]
```

Some minor changes have had to be made to the CALF77 routines, to
allow compatibility with Pascal. For instance, the name of the
routine "LABEL" has had to be changed to "LABLN", because "label"
is a reserved word in Pascal.
Other changes, mostly minor, are explained in a help file on the
VAX, which can be read by using the command:

```
$ TYPE (or PRINT) SYS$DOC:PASCALF.DOC
```

To compile and link a graphics application written in Pascal, you
should use the following DCL commands.

```
$ PASCAL prog
$ LINK prog, SYS$LIBRARY:CALF77/LIB
```

2.2.3 On the IBM (Host UQVM) ...

On the IBM system, CALF77 routines can be called from Fortran or
Pascal programs. The CALF77 routines are in PLOTLIB TXTLIB which
is invoked automatically by FORTVS EXEC.

Compile your program using a command like:

```
FORTVS fname ftype fmode
```

Then load and run your program with a command like:

        LOAD fname (START CLEAR

Of course, you may wish to use some of the options to the FORTVS
command, and if you don't know much about them (or about FORTVS)
then say

        INFO FORTVS

2.2.4 On PDP-11 (RSX-11) systems ...

For PDP-11 systems (running the RSX-11M or RSX-11M+ operating
system), the CALF77 library had to be extensively revised, with
alterations made to several of the subroutines.   Development was
still under way at the time of publication of this manual, and
before attempting to use CALF77 at all, you should consult either
your systems manager, or the Prentice Computer Centre, for advice
as to the current status of the library on PDP-11 systems.

If you do decide to use the library, it should be present on the
LB:[1,1] area on your PDP-11 system and you can link it to your
FORTRAN-77 program in the normal way, using the TKB or LINK
command.

   From DCL, use the following command:

     $ LINK prog,LB:[1,1]CALF77/LIB/CODE:FPP

   From MCR, use the following command:

     >TKB prog=prog,LB:[1,1]CALF77/LB/FP

Note: The /CODE:FPP and /FP switches to LINK and TKB must be used
to instruct the linker to include floating-point processor support.

2.3 GRAPHICS UTILITIES AND APPLICATION PROGRAMS

2.3 The VUEPLT Program

VUEPLT is a utility program which will convert a plot file which
was written for a particular type of plotter into a file for
another type of plotter, or alternatively, will display it on the
screen of a graphics terminal.

To run VUEPLT on the DECsystem-10, type the command:

     .RUN PLO:VUEPLT

and to run it on the VAX, type:

     $ RUN SYS$UTI:VUEPLT

You will be prompted firstly to enter the device name, which can be
any string which follows the format of the DEVICE field in a call
to PLOTIN (see Section 3.2.1).   Secondly, you should enter the name
of the plot file which is to be converted and then a scaling factor
for the output plot.

Note that VUEPLT is NOT available on the IBM system.

For further information on the use of VUEPLT, on restrictions and
on possible bugs and problems, see Appendix D, which contains a
copy of the on-line help file for VUEPLT.
To access this information on-line, type one of the following
files:

     .TYPE DOC:VUEPLT.DOC, on the KL-10, or
     $ TYPE SYS$DOC:VUEPLT.DOC, on the VAX.

2.4 GENERAL CONSIDERATIONS AND WATCHITS

2.4.1 Versions of FORTRAN on the KL10

The extended CalComp routines in the CALF77 library were written in
FORTRAN-77. Hence, they can only be linked to programs written in
FORTRAN-77 (or PASCAL on VAX or IBM systems). This won't cause
VAX/VMS or IBM users any problems, as that's the only FORTRAN
available on the these machines, but on the KL-10, there are three
versions of FORTRAN.   To ensure that you use the right one, the
easiest thing is to give the monitor command,

        .PATH/NEW

From then on, the FORTRAN you get will be FORTRAN-77.

2.4.2 FORTRAN Units

The plotting routines reserve FORTRAN's unit 1 for internal use at
all times. Your applications programs must leave this unit alone,
if they are to work correctly.

2.4.3 Using /TERMINAL

The "/TERMINAL" switch is available on all implementations of
CALF77 except the IBM implementation.

When you use the "/TERMINAL" switch in your call to PLOTIN, the
plotter and terminal are set up to interpret all characters
received as plotter commands. If, while in this mode, your program
tries to output textual information to the terminal, the likely
result will be a lot of meaningless scribble appearing in and
around your plot.

To prevent this, you must tell the software when you wish to output
text, and then tell it when you want plot output again. To do
this, you should call the TERMON subroutine to prepare the terminal
to accept normal characters and then to call TERMOF to get back
into the "graphics mode", where output will be sent to the plotter.

2.4.4   TPRINT command on UQVM

The TPRINT command provides support for output to devices connected
to an ASCII terminal's auxiliary port. In this way, plots
generated to a disk file may be plotted on a terminal attached
plotter.

The syntax is:
    TPRINT filename <filetype|LISTING> <filemode|*> <(options...<)>>

The TPRINT command will output a CMS file to a printer or plotter
attached to an ASCII full-screen terminal. (Note that TPRINT
supports Yale connected terminals only.)

Options:
    VI200   Output is being directed to a device connected to a
             Visual 200 terminal. The proper commands to enable and
             then disable the device will be sent by TPRINT.

    VI500   Output is being directed to a device connected to a
             Visual 500 terminal. The proper commands to enable and
             then disable the device will be sent by TPRINT.

    GIGI   Output is being directed to a device connected to a DEC
             GIGI terminal. The proper commands to enable and then
             disable the device will be sent by TPRINT.

    VT100   Output is being directed to a device connected to a DEC
             VT-100 terminal. The proper commands to enable and
             then disable the device will be sent by TPRINT.

    FR100   Output is being directed to a device connected to a
             FREEDOM 100 terminal. The proper commands to enable
             and then disable the device will be sent by TPRINT.

    VT52   Output is being directed to a device connected to a DEC
             VT-52 terminal. The proper commands to enable and then
             disable the device will be sent by TPRINT.

    IBMPC   Output is being directed to a device connected to an
             IBM Personal Computer using YTERM 1.0 or a compatible
             package. Commands to enable and then disable the
             device will be sent by TPRINT.

                              NOTE

             If none of the options IBMPC, VT100,
             VT52, VI200, VI500, GIGI or FR100 is
             specified, then the device attached
             to the terminal must be enabled by

the user prior to invoking TPRINT, if
output to the device is desired.

The TPRINT command supports other options which effect and control
output. For a complete list of options and usage notes, type

INFO TPRINT

when logged onto UQVM.

2.4.5 Tektronix terminals

Before running a program which uses a Tektronix graphics terminal
(such as a 4014) you must remember to correctly set up the terminal
line characteristics. If this is not done, the screen will not
clear properly on initialization and your plot will be spoilt by
seemingly random characters overwriting it.

On UQKL10, type the command:

.SET TTY FORM NO CRLF

On UQVAX, type the command:

$ SET TERMINAL/FORM/NOWRAP

On the IBM system, it is not possible to use Tektronix terminals,
as the output processor cannot send the command codes which they
require.

2.4.6 PDP-11 systems

Although the CALF77 library is available on PDP-11 systems (running
under the RSX-11 operating system), it has undergone modifications
to get there. Several subroutines (most notably, PLOTIN) have been
changed and must be used differently under RSX.

For further information about CALF77 on your RSX system, you should
contact your system manager or the Prentice Computer Centre.

CHAPTER 3

BASIC SOFTWARE

3.1 INTRODUCTION

This chapter describes in detail the basic CalComp subroutines,
some of which have been customized at the University of Queensland.
The graphics library comprises fifteen subroutines which can be
called from FORTRAN (or, in some cases, from PASCAL) programs.
They are PLOTIN, PLOT, FACTOR, WHERE, SYMBOL, CENTRE, NUMBER,
SCALE, AXIS, LINE, NEWPEN, SETPEN, PLTOPR, WINDOW, TERMON and
TERMOF. The software will drive any of the plot devices which are
available to users at the Prentice Computer Centre, provided that
the required device is initially specified.

3.1.1 SUBROUTINE OVERVIEW

PLOTIN   is used to initialize the graphics system for output to a
          particular device. A call to PLOTIN must occur once for
          each separate plot file and must occur before any other
          graphics subroutine calls.

PLOT   converts a pen movement specification into   plotter
          commands for the device in use (that specified in a call
          to PLOTIN) and outputs these to the current plotter file.

FACTOR   enables the user to increase or reduce the scale of
          subsequent plot commands.

WHERE   is used to determine the present position of the pen, the

3-1

current scaling factor and, optionally, the current physical origin coordinates.

SYMBOL   plots a sequence of alphanumeric characters, at a specified position and angle.

CENTRE   plots one symbol - either a special symbol or an alphanumeric character - centred over a specified point.

NUMBER   plots the decimal representation of a floating-point number.

SCALE   will examine a data array to determine the best starting value and scaling factor to use in converting that data into physical plotter page coordinates.

AXIS   draws an axis line with the appropriate title and scale labeling. There are three other routines, called AXISI, VAXIS and VAXISI, which perform the same function, but with differences in scaling and labeling styles.

LINE   plots a series of scaled data points defined by two arrays, optionally plotting specified symbols at each point and connecting the points with straight lines.

NEWPEN   is used to select a new pen from the pen set on the plotter which is currently in use.

SETPEN   generates a message for the operator specifying which pen set will be required for the following plot.

PLTOPR   enables short messages to be sent to the operator during the plot.

WINDOW   is used to define windowing regions during the course of a plot.

TERMON   is used when a plotter is connected in series with your terminal, or when a graphics terminal is in use. It forces suspension of plotter operation and allows normal communication with the terminal.

TERMOF   reverses the effect of TERMON. That is, normal terminal I/O is stopped and communication then proceeds with the plotting device.

3.2 SUBROUTINE DETAILS

The argument names used in the following subroutine descriptions
conform to standard FORTRAN conventions. That is, if the initial
letter is I, J, K, L, M, or N, then the argument type is integer,
but if the initial is any other letter, the argument type is real
(floating-point).

3.2.1 PLOTIN SUBROUTINE

The PLOTIN subroutine initialises a plotter file with a specified
name and for a specified device and allows various features to be
set up for the subsequent plot. The plotting environment is
restored to the default configuration for the specified device, but
it is possible to specify configuration features, such as paper
size, paper type and pen types. The standard configurations are
shown in Appendix A.

The calling sequence for PLOTIN (in FORTRAN) is:

                    CALL PLOTIN(FNAME,DEVICE)

FNAME   specifies the name to be used in identifying the plot. The
         default plot file extension is ".PLT" on DEC systems. On
         UQVM the output file will be "FILE fname A" though the CMS
         command, FILDEF, may be used to change the default output.

DEVICE   is a CHARACTER variable or literal which specifies which
         one of the available plotting devices is to be used.
         Switches specifying setup and configuration features can be
         appended to the device name and these can significantly
         enhance the power of this routine. The most commonly used
         devices and switches are described below, with full
         descriptions being provided in Appendix A.

Device names:

HP7585	the HP7585, at the Prentice Computer Centre.
PLT3	another name for the HP7585.
HP7475	the HP7475, at the Prentice Computer Centre.
PLT4	another name for the HP7475.
HP	either type of HP plotter.
PLT5	the HP7475, at Griffith University.

TEK4010	Tektronix 4010 graphics terminal (not available on UQVM).
TEK4014	Tektronix 4014 graphics terminal (not available on UQVM).
TEK4015	Tektronix 4015 graphics terminal (not available on UQVM).
TEK4105	Tektronix 4105 graphics terminal (not available on UQVM).
TEK4107	Tektronix 4107 graphics terminal (not available on UQVM).
TEK4662	Tektronix 4662 plotter (not available on UQVM).
GIGI	DEC GIGI graphics terminal.

Switches:

/CAROUSEL:x	specifies a non-default pen set to be used for the plot.
/FORMS:x	specifies paper size and type.
/BOUNDS	specifies that out-of-bounds errors should be fatal.
/XYMIN:n,m	specifies the X and Y minimum values for the plot.
/XYMAX:n,m	specifies the X and Y maximum values for the plot.
/TERMINAL	indicates that the specified plotter is attached to your terminal (not available on UQVM - see comments on TPRINT command).
/BAUD:n	specifies the speed at which your terminal graphics device runs (not available on UQVM).

example:

```
C
C     Initialise a plot file for PLT3: (the HP7585) with
C     fibre pens, and normal, A4-size paper. 'GRAPH' is
C     the name of the plot file which will be created.
C
      CALL PLOTIN('GRAPH','PLT3/CAROUSEL:FIBRE/FORMS:A4NORMAL')
C
```

NOTE: The "PLOTIN" subroutine was developed at the Prentice Computer Centre and replaces the standard CalComp "PLOTS" routine.

3.2.2 <u>PLOT SUBROUTINE</u>

The PLOT subroutine is used primarily to move the plotting head in
a straight line to a new position, with the pen either up or down
during the movement. It is possible to reset the coordinate origin
to be at that new pen position.

Note that the PLOTIN subroutine must be called before any call is
made to this or any other graphics routine.

The calling sequence is:

                CALL PLOT(XPAGE,YPAGE,IPEN)

XPAGE,YPAGE are real values that indicate the horizontal and
     vertical coordinates (relative to the current origin) of the
     position to which the pen is to be moved. A new origin may
     be established at the point (XPAGE,YPAGE), as explained
     below.

IPEN is a signed integer which controls pen UP/DOWN status and
     origin definition. The values of IPEN are explained as
     follows:

        1   - there is no change in the pen's UP/DOWN status.
        2   - the pen is down during movement, drawing a visible
               line.
        3   - the pen is up during movement, drawing no line.
        <0   - At the end of the movement, the origin is relocated
               to the new pen position. The effective IPEN value
               used for the movement is the absolute magnitude of
               the value specified.
        999 - this IPEN value signifies that the current plot file
               is to be closed. Another plot file can then be
               initialized, by calling PLOTIN.

See Plate 3-1 for an illustration of the effects of calling this
routine in various ways.

Because it is possible to move the origin to an arbitrary position
within the current bounds, negative coordinates for a call to PLOT
are quite legitimate. For example, suppose the call is:

        CALL PLOT(100.0,150.0,-2)

This resets the origin at coordinates (100.0,150.0). A subsequent
call of

        CALL PLOT(-90.0,-150.0,IPEN)

would cause no problems, but the call

        CALL PLOT(-120.0,-140.0,IPEN)

would result in clipping of the line as the X coordinate is out of bounds.

<u>WARNING:</u> If a closing call to PLOT (with IPEN = 999) is not given
before the end of your graphics application, the plot file may be
lost. In case of an error (e.g. out-of-bounds), the closing call
will be done implicitly.

3.2.3   <u>FACTOR SUBROUTINE</u>

The FACTOR subroutine enables the user to increase or reduce the
scale of subsequent plot commands by setting the ratio of the
desired plot size to the normal plot size.   For example, if the
scaling factor is 2.0, then all subsequent pen coordinate values
will be doubled. During the debugging of a plotting application
program, processing and plotting time can be saved by reducing the
size of the entire plot output. This is done by calling FACTOR
with a value less than 1.0, after calling PLOTIN. When debugging
is completed, this statement can be removed.

The PLOTIN routine initially sets the scaling factor to 1.0, as if
a FACTOR(1.0) call had been made.

The calling sequence is:

              CALL FACTOR(FACT)

FACT is a REAL value which represents the desired scaling factor,
      as described above.

An example FACTOR subroutine call is included in Plate 3-1 along
with the calls to the PLOT subroutine.

3.2.4 <u>WHERE SUBROUTINE</u>

The WHERE subroutine returns six values: the current pen position
coordinates, the current scale factor, the current unit of length
and the physical coordinates of the current origin.

The calling sequence is:

        CALL WHERE(XPAGE,YPAGE,FACT,MMFLAG,ORGX,ORGY)

XPAGE,YPAGE will contain the current pen coordinates, after the
        call to WHERE.

FACT   will contain the current scale factor.   (That is, the
          value last given in a call to FACTOR, or 1.0, if FACTOR
          has not been called.)

MMFLAG   returns a logical value indicating whether the plotting
          unit is the millimetre or the inch. The value should
          always be .TRUE., since inches can no longer be used.

ORGX,ORGY will return the current physical origin coordinates.

<u>EXAMPLE</u>:

The following example uses the WHERE subroutine to determine the
current pen position and scale factor, which are then used to move
the origin to a point located 50mm to the right and half-way to
the bottom of the plotter page. After the move, the pen position
and factor are restored to their old values.

        CALL WHERE(XPAGE,YPAGE,FACT,MMFLAG,ORGX,ORGY)
        CALL FACTOR(1.0)
        CALL PLOT(XPAGE+50.0,-(YPAGE/2.),-3)
        CALL FACTOR(FACT)
        CALL PLOT(XPAGE,YPAGE,3)

<u>Note:</u> If WHERE is called after an out-of-bounds condition, the
values returned will indicate the pen position that is out of
bounds. Of course, if the /BOUNDS switch isn't given to PLOTIN,
then any out-of-bounds condition will cause a fatal error.

This version of the WHERE routine is an extended version of the
original CalComp routine, which took only three arguments.

3.2.5 SYMBOL SUBROUTINE

The SYMBOL subroutine produces plotted annotation at any angle and
of practically any size. It is generally used to plot text, such
as titles, captions and legends.
The standard characters that are drawn by SYMBOL include
alphanumerics (the letters A-Z and digits 0-9) and a number of
special characters.

The calling sequence is:

        CALL SYMBOL(XPAGE,YPAGE,HEIGHT,TEXT,ANGLE,NCHAR)

XPAGE,YPAGE are the coordinates of the lower, left-hand corner
        (before character rotation) of the first character to be
        plotted. The pen is automatically raised before moving
        to this point.

HEIGHT   is the height, in millimetres, of the characters to be
        plotted. The width of each character, including spacing,
        is the same as its height.

TEXT   is a character variable or literal containing the text to
        be plotted.

ANGLE   is the angle (in degrees, anticlockwise, from the X-axis)
        at which the text is to be plotted.

NCHAR   is the number of characters of TEXT to be plotted.

It is possible to continue plotting text from the position at which
the last text ended, by passing the value 4HCONT in XPAGE and/or
YPAGE.. Passing this value in only one parameter will cause that
coordinate to be retained, while the other will have to be
specified.

Note: This SYMBOL routine differs from the standard CalComp SYMBOL
routine, which is used to plot special symbols as well as text.
You must now use the CENTRE routine (section 3.2.6) to plot special
symbols.

See Plate 3-2 and Plate 3-4 for some example calls to SYMBOL.

3.2.6 CENTRE SUBROUTINE

The CENTRE subroutine produces a single symbol, which is specified
by an integer value. Values from 0 to 15 produce centred symbols,
while others produce ASCII characters with the specified code. The
available symbols are shown in Plate 3-3.

The calling sequence is:

        CALL CENTRE(XPAGE,YPAGE,HEIGHT,INTEQ,ANGLE,ICODE)

XPAGE,YPAGE,ANGLE have the same purpose as for a SYMBOL call,
        except that XPAGE and YPAGE represent the coordinates to
        be used for the geometric centre of the symbol.

HEIGHT   is the height (and width), in millimetres, of the symbol
          to be drawn.

INTEQ   controls what is printed. A value in the range 0 to 15
          causes a centred symbol to be displayed. Any printable
          character can also be displayed by giving the
          corresponding decimal ASCII code for INTEQ.

ICODE   determines the pen status during the move to
          (XPAGE,YPAGE).

          When ICODE is:

          -1, the pen is up during the move, after which a single
          symbol is produced.

          <-1, the pen is down during the move, after which a
          single symbol is produced.

          ICODE has no effect when ASCII characters are displayed,
          the pen always being up during the move.

          Note: CENTRE is a subroutine developed at the Prentice
          Computer Centre, and is not included in any standard
          CalComp library.

Some sample calls to the CENTRE subroutine are shown in Plate 3-4.

3.2.7 NUMBER SUBROUTINE

NUMBER converts a floating-point number to the appropriate decimal equivalent and plots it as a string of characters in the FORTRAN F-type format.

The calling sequence is:

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

XPAGE,YPAGE,HEIGHT,ANGLE are the same as the arguments described for the CENTRE subroutine. The continuation feature, where XPAGE and/or YPAGE has the value 4HCONT, may be used, as for the SYMBOL routine.

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

NDEC   controls the precision of the conversion of the number FPN. If the value of NDEC is greater than 0, it specifies the number of digits to be plotted to the right of the decimal point, after proper rounding.

            If NDEC = -1, only the number's integer portion (without a decimal point) is plotted, after rounding.

            If NDEC < -1, then the last (|NDEC|-1) digits of the integer portion are zeroed, after rounding.

            NOTE: The magnitude of NDEC should not exceed 9.

See Plates 3-2, 3-3 and 3-4 for illustrations of various uses of SYMBOL, CENTRE and NUMBER.

3.2.8 SCALE SUBROUTINE

Typically, the user's program will accumulate plotting data in two arrays:

  (i)   an array of independent variables, Xi; and
  (ii) an array of dependant variables, Yi = f(Xi).

It would be unusual, however, if the range of values in each array corresponded exactly with the number of millimetres available in

the actual plotting area. In general, the range of data is
unpredictable and scaling and translation factors must be used to
map data points onto the plotter page.

The SCALE subroutine is used to examine the data values in an array
and to determine a starting value, either minimum or maximum, and a
scaling factor, positive or negative, such that:

(i)   The scale annotation drawn by the axis subroutine at each
       division will properly represent the range of real data
       values in the array.

(ii)   The data points, when plotted by the LINE subroutine, will
       fit in a given plotting area.

These two values are computed and stored by SCALE in the last two
elements of the array. The array must be dimensioned to
accommodate these two values.

The scaling factor (DELTAV) that is computed, represents the number
of data units per millimetre of axis, but is adjusted so that it is
always an interval of 1, 2, 4, 5, or 8 times 10^n (where n is an
exponent consistent with the original unadjusted scaling factor).
As an example, an array may have a range of values from 301 to 912,
to be plotted over an axis of 200 mm. The unadjusted scaling
factor is (912-301)/200 = 3.055 units/mm. The adjusted DELTAV
would be 4×10^0 = 4.0 units/mm .

The starting value (FIRSTV), which will appear as the first
annotation on the axis, is computed as a multiple of DELTAV which
is equal to or outside the limits of the data in the array. In the
example given above, if a minimum is wanted for FIRSTV, 300 would
be chosen as the best value, while if a maximum is desired instead,
912 would be selected.

The calling sequence for SCALE is:

                CALL SCALE(ARRAY,AXLEN,NPTS,INC)

ARRAY is the name of the array of data points to be scaled.

AXLEN is the length of the axis to which the data points are to be
       scaled. Its value must be greater than 25.0 mm.

NPTS   is the number of data values in the array. The array should
       be dimensioned to contain at least two extra elements, to
       allow room for SCALE to store the computed starting value
       and scaling factor at the end of the array.

INC   is an integer whose magnitude is used by SCALE as the

interval at which to select data values from the array.

If INC>=0, the selected starting value (FIRSTV) will approximate a minimum, and DELTAV is positive.

If INC<=0, FIRSTV will approximate a maximum and the scaling factor (DELTAV) is negative.

For any value of INC, the array must be dimensioned at least $(2*|INC|)$ elements larger than the value of NPTS. The computed starting point value is stored at $|INC|$ elements past the end of the data, or at element ARRAY(NPTS*INC+1), and DELTAV is stored at $(2*|INC|)$ past the end, or at element ARRAY(NPTS*INC+INC+1).

NOTE: If you know the range of your data values you do not have to call SCALE for that array so long as you supply an appropriate FIRSTV and DELTAV when AXIS and LINE are called.

EXAMPLES:

(i)   Given an array of 24 data values to be plotted over a 125 mm axis, assume the minimum value in the array is 1.0 and the maximum is 42.0. The statement

            CALL SCALE(ARRAY,125.0,24,1)

would give the following results.

        Units/25 mm = (42.0-1.0)*125.0/25.0 = 8.2
           (rounded to next interval) -> 10.0
        DELTAV (in units/mm) = 10.0/25 = 0.4
        FIRSTV (interval below minimum) = 0.0
        FIRSTV value is stored in ARRAY (25) = 0.0
        DELTAV value is stored in ARRAY (26) = 0.4

Using these values, the first axis in Plate 3-6 shows the data range.

(ii)   Assume that the array of example (i) is to be plotted on a 100 mm axis, from maximum to minimum.

            CALL SCALE(ARRAY,100.0,24,-1)

would give these results:
        Units/25mm = (1.0-42.0)*100.0/25.0 = -10.25
        DELTAV (adjusted) = -20.0/25 = 0.8 units/mm
        Multiple below minimum = 0.0
        FIRSTV = Min. multiple + (AXLEN * |DELTAV|) = 80.0

In this case, the second axis in Plate 3-6 would be suitable.

(iii) Assume 100 points in the range 22.0 to -9.0 are to be plotted
      an a 100 mm axis. We wish to use every second element in the
      array, so a DIMENSION statement should specify ARRAY(204) or
      ARRAY(2,102). The call required is:

              CALL SCALE(ARRAY,100.0,100,-2)

          Initial units/25mm = (-9.0-22.0)*100.0/25.0 = -7.75
          Adjusted DELTAV = -8.0/25 = 0.032 units/mm
          Initial FIRSTV = 16.0

          Last value on axis = -16.0

      The axis range is inadequate for the data range, so DELTAV is
      automatically revised to the next higher interval.

          Revised DELTAV = -10.0
          Revised FIRSTV = 30.0
          FIRSTV is stored in ARRAY(201) or ARRAY(1,101).
          DELTAV is stored in ARRAY(203) or ARRAY(1,102).

      The third axis in Plate 3-6 shows the data range.

3.2.9   THE AXIS SUBROUTINE

Most graphs require axis lines and scales to indicate   the
orientation and values of the plotted data points. The most common
type of scaled axis is easily produced by the AXIS subroutine,
which draws any length line at any angle, divides it into 25 mm
segments, annotates the divisions with appropriate scale values,
and labels the axis with a centred title. When both the X and the
Y axes are needed, AXIS must be called separately for each one.

The calling sequence is:

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

XPAGE,YPAGE are the coordinates, in mm, of the axis starting point.
          Usually, both the X and Y axes are joined at the origin
          of the graph, (where XPAGE and YPAGE both equal zero) but
          other starting points can be used if desired. When using

the LINE subroutine to plot data on an axis, at least one
of the coordinates must be 0, so that for an X-axis,
YPAGE = 0, and for a Y-axis, XPAGE = 0.

Note that the entire axis line and its ends should be at
least 13mm from the page boundary so as to allow space
for the scale annotation and axis title.

TEXT   is the title, which is centred and placed parallel to the
axis line. This parameter may be a character variable or
a literal. The title characters have a fixed height of
3.5mm.

NCHAR   specifies by its magnitude, the number of characters in
the axis title, and by its sign, the side of the line on
which the scale marks and labelling information shall be
placed.

If NCHAR is positive, all annotation appears on the
"positive" side of the line (that is, on the side of
increasing angle), which is the condition normally
required for the Y-axis.

If NCHAR is negative, all annotation appears on the
"negative" side of the axis, which is the condition
normally required for the X-axis.

AXLEN   is the length of the axis line, in millimetres. AXLEN is
adjusted up to the nearest integer multiple of 25mm,
before the axis is plotted.

ANGLE   is the angle in degrees at which the axis is to be drawn.
Normally, this value is 0.0 for the X-axis, since that
axis is often drawn as a horizontal line, and 90.0 for
the Y-axis.

FIRSTV is the value which will appear at the first scale mark on
the axis. This value may be computed by the SCALE
subroutine, which will store it in element NPTS*INC+1 of
ARRAY, but it must still be explicitly passed as an
argument to AXIS.

Scale values along the axis are always represented with
two decimal places. Since the digit size is 2.5mm and
since a scale value appears every 25 mm, no more than six
digits and a sign should appear to the left of the
decimal point.

DELTAV represents the size of data unit increment (positive or
negative) per millimetre on the axis. Like FIRSTV, it
may be computed by SCALE, which will store it beyond

FIRSTV at ARRAY(NPTS*INC+INC+1), or it may simply be determined by the user.

In order to use a standard format of two decimal places, the size of DELTAV is adjusted so that the interval between tick marks is less than 100 but not less than 0.01. As a result, the decimal point may be shifted left or right in the scale values as drawn and the axis title is followed by an indication of the adjustment factor. (See X-axis example in Plate 3-5).

Plate 3-6 illustrates typical X and Y axis elements controlled by the arguments of AXIS.

3.2.10 AXISI SUBROUTINE

The AXISI subroutine is similar in most respects to the AXIS subroutine, except in the following:

(i)   integer notation is used for values at scale marks where it is appropriate. That is, if all the scale values would be 'xx.00', then the '.00' is not plotted for any of them.

(ii) when AXISI adjusts the scale increment by a factor of ten (and follows the axis title by that factor), the exponent of the factor is always a multiple of three.

The calling sequence for AXISI is identical to that for AXIS, described in section 3.2.9.

3.2.11 VAXIS AND VAXISI SUBROUTINES

These subroutines function in just the same way as the AXIS and AXISI subroutines, except that they are enhanced to provide variable spacing of the axis scale marks and annotations.

For VAXIS, the calling sequence is:

```
CALL VAXIS(XPAGE,YPAGE,TEXT,NCHAR,AXLEN,
1          ANGLE,FIRSTV,DELTAV,TDIST)
```

TDIST   is the distance in millimetres required between
           successive scale marks. The axis length (AXLEN)
           specified will be adjusted upwards to the nearest integer
           multiple of TDIST, before the axis is drawn.

The calling sequence for VAXISI is identical to that for VAXIS.

NOTE: The AXISI, VAXIS and VAXISI subroutines were developed at
the Prentice Computer Centre, University of Queensland, and are not
part of any standard CalComp package.

3.2.12 LINE SUBROUTINE

The LINE subroutine produces a plot of the pairs of data values
contained in two arrays. It computes the page coordinates of each
plotted point, according to the data values and scaling parameters
contained in each array. The data points are represented by
centred symbols and connecting lines can optionally be drawn
between the points.

The scaling parameters corresponding to FIRSTV and DELTAV must
immediately follow the data in each array (as described in section
3.2.8). If these parameters have not been computed by the SCALE
subroutine, then they must be supplied by the user (see example (i)
in section 3.2.8).

The calling sequence is:

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

XARRAY   is the array containing the abscissa values and scaling
           factors for the X-array.

YARRAY   is the array containing the ordinate values and scaling
           factors for the Y-array. The number of values in this
           array should be the same as that in XARRAY.

NPTS   is the number of data points in each array. The number
           does not include the extra two array elements containing
           the scaling parameters.

INC   is the index increment to be used between points taken
           from each array, as described previously for the SCALE
           subroutine.

LINTYP   describes the type of line to be drawn through the data

3-16

points, with the magnitude determining how frequently
symbols are plotted, and the sign determining whether or
not connection lines are to be drawn. For example, if
LINTYP = 4, a symbol is plotted at every fourth data
point and straight lines connect successive data points.
If LINTYP=-2, every second data point is represented by a
symbol and no connecting lines are drawn.

If LINTYP is zero, the points are connected by straight
lines and no symbols are plotted.

INTEQ   is an integer representing the special plotting symbol
            centred at each data point. It has meaning only when
            LINTYP is not zero. Plate 3-3 lists the symbols that are
            available.

Plate 3-7 illustrates the types of lines drawn by various
combinations of LINTYP and INTEQ.

3.2.13   PLTOPR SUBROUTINE

The PLTOPR subroutine causes the spooling program to halt the
current plot, type the specified message on the operator's terminal
and wait for the operator's response. Note that PLTOPR is not part
of the standard CalComp library.

        CALL PLTOPR(MESSAGE,LENGTH)

MESSAGE   is a character variable or literal containing the ASCII
            message.

LENGTH   is the length of the message in characters.

3.2.14   NEWPEN SUBROUTINE

The NEWPEN subroutine is used to tell the plotter to load a new
pen, which will be used "until further notice". The pens which can
be selected depend upon which plotting device is being used, and
for some plotters, upon which carousel is currently loaded.
Available pens and carousels are listed in Appendix A.

The calling sequence is:

        CALL NEWPEN(NPEN)

NPEN   is an integer argument which specifies the new pen to be
             used.   The pen position is unchanged after the plot, but
             the pen is left in the UP position. Values which do not
             correspond to a valid pen for the current plotter result
             in the subroutine call being ignored.

3.2.15   SETPEN SUBROUTINE

The SETPEN subroutine allows the user to specify a pen to be
included in the carousel for the duration of the current plot.   An
entire carousel setup may be specified by multiple calls to SETPEN,
but for this purpose, it is preferable to use the /CAROUSEL (see
Appendix A) switch to PLOTIN, if possible.

SETPEN, like the PLTOPR subroutine forces plotting to halt until
the operator responds.

The calling sequence is:

        CALL SETPEN(NPEN,TYPE,SIZE,INK)

NPEN   is the pen number, as described   for   the   NEWPEN
             subroutine, above.   A value of zero is allowed if the
             other arguments are blank and this causes the pen setting
             to be reset to its initial state.

TYPE   is a character variable, specifying the pen type.

SIZE   is a character variable; specifying the pen size.

INK   is a character variable, specifying the ink colour.

             If blank values are given for TYPE, SIZE or INK, then the
             'standard' values are selected.   These standard values
             for plotter stationery depend upon the device in use and
             are described fully in Appendix A. Users will of course
             be notified through the Computer Centre Newsletter of any
             changes or extra options, if they become available.

3.2.16 WINDOW SUBROUTINE

The WINDOW subroutine allows you to define the windowing region
which will be in effect for the subsequent plot. CALF77 will
automatically clip all pen movements at the boundary of this
region, so that no part of the plot which appears outside the
window will actually be displayed.

                 The calling sequence for WINDOW is:

                 CALL WINDOW(XMIN,YMIN,XMAX,YMAX)

   XMIN,YMIN   specify the coordinates of the lower, left-hand
               corner of the windowing region.

   XMAX,YMAX   give the coordinates of the upper, right-hand corner
               of the windowing region.

NOTE: The WINDOW subroutine was written at the University of
Queensland, and is not part of the standard CalComp package.

3.2.17 TERMON AND TERMOF SUBROUTINES

The TERMON and TERMOF subroutines are used in cases where a user
has a plotting device connected in series with a terminal (and so
has given the /TERMINAL switch to PLOTIN) and wishes to use both
devices during the execution of a plotting program (not applicable
on the IBM host UQVM). Calling TERMON has the effect of forcing
program I/O to go to the connected terminal, rather than to the
plotter, while calling TERMOF has the opposite effect, disabling
the terminal and forcing communication with the plotter.

The calling sequence for either routine is simple:

             CALL TERMON

                 .
                 .
             CALL TERMOF

NOTE: The TERMON and TERMOF subroutines were developed at the
Prentice Computer Centre, University of Queensland, and are not
part of any standard CalComp package.

3.3 SAMPLE PLOTTING PROGRAMS

To illustrate the use of the CALF77 routines within FORTRAN programs, several programs, each written for the FORTRAN-77 compiler, are shown below, along with their output.

NOTE: Non-ANSI standard Features of FORTRAN-77 used in the examples are:

(i) Text following an exclamation mark (!) on a line is regarded by the compiler to be a comment.

(ii) Compile-time constants may be named in a 'PARAMETER' statement.

3.3.1 EXAMPLE 1

The program that produced the graph in Plate 3-8 is shown below. The only assumption made is that the 24 pairs of TIME and VOLTAGE data values are contained in a file of 24 records. Notice that only 11 executable statements are required to produce the graph.

```
        PARAMETER (NUMPTS=24)            !Array size
        DIMENSION XARRAY(NUMPTS+2),YARRAY(NUMPTS+2)
C
C  Initialise plotter file
C
        CALL PLOTIN
     1      ('PERFM','PLT3/FORMS:A4NORMAL/ROTATE/CAROUSEL:FIBRE')
C
C  Read data into arrays
C
        READ(5,25)(XARRAY(I),YARRAY(I),I=1,NUMPTS)
 25     FORMAT(2F)
C
C  Establish new origin at 20.,20.
C
 30     CALL PLOT(20.,20.,-3)
C
C  Compute scale factors
C
        CALL SCALE(XARRAY,125.,NUMPTS,1)
        CALL SCALE(YARRAY,150.,NUMPTS,1)
C
```

```
C  Draw time and voltage axes
C
       CALL AXIS(0.,0.,'TIME IN MILLISECONDS',-20,125.0,0.,
      1          XARRAY(NUMPTS+1),XARRAY(NUMPTS+2))
       CALL AXIS(0.,0.,'VOLTAGE',7,150.,90.,
      1          YARRAY(NUMPTS+1),YARRAY(NUMPTS+2))
C
C  Plot the curve
C
       CALL LINE(XARRAY,YARRAY,NUMPTS,1,2,1)
C
C  Plot the graph title
C
       CALL SYMBOL(20.,140.,5.25,'PERFORMANCE TEST',0.,16)
       CALL SYMBOL(20.,130.,3.50,'REF. NO. 1623-46',0.,16)
C
C  Close the plot output file
C
       CALL PLOT(0.,0.,999)
       END
```

3.3.2 EXAMPLE 2

The following FORTRAN program plots the graph of Y = X**2 (see Plate 3-9).

```
        PARAMETER (INC=1)              !Single dimension array
        PARAMETER (NPTS=30)            !Points on graph
        PARAMETER (HEIGHT=5.25)        !Height of annotation
        PARAMETER (XLEN=250.)          !Length of x-axis
        PARAMETER (YLEN=225.)          !Length of y-axis
        DIMENSION X(NPTS+2),Y(NPTS+2)
        CHARACTER*17 XLABEL,YLABEL
        CHARACTER*17 TITLE
        CHARACTER*5  FILE
        DATA TITLE/' GRAPH OF Y=X**2 '/
        DATA XLABEL/' X AXIS '/
        DATA YLABEL/' Y AXIS '/
        DATA NTITLE,NLABEL/17,8/
        DATA FILE/'GRAPH'/
C
C  Create the X and Y coordinates
C
        DO 10 I=1,NPTS
            FI = FLOAT(I-1)
            X(I) = FI
            Y(I) =  FI*FI
 10     CONTINUE
C
C  Open a plot file and move origin to allow
C  for axis annotation
C
        CALL PLOTIN(FILE,'PLT4/FORMS:A4NORMAL/ROTATE')   !Device is
hp7475
        CALL PLOT(20.,20.,-3)
C
C  Scale the X and Y coordinates
C
        CALL SCALE(X,XLEN,NPTS,INC)
        CALL SCALE(Y,YLEN,NPTS,INC)
C
C  Plot the X and Y axes
C
        CALL AXIS(0.,0.,XLABEL,-NLABEL,XLEN,0.,X(NPTS+1),
 1      X(NPTS+2))
        CALL AXIS(0.,0.,YLABEL, NLABEL,YLEN,90.,Y(NPTS+1),
 1      Y(NPTS+2))
C
C  Plot the line
C
        CALL LINE(X,Y,NPTS,INC,0,0)
```

```
C
C  Plot the title and close the plot file
C
       CALL SYMBOL(25.,YLEN-25.,HEIGHT,TITLE,0.,NTITLE)
       CALL PLOT(0.,0.,999)
       END
```

3.3.3 EXAMPLE 3

This third program produces two graphs, both of the function

$$Y = EXP(-X/10.)*SIN(X)$$

Different sets of axes are used for each graph.

```
       PARAMETER (NPTS=200)               !No of graph points
       PARAMETER (NELTS=NPTS+2)           !Array size
       PARAMETER (MINOFF=NPTS+1)          !Index of min
       PARAMETER (INCOFF=NPTS+2)          !Index of inc
       PARAMETER (XLEN=325.)              !Axis length
       PARAMETER (YLEN=225.)
       PARAMETER (HEIGHT=5.25)            !Character height
       PARAMETER (NTITLE=32)              !Chars in title
       PARAMETER (NLABEL=8)               !Chars in label
       PARAMETER (INC=1)
       DIMENSION X(NELTS),Y(NELTS)
       CHARACTER*8 XLABEL,YLABEL
       CHARACTER*32 TITLE
       EQUIVALENCE (XMIN,X(MINOFF))       !Minimum X
       EQUIVALENCE (DX,X(INCOFF))         !Delta-X
       EQUIVALENCE (YMIN,Y(MINOFF))       !Minimum Y
       EQUIVALENCE (DY,Y(INCOFF))         !Delta-Y
       DATA XLABEL/' X AXIS '/, YLABEL/' Y AXIS '/
       DATA TITLE/' GRAPH OF Y=EXP(-X/10.0)*SIN(X) '/
C
       X0=0.
       Y0=0.
       DO 10 I=1,NPTS                     !Create coordinates
       X(I)=FLOAT(I-1)/5.
       Y(I)=EXP(X(I)/10.)*SIN(X(I))
   10  CONTINUE
C
C  Open plot file and shift origin to allow for axis annotation
C
       CALL PLOTIN('OSCI1','PLT3/SIZE:A4')
```

```
        CALL PLOT(13.,13.,-3)
C
C   Scale the coordinates
C
        CALL SCALE(X,XLEN,NPTS,INC)
        CALL SCALE(Y,YLEN,NPTS,INC)
C
C   Plot the axes
C
        CALL AXIS(X0,Y0,XLABEL,-NLABEL,XLEN,0.,XMIN,DX)
        CALL AXIS(X0,Y0,YLABEL,NLABEL,YLEN,90.,YMIN,DY)
C
C   Plot the line
C
        CALL LINE(X,Y,NPTS,INC,0,0)
C
C   Plot the title and close the file
C
        CALL SYMBOL(25.,YLEN-25.,HEIGHT,TITLE,0.,NTITLE)
        CALL PLOT(0.,0.,999)
C
C   Open a second plot file
C
        CALL PLOTIN('OSCI2','PLT3/SIZE:A4')
        CALL PLOT(13.,13.,-3)
C
C   Determine the origin of the X-axis and plot
C        the axes
C
        CALL AXIS(X0,Y0,YLABEL,NLABEL,YLEN,90.,YMIN,DY)
        IF(YMIN .GE. 0)GO TO 20
        TMP=ABS(YMIN)/DY+0.5
        Y0=FLOAT(IFIX(TMP))
  20    CALL AXIS(X0,Y0,XLABEL,-NLABEL,XLEN,0.,XMIN,DX)
C
C   Plot the line
C
        CALL LINE(X,Y,NPTS,INC,0,0)
C
C   Plot the title and close the file
C
        CALL SYMBOL(25.,YLEN-25.,HEIGHT,TITLE,0.,NTITLE)
        CALL PLOT(0.,0.,999)
        END
```

CALLS   EFFECT

10   CALL PLOT (0.0, 0.0, -3)

20   CALL PLOT (20.0, 30.0, 2)
30   CALL PLOT (30.0, 0.0, 2)

40   CALL PLOT (0.0, 0.0, 3)
C   returns pen to the origin
50   CALL PLOT (20.0, 30.0, -2)
C   defines a new origin
60   CALL PLOT (30.0, 0.0, 2)
C   note that the X, Y values in statements
C   50, 60 are the same as those used in 20, 30

70   CALL PLOT (-20.0, -30.0, -3)
C   positions the pen at the initial origin
C   and defines this as the new origin

80   CALL FACTOR (0.5)
C   reduces subsequent pen movement by half
90   CALL PLOT (100.0, 100.0, 2)
100 CALL PLOT (100.0, 0.0, 3)

LEGEND

——— Carriage movement with pen down

  ⊙   Terminal position of pen

Plate 3-1
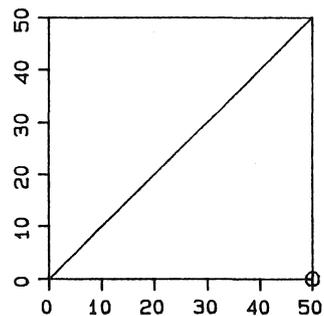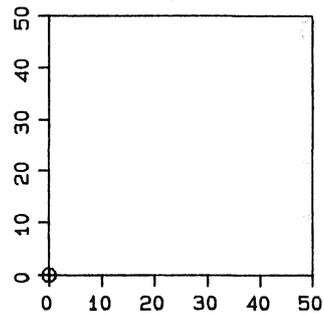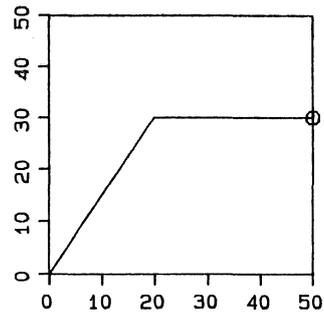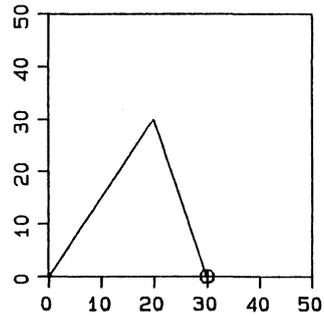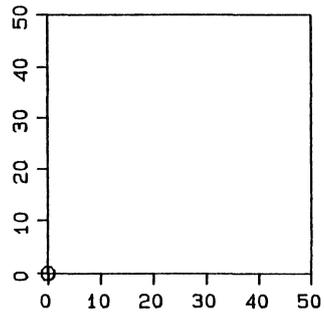
```
CALL SYMBOL (X, Y, 6.0, 'A SAMPLE TITLE', 0.0, 14)
```

A SAMPLE TITLE

```
CHARACTER*16 TEXT
TEXT = 'A SAMPLE TITLE   '
CALL  SYMBOL (X, Y, 4.0, TEXT, 0.0, 16)
CALL  SYMBOL (4HCONT, 4HCONT, 4.0, TEXT, 90.0, 16)
CALL  SYMBOL (4HCONT, 4HCONT, 4.0, TEXT, 180.0, 16)
CALL  SYMBOL (4HCONT, 4HCONT, 4.0, TEXT, 270.0, 16)
```

Plate 3-2

CHARACTERS AVAILABLE FROM THE CENTRE ROUTINE
AT THE PRENTICE COMPUTER CENTRE, U OF Q

Int	Sym	Int	Sym	Int	Sym	Int	Sym	Int	Sym	Int	Sym	Int	Sym	Int	Sym
0	⌶	16	_	32		48	0	64	@	80	P	96		112	p
1	⬡	17	_	33	!	49	1	65	A	81	Q	97	a	113	q
2	△	18	_	34	"	50	2	66	B	82	R	98	b	114	r
3	+	19	_	35	#	51	3	67	C	83	S	99	c	115	s
4	✕	20	_	36	$	52	4	68	D	84	T	100	d	116	t
5	◇	21	_	37	%	53	5	69	E	85	U	101	e	117	u
6	⬘	22	_	38	&	54	6	70	F	86	V	102	f	118	v
7	⊠	23	_	39	'	55	7	71	G	87	W	103	g	119	w
8	⬙	24	_	40	(	56	8	72	H	88	X	104	h	120	x
9	⅄	25	_	41	)	57	9	73	I	89	Y	105	i	121	y
10	⬚	26	_	42	✳	58	:	74	J	90	Z	106	j	122	z
11	✳	27	_	43	+	59	;	75	K	91	[	107	k	123	{
12	⬲	28	_	44	,	60	<	76	L	92	\	108	l	124	|
13	|	29	_	45	—	61	=	77	M	93	]	109	m	125	}
14	✡	30	_	46	.	62	>	78	N	94	^	110	n	126	~
15	—	31	_	47	/	63	?	79	O	95	_	111	o	127	

INTEGERS TO USE IN CALLS TO CENTRE ARE SHOWN TO LEFT OF EACH SYMBOL

Plate 3-3

Plotting Data Points

```
CALL CENTRE (0.0, 0.0, 5.0, 6, 0.0, -1)
CALL CENTRE (40.0, 30.0, 5.0, 10, 0.0, -1)
CALL CENTRE (70.0, 15.0, 5.0, 1, 0.0, -2)
CALL CENTRE (110.0, 15.0, 5.0, 35, 0.0, -1)
```
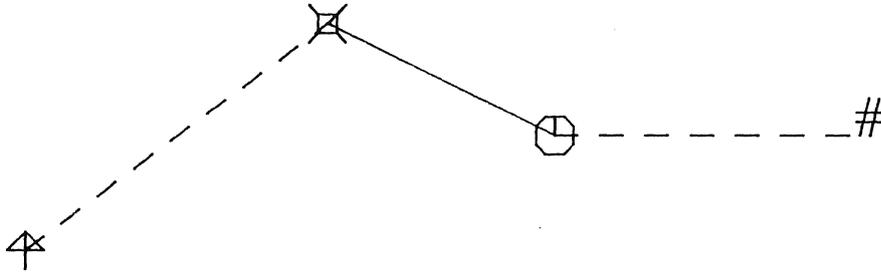
Combining CENTRE and NUMBER and drawing a superscript

```
VALUE = 12.123
CALL SYMBOL (X, Y, 6.0, 'VALUE OF X', 0.0, 10)
CALL SYMBOL (4HCONT, Y+3.0, 3.0, '2', 0.0, 2)
CALL SYMBOL (4HCONT, Y, 6.0, ' = ', 0.0, 2)
CALL NUMBER (4HCONT, 4HCONT, 6.0, VALUE, 0.0, 3)
```

VALUE OF X^2 = 12.123

Drawing text and numbers at various angles

```
      DO 10 I=0, 315, 45
        ANGLE = I
        CALL SYMBOL (X, Y, 4.0, ' ANGLE=', ANGLE, 7)
10    CALL NUMBER (4HCONT, 4HCONT, 4.0, ANGLE, ANGLE, -1)
```

Plate 3-4

Y-AXIS

CALL AXIS (0.0, 0.0, 'X-AXIS', -3.0, 125.0, 0.0, -900.0, 40.0)

CALL AXIS (0.0, 0.0, 'Y-AXIS', 3.0, 75.0, 90.0, 100.0, 0.02)

$X-AXIS*10^2$

Plate 3-5

```
      r---------------- (RANGE OF DATA) -------------¬
      ¦                                              ¦
   0.00      10.00      20.00      30.00      40.00      50.00
   L_____|_____|_____|_____|_____|

   (i)


                              r-(RANGE OF DATA)--¬
                              ¦                  ¦
   80.00      60.00      40.00      20.00       0.00
   L_____|_____|_____|_____|

   (ii)


                        r------ (RANGE OF DATA) -------¬
                        ¦                              ¦
   30.00      20.00      10.00       0.00      -10.00
   L_____|_____|_____|_____|

   (iii)
```

Plate 3-6
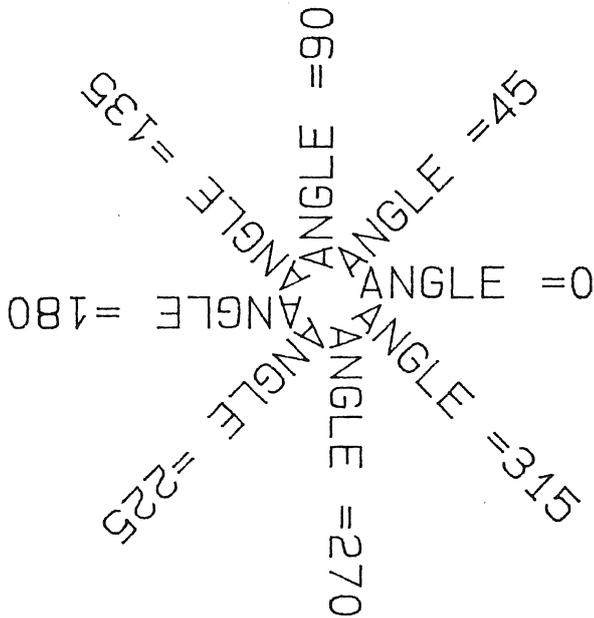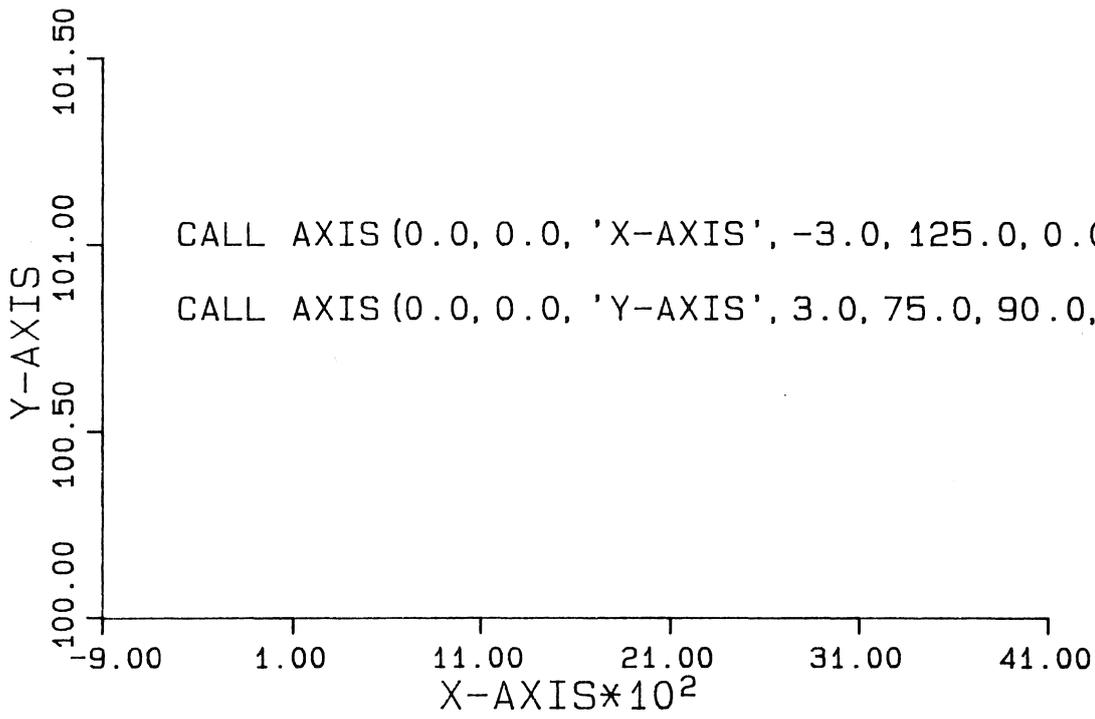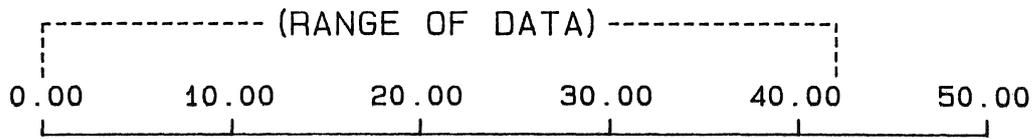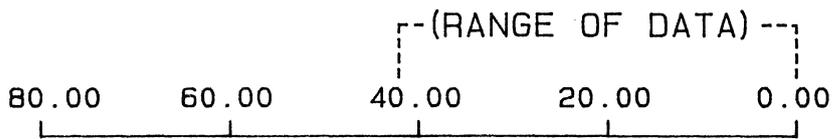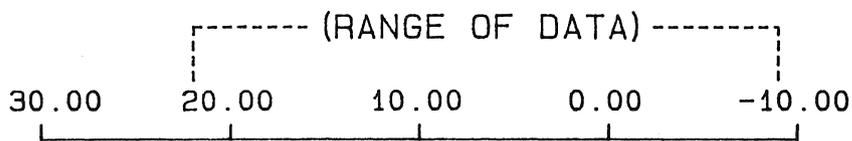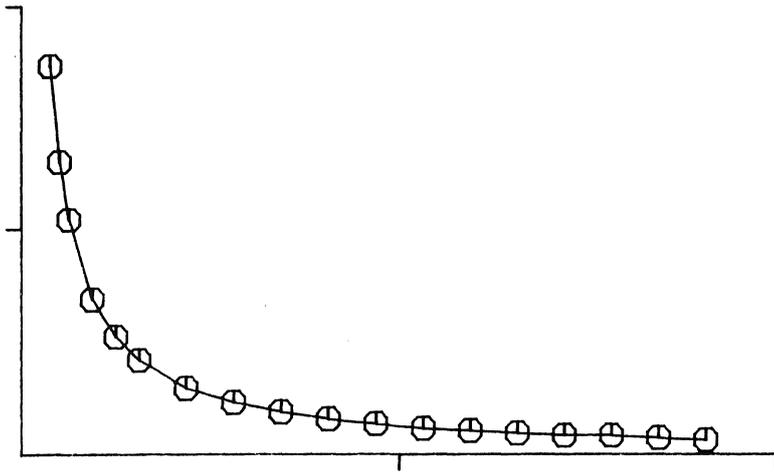
(A) CALL LINE (XARRAY, YARRAY, 18, 1, 1, 1)
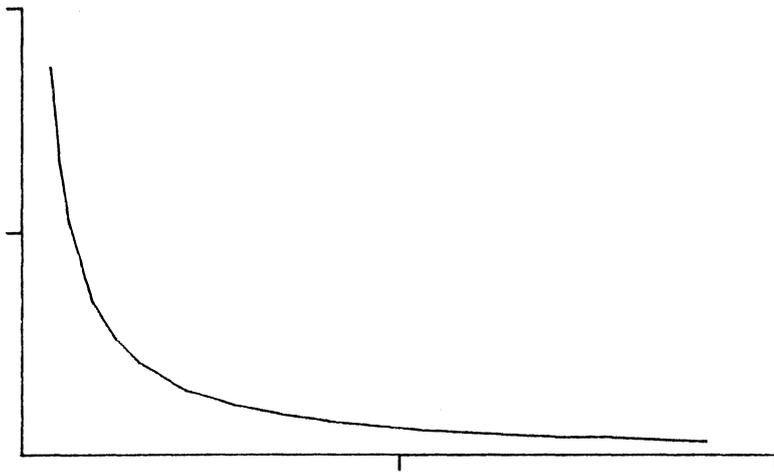
(B) CALL LINE (XARRAY, YARRAY, 18, 1, 0, 0)

(C) CALL LINE (XARRAY, YARRAY, 18, 1, -2, 1

Plate 3-7

PERFORMANCE TEST

REF. NO. 1623-46

VOLTAGE*10³

TIME IN MILLISECONDS

Plate 3-8

GRAPH OF Y=X**2

Y AXIS *10^1

X AXIS

Plate 3-9

GRAPH OF Y=EXP (-X/10.0) *SIN (X)

Y AXIS

X AXIS

Plate 3-10

GRAPH OF Y=EXP(-X/10.0)*SIN(X)

X AXIS

Y AXIS

Plate 3-11
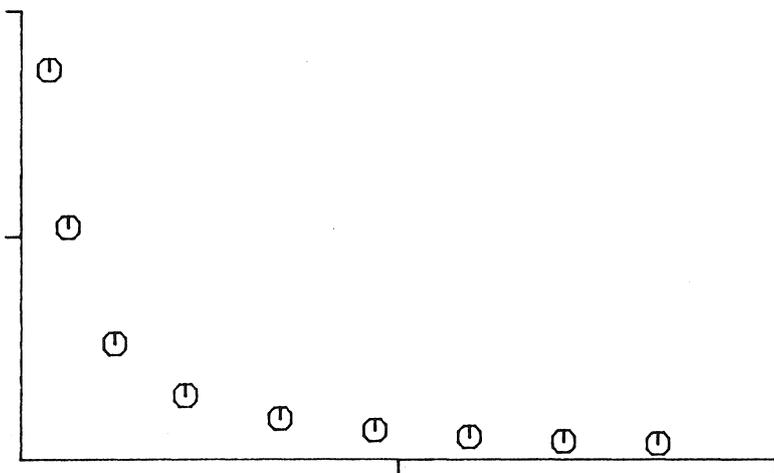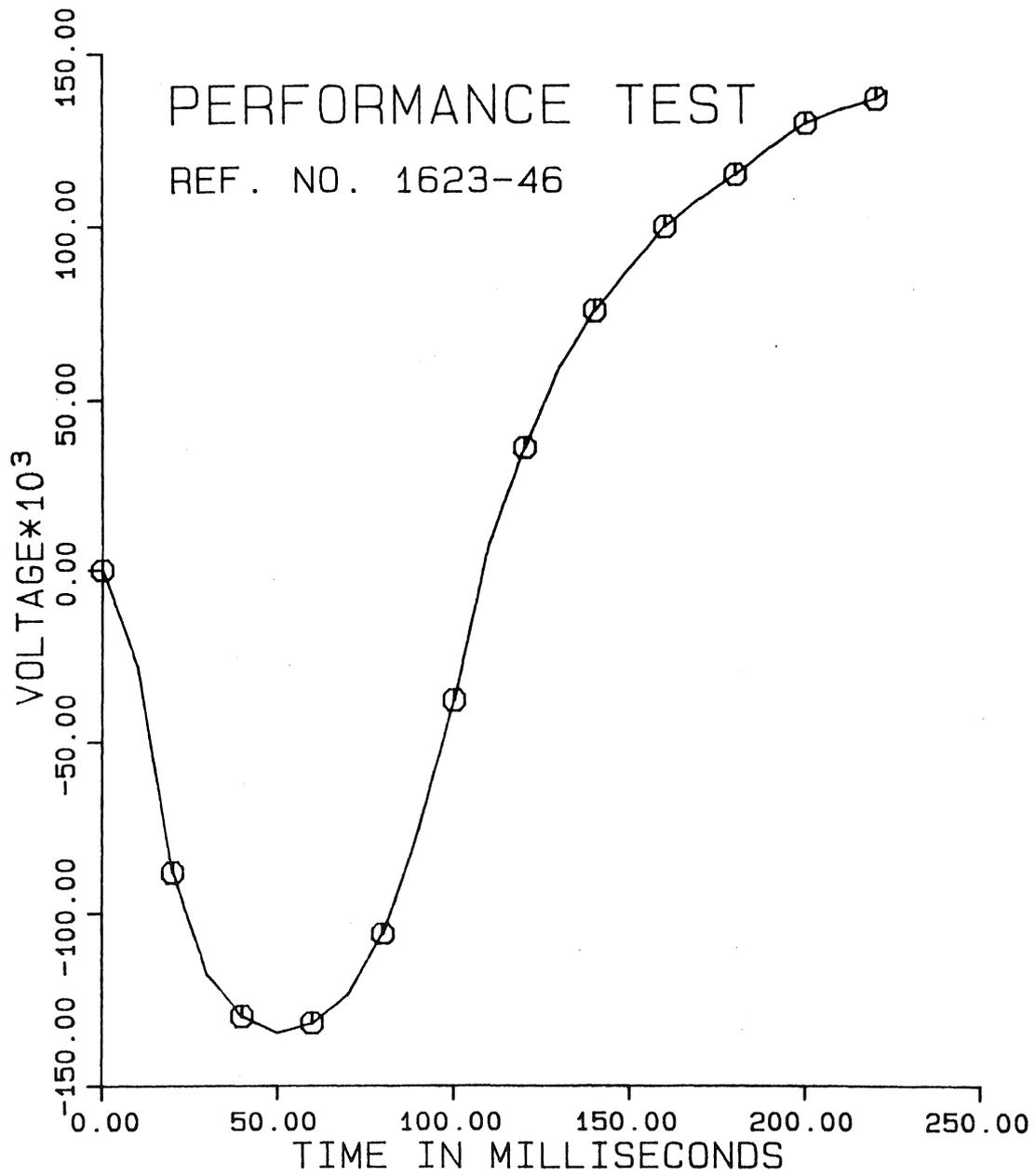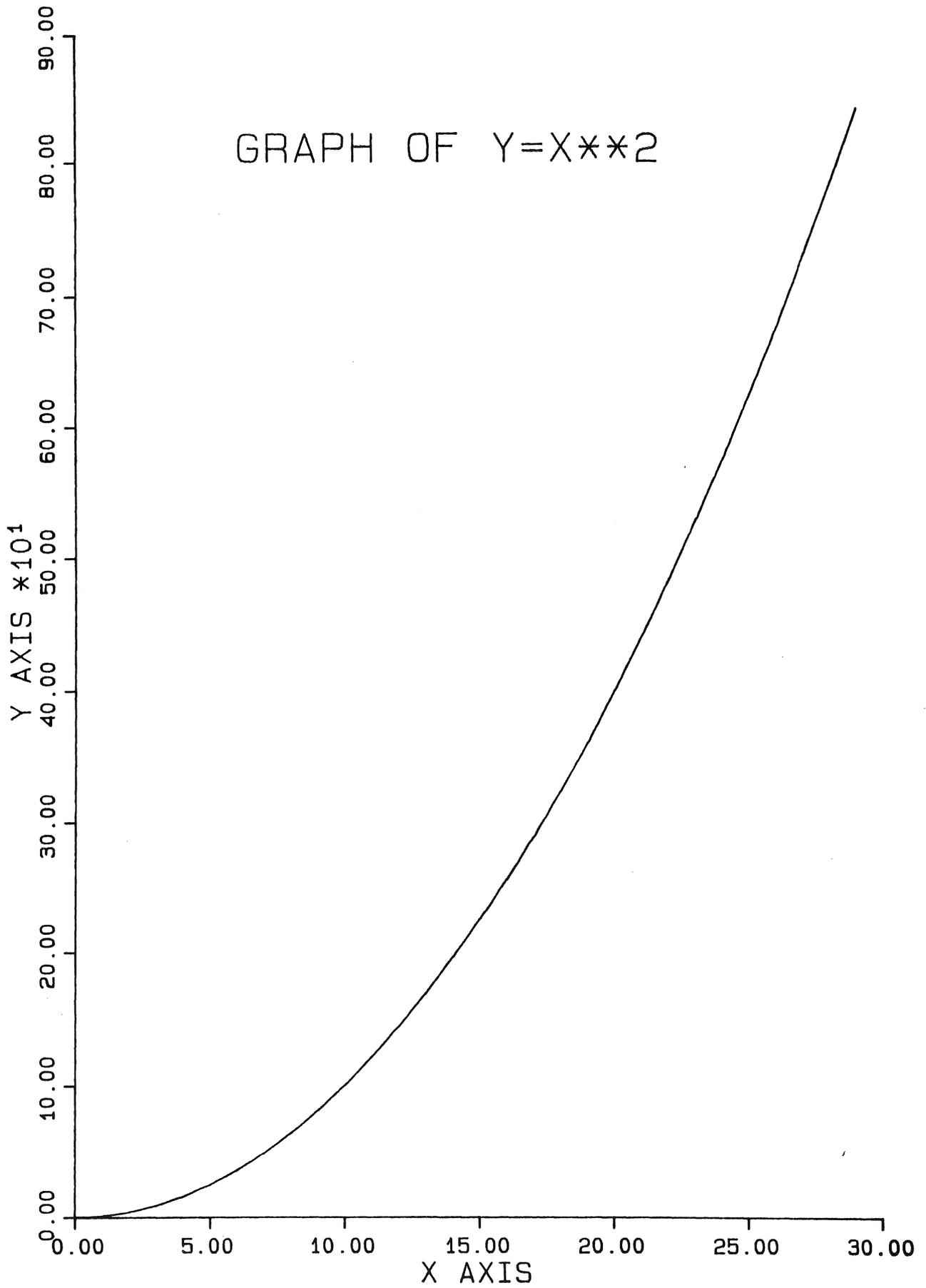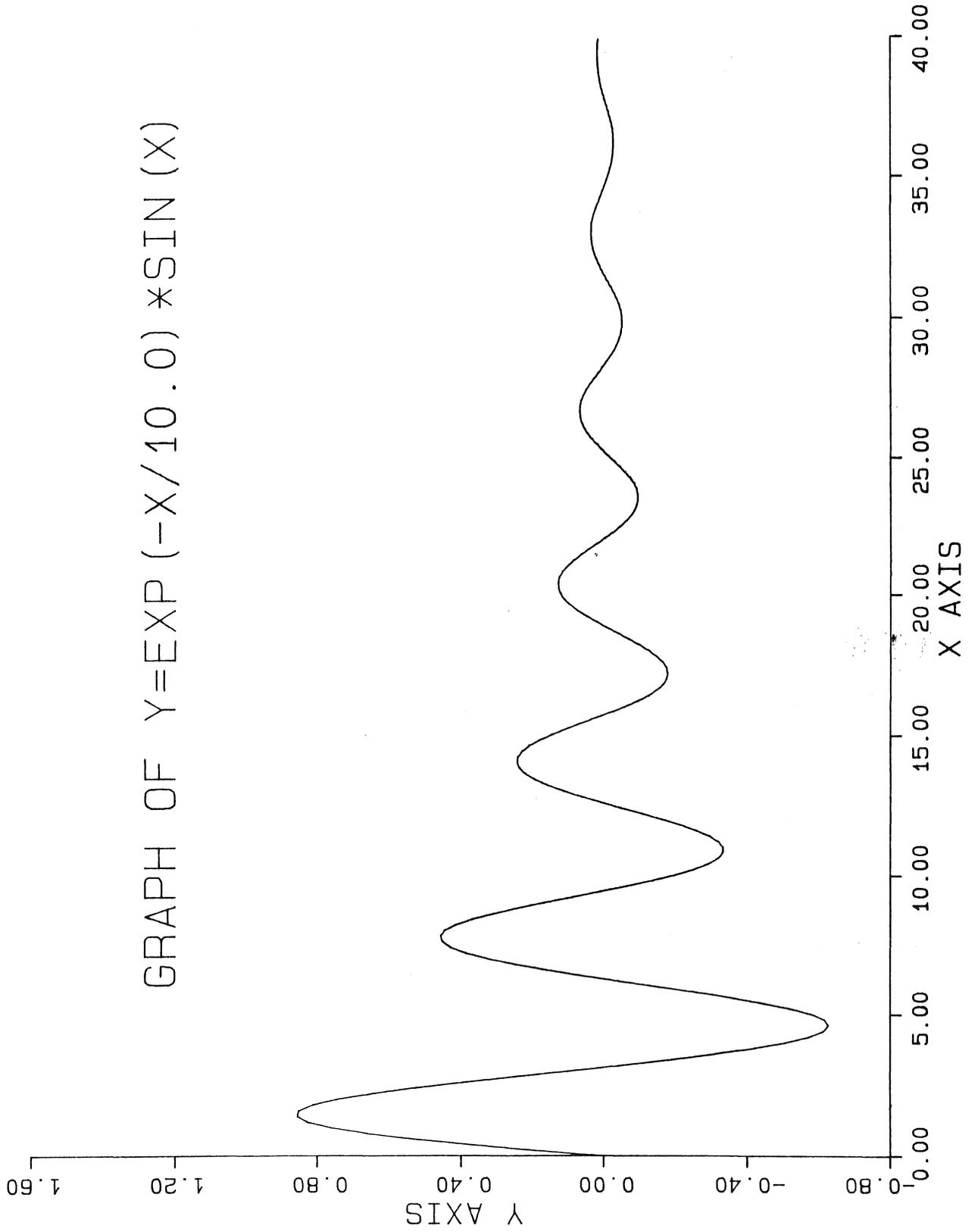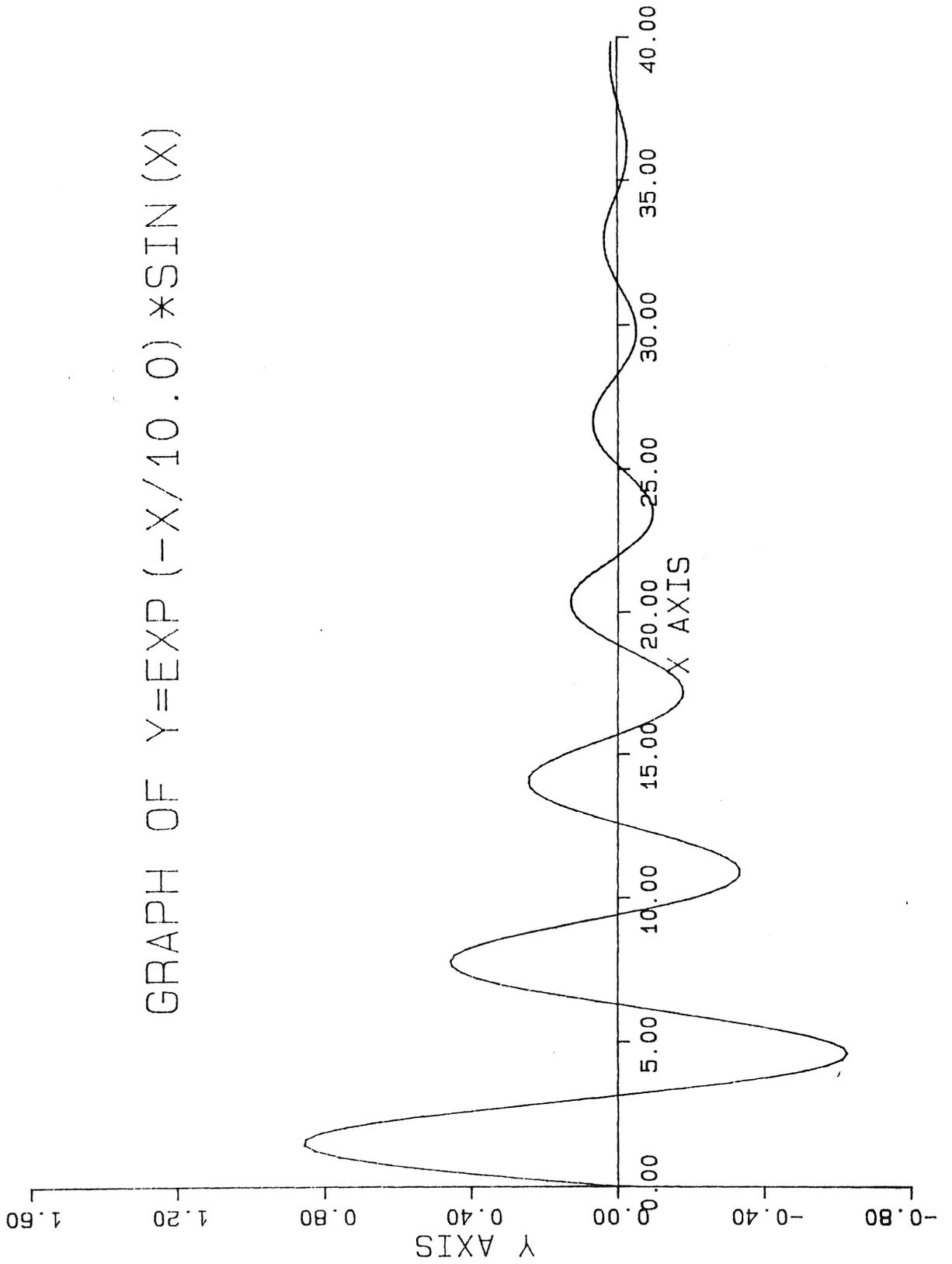
CHAPTER 4

CALCOMP GRAPHICS FUNCTIONAL SOFTWARE LIBRARY

4.1 INTRODUCTION

This chapter describes the calling sequences and arguments for the
subroutines in CalComp's Functional Software library. The library
is comprised of twenty-seven subroutines, written originally in USA
Standard FORTRAN, and since translated into FORTRAN-77.

Plate 4-1 shows a bar graph and a simple line plot drawn using
these routines.

4.1.1 SUBROUTINE OVERVIEW

AROHD	draws arrowheads.
ARROW	draws lines terminated with arrows.
AXISB	draws an axis with business-oriented annotation.
AXISC	draws an axis with calendar-month annotation.
BAR	draws bars for bar-graph plotting.
CIRCL	draws a circle, arc or spiral.
CNTRL	draws centre lines.
CRVPT	fits a polynomial curve to a set of data points.
CURVX	plots a function of X over a given range.
CURVY	plots a function of Y over a given range.
DASHL	draws dashed lines connecting a series of data points.
DASHP	draws a dashed line to a specified point.
DIMEN	draws annotated dimension lines.
ELIPS	draws an ellipse or elliptical arc.

FIT	draws a curve through three points.
FLINE	draws a smooth curve through a set of data points.
GRID	draws a linear grid.
LABEL	draws annotation between specified points.
LBAXS	draws a logarithmic axis with business annotation.
LGAXS	draws a logarithmic axis with annotation.
LGLIN	plots data either in log-log or in semi-log mode.
POLAR	plots data points, using polar coordinates.
POLY	draws an equilateral polygon.
RECT	draws a rectangle.
SCALG	performs scaling for logarithmic plotting.
SHADE	draws shading between designated lines.
SMOOT	draws a spline curve through a set of data points.

4.2 SUBROUTINE DETAILS

With each of the following subroutine descriptions, plotted output
samples are included, which show the results of trials using
various argument values. As in chapter 3, the standard FORTRAN
default variable type conventions are used - if the initial letter
of an undeclared variable's name is I, J, K, L, M or N, then that
variable's type is Integer. Otherwise it is Real.

According to the standards of USA Standard FORTRAN, the subroutine
names have been restricted to a maximum of five letters in length.

4.2.1 AXISB SUBROUTINE

AXISB is a FORTRAN subroutine which draws an annotated axis with
business-oriented labeling. The calling sequence is:

```
    CALL AXISB
  1  (XPAGE,YPAGE,IBCD,NCHAR,AXLEN,ANGLE,FIRSTV,DELTAV)
```

XPAGE,PAGE	are the coordinates, in millimetres, of the axis starting point.
IBCD	is a CHARACTER variable or literal containing the axis title.
NCHAR	is the number of characters in the axis title. If NCHAR is negative, the annotation is placed on the

4-2

clockwise side of the axis, while if it is positive,
annotation is placed on the counter-clockwise side of
the axis.

AXLEN   is the axis length, in millimetres.

ANGLE   is the angle, in degrees, at which the axis is to be
               drawn.

FIRSTV   is the value of annotation at the first tick mark on
               the axis.

DELTAV   is the difference in values from one tick mark to the
               next.

COMMENTS

The value in DELTAV is scaled so that the adjusted delta value lies
between 0.01 and 100. The appropriate scaling factor annotation
appears after the axis title, as shown in the table below.

VALUE OF DELTAV	ADJUSTED DELTAV VALUE	SCALING FACTOR ANNOTATION
DELTAV<0.01	DELTAV*10¬N	*10¬-N
0.01<DELTAV<100	DELTAV	NONE
100<DELTAV<1,000	DELTAV/100	HUNDREDS
1,000<DELTAV<10,000	DELTAV/1,000	THOUSANDS
10,000<DELTAV<100,000	DELTAV/10,000	TEN THOUSANDS
100,000<DELTAV<1,000,000	DELTAV/100,000	HUNDRED THOUSANDS
1,000,000<DELTAV<10,000,000	DELTAV/1,000,000	MILLIONS
10,000,000<DELTAV	DELTAV/10¬N	*10¬N

Scaling factors applied to DELTAV.

Plates 4-2 and 4-3 show the effect of varying the parameters
supplied to AXISB.

4.2.2 AXISC SUBROUTINE

AXISC is a FORTRAN subroutine which draws a labelled axis similar to
that drawn by AXISB (described above) except that it has calendar
annotation.   The calling sequence is:

     CALL AXISC
    1   (XPAGE,YPAGE,IBCD,NCHAR,AXLEN,ANGLE,FIRSTV,DELTAV)

XPAGE,YPAGE   are the coordinates of the axis starting point.

IBCD   is a CHARACTER variable or literal containing the axis title.

NCHAR   is the number of characters in the axis title. If NCHAR<0 then axis annotation is placed in the clockwise side of the axis; otherwise, it is placed on the counter-clockwise side.

AXLEN   is the axis length, in millimetres.

ANGLE   is the angle, in degrees, at which the axis is to be drawn. (The axis is rotated about XPAGE,YPAGE.)

FIRSTV   is a value representing the month of the year, to establish annotation at the first tick mark. The value 1.0 represents January, the value 2.0, February, and so on. Only the integer part of FIRSTV is used.

DELTAV   is the number of months between tick marks. It must be positive and, like FIRSTV, only the integer part is used.

Plate 4-4 shows how changes to these parameters affect the resulting plot.

4.2.3 BAR SUBROUTINE

BAR is a FORTRAN subroutine which draws bars (for use in plotting bar graphs) with optional hatching. The calling sequence is:

    CALL BAR (XPAGE,YPAGE,ANGLE,HEIGHT,WIDTH,SH,IHAT,NPI)

XPAGE,YPAGE   are the coordinates in millimetres of the lower, left-hand corner of the bar, before rotation.

ANGLE   is the angle, in degrees to the horizontal, at which the base of the bar is to be drawn.

HEIGHT   is the height of the bar, in millimetres.

WIDTH   is the width of the bar, in millimetres.

SH   is the height, in millimetres, of the intermediate bar, which is hatched according to the IHAT code.

This value may, of course, be equal to the value of HEIGHT, above.

IHAT   is the hatching code, as follows:
                       IHAT=1   Draw bar only (no hatching).
                       IHAT=2   Hatch with positive gradient.
                       IHAT=3   Hatch with negative gradient.
                       IHAT=4   Hatch both ways.

NPI   is the number of lines of hatching per millimetre, along the base.

The hatching produced for the intermediate bar is parallel to the diagonal of that bar. See Plates 4-5 and 4-6 for examples.

4.2.4 LBAXS SUBROUTINE

LBAXS is a FORTRAN subroutine which draws a logarithmic axis with business annotation of powers of ten. The calling sequence is:

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

XPAGE,YPAGE   are the coordinates of the axis starting point.

IBCD   is a CHARACTER variable containing the axis title.

NCHAR   is the number of characters in the axis title. As for the AXISB and AXISC routines (described previously), the sign of NCHAR determines on which side of the axis the annotation is placed.

AXLEN   is the axis length, in millimetres.

ANGLE   is the angle, in degrees, at which the axis is to be drawn.

FIRSTV   is the value of annotation at the beginning of the axis.

DELTAV   is the number of log cycles per mm (i.e. the reciprocal of the length of one cycle in millimetres).

COMMENTS

A tick mark is placed along the axis for each power of ten and at

each of the nine integer values between.

Annotation is placed at the tick marks as follows:

If a cycle is not less than two millimetres long, the integer tick marks are annotated.

For powers of ten from -14 to +14, the power-of-ten tick marks are annotated with words (e.g., HUNDREDTHS, MILLIONS).

The power-of-ten tick marks are annotated in the form "10¬N" for N less than -14 or N greater than +14.

The SCALG routine may be used for determining FIRSTV and DELTAV, if desired.

Plate 4-7 illustrates logarithmic axes drawn by LBAXS.

4.2.5   SHADE SUBROUTINE

The SHADE subroutine shades any polygon formed by two lines defined by two sets of points. SHADE assumes that an imaginary line connects the first points of the two lines and that another connects the last points of the two lines. Shading is done in the area(s) enclosed by the defined lines and the imaginary lines. The calling sequence is:

```
      CALL SHADE(XARAY1,XARAY2,YARAY1,YARAY2,DLIN,ANGLE,
     1            NPTS1,INC1,NPTS2,INC2)
```

XARAY1,YARAY1 are arrays containing the X and Y coordinates of the data points to be plotted for the first line.

XARAY2,YARAY2 are arrays containing the coordinates for the second line.

DLIN   is the distance between shading lines.

ANGLE   is the shading line inclination angle, in degrees.

NPTS1   is the number of data points forming the first line.

INC1   is the increment between elements of the arrays (XARAY1 and the YARAY1) which are considered in forming the first line.

NPTS2   is the number of data points forming line 2.

INC2   is the increment between elements of the arrays
                  forming line 2.

COMMENTS

The arrays must be dimensioned with at least NPTS+2 elements. The
adjusted minimum value (FIRSTV) and the adjusted delta value
(DELTAV), normally provided by the SCALE subroutine, must be stored
following the data array.

For each array, the adjusted minimum is stored in element
(NPTS*INC+1) of the array (e.g. XARAY1(NPTS1*INC+1)) and the
adjusted delta is stored in element (NPTS*INC+INC+1).

If scaling is not required, the user must place the appropriate
minimum and delta values in the specified elements of the arrays.
For a one-to-one correspondence between array data and plotted
data, these values should be 0.0 (minimum) and 1.0 (delta).

Output produced by SHADE may be seen in Plates 4-8 and 4-9.

4.2.6   CRVPT SUBROUTINE

CRVPT is a subroutine which fits a polynomial curve to a set of
data points. The fitted curve and data are then plotted, along
with reference axes and the equation of the curve. The calling
sequence (in FORTRAN) is:

      CALL CRVPT(XARRAY,YARRAY,INTEQ,NPTS,INC,SH,SW,IBCDT,
     1   NCHAR,IBCDX,NCHARX,IBCDY,NCHARY,INT)

XARRAY,YARRAY are the arrays containing the X and Y coordinates of
              the data. They must be dimensioned with at least
              NPTS+2 elements.

INTEQ   is an array of integers representing the symbols to
                  be plotted at each data point. (See Appendix B)

NPTS   is the number of points to be plotted.

INC   is the increment between elements of XARRAY and
                  YARRAY. INC must be greater than 1 if the values to
                  be plotted are in a mixed or multidimensioned array.
                  (Normally INC=1.)

SH   is the maximum height of the graph, in millimetres.
                 The Y-axis will have a length of (SH+25.0) mm.

SW   is the maximum width of the graph (and the X-axis).

IBCDT   is a CHARACTER variable containing the graph title.

NCHAR   is the number of characters in IBCDT.

IBCDX   is a CHARACTER variable containing the X-axis title.

NCHARX   is the number of characters in IBCDX.

IBCDY   is a CHARACTER variable containing the Y-axis title.

NCHARY   is the number of characters in IBCDY.

INT   is an integer array of digits (ending with a zero)
                 giving the user-specified degrees of the desired
                 polynomials.

COMMENTS

Special uses of NPTS, INC, INTEQ and INT:

NPTS   If NPTS is negative, the absolute value of NPTS is
                 used for the number of data points and the current
                 data set will be plotted with the same scaling used
                 for the preceding data set.

INC   If INC is negative, the absolute value of INC is used
                 for the array increment and the pen is repositioned
                 at the origin of the present graph before returning
                 to the calling program. This allows the next set of
                 data to be plotted on the same graph.

INTEQ   If INTEQ is negative, it is treated as a
                 single-integer variable, instead of as an array, and
                 the plot symbol denoted by the integer equivalent is
                 printed at each data point.

INT   If the first value of INT is zero, the data points
                 are plotted with connecting lines and the second
                 value of INT is interrogated for the degree of the
                 first polynomial to be fit.
                 Any zero INT value following the initial INT value
                 will cause the program to exit from CRVPT.
                 If INT is negative, it is treated as a single-integer
                 variable and each digit of INT is considered an
                 element of an array starting with the units position.

HOW CRVPT WORKS

To produce a plot, CRVPT performs the following steps:

1. Tests the sign of NPTS. If NPTS is positive, CRVPT proceeds to Step 2. If it is negative, CRVPT uses the scaling factors that were determined for the previous data set, then proceeds to Step 3.

2. Uses the SCALE subroutine to determine scaling factors and uses AXIS to plot axes for X and Y and to plot titles. (SCALE and AXIS are CALF77 subroutines described in Chapter 3 of this manual).

3. Plots the title of the page at the top.

4. Calculates the average value of XARRAY and plots below the title the line "Z = X - (average value of XARRAY)".

5. Tests the first digit of INT. If it is zero, the data points are plotted with connecting lines. If it is not zero, connecting lines are not used. The special symbol specified by the integer in array INTEQ is plotted at the corresponding data point.

6. Sets J (the degree of the polynomial) to the first non-zero digit of INT.

7. Computes the coefficients for a polynomial of degree J in powers of Z (calculated in Step 4 above), thus producing the least-squares approximation to the data.

8. Plots the equation of the resulting polynomial below the page title and plots the curve on the page.

9. Causes the special symbol specified by J to be plotted along the curve at 50 mm intervals and alongside the equation of the polynomial.

10. Sets J to the next digit of INT and tests it. If J is not zero, Steps 7 through 10 are repeated. Otherwise CRVPT proceeds to Step 11.

11. Tests the sign of INC. If INC is positive, the pen is repositioned beyond the present graph to the origin of the next graph. If INC is negative, the pen remains positioned on the current graph.

12. Writes a Search Address and returns to the calling program.

NOTE

Before CRVPT is called, the origin should be positioned at least 12.5 millimetres from the bottom of the page to allow room for the X-axis annotation.

OUTPUT DESCRIPTION

OUTPUT GRAPH FORMAT:

The output graph consists of the following components.

1. A user-specified title positioned at the top of the graph and plotted in characters 3.5 mm high.

2. The equation "Z = X - (average value of Xarray)", under the graph title.

3. The equation of the polynomial of best fit for each degree polynomial specified, plotted below the equation "Z = X - (average value of ARRAY)".

4. X and Y axes, with user-specified titles and lengths.

5. The X,Y coordinate data points, with a plot symbol at each point. The data points may or may not be connected with a line as specified by INT.

6. A curve for each degree polynomial specified, representing
    the polynomial that best fits the data points.

7. A plot symbol on the curve, at 50 mm intervals.

ERROR MESSAGES -

Error messages are plotted in the lower right corner of the
output graph and are of the form "ERROR NO. NN", where NN is
as follows:

    NN=J+10 if the degree of the desired polynomial (J) is
    greater than the number of data points.

    NN=I+100 if the Ith data point lies outside the range of
    the plot.

If an error is found, SW will be increased by 3 millimetres so
that the next plot may be origined at (SW,0.0) and it will
clear the error comment. The subroutine will be exited when an
error occurs.

RESTRICTIONS:

1. The value of NPTS in the first call to CRVPT must be
    greater than zero.

2. The highest allowable degree of the polynomial to be fit is
    9.

3. The degree of the polynomial to be fit must be at least one
    less than the number of distinct data points. That is, if
    two or more points have the same coordinates, those points
    must be considered as one.

Plate 4-10 shows a sample output graph.

4.2.7 AROHD SUBROUTINE

AROHD is a FORTRAN subroutine which draws an arrowhead at the end
of a line segment. The calling sequence is:

    CALL AROHD (XPAGE,YPAGE,XTIP,YTIP,AHLEN,AHWID,ICODE)

XPAGE,YPAGE   are the coordinates of the starting point of the line segment that determines the direction of the arrowhead.

XTIP,YTIP   are the coordinates of the tip of the arrowhead. The arrowhead will point in the direction of the line from XPAGE,YPAGE to XTIP,YTIP.

AHLEN   is the arrowhead's length, in millimetres.

AHWID   is the arrowhead's width, in millimetres. If AHWID is zero, the width of the arrowhead is 0.666*AHLEN.

ICODE   is a two-digit decimal code, the absolute value of which determines the type of line and arrowhead drawn. If ICODE is negative, then XPAGE and YPAGE will be ignored and the current pen position used in their place.

               The first digit specifies the line joining XPAGE,YPAGE and XTIP,YTIP:
                   0 if no line is desired.
                   1 if a line is desired.
                   2 if a line is desired and a second arrowhead, pointing in the opposite direction, is desired at XPAGE,YPAGE.

               The second digit specifies the type of arrowhead desired, as shown in Plate 4-11.

4.2.8 ARROW SUBROUTINE

ARROW is a FORTRAN subroutine which draws a line through a series of points and places an arrow on the end. The calling sequence is:

        CALL ARROW(XARRAY,YARRAY,NPTS,INC,IATYPE)

XARRAY,YARRAY are the names of arrays containing the X and Y coordinates of the points to be plotted.

NPTS   is the number of data points to be plotted.

INC   is the increment between array elements. INC is greater than 1 if the values to be plotted are in a mixed or multidimensioned array.

IATYPE   specifies the type of arrow desired, as illustrated

in Plate 4-11.

COMMENTS

The tip of the arrow appears at the point (XARRAY(NPTS*INC-INC+1), YARRAY(NPTS*INC-INC+1)).

The arrays XARRAY and YARRAY must be dimensioned with at least (NPTS+7)*INC elements. The last five elements in the arrays are used by the ARROW subroutine.

The adjusted minimum value (FIRSTV) and the adjusted delta value (DELTAV), (which can be provided by the SCALE subroutine) must be stored in the array. For the X array, the adjusted minimum is stored in XARRAY(NPTS*INC+1) and the adjusted delta in XARRAY(NPTS*INC+INC+1), and similarly for the Y array. For more information about the SCALE routine, see Chapter 3.

The length of the arrowhead is 8 millimetres before scale factors are applied. The plotting is optimized and may start at either end of the line or arrow.

4.2.9 CNTRL SUBROUTINE

CNTRL is a FORTRAN subroutine which draws "centre lines" between successive data points. Each "centre line" is a broken line consisting of alternating long dashes and short dashes. The long dashes are 4/11 and the short dashes 1/11 of the distance between points. The calling sequence is:

          CALL CNTRL(XARRAY,YARRAY,NPTS,INC)

XARRAY,YARRAY are the arrays containing the X and Y coordinates of the points to be plotted.

NPTS   is the number of points to be plotted.

INC   is the increment between elements of the array. INC is greater than 1 if the values to be plotted are in a mixed or multidimensioned array.

COMMENTS

The arrays must be dimensioned with at least NPTS+2 elements. The adjusted minimum value (FIRSTV) and the adjusted delta value (DELTAV) must be stored in the array, as described in the previous section.

Centre lines drawn between successive points are seen in Plate 4-12.

4.2.10 DIMEN SUBROUTINE

DIMEN is a FORTRAN subroutine which generates a dimension line with short extension lines at the tips of the line's arrowheads. The calling sequence is:

CALL DIMEN(XPAGE,YPAGE,DIME,ANGLE,SCALER)

XPAGE,YPAGE   are the coordinates of the dimension line's starting point.

DIME   is the value to be plotted along the line.

ANGLE   is the dimension line's angle, in degrees.

SCALER   is the scale ratio used in the drawing.

COMMENTS

The actual length of the dimension line is the product of DIME and SCALER. If the actual length is 30 millimetres or longer, the annotation and arrows are plotted in the middle of the dimension line. If the dimension line is between 20 and 30 mm long the annotation is placed outside; if it is under 20 mm long the arrowheads are placed outside as well.

Plate 4-13 illustrates these points.

4.2.11 LABEL SUBROUTINE

LABEL is a FORTRAN subroutine which plots an array of alphanumeric
characters along a straight-line segment. The segment may be at
any angle. The characters are automatically centred and their size
is adjusted to fit between the specified end points of the segment.
A floating-point number may be plotted in addition to the
alphanumeric characters.

The calling sequence is:

     CALL LABEL(XPAGE1,YPAGE1,XPAGE2,YPAGE2,IBCD,NCHAR,
     1   HEIGHT,ISIDE,DSTFLN,FPN,NDEC)

XPAGE1,YPAGE1 are the coordinates of the starting point, P1, of the
          line segment along which the characters are to be
          plotted.

XPAGE2,YPAGE2 are the coordinates of the ending point, P2, of the
          line.

IBCD   is a CHARACTER variable containing the text to be
          plotted.

NCHAR   is the number of characters from IBCD to be plotted.

HEIGHT   is the height of the characters, in millimetres.

ISIDE   is a character-positioning code. If the units digit of
          ISIDE is:

          1, the characters are placed on the clockwise side of
              the line (viewed looking in the direction P1-P2);
          2, the characters are plotted on the counter-clockwise
              side of the line.

          If the tens digit of ISIDE is 1, FPN is plotted along
          with IBCD.

          If ISIDE is negative, the text is inverted for
          readability. (Only for lines that lie in the second or
          third quadrants.)

DSTFLN   is the distance of IBCD from line P1-P2.

          If ISIDE is +1 or +11, DSTFLN is measured from P1-P2 to
          the top of the characters.

          If ISIDE is +2 or +12, DSTFLN is measured from P1-P2 to
          the bottom of the characters.

FPN is the floating-point number to be converted and plotted
    when ISIDE is equal to 11 or 12.

NDEC is the number of places to the right of the decimal point
    in FPN.

COMMENTS

LABEL does not plot the line between P1 and P2.

If the characters in array IBCD do not fit within the line
P1-P2 as indicated, LABEL adjusts HEIGHT until it fits, or down
to the minimum, 1.75 mm. If HEIGHT has to be made less than
1.75 mm, then nothing is plotted.

In centring the characters along the line, both arrays IBCD and
FPN are considered.

4.2.12 CIRCL SUBROUTINE

CIRCL is a FORTRAN subroutine that draws, starting at a given
point, an arc which may be extended to form a circle or spiral.
The calling sequence is:

   CALL CIRCL(XPAGE,YPAGE,THO,THF,RO,RF,DI)

XPAGE,YPAGE are the coordinates of the arc's starting point.

THO is the arc's initial radius angle, in degrees from the
    X-axis.

THF is the arc's final radius angle, in degrees from the
    X-axis.

RO is the arc's initial radius length, in millimetres.

RF is the arc's final radius length.

DI is used to specify the type of line desired.
     If DI = 0.0, a solid arc is drawn.
     If DI = 0.5, a dashed arc is drawn.

COMMENTS

> THO and THF may be positive or negative. If THO is less than THF, the arc is drawn in a counter-clockwise direction. Otherwise, the arc is drawn in a clockwise direction.

> Plate 4-14 shows the effect of varying the parameters.

4.2.13 DASHL SUBROUTINE

DASHL is a FORTRAN subroutine which draws dashed lines connecting a series of points. The calling sequence is:

        CALL DASHL(XARRAY,YARRAY,NPTS,INC)

XARRAY,YARRAY are arrays containing the X and Y coordinates of the points to be plotted.

NPTS   is the number of points to be plotted.

INC   is the increment between array elements. INC is greater than 1 if the values to be plotted are in a mixed or multidimensioned array.

COMMENTS

> A dashed line, with dashes approximately 2.5 mm long, is drawn connecting sequential points. Coding is optimized so that plotting may either begin at either the first or the last point, and progress forward or backward.

> The arrays must be dimensioned with at least NPTS+2 elements, to allow the adjusted minimum value (FIRSTV) and the adjusted delta value (DELTAV), (which can be provided by the SCALE subroutine) to be stored. For the X array, the adjusted minimum is stored in XARRAY(NPTS*INC+1) and the adjusted delta in XARRAY(NPTS*INC+INC+1), and similarly for the Y array. For more information about the SCALE routine, see Chapter 3.

DASHL is illustrated with DASHP in Plate 4-15.

4.2.14 DASHP SUBROUTINE

DASHP is a FORTRAN subroutine which draws a dashed line from the pen's present position to a specified point.

The calling sequence is:

        CALL DASHP(XPAGE,YPAGE,DASH)

XPAGE,YPAGE   are the coordinates of the point to which the dashed line is to be drawn.

DASH   is the length of each dash and of the space between dashes. If the dash length is more than half the line length, it is adjusted to half the line length.

See Plate 4-15 for the types of dashed lines produced.

4.2.15 ELIPS SUBROUTINE

ELIPS is a FORTRAN subroutine which draws an ellipse or elliptical arc. The calling sequence is:

        CALL ELIPS(XPAGE,YPAGE,RMAJ,RMIN,ANGLE,THO,THF,IPEN)

XPAGE,YPAGE   are the coordinates of the starting point of the ellipse or arc.

RMAJ,RMIN   are the lengths of the semimajor and semiminor axes, respectively.

ANGLE   is the angle of the major axis, in degrees.

THO,THF   are the angles, in degrees with respect to ANGLE, of the arc's starting and ending points. They determine the direction of the arc, just as they do in the CIRCL routine.

IPEN   determines the pen's UP/DOWN status during the move to the arc's starting point. If IPEN=3, the pen is up for the move, while if IPEN=2, it is down.

Various ellipses and arcs are shown in Plate 4-16.

4.2.16 FIT SUBROUTINE

FIT is a FORTRAN subroutine which draws a semi-hyperbolic curve
through three points. The calling sequence is:

        CALL FIT(XPAGE1,YPAGE1,XPAGE2,YPAGE2,XPAGE3,YPAGE3)

XPAGEn,YPAGEn   are the coordinates of one of the three points
                    through which the curve passes.

    COMMENTS

    This subroutine generates a semi-hyperbolic fit using the three
    given points. A set of points for which a fit is not possible
    is drawn with straight-line segments.

    RESTRICTIONS

    The curve through the three points must not be multi-valued in
    both X and Y. That is, the middle point (XPAGE2,YPAGE2) must
    be between the endpoints along the X-axis or the Y-axis:

        XPAGE1<XPAGE2<XPAGE3

            or

        XPAGE1>XPAGE2>XPAGE3

            or

        YPAGE1<YPAGE2<YPAGE3

            or

        YPAGE1>YPAGE2>YPAGE3

Plate 4-17 illustrates the abilities of the FIT subroutine.

4.2.17 GRID SUBROUTINE

GRID is a FORTRAN subroutine which draws a linear grid, of
selectable density. The calling sequence is:

CALL GRID(XPAGE,YPAGE,DELTAX,DELTAY,NXSP,NYSP)

XPAGE,YPAGE   are the coordinates of grid's lower left corner.

DELTAX,DELTAY   are the sizes of the squares in the X and Y directions, respectively.

NXSP,NYSP   are the numbers of squares in the X and Y directions, respectively.

COMMENTS

GRID generates a linear grid of any size. The number of lines drawn is NXSP+1 in the X direction and NYSP+1 in the Y direction.

Various grids are illustrated in Plate 4-18.

4.2.18   POLY SUBROUTINE

POLY is a FORTRAN subroutine used to draw equilateral, or regular, polygons. The calling sequence is:

CALL POLY(XPAGE,YPAGE,SLEN,SN,ANGLE)

XPAGE,YPAGE   are the coordinates of the polygon's starting point.

SLEN   is the length of a side of the polygon.

SN   is the number of sides of the polygon. If SN is negative, a star with |SN| points is drawn.

ANGLE   is the angle, in degrees, of the first side of the polygon.

The effects of changing SLEN, SN and ANGLE are shown in Plate 4-19.

4.2.19   RECT SUBROUTINE

RECT is a FORTRAN subroutine used to draw rectangles. The calling
sequence is:

          CALL RECT(XPAGE,YPAGE,HEIGHT,WIDTH,ANGLE,IPEN)

XPAGE,YPAGE   are the coordinates of the rectangle's lower left
                corner, before rotation.

HEIGHT   is the rectangle's height, in millimetres.

WIDTH   is the rectangle's width, in millimetres.

ANGLE   is the angle, in degrees, at which the rectangle's
                base is to be drawn. (Rectangle is rotated about
                XPAGE,YPAGE.)

IPEN   is the code that moves the pen to the rectangle's
                starting point.   If IPEN=3, the plotter pen is up
                during the move, while if IPEN=2, the pen is down.

4.2.20   CURVX SUBROUTINE

CURVX is a FORTRAN subroutine which plots a function of X over a
given range. The calling sequence is:

          CALL CURVX(XO,XF,COEFF1,EXP1,COEFF2,EXP2,
          1   COEFF3,EXP3,COEFF4,EXP4)

XO,XF   are the starting and ending values of X.

COEFF1..COEFF4 are the coefficients of the polynomial that defines
                the function to be plotted.

EXP1..EXP4   are the exponents of the polynomial that defines
                the function to be plotted.

COMMENTS

The polynomial that defines the function to be plotted is:

Y = COEFF1*(X¬EXP1)+COEFF2*(X¬EXP2)+
    COEFF3*(X¬EXP3)+COEFF4*(X¬EXP4)

for values of X from XO to XF, where delta X = 0.01.

If X is zero or negative, errors may be generated.

CURVX is illustrated in Plate 4-20 and with CURVY in Plate 4-21.

4.2.21 CURVY SUBROUTINE

CURVY is a FORTRAN subroutine which plots a function of Y over a given range. The calling sequence is:

        CALL CURVY(YO,YF,COEFF1,EXP1,COEFF2,EXP2,
        1   COEFF3,EXP3,COEFF4,EXP4)

The arguments to this routine are similar to those for the CURVX routine, in the previous section.

COMMENTS

The polynomial that defines the function to be plotted is:

X = COEFF1*(Y¬EXP1)+COEFF2*(Y¬EXP2)+
    COEFF3*(Y¬EXP3)+COEFF4*(Y¬EXP4)
        for values of Y from YO to YF, where delta Y = 0.01.

NOTE

If Y is zero or negative, errors may be generated.

See Plate 4-21 for an example of CURVY.

4.2.22   FLINE SUBROUTINE

FLINE plots points from an array, in much the same way as does the
LINE subroutine, with the additional option of drawing straight
lines or smooth curves through each successive point.

The calling sequence is:

        CALL FLINE(XARRAY,YARRAY,NPTS,INC,LINTYP,INTEQ)

XARRAY,YARRAY are arrays containing the X and Y coordinates of the
              points to be plotted. They must be dimensioned with
              at least NPTS+2 elements.

              The adjusted minimum and the adjusted delta must be
              stored in the X and Y data arrays. Refer to Comments
              below.

NPTS   is the number of data points to be plotted.   If
              NPTS>0, a straight line is drawn between the points.
              If NPTS<0, a smooth curve, drawn using a modified
              spline-fitting technique, is drawn between the
              points.

INC   is the increment between elements to be taken from
              the array. INC>1 if the values to be plotted are in
              a mixed array.

LINTYP   is used to control the type of graph produced, as
              follows:
                  0 a line is plotted between successive data
                     points.
                  1 a line plot is produced, with a symbol at each
                     data point.
                  2 a line plot is produced, with a symbol at every
                     second data point.
                  n a line plot is produced, with a symbol at every
                     nth data point.
                 -n a symbol appears at every nth data point, but
                     connecting lines are not plotted.

INTEQ   is the integer used to specify the symbol to be
              plotted   at   each   point.   (Refer Appendix B for
              possible values of INTEQ.)

Plate 4-22 shows eight points plotted with straight lines, smooth
curves and with the data points indicated.

4.2.23   LGLIN SUBROUTINE

LGLIN is a FORTRAN subroutine used to plot data either in log-log
or in semi-log mode. The calling sequence is:

        CALL LGLIN(XARRAY,YARRAY,NPTS,INC,LINTYP,INTEQ,LOGTYP)

XARRAY,YARRAY   are the arrays containing the X and Y coordinates,
                respectively, to be plotted.

NPTS   is the number of points to be plotted.

INC   is the increment between elements in the array.
                INC>1 if the values to be plotted are in a mixed
                array.

LINTYP   is used to control the type of graph produced, in
                just the same way as it is in the FLINE subroutine
                (previous section).

INTEQ   is the integer used to specify the symbol to be
                plotted at each point. (Refer to Appendix B for
                possible values of INTEQ.)

LOGTYP   specifies the type of plot, as follows:

                -1 a semi-log plot logarithmic in X and linear in Y
                    is produced.
                0   a log-log plot is produced.
                +1 a semi-log plot linear in X and logarithmic in Y
                    is produced.

COMMENTS

        The arrays XARRAY and YARRAY must be dimensioned with at least
        NPTS+2 elements.   The adjusted minimum values and the delta
        values (normally provided by the SCALG subroutine   for
        logarithmic data and by the SCALE subroutine for linear data)
        must be stored in the data arrays.

        For the X array, the adjusted   minimum   is   stored   in
        XARRAY(NPTS*INC+1)   and   the   adjusted   delta   is   in
        XARRAY(NPTS*INC+INC+1).   Similarly, for the Y array,   the
        minimum   is   in   YARRAY(NPTS*INC+1) and the delta is in
        YARRAY(NPTS*INC+INC+1).

        If scaling is not required, the user must place the appropriate
        minimum and delta values in the specified locations in the
        arrays. For linear arrays, these values should be   0.0

4-24

(minimum) and   1.0   (delta),   to   ensure   a   one-to-one
correspondence between array data and plotter data.

LGLIN is illustrated in Plate 4-23.

4.2.24 LGAXS SUBROUTINE

LGAXS is a FORTRAN subroutine which draws a logarithmic axis with
annotation in powers of ten. The calling sequence is:

        CALL LGAXS(XPAGE,YPAGE,IBCD,NCHAR,AXLEN,
        1   ANGLE,FIRSTV,DELTAV)

XPAGE,YPAGE   are the coordinates of the axis starting point.

IBCD   is a CHARACTER variable containing the axis title.

NCHAR   is the number of characters in the axis title.   If
               NCHAR is negative, annotation is placed on the
               clockwise side of the axis. Otherwise, annotation is
               placed on the counter-clockwise side of the axis.

AXLEN   is the length of the axis, in millimetres.

ANGLE   is the angle, in degrees, at which the axis is to be
               drawn.

FIRSTV   is the value of annotation at the first tick mark.

DELTAV   is the number of log cycles per mm (the reciprocal of
               the length of one cycle, in millimetres).

   COMMENTS

   A tick mark is placed of the axis for each power of ten and for
   each of the nine integer values between.

   Annotation is placed at the tick marks as follows:

        If a cycle is not less than two millimetres long, the
        integer tick marks are annotated.
        The power-of-ten tick marks are annotated in the form
        10^{-N} .

   SCALG may be used for determining FIRSTV and DELTAV, if

desired.   SCALG and LGAXS were used in producing Plate 4-23.

Plate 4-24 shows a variety of axes drawn with LGAXS.

4.2.25 POLAR SUBROUTINE

POLAR is a FORTRAN subroutine which scales and plots a radial
variable of any magnitude against an angular variable (angle in
radians) as polar coordinates. POLAR produces either a line plot
(with lines connecting data points) or a point plot, centred on the
origin.

The calling sequence is:

    CALL POLAR(RADAR,ANGAR,NPTS,INC,LINTYP,INTEQ,RMAX,DR)

RADAR   is an array containing the radial values, in mm.

ANGAR   is an array of angular values, in radians.

NPTS   is the number of points to be plotted.

INC   is the increment between elements in the array.
                Every INCth point in the array is plotted.

LINTYP   is used to control the type of graph produced, in
                just the same way as it is in the FLINE routine.

INTEQ   is an integer specifying the symbol to be plotted at
                each point. (Refer to Appendix B for possible values
                of INTEQ.)

RMAX   is the maximum radius for the plotting area, in page
                millimetres.   If RMAX$\leq$0.0, then DR must contain the
                scale factor.

DR   is expressed in units of data per millimetre.

COMMENTS

    Angles are measured in radians counterclockwise around the
    origin, relative to the X axis.

    Radial values are measured from the origin, with negative
    values being plotted radially opposite from positive values.

4-26

POLAR is shown in Plate 4-23.

4.2.26 SCALG SUBROUTINE

SCALG is a FORTRAN subroutine used to determine scale factors of a data array to be plotted on a logarithmic scale. The scale factors are those used by the LGLIN, LGAXS and LBAXS routines.

The calling sequence is:

    CALL SCALG(ARRAY,AXLEN,NPTS,INC)

ARRAY   is the array containing the data to be scaled.

AXLEN   is the maximum length over which the data is to be plotted.

NPTS   is the number of values in ARRAY to be scaled.

INC   is the increment between elements of the array.   INC is greater than 1 if the values to be plotted are in a mixed or multidimensioned array.

COMMENTS

    The array must be dimensioned with at least NPTS+2 elements. The adjusted minimum value (FIRSTV) and the delta value (DELTAV) are provided by the SCALG routine in the data array.

    The adjusted minimum is stored in ARRAY(NPTS$*$INC+1) and the adjusted delta is in ARRAY(NPTS$*$INC+INC+1).

    Every INCth element of the ARRAY, beginning with the first, is scanned to find the minimum and maximum values of the array. Next, the greatest value of (10^{-n}) (integer n) less than or equal to the minimum value is found and then stored in ARRAY(NPTS$*$INC+1).   Finally, the smallest value of (10^{-m}) (integer m) greater than or equal to the maximum value is established.

    The delta value is the difference between the minimum and maximum powers of ten, divided by AXLEN, yielding log cycles per inch.   The delta value is stored in ARRAY(NPTS$*$INC+INC+1).

EXAMPLES

A.

    For the following array of values:

```
ARRAY(1) = 1500.0
ARRAY(2) = 3000.0
ARRAY(3) = 2500.0
ARRAY(4) =  300.0
```

    and the following argument values:

```
AXLEN = 2.0
NPTS  = 4
INC   = 1
```

    the FIRSTV and DELTAV values stored by SCALG are:

```
FIRSTV:     ARRAY(5) = 100.0
DELTAV:     ARRAY(6) =   1.0.
```

B.

    If the value of AXLEN in Example A were changed to 4.0, the
resultant values stored would be:

```
FIRSTV:     ARRAY(5) = 100.0
DELTAV:     ARRAY(6) =   0.5
```

C.

    For the following array of values:

```
ARRAY(1) =    1.2
ARRAY(2) = 100.0
ARRAY(3) =    2.3
ARRAY(4) =  88.0
ARRAY(5) =    1.8
ARRAY(6) =    0.0
ARRAY(7) =    0.8
ARRAY(8) =  20.0
ARRAY(9) =    0.7
ARRAY(10)=  10.0
```

    and the following argument values:

```
AXLEN = 10.0
NPTS  = 5
INC   = 2,
```

    the FIRSTV and DELTAV values are determined from the value
of items 1, 3, 5, 7, and 9 in the above array. Note that

the computed values are also stored two subscript elements
apart.

               FIRSTV: ARRAY(11) = 0.1
               DELTAV: ARRAY(13) = 0.2

4.2.27 SMOOT SUBROUTINE

SMOOT is a FORTRAN subroutine which draws a smooth curve through a
set of points, using a modified spline-fitting technique. The
subroutine receives a single coordinate pair on each call and
accumulates the points until it has received a sufficient number to
compute a pair of cubic parametric equations for a smooth curve.
This accumulation method requires the user to specify an initial
and a terminal call to the subroutine.

The SMOOT subroutine operates in either of two modes: Smooth Mode
and Plot Mode. In smooth mode, SMOOT plots the smooth curve, while
in plot mode, SMOOT acts just like the PLOT routine, with IPEN
equal to 2 or 3.

The calling sequence is:

    CALL SMOOT(XPAGE,YPAGE,IPEN)

XPAGE,YPAGE   are the coordinates of a single point on the curve.

IPEN   determines the mode and action of the SMOOT
               subroutine. See details below.

DETAILED DESCRIPTION

     The first call to SMOOT must use an IPEN value of 0 or -1 to
     put SMOOT in the Smooth Mode.

        IPEN = 0 XPAGE,YPAGE define the initial point (P1) on the
                  curve. The smoothing function ends at the last
                  point (Pn). An open curve is produced.

        IPEN = -1 XPAGE,YPAGE define the initial point (P1) on the
                  curve. The smoothing function continues from the
                  last point (Pn) back to the initial point (P1).   A
                  closed curve is produced.

     When SMOOT is in the Smooth Mode, IPEN performs the following

functions:

IPEN = -2 XPAGE,YPAGE are used to define points and a smoothed curve is drawn through the points on the curve.

IPEN = -3 XPAGE,YPAGE are used to define points and the pen, in the up position, is moved through these points. The smoothing function is maintained.

IPEN = 2,3 The call is treated as a normal CALL PLOT(XPAGE,YPAGE,IPEN), and the point is not considered a point on the curve. The point of departure from the curve is the second-last point received by SMOOT, not the last point.

When the next call to SMOOT with IPEN = -2 or -3 is received, the pen is repositioned to the point where it left the smooth curve. The smooth curve is then continued as though calls with IPEN = 2 or 3 had not occurred.

IPEN ≤ -24 is used for the terminal call while SMOOT is in the Smooth Mode. XPAGE,YPAGE specify the terminal point. The curve is finished and the subroutine returns to the plot Mode.

COMMENTS

When SMOOT is called in Smooth Mode, the pen is not moved until three points on an open curve or four points on a closed curve have been received. For subsequent calls to SMOOT, the actual pen position is the second-last point received.

Calls to other plotting subroutines may be intermixed with calls to SMOOT. Point-of-departure restrictions are the same as noted in the Smooth Mode description above.

The first call to SMOOT must be with IPEN = 0 or -1.

Plate 4-25 shows similar curves drawn with SMOOT and FLINE subroutine calls.

SAMPLE OF BUSINESS SUBROUTINES
USING AXISB, AXISC AND BAR.
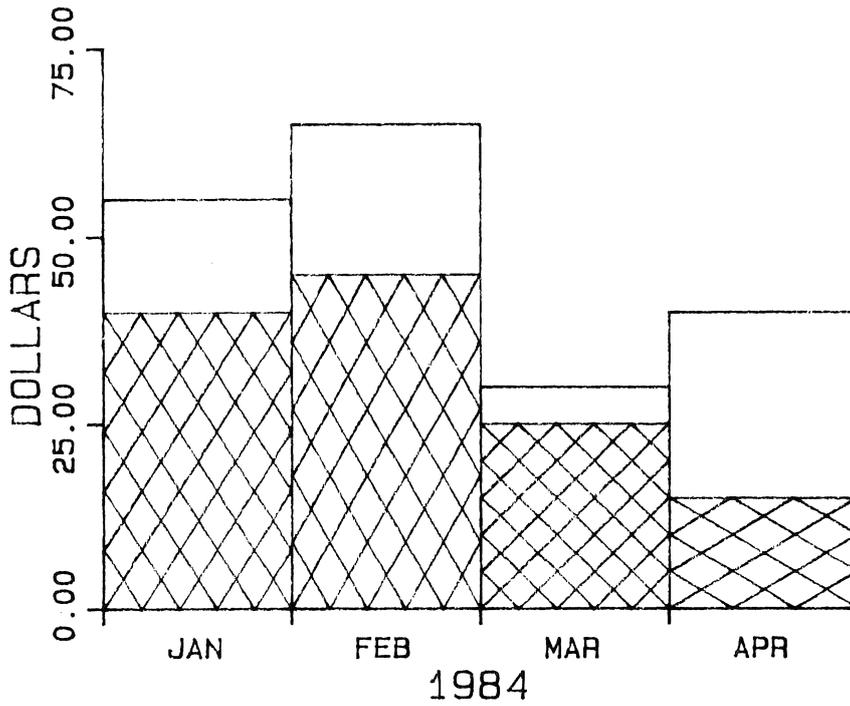
USING SCALG, LBAXS AND LGLIN

Plate 4-1
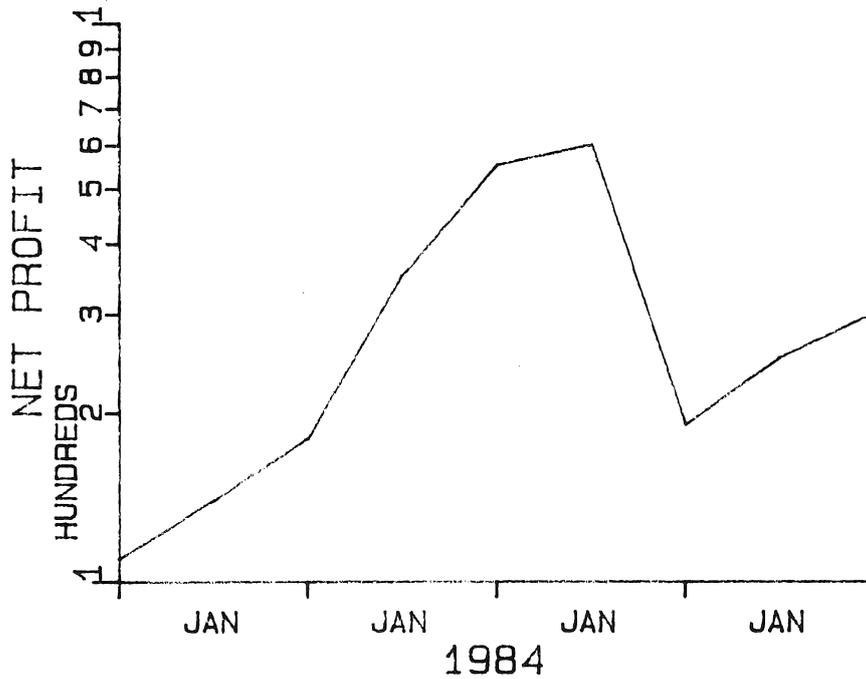
CALL AXISB (XPAGE, YPAGE, IBCD, NCHAR, AXLEN, ANGLE, FIRSTV, DELTAV)

VARYING IBCD, NCHAR, AXLEN, ANGLE, AND DELTAV

IBCD='AN AXIS WITH ITS TITLE LONGER THAN ITS LENGTH'
NCHAR=-45
AXLEN=100.
ANGLE=90.
DELTAV=.4
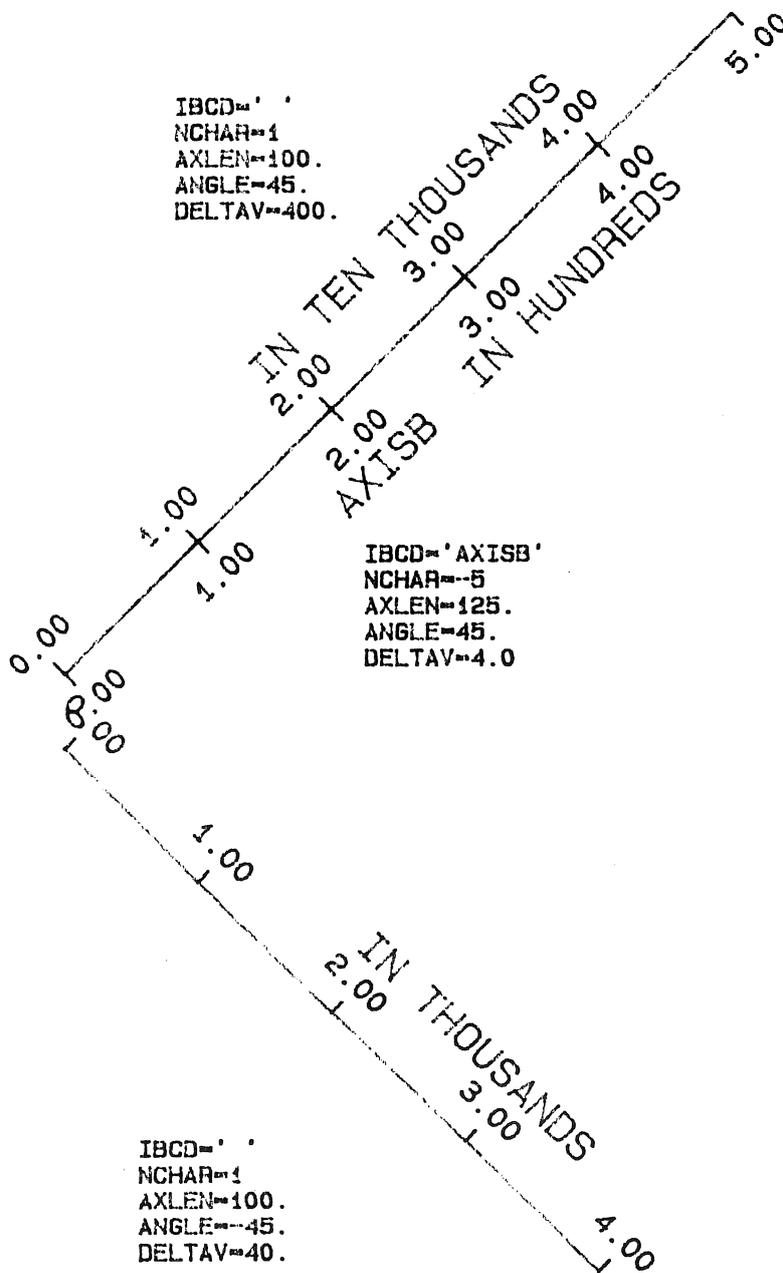
IBCD=' '
NCHAR=1
AXLEN=100.
ANGLE=45.
DELTAV=400.

IBCD='-AXISB'
NCHAR=-6
AXLEN=125.
ANGLE=90
DELTAV=-.04

IBCD='AXISB'
NCHAR=-5
AXLEN=125.
ANGLE=45.
DELTAV=4.0

IBCD=' '
NCHAR=1
AXLEN=100.
ANGLE=-45.
DELTAV=40.

Plate 4-2

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

VARYING FIRSTV AND DELTAV

IBCD='AXISB'
NCHAR=-5
AXLEN=100.

AXISB

FIRSTV=.001   0.00   0.03   0.05   0.08   0.10
DELTAV=.001

AXISB   IN HUNDREDS

FIRSTV=90.   0.90   3.40   5.90   8.40   10.90
DELTAV=10.

AXISB   IN THOUSANDS

FIRSTV=-1000.   -1.00   1.50   4.00   6.50   9.00
DELTAV=100.

AXISB $*10^7$

FIRSTV=-900000.   -0.09   4.91   9.91   14.91   19.91
DELTAV=2000000.

AXISB $*10^{10}$

FIRSTV=0.0   0.00   2.50   5.00   7.50   10.00
DELTAV=1000000000.

Plate 4-3
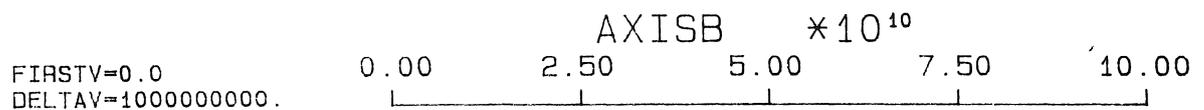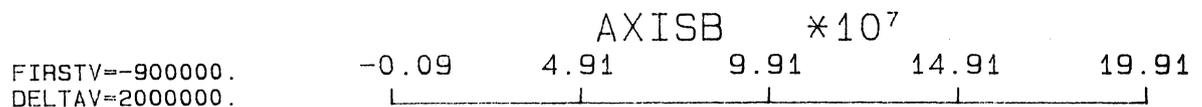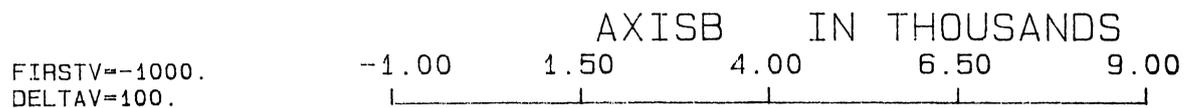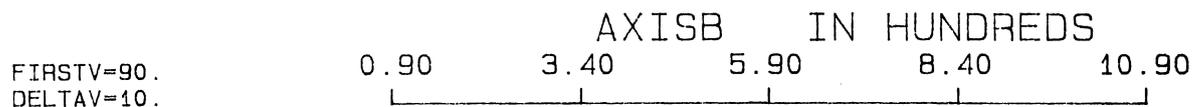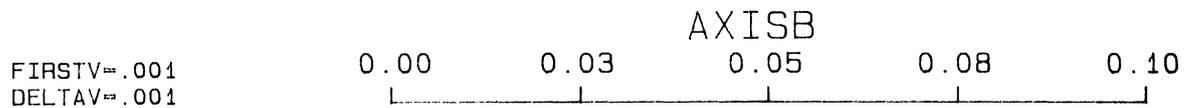
CALL AXISC (XPAGE, YPAGE, IBCD, NCHAR, AXLEN,
ANGLE, FIRSTV, DELTAV)

VARYING NCHAR, AXLEN, ANGLE, FIRSTV AND DELTA

IBCD='AXISC'
AXLEN=100.
FIRSTV=1.0

NCHAR=5
ANGLE=0.0
DELTAV=3.0

AXISC

JAN   APR   JUL   OCT

NCHAR=-5
ANGLE=270.
DELTAV=4.0

JAN

MAY

AXISC

SEP

JAN

POSITIVE NCHAR (19)

MAR   JUL   NOV

MAR   JUL   NOV

NEGATIVE NCHAR (-20)

NCHAR=-20
ANGLE=45.
DELTAV=4.0
AXLEN=75.

NCHAR=-5
ANGLE=90.
DELTAV=2.0

JAN

SEP

MAY

AXISC

JAN

JAN   FEB   MAR   APR

AXISC

NCHAR=-5
ANGLE=0.0
DELTAV=1.0
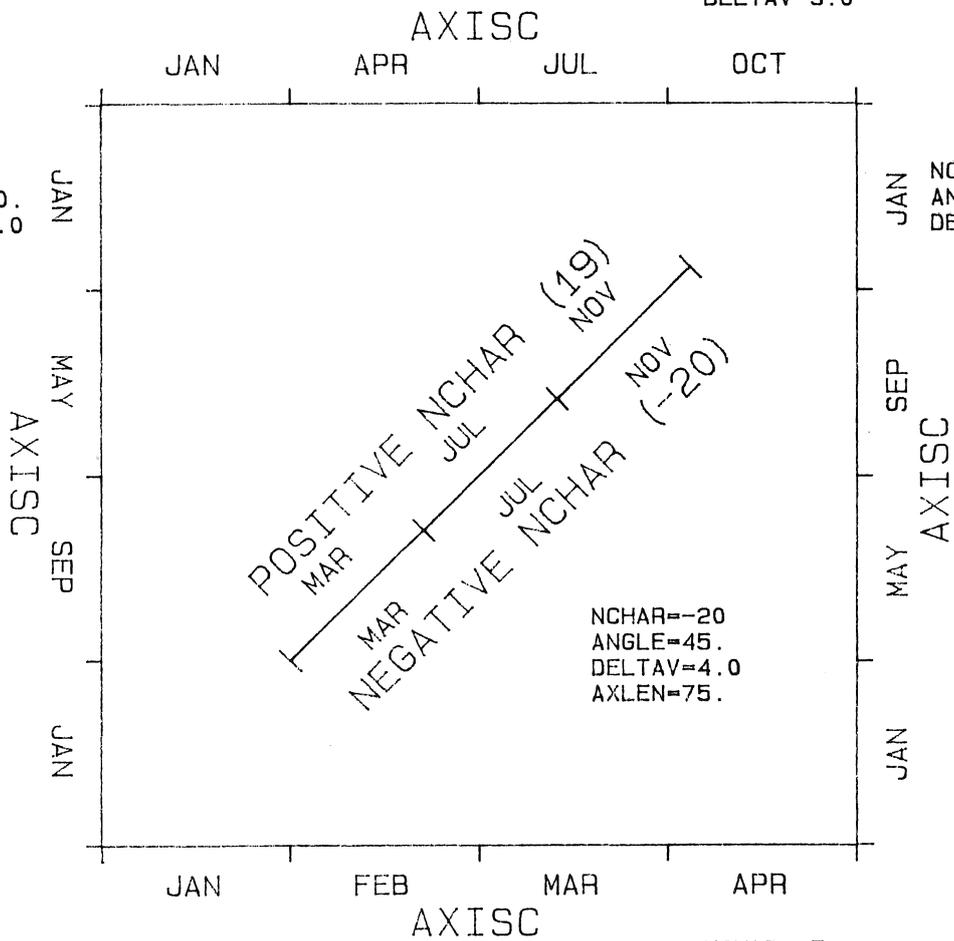
IBCD='1984 (24BLANKS) 1985'
NCHAR=-32
AXLEN=175.
ANGLE=0.0
DELTAV=4.0
(FACTORED BY .75)

JAN   MAY   SEP   JAN   MAY   SEP   JAN
       1984   1985

Plate 4-4
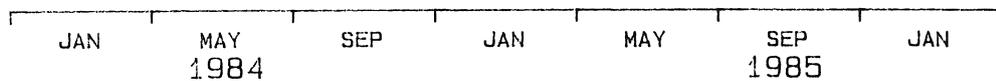
CALL BAR (XPAGE, YPAGE, ANGLE, HEIGHT, WIDTH,
              SH, IHAT, NPI)

VARYING ANGLE, HEIGHT, WIDTH AND DELTA

NPI=.125
SH=40.

ANGLE=45.
HEIGHT=80.
WIDTH=30.
IHAT=2

ANGLE=0.
HEIGHT=100.
WIDTH=40.
IHAT=1

ANGLE=90.
HEIGHT=60.
WIDTH=20.
IHAT=3

ANGLE=180.
HEIGHT=50.
WIDTH=10.
IHAT=4

Plate 4-5
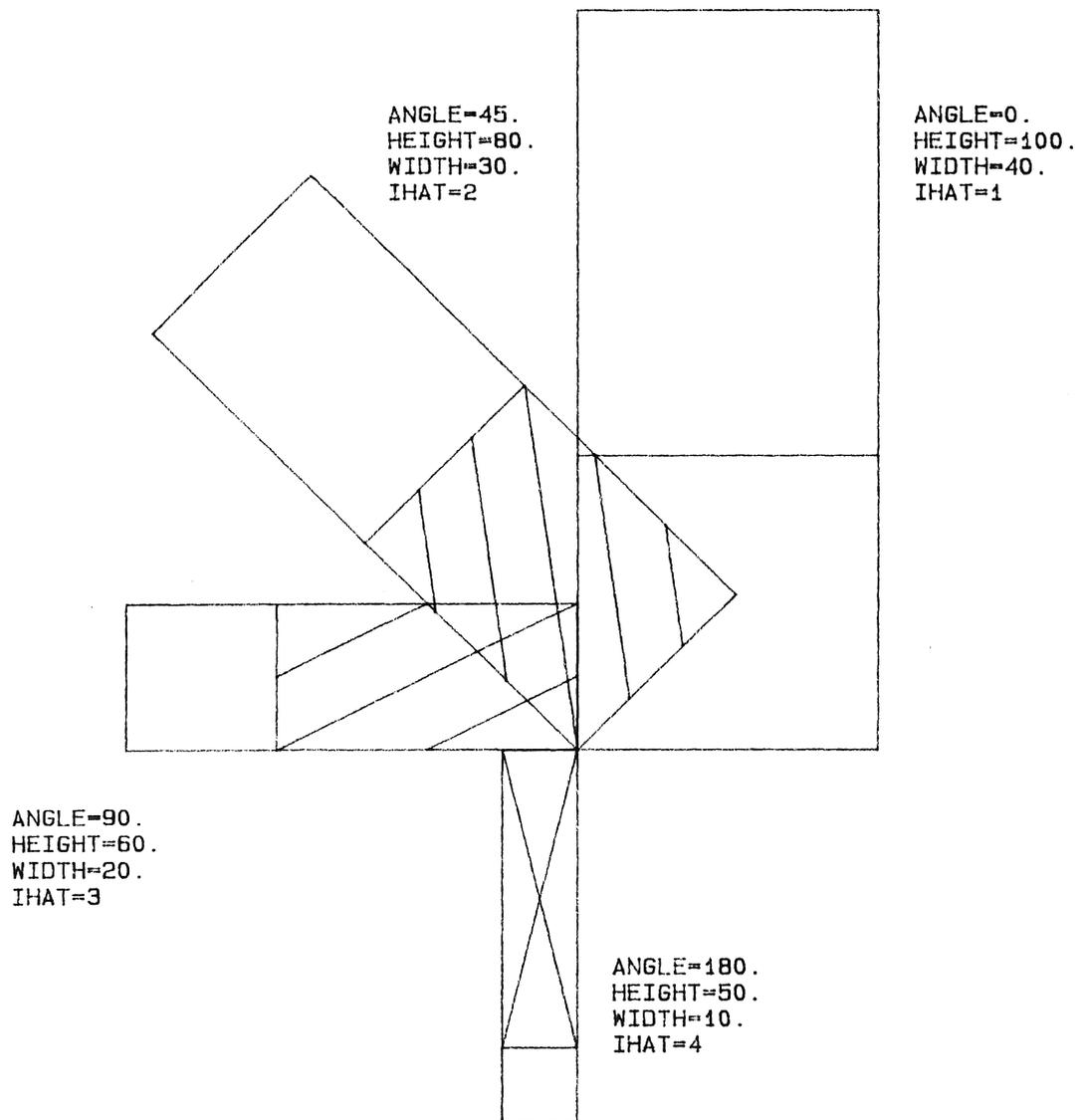
CALL BAR (XPAGE, YPAGE, ANGLE, HEIGHT, WIDTH,
SH, IHAT, NPI)

VARYING HEIGHT, SH AND NPI

ANGLE=0.
WIDTH=25.
IHAT=2
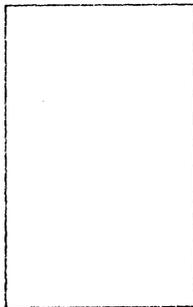
HEIGHT=40.
SH=0.0
NPI=0
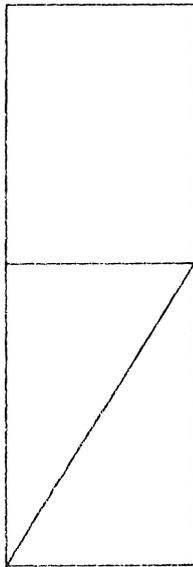
HEIGHT=75.
SH=40.
NPI=0
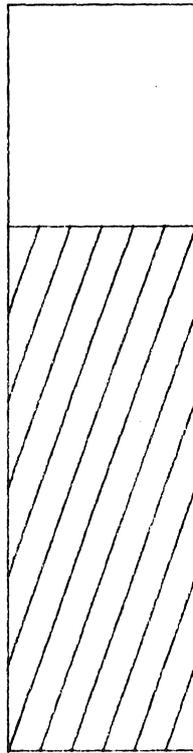
HEIGHT=100.
SH=70.
NPI=.25
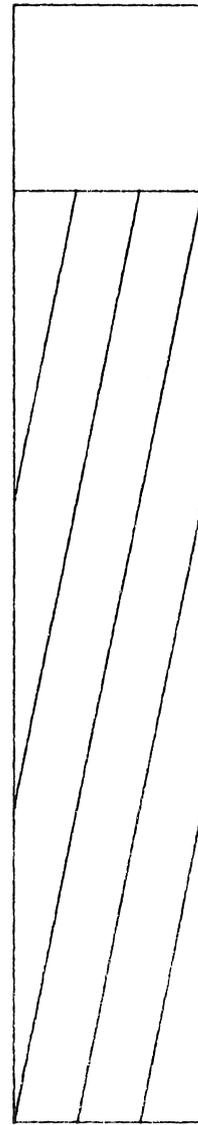
HEIGHT=150.
SH=125.
NPI=.125

Plate 4-6

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

VARYING NCHAR, ANGLE, FIRSTV AND DELTAV

IBCD='LGAXS'
AXLEN=100.
NCHAR=-5

ANGLE=45.
FIRSTV=100.
DELTAV=.02
NCHAR=5
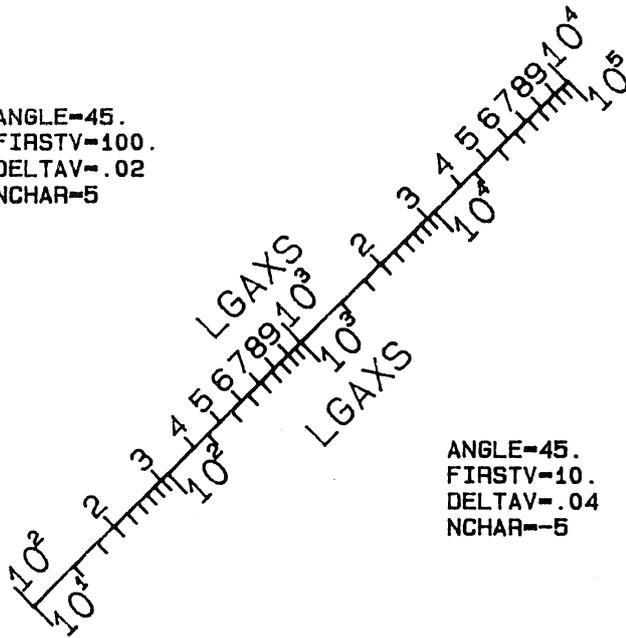
ANGLE=45.
FIRSTV=10.
DELTAV=.04
NCHAR=-5

ANGLE=0.0
FIRSTV=.0001
DELTAV=0.03

10^{-4} 10^{-3} 10^{-2} 10^{-1}

LGAXS

FIRSTV=30.
DELTAV=.02

3 4 5 6 7 8 9 10^{2} 2   3 4 5 6 7 8 9 10^{3} 2   3

LGAXS

FIRSTV=1000000.
DELTAV=.01

10^{6} 2   3   4   5   6   7   8   9 10^{7}

LGAXS

Plate 4-7

USING SHADE SUBROUTINE

△ NET SALES

⊕ COST OF SALES

IN MILLIONS

1984
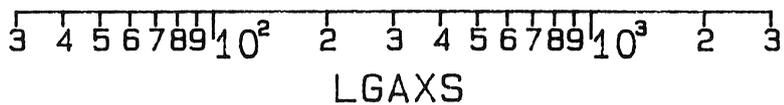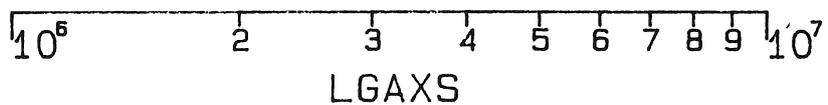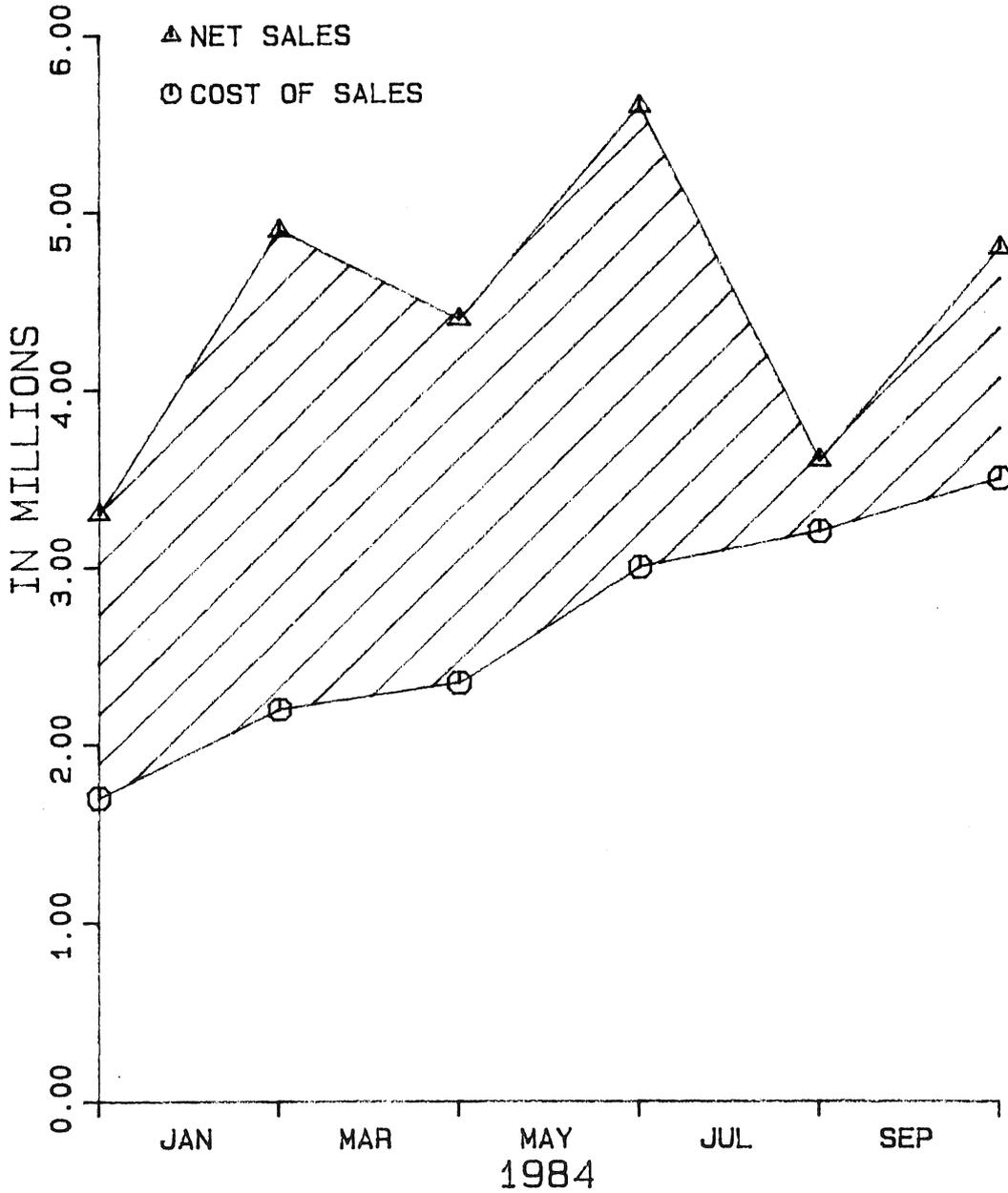
SHADED REGION REPRESENTS GROSS PROFIT

Plate 4-8

CALL SHADE (XARAY1, YARAY1, XARAY2, YARAY2, DLIN,
ANGLE, NPTS1, INC1, NPT

NPTS1=30
NPTS2=2
DLIN=3.
ANGLE=-30.
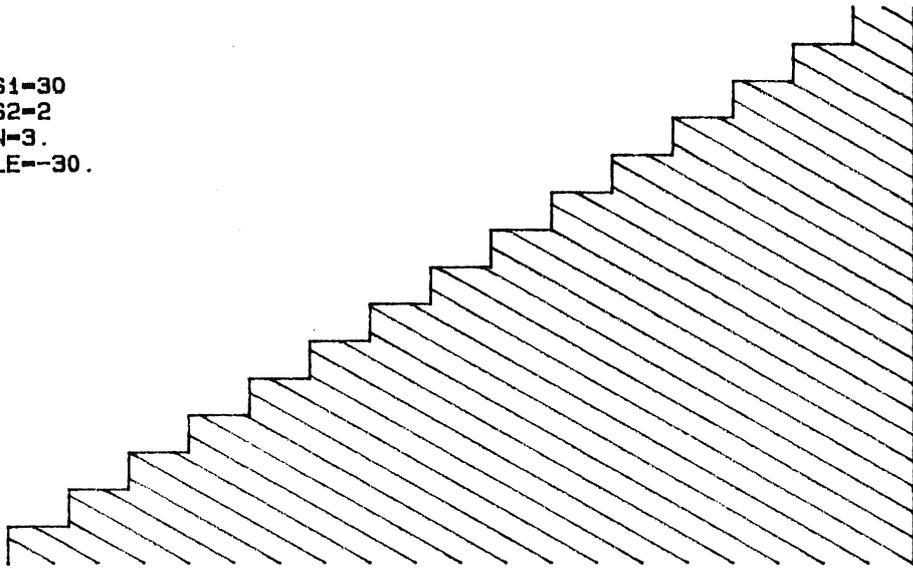
Y1 (X) =SIN (X)
Y2 (X) =SIN (X/2)
X=0. TO 720.
ANGLE=0.
DLIN=2.

Plate 4-9
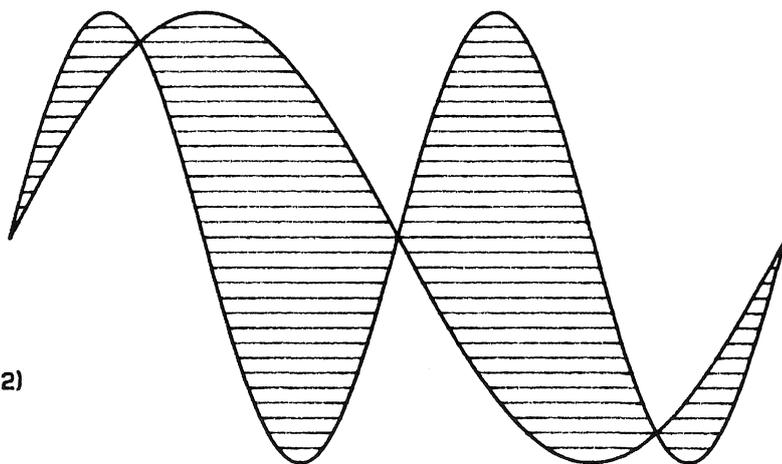
CALL CRVPT (XARRAY, YARRAY, INTEQ, NPTS,
            INC, SH, SW, IBCDT, NCHAR, IBCDX,
            NCHARX, IBCDY, NCHARY, INT)

CRVPT TITLE
  Z=X-79.688

Plate 4-10
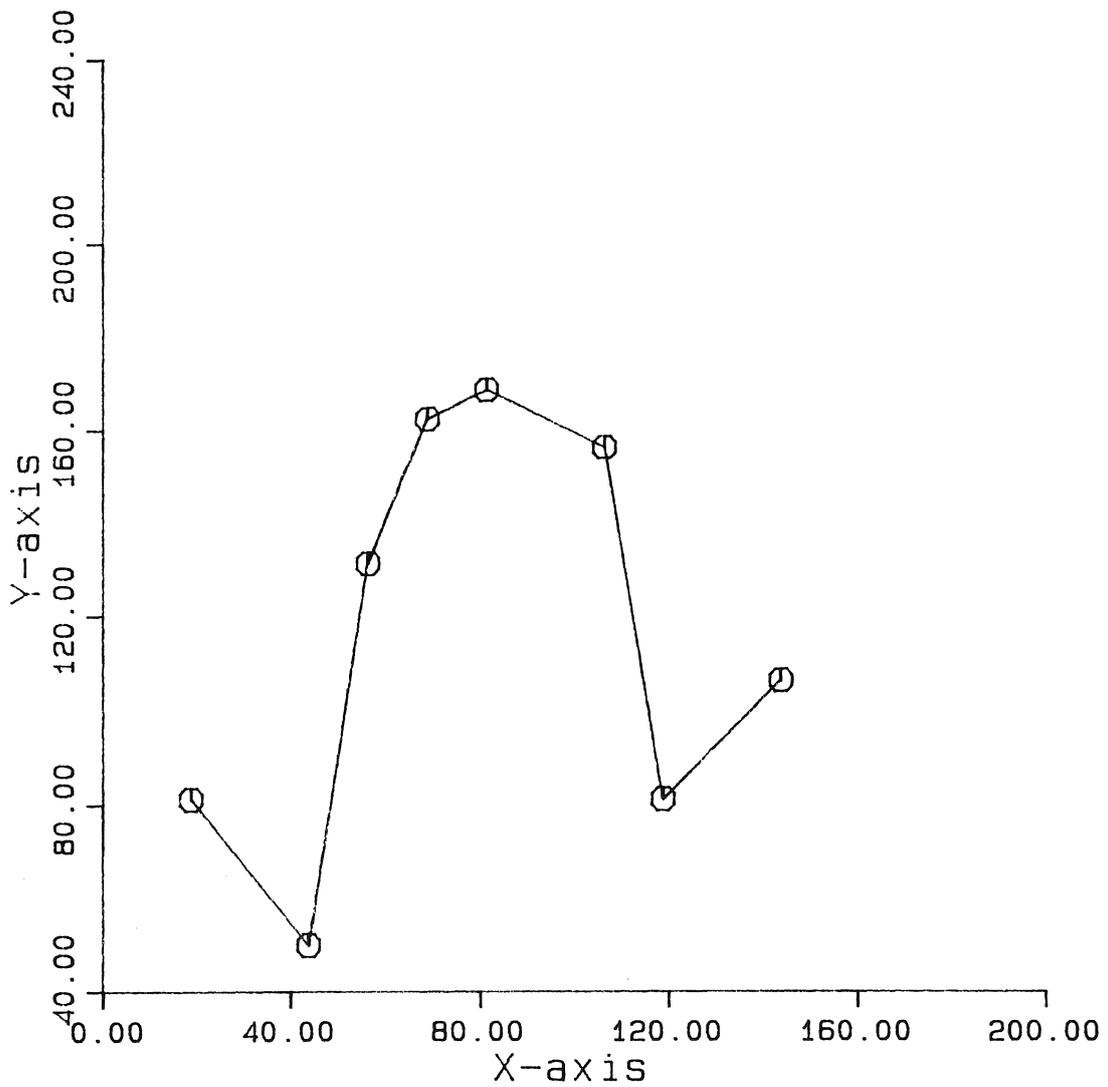
CALL AROHD (XPAGE, YPAGE, XTIP, YTIP, AHLEN, AHWID, ICODE)

AHWID=5.0
AHLEN=3.5

ICODE-11   ICODE-12   ICODE-13   ICODE-14   ICODE-15   ICODE-16   ICODE-17
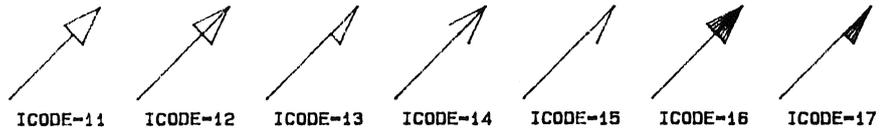
YTIP =
YPAGE = Y
AHLEN = 5.0
AHWID = 5.0
ICODE = 22

           Y+15.0   Y+7.0   Y-15.0

AHLEN =
AHWID = 8.0

           10.0   5.0   00.0

AHWID =
AHLEN = 8.0

           00.0   8.0   15.0
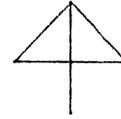
ICODE =
XPAGE = X
YPAGE = Y - 40.
XTIP = X
YTIP = Y
AHLEN = 15.0
AHWID = 0.

           -12   -1   -6

✳ CURRENT PEN POSITION

Plate 4-11

CALL CNTRL (XARRAY, YARRAY, NPTS, INC)

PLOTTED POINTS ARE SIGNIFIED BY THE SYMBOL+

NPTS=7
INC=1
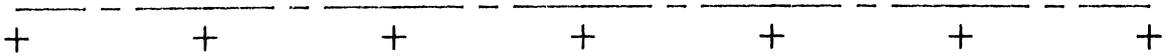
NPTS=4
INC=2

NPTS=3
INC=3
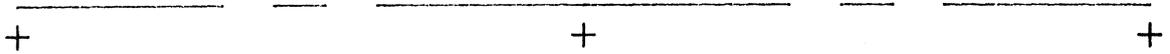
(IRREGULAR DATA POINTS)

Plate 4-12
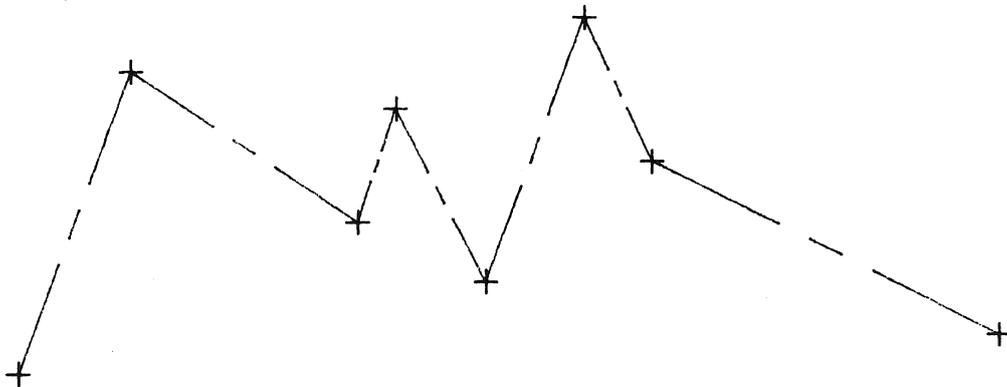
CALL DIMEN (XPAGE, YPAGE, DIME, ANGLE, SCALER)

ANGLE=270.
SCALER=1.
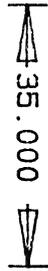
DIME=35.   DIME=21.   DIME=15.
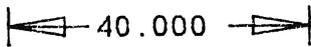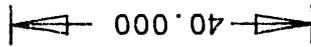
ANGLE=0.   ANGLE=180.   ANGLE=-135.
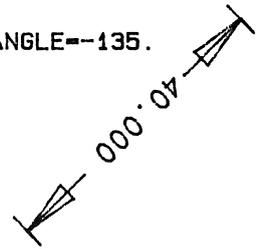
DIME=1.
ANGLE=90.

SCALER=2.   SCALER=1.   SCALER=.5

Plate 4-13

CALL CIRCL (XPAGE, YPAGE, THO, THF, RO, RF, DI)

THO=0.   THO=90.   THO=180.
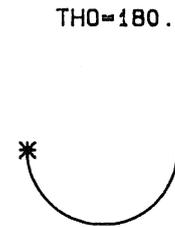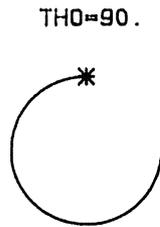
THF=360.
RO=10.
RF=10.
DI=0.

THF=360.   THO=-360.
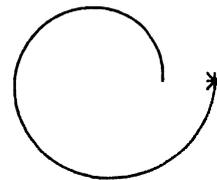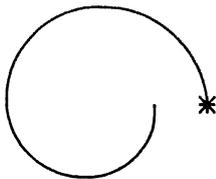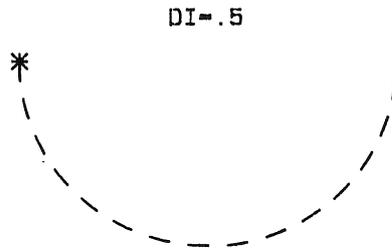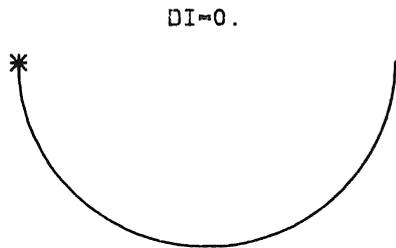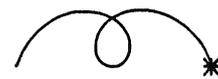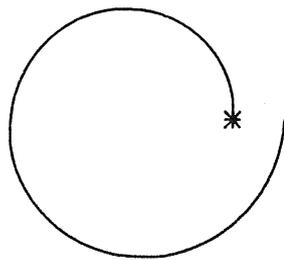
THO=0.
RO=15.
RF=8.
DI=0.

RF=20.   RF=0.   RF=-13.

THO=0.
THF=360.
RO=13.
DI=0.

DI=0.   DI=.5

XPAGE, YPAGE INDICATED BY SYMBOL ✳

Plate 4-14

CALL DASHP (XPAGE, YPAGE, DASH)

XPAGE, YPAGE INDICATED BY SYMBOL X
ORIGINAL PEN POSITION BY SYMBOL I

DASH=80   X————————————————————   I

DASH=25   X———————   ———————   I

DASH=10   X—— —— —— —— ——   I

DASH=5   X— — — — — — — — —   I

DASH=4   X— — — — — — — — — — —   I

DASH=3   X— — — — — — — — — — — —   I

DASH=2   X– – – – – – – – – – – –   I

DASH=1   X-------------------------------   I

CALL DASHL (XARAY, YARAY, NPTS, INC)

NPTS=8.
INC=1

Plate 4-15

CALL ELIPS (XPAGE, YPAGE, RMAJ, RMIN, ANGLE, THO, THF, IPEN)

RMAJ=15. RMAJ=10. RMAJ=5.

RMIN=10.
ANGLE=0.
THO=0.
THF=360.
IPEN=3

RMAJ=10.
RMIN=5.
THO=0.
THF=360.
IPEN=3

ANGLE=90. ANGLE=135. ANGLE=-135.

RMAJ=4.
RMIN=12.
ANGLE=0.
THF=360.
IPEN=3

THO=30. THO=120. THO=200.

RMAJ=10.
RMIN=8.
ANGLE=90.
THO=30.
IPEN=3

THF=210. THF=300. THF=-240.

XPAGE, YPAGE INDICATED BY SYMBOL ✳

Plate 4-16
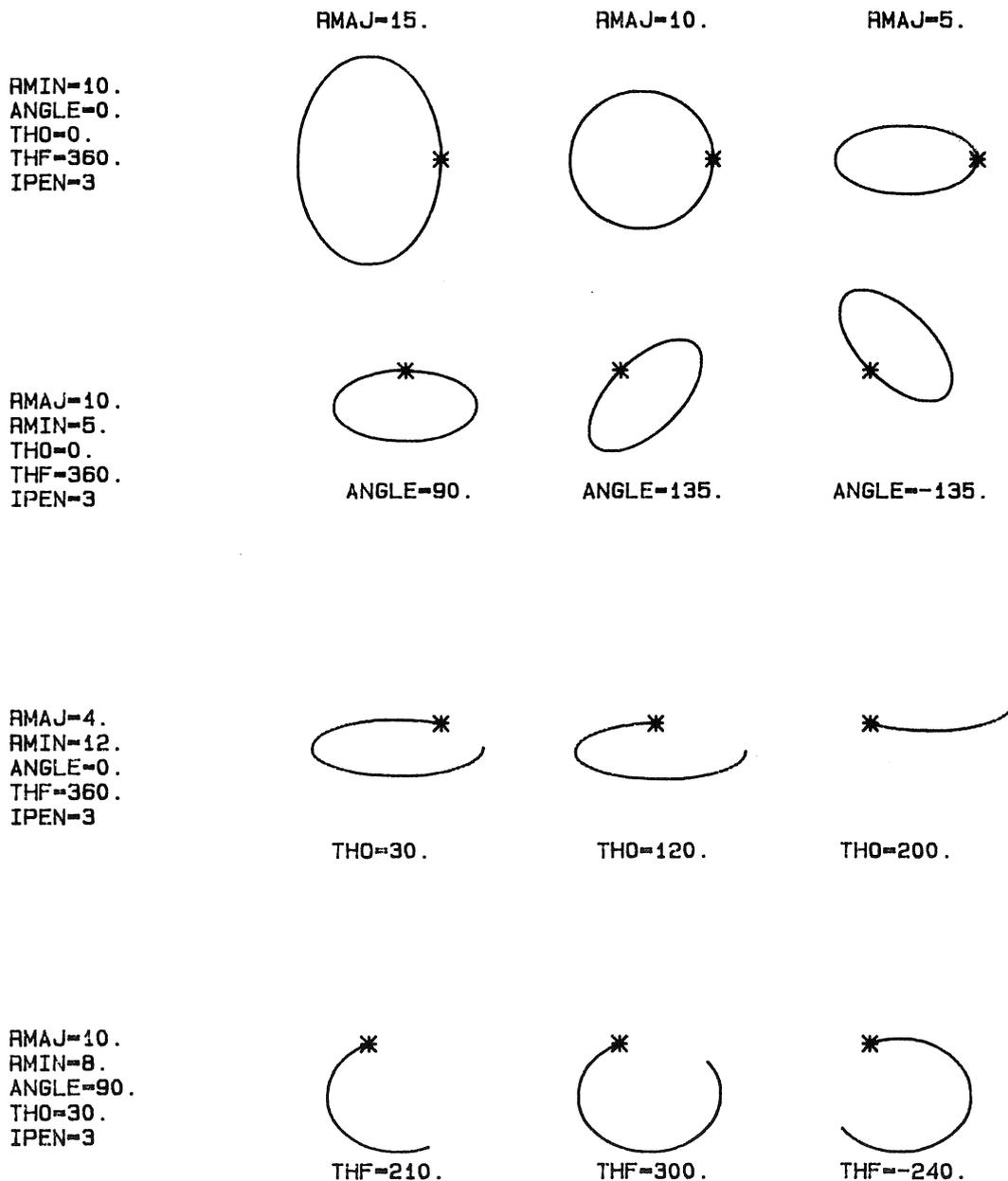
CALL FIT (XPAGE1, YPAGE1, XPAGE2, YPAGE2,
XPAGE3, YPAGE3)
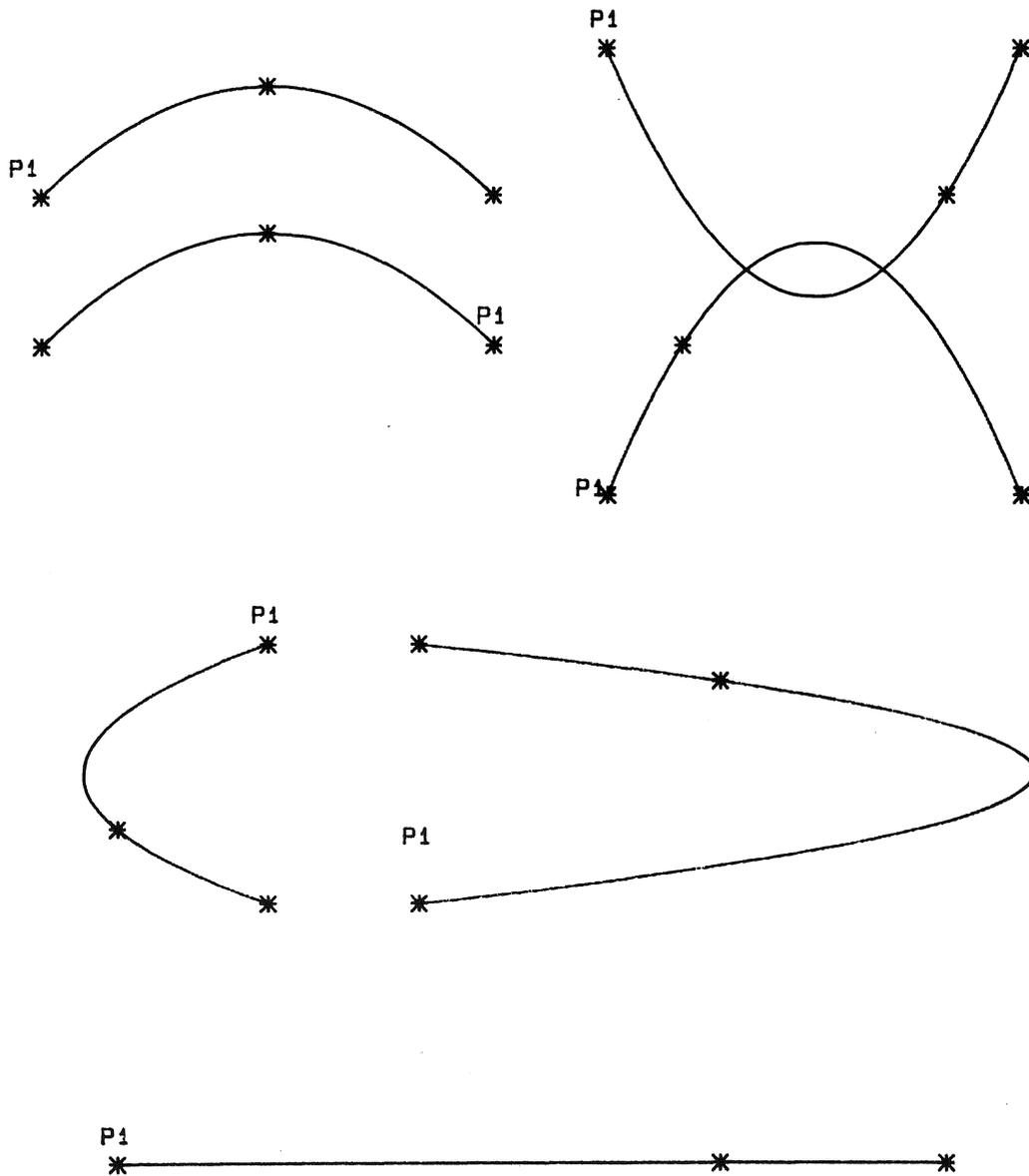
DATA POINTS INDICATED BY SYMBOL ✳

Plate 4-17

CALL GRID (XPAGE, YPAGE, DELTAX, DELTAY, NXSP, NYSP)

DELTAX=4.
DELTAY=4.
NXSP=8
NYSP=8
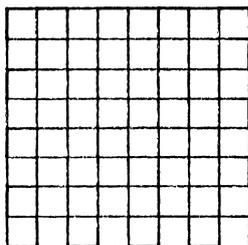
DELTAX=12.
DELTAY=8.
NXSP=7
NYSP=1
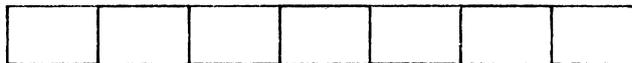
DELTAX=6.
DELTAY=6.
NXSP=12
NXSP=8
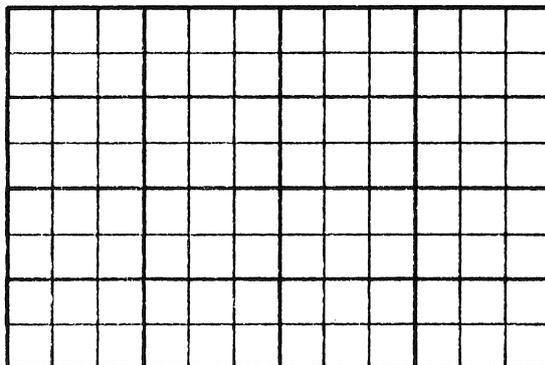
DELTAX=18.
DELTAX=12.
NXSP=4
NYSP=4

TWO GRIDS OVERLAYED, WITH XPAGE,YPAGE OFFSET BY 0.2 MM

Plate 4-18

CALL POLY (XPAGE, YPAGE, SLEN, SN, ANGLE)

SLEN=10.   SLEN=15.   SLEN=20.

SN=3.
ANGLE=0.
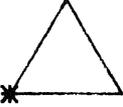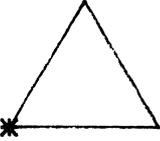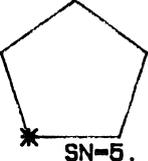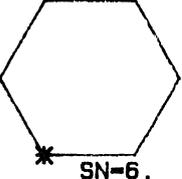
SLEN=12.
ANGLE=0.

SN=4.   SN=5.   SN=6.

SLEN=7.
ANGLE=0.
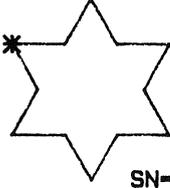
SN=-4.   SN=-5.   SN=-6.

SN=3.
SLEN=12.
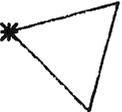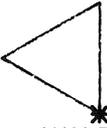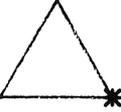
ANGLE=-45.   ANGLE=90.   ANGLE=120.

XPAGE, YPAGE INDICATED BY SYMBOL ✳

Plate 4-19

CALL CURVX (XO, XF, COEFF1, EXP1, COEFF1, EXP2,
COEFF3, EXP3, COEFF4, EXP4)

XO=1.
XF=100.

+
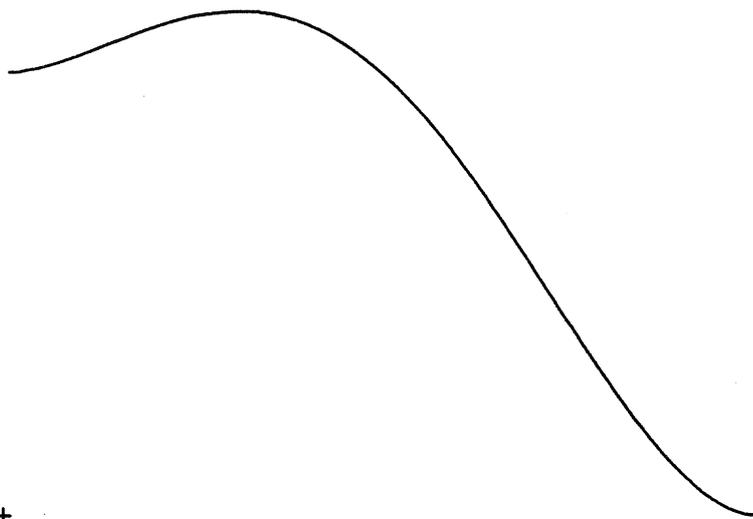
XO=21.
XF=120.

COEFF1=.0000048
EXP1=4.0
COEFF2=-.00084
EXP2=3.0
COEFF3=.03
EXP3=2.0
COEFF4=60.
EXP4=0.0

+
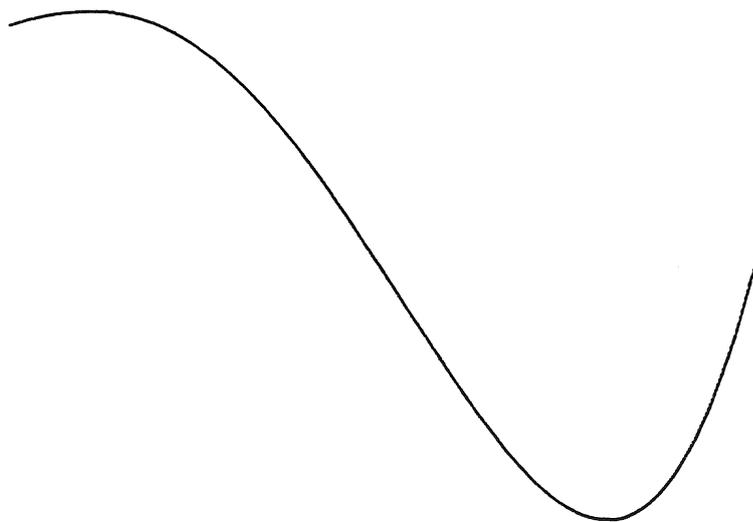
ORIGIN MARKED BY SYMBOL +

Plate 4-20

SAMPLE OF SCIENTIFIC TYPE SUBROUTINES
USING CURVX SUBROUTINE

$Y=0.075X**4-0.525X**3+0.75X**2+2.40$

USING CURVY SUBROUTINE

$X=Y**3.78-6Y**2.52+9Y**1.26$

Plate 4-21
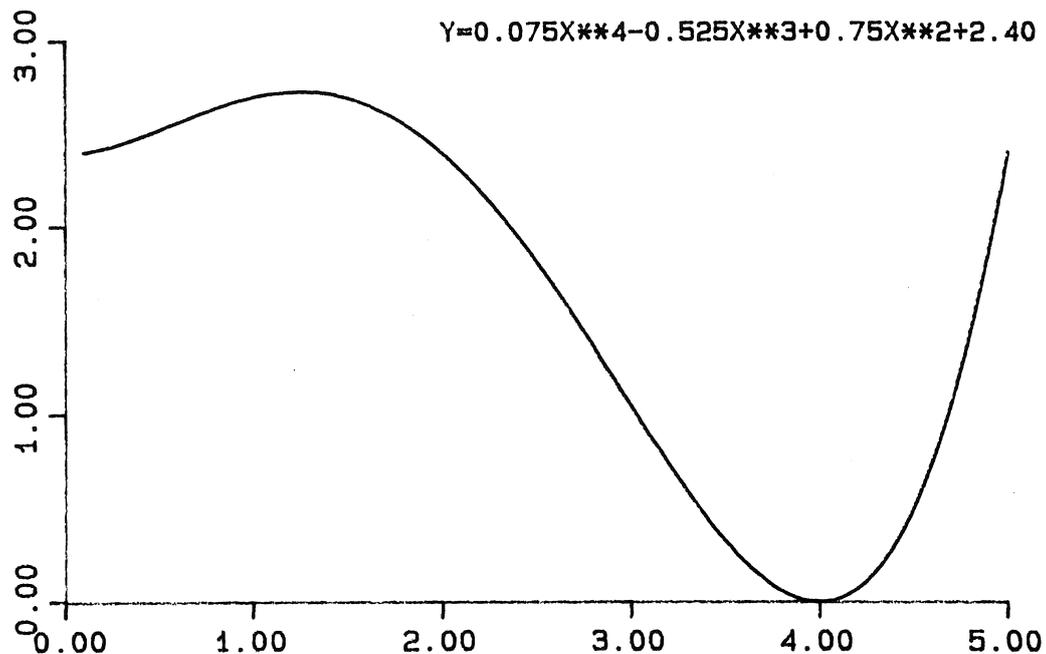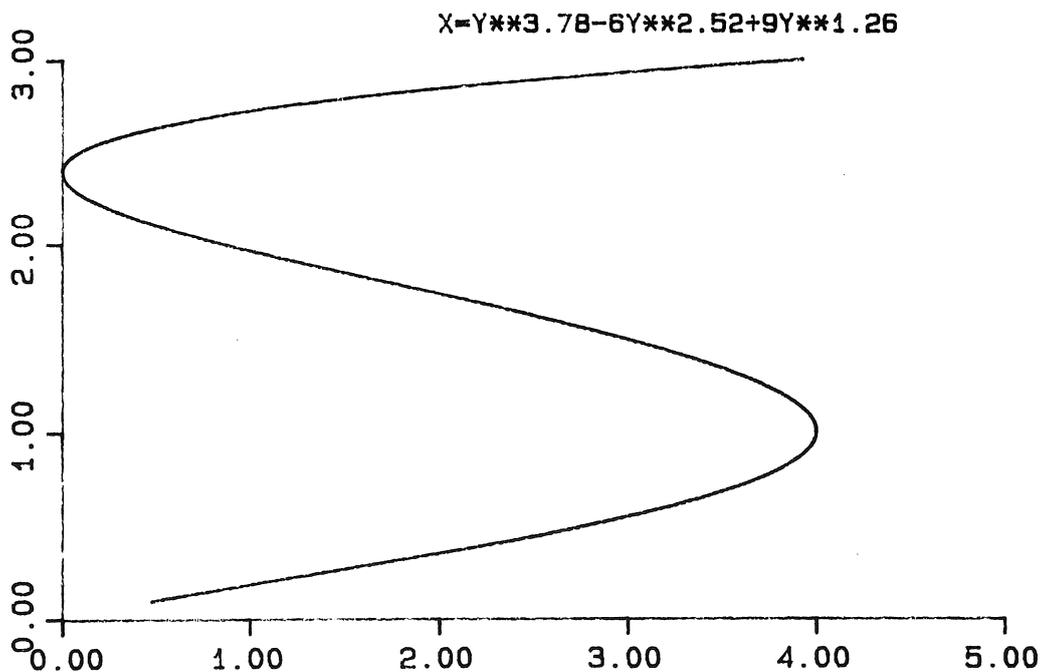
CALL FLINE (XARRAY, YARRAY, NPTS, INC,
LINTYP, INTEQ)

NPTS=8
INC=1
LINTYP=0
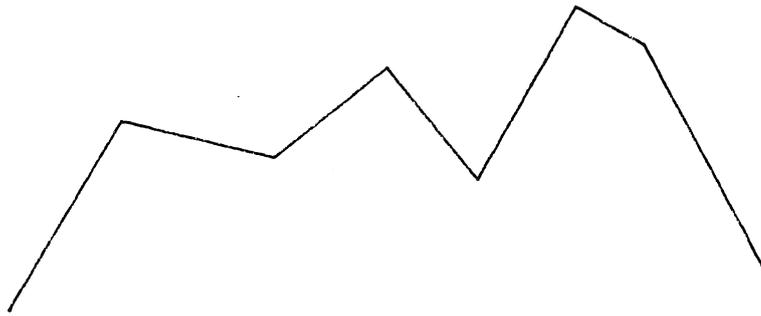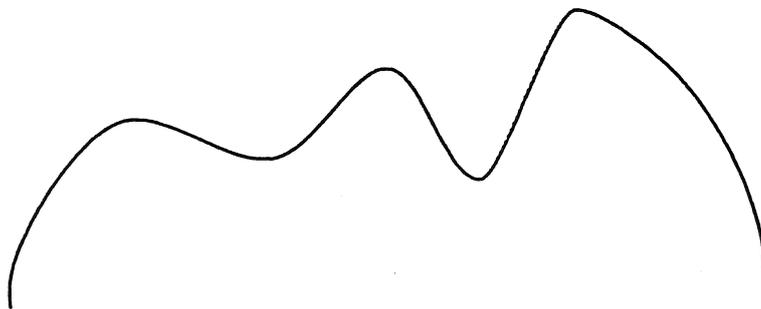
NPTS=-8
INC=1
LINTYP=0

NPTS=-8
INC=1
LINTYP=1
INTEQ=2

Plate 4-22
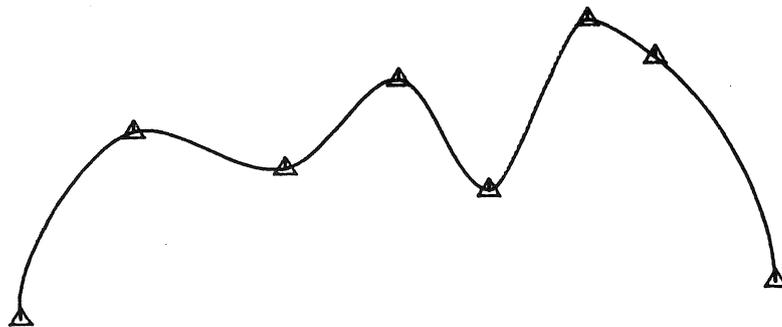
USING SCALG, LGAXS, AND LGLIN SUBROUTINES

TEMPERATURE vs ALTITUDE
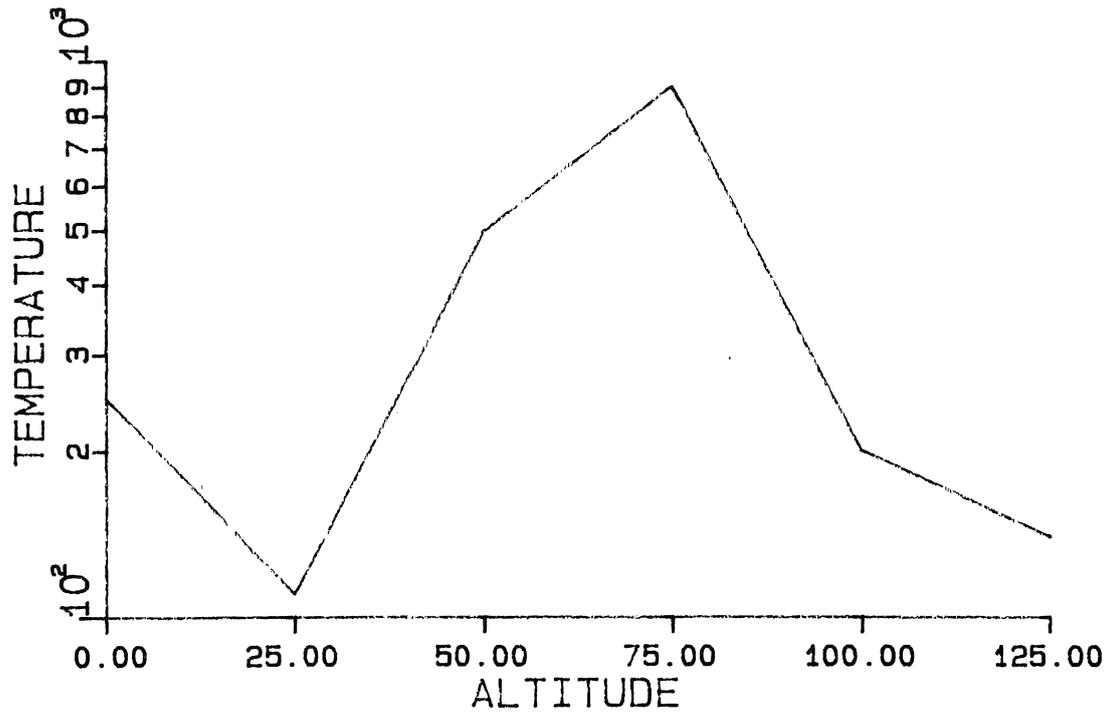
USING POLAR SUBROUTINE

RADIUS=2*(1-COS(ANGLE))
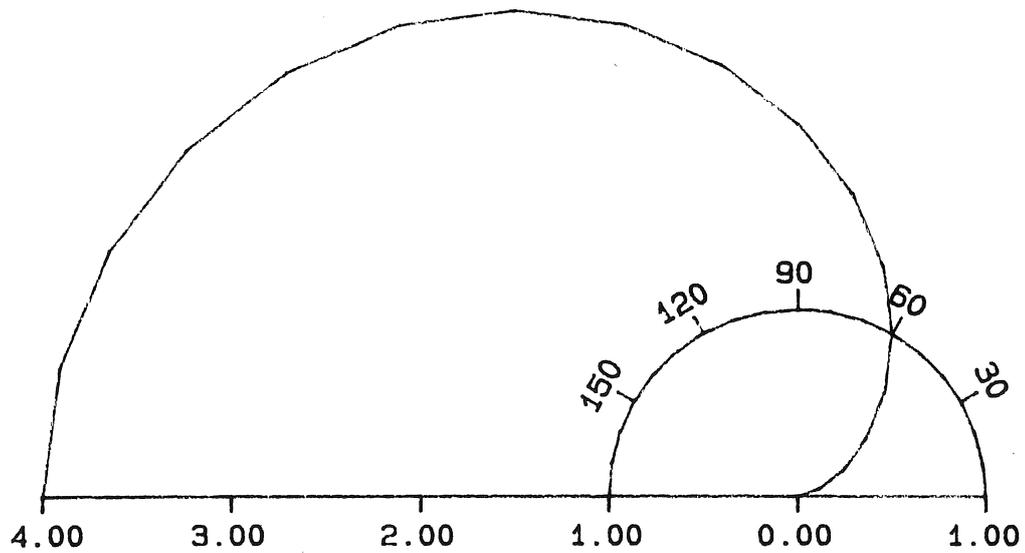
Plate 4-23

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

VARYING NCHAR, ANGLE, FIRSTV AND DELTAV

IBCD='LBAXS'
AXLEN=100.
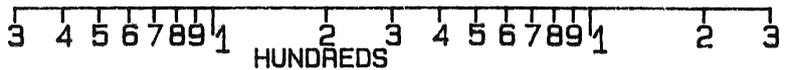NCHAR=-5

ANGLE=45.
FIRSTV=100.
DELTAV=.02
NCHAR=5
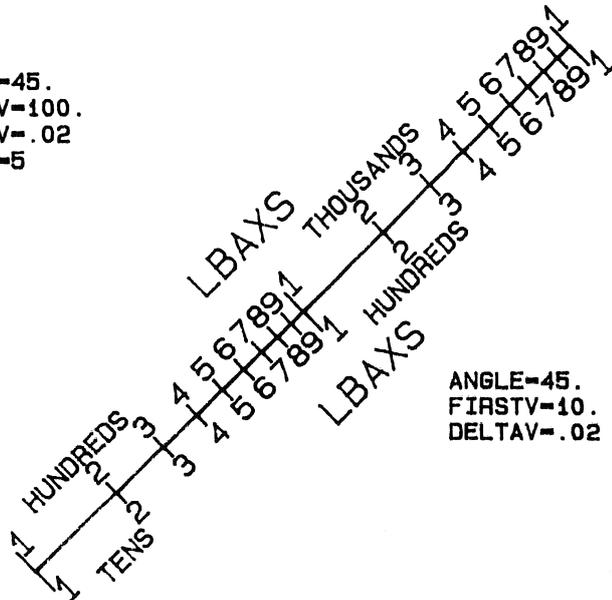
ANGLE=45.
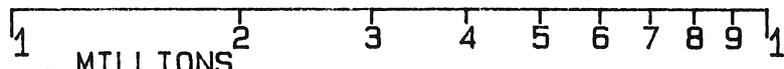FIRSTV=10.
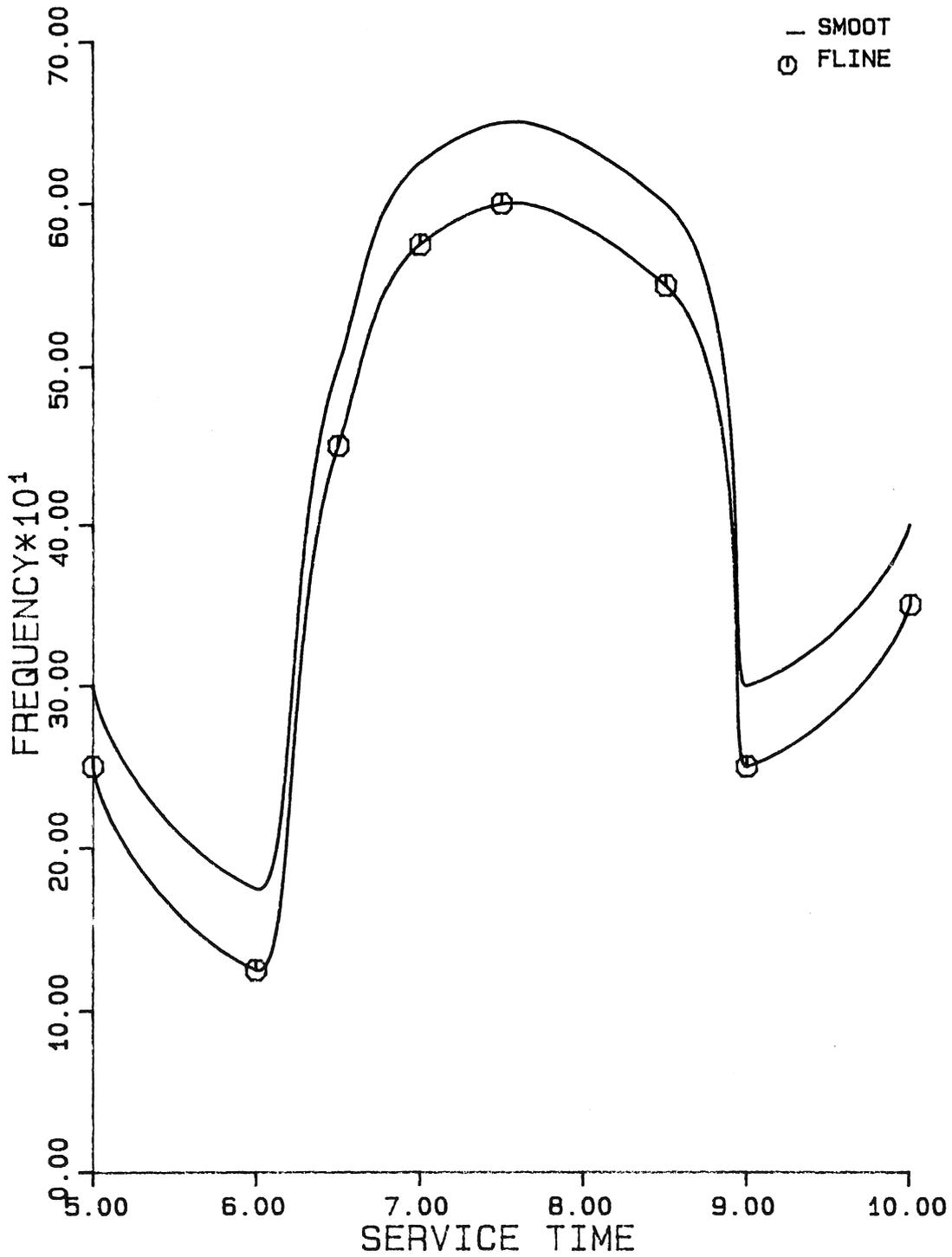DELTAV=.02

ANGLE=0.0
FIRSTV=.0001
DELTAV=0.03

ᵢ1 TEN THOUSANDTHS   ᵢ1 THOUSANDTHS   ᵢ1 HUNDREDTHS   1ᵢ

LBAXS

FIRSTV=30.
DELTAV=.02

3 4 5 6 789ₗ1   2   3 4 5 6 789ₗ1   2   3
              HUNDREDS

LBAXS

FIRSTV=1000000.
DELTAV=.01

ₗ1   MILLIONS 2   3   4   5   6   7   8   9 ₗ1

LBAXS

Plate 4-24

USING FLINE AND SMOOT SUBROUTINES

SMOOT
FLINE

FREQUENCY*10^1

SERVICE TIME

Plate 4-25

CHAPTER 5

PLOTTING CONTOURS AND SURFACES

5.1 INTRODUCTION

This chapter describes two subroutines within the CALF77 library
which may be used to plot a function Z of two independent variables
X and Y. Like the other subroutines in the graphics library (see
chapters 3 and 4), the CONTOR and SURFAS routines are callable from
FORTRAN-77 programs. The procedures for compiling and linking
graphics applications programs with the routines in the CALF77
library are described in Chapter 2.

5.2 CONTOR SUBROUTINE

CONTOR plots the regularly spaced equipotential (contour) lines of
a function Z of two independent variable X and Y, given its values
at regularly spaced data points. The user must supply a
two-dimensional array which contains at position (I,J) the value of
the function Z at (X(I),Y(J)); where both X(I+1)-X(I) and
Y(J+1)-Y(J) are constant for all I' and J. CONTOR scans this array,
plotting the required contour lines.

The calling sequence is:

  CALL CONTOR(A,M,N,CL0,DELTA,CL1,IPEN,HT,RATIO,THETA,INUM,IBOX)

A   is the two-dimensional array to be contoured.

M   is the highest subscript in the X-direction.

N   is the highest subscript in the Y-direction.

CL0   is the lowest contour level to be plotted.

DELTA is the vertical increment value between contour lines.

CL1   highest contour level (highest function value) to be plotted.

IPEN   is 1 for a plot with a continuous line contour and 2 for a dashed line contour. These values should be negated if superposition on a previous plot is desired.

HT   height of the contour plot (in millimetres).

RATIO ratio of the lengths of one X-unit to one Y-unit on the desired plot.

THETA angle (in degrees) between the X and Y axes (10<THETA<170).

INUM   if non-zero, each contour line will be labelled with its numerical value.

IBOX   is positive for a box around the plot, negative for two axes marking the subscripts of the array and zero for no boundary around the contour plot.

NOTES

Since CONTOR makes use of a common area named "CONTUR", the calling program should not use a common area of this name.

The array to be contoured must be dimensioned at exactly (M,N) in the calling program. If contouring two arrays with different dimensions, two different array names must be specified. There can be no unused points in the array.

CONTOR assumes the user has opened a plot file and draws the contour map with its lower left corner at the existing origin (NOT at the current pen position). The user should ensure the origin is at least 5mm away from the page boundaries, to allow for CONTOR's labels.

When calling CONTOR twice in succession, it is necessary to reset the origin between calls in order to prevent superposition. Merely specifying a negative IPEN value is not sufficient, since CONTOR leaves the location of the origin unchanged, regardless of the pen position.

```
C       Program which uses the CONTOR subroutine to plot
C       the equipotential curves of a function of 2 variables
C
        DIMENSION A(13,13),B(7,5)
        EQUIVALENCE(A(1,1),B(1,1))        !Save space
C
C       Calculate the data points, 1 every 30 degrees
C       f(x,y) = sin(x) + sin(y), 0<x<360, 0<y<360
C
        DO 360 I=1,13
        DO 360 J=1,13
360     A(I,J) = SIND(FLOAT(30*(I-1)+1)) +
        1         SIND(FLOAT(30*(J-1)+1))
C
C       Initialize the plotter, set an origin in top of page
C
        CALL PLOTIN('CONTOR','PLT4/FORMS:A4HEAVY/ROTATE')
        CALL PLOT(15.0,150.0,-3)          !Allow for labels
C
C       Call the contouring routine:
C               array dimensions are (13,13)
C               plot will 100 mm high, x-axis scaling 1.5
C               axes will be perpendicular
C               lines will be labelled
C               a box will surround the plot.
C
        CALL CONTOR(A,13,13,-2.0,0.25,2.0,1,100.,1.5,90.0,1,1)
C
C       On the bottom of the page, plot another contour map
C       with data points generated at random.
C
        ISEED = -27409163                 !For VAX-Fortran only
        DO 820 I=1,7
        DO 820 J=1,5
        B(I,J)=10.*RAN(ISEED)
820     CONTINUE
        CALL PLOT(0.,-125.,-3)            !Move the origin
C
C       Contour the 7x5 random array, plotting levels from 0 to 10
C       100 mm high, with each X-unit the same size as each Y-unit.
C
        CALL CONTOR(B,7,5,0.0,1.0,10.0,1,100.0,1.0,90.0,1,1)
C
        CALL PLOT(0.,0.,999)              !Close the plot file
        END
```

See Plate 5-1 for the output from this program.

5.3 SURFAS SUBROUTINE

The SURFAS subroutine displays a function Z of two independent variables X and Y (a surface) in "three dimensions". The graph is derived from the data representing the surface, and is portrayed using the projection of that surface onto a plane. This plane is placed at a specific distance from an "observer" located at a given point in space.

Lines which are hidden from the observer are not plotted and those which are partially obscured are "clipped" at some point along their length. The routines which plot the three-dimensional surfaces perform a binary search to determine where each line disappears from view and the accuracy of that search can be determined by the user, in the argument IACC. This argument is an integer representing to what accuracy the user wishes to eliminate hidden lines. For example, IACC=8 would eliminate hidden lines with the accuracy of 1/8 of their length.

The calling sequence is:

        CALL SURFAS(X,Y,Z,MPTS,NPTS,XSIZE,YSIZE,OBS,IACC)

X   is the one-dimensional array containing the different
             values which X can take on in the domain.

Y   is an array containing the values Y can take on in the
             domain.

Z   is the two-dimensional array containing the function
             values. The element Z(I,J) is the value of the function
             at the point (X(I),Y(J)).

MPTS   the maximum subscript of X, so that X is dimensioned at
             X(MPTS).

NPTS   the maximum subscript of Y, so that Y is dimensioned at
             Y(NPTS). Z would then have to be dimensioned at
             Z(MPTS,NPTS).

XSIZE   the X-dimension of the drawn surface. The annotation
             will begin at a distance of 50mm*(current factor) to the
             right of the drawn surface.

YSIZE   the Y-dimension of the drawn surface.

OBS   a 4-word array containing the desired position of the
             observer.
             If OBS(1)=0.0, "SURFAS" will provide a set of observation

5-4

coordinates dependant upon the range of the surface by
default.
If OBS(1) is non-zero, then the user must specify their
own observation coordinates by setting:

> OBS(2) = Observer's X coordinate.
> OBS(3) = Observer's Y coordinate.
> OBS(4) = Observer's Z coordinate.

IACC   the degree of accuracy of hidden line elimination, as
              described above.

NOTES:

The observation coordinates calculated by default will always place
the observer to the positive side of the surface in all three
directions, looking down at the figure from the right. The
positive X-axis will be coming out of the paper, the positive
Y-axis will be approximately horizontal and to the right, and the
positive Z-axis will be vertically upwards, as in the traditional
representation of three perpendicular axes in space.

SURFAS scales the plot with all three directions being given the
same scale factor. It is therefore important that the ranges of
the three dimensions cover approximately the same number of units,
or else the change in one dimension may be so small as to be
imperceptible. This problem can be avoided by multiplying the
function by an appropriate scale factor, as in the example which
follows. The scaling is chosen to enable the surface drawn to fit
within a box of the size specified by the user. SURFAS
automatically plots a summary to the right of the graph, which
includes the observation coordinates and angles with respect to the
coordinate axes and the image distance.

```
C       Program which uses the SURFAS subroutine to plot
C       a "3D" representation of a function of 2 variables.
C
C       Produces MNT-11 Plate 5-2
C
        DIMENSION X(33),Y(33),Z(33,33),OBS(4)
C
C       Calculate all the data points
C       let -8.0 < x < 8.0 in increments of 0.5
C       let -8.0 < y < 8.0 in increments of 0.5
C       f(x,y) = 8.0 * sin r / r, where r=sqrt(x**2+y**2)
C
        DO 100 I=1,33
        X(I) = FLOAT(I-17) / 2.0
        Y(I) = X(I)
100     CONTINUE
        DO 200 J=1,33
        DO 200 I=1,33
```

```
        R = SQRT(X(I)**2+Y(J)**2)
        Z(I,J) = 8.0
        IF (R .NE. 0) Z(I,J) = 8.0 * SIN(R) / R
200     CONTINUE
        Z(17,17) = 8.0                      !For continuity
C
C       call the subroutine "SURFAS" to do the plotting
C       make surface drawn be 200mm * 250mm
C       let it choose its own observation coordinates (OBS(1)=0.0)
C       eliminate hidden lines accurate to 1/8 of a grid
C
        CALL PLOTIN('SURFAS','PLT4/FORMS:A3NORMAL')
        CALL PLOT(10.,10.,-3)
        CALL SURFAS(X,Y,Z,33,33,200.0,250.0,OBS,8)
        END
```

Plate 5-2 shows the output from this program.

Plate 5-1
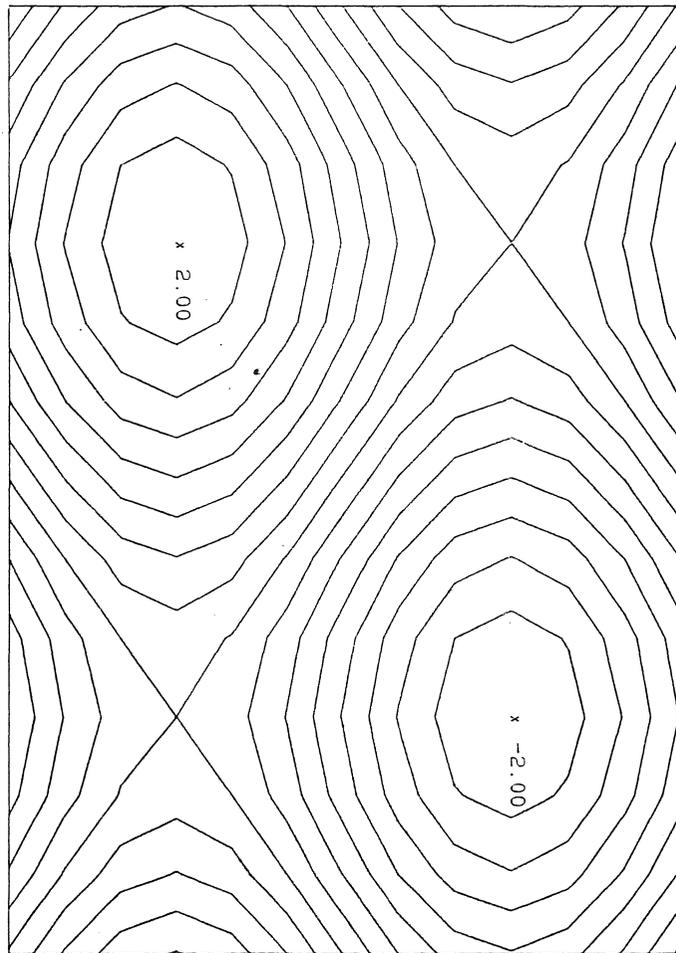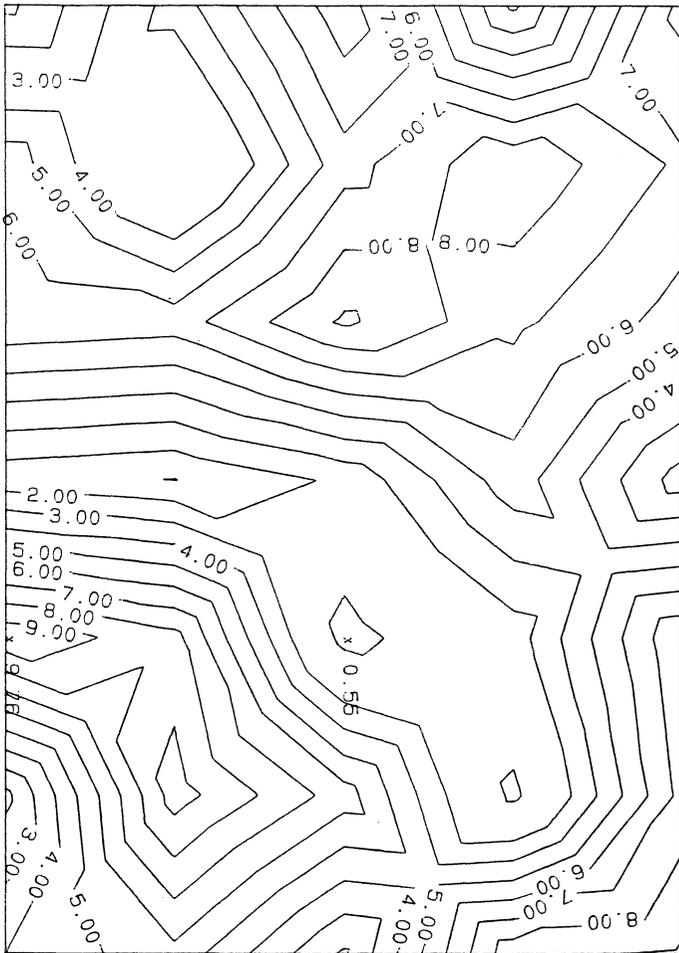
OBSERVER'S
COORDINATES
X = 1600.00
Y = 800.00
Z = 490.02

OBSERVATION
ANGLES
= 149.66
= 115.56
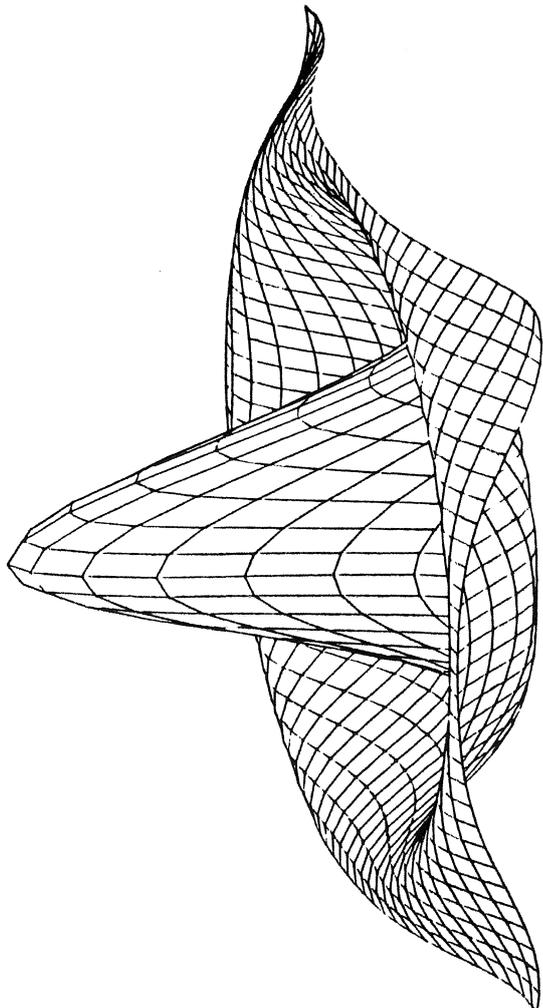= 105.23

IMAGE
DISTANCE
D = 1853.93

Plate 5-2

APPENDIX A

DETAILS OF THE PLOTIN SUBROUTINE

A.1 Introduction

The PLOTIN subroutine, used to initialize the plotter and plot file before commencement of a plot, is described in Section 2.2.1. In this section we give the details of the subroutine, regarding plotter types, stationery and plot control.

As shown in Section 2.2.1, the PLOTIN calling convention is:

                    CALL PLOTIN(FILE,DEVICE)

FILE   is a CHARACTER variable containing the name to be used
          for the output plot file, if there is one.
DEVICE   specifies the output device and other characteristics of
          the plot.

The DEVICE field is a CHARACTER variable, which is of the following format:

                    "device/switch/switch..."

The switches which may be appended to the device name are many and may not all be used with every plotting device.

A.2   Plotting Devices

The device specified in the PLOTIN call may, at present, be one of
the following:

```
PLT3     - HP7585 plotter, at the Prentice Computer Centre
PLT4     - HP7475 plotter, at the Prentice Computer Centre
PLT5     - HP7475 plotter, at Griffith University
HP       - Either one of the HP plotters
HP7585   - HP7585 plotter, possibly in series with your terminal
HP7475   - HP7475 plotter, possibly in series with your terminal
TEK4010  - Tektronix 4010 terminal, or lookalike
TEK4014  - Tektronix 4014 terminal
TEK4015  - Tektronix 4015 terminal
TEK4105  - Tektronix 4105 terminal
TEK4107  - Tektronix 4107 terminal
TEK4662  - Tektronix 4662 plotter, in series with your terminal
GIGI     - DEC GIGI terminal, or other ReGIS device
```

(Tektronix equipment is not currently supported by CALF77 on UQVM.)

This list may, of course, be augmented with more devices, as they
become available for use from the Prentice Computer Centre.

A.3   General-Purpose Switches

A.3.1   '/BOUNDS' switch

This switch indicates that bounds checking is to be performed and
any out-of-bounds condition will result in a fatal error. (A
"fatal" error is one which results in the running program being
halted).

A.3.2   '/XYMIN:n,m' switch

This switch sets the minimum X and Y bounds for the windowing
region.   The values may be real or integer and must be within the
range (0.0,9999.9). The default values correspond to the minimum
bounds of the hard-clip region of the plotter being used.

A.3.3 '/XYMAX:n,m' switch

This switch is similar to the XYMIN switch (above), but specifies
the maximum X and Y values for the windowing region. The values
given for this switch must each be greater than the corresponding
values given for the XYMIN switch.

A.3.4 '/TERMINAL' switch

This switch indicates that the plotting device is connected in
series with your terminal, and on which the plot output will appear
as it is produced by your plotting application.

A.3.5 '/HEADER' switch

This switch specifies that you require an identifying name on the
plotted page. This is the default condition, which can be negated
with the '/NOHEADER' switch.

A.4 Switches for Hewlett-Packard (HP) Devices

The devices PLT3, PLT4, PLT5, HP7475 and HP7585 each specify a
particular type of Hewlett-Packard plotter. The device 'HP' refers
to either type of plotter, with the particular model being chosen
by the software, depending on the values of certain switches, such
as '/NPENS:', '/FORMS:' and '/PLOTTER:'.

A.4.1 '/FORMS:xxx' switch

This switch forces a particular paper size and type to be used in
the plotter. The first two letters of the specification give the
required paper size, such as 'A3', while the rest give the paper
type, which may be either 'NORMAL', 'HEAVY', 'TRACING' or
'DRAFTING'. ('DRAFTING' is the same as 'TRACING'.)
'A4TRANSPARENCY' is also available. Other stationery may be
available from time to time. You should check with the Prentice
Computer Centre before assuming the availability of any paper size
and type. Up-to-date information on currently available stationery
can be obtained on UQKL10 by typing .HELP PCOST, or on UQVAX by
examining the file SYS$ACCOUNTS:PLOTCOST.DAT.

A-3

International Paper Sizes, A Series

	Y	X
A0	841 mm	1189 mm
A1	594 mm	841 mm
A2	420 mm	594 mm
A3	297 mm	420 mm
A4	210 mm	297 mm

NOTE

These are the actual paper sizes.
The plotting surface is somewhat
smaller. It is advisable to allow
10-15 mm from each edge.

A.4.2 '/CAROUSEL:xxx' switch

This switch is used to specify the required pen set for the HP
plotter. The available carousels are 'ROLLER', 'FIBRE' and
'DRAFTING'. The default carousel for the HP7585 plotter is
'ROLLER' and the default carousel for the HP7475 plotters is
'FIBRE'. The default pen setups are as follows:

    Roller Carousel
       Pen 1 - Black
       Pen 2 - Red
       Pen 3 - Blue
       Pen 4 - Green
       Pen 5 - Black
       Pen 6 - Red
       Pen 7 - Blue
       Pen 8 - Green

    (Roller pens are available in 0.3mm only)

    Fibre Carousel
       Pen 1 - Black, 0.3mm
       Pen 2 - Red, 0.3mm
       Pen 3 - Blue, 0.3mm
       Pen 4 - Green, 0.3mm
       Pen 5 - Black, 0.7mm
       Pen 6 - Red, 0.7mm
       Pen 7 - Blue, 0.7mm
       Pen 8 - Green, 0.7mm

Drafting Carousel
  Pen 1 - Black, 0.35mm
  Pen 2 - Black, 0.5mm
  Pen 3 - Black, 0.7mm
  Pen 4 - Black, 0.25mm

Other fibre pens are available, including orange, lime, gold, turquoise, violet and brown in 0.3mm, and blue and green in 0.7mm.

Overhead transparency pens are also available in colours black, red, blue and green, in size 0.6mm.

Drafting pens are also available in sizes 0.4mm, 0.6mm and 0.8mm.

A.4.3 '/NPENS:n' switch

This switch lets the software know how many pens are required for your plot. The number specified must, of course, be smaller than that available in the selected (or default) carousel.

A.4.4 '/PLOTTER:xxx' switch

This switch is used with the 'HP' device, to indicate that only the specified device is to be used.

A.4.5 '/ROTATE' switch

This switch specifies a rotation of the plotter's coordinate system through 90 degrees. 'NOROTATE' specifies a zero rotation, which is the default condition.

A.5 Tektronix Devices

A.5.1 '/BAUD:n' switch

This switch specifies the baud rate (speed) at which your terminal is running.

Note that the '/TERMINAL' switch is implied when a TEKxxxx device is selected.

A.6 GIGI terminals

A.6.1 '/TERMINAL' switch

This switch must be used, if you desire the plot output to be displayed directly on the terminal. If this switch is not given, a plot file containing ReGIS graphics commands will be produced. Such a file would be suitable for subsequent processing by the GIGI Graphics Editor or the Slide Projection System.

A.7 PDP-11 (RSX) systems

Due to incompatibilities between standard FORTRAN-77 and FORTRAN-77 on RSX systems, major changes had to be made to the CALF77 library routines before they could be run on PDP-11 systems. Before attempting to use CALF77 on your PDP-11 system, you should consult either your system manager, or the Prentice Computer Centre.

APPENDIX B

SUMMARY OF STANDARD CALLING SEQUENCES FOR BASIC ROUTINES

CALL PLOTIN(FILE,DEVICE)

FILE   is the name to be used for the output plot file, if there
            is one.
DEVICE   specifies the output device and other characteristics of
            the plot. See appendix A.

CALL PLOT(XPAGE,YPAGE,IPEN)
CALL FACTOR(FACT)
CALL WHERE(RXPAGE,RYPAGE,RFACT,MMFLAG,ORGX,ORGY)

XPAGE,YPAGE are the destination coordinates, in millimetres
            relative to the current origin, of the pen movement.
IPEN   magnitude specifies the pen up/down status during
            movement (1=no change, 2=down, 3=up) and, if negative,
            causes the origin to be redefined at the new position.
FACT   is a factor which determines the scaling of the
            subsequent plot.
RXPAGE,RYPAGE,RFACT are the locations which will contain the
            current values of XPAGE, YPAGE and FACT after WHERE is
            called.
MMFLAG   should always be set to .TRUE. by the call to WHERE.
ORGX,ORGY are variables set by the call to WHERE to the physical
            coordinates of the current origin.

CALL SYMBOL(XPAGE,YPAGE,HEIGHT,IBCD,ANGLE,NCHAR)
CALL CENTRE(XPAGE,YPAGE,HEIGHT,INTEQ,ANGLE,IPEN)
CALL NUMBER(XPAGE,YPAGE,HEIGHT,FPN,ANGLE,NDEC)

XPAGE,YPAGE define the position of the start of the string.
HEIGHT   is the height (and width) of a single plotted character.
IBCD   is the CHARACTER variable or literal string.
NCHAR   is the number of characters in IBCD.
ANGLE   is the angle at which the string is to be plotted.
INTEQ   is the integer code for the symbol to be plotted.
IPEN   specifies the pen up/down status for movement to the
           symbol location.
FPN   is a real (floating-point) value to be plotted.
NDEC   specifies the number of decimal places to be plotted.

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

ARRAY   is the array of data values to be plotted.
NPTS   is the number of array entries to be plotted.
INC   is the increment between the indices of entries to plot.
AXLEN   is the length of the axis line.
XPAGE,YPAGE are the coordinates of the axis origin.
IBCD   is the alphanumeric title for the axis.
NCHAR   is the number of characters in IBCD.
           It also specifies by its sign the side of the axis line
           on which the text is to be drawn.
ANGLE   is the angle of the axis line, from the X axis.
FIRSTV   is the first scale value printed on the axis.
DELTAV   is the increment between scale values drawn on the axis.
XARRAY,YARRAY contain the pairs of data values to be plotted.
LINTYP   specifies the type of line to be drawn.
INTEQ   is an integer specifying the symbol to be plotted.

CALL NEWPEN(NPEN)
CALL SETPEN(NPEN,TYPE,SIZE,INK)
CALL PLTOPR(MESSAG,LENGTH)

NPEN   is the number of the required pen.
TYPE   is an alphanumeric description of the pen type.
SIZE   is an alphanumeric description of the pen size.
INK   is an alphanumeric description of the pen ink colour.
MESSAG   is an alphanumeric message for the operator.
LENGTH   is the length in characters of MESSAG.

B-2

APPENDIX C

VUEPLT DOCUMENTATION FILE - VUEPLT.DOC

CONTENTS

C.1 Introduction

VUEPLT is a wonderful utility program used to preview or to convert
previously generated HP plot files. It is available on the KL10
and VAXen only at this stage (i.e. it is NOT available on the IBM).

C.2 History

There have been quite a conglomeration of preview programs stored
on the various systems, each performing slightly different
functions - although nobody seems quite sure what all these
slightly different functions are. This seems unnecessary and is
actually very confusing. So the versatile VUEPLT program
supersedes Vplot, Viewpl and Tekplo. Now there is no decision as
to which program to use -- if you want to output a previously
generated plot file to a graphics terminal or to a different
plotter, then USE VUEPLT. In particular, VUEPLT supports a number
of output devices including Tektronix terminals and plotters, HP
plotters, plus the GIGI graphics terminals.

C.3 Using VUEPLT

The program VUEPLT is very straightforward for the user. It is
invoked as follows,

on the KL-10:
        .run plo:vueplt

on the VAX:
        $ plot/preview
or
        $ run sys$uti:vueplt

Thereafter, the information requested is independent of the
machine. That is, VUEPLT requests the following information:
Output device, please? Plot file? Factor?

Wherever possible, if an invalid response is entered, the program
specifically states the problem and allows the user to enter a
correct value. Valid answers to these questions are now explained.

C.3.1 TO WHAT DEVICES CAN I SEND MY PLOT FILE ?

Valid output device The output devices that VUEPLT understands are
all those valid in a call to the PLOTIN subroutine. It even
understands the PLOTIN device switches. So, if you have a HP plot
file, you can output it to any one of the following devices:

              PLT3   -- the HP7585B plotter at the Computer Centre
              PLT4   -- the HP7475A plotter at the Computer Centre
              PLT5   -- the HP7475A plotter at Griffith University
              HP   -- either HP plotter at the Computer Centre
              HP7585 -- a HP7585 plotter (possibly in series with
                         your terminal)
              HP7475 -- a HP7475 plotter (possibly in series with
                         your terminal)
              TEK4010 -- Tektronix 4010 terminal (or lookalike)
              TEK4014 -- Tektronix 4014 terminal
              TEK4015 -- Tektronix 4015 terminal
              TEK4105 -- Tektronix 4105 colour terminal
              TEK4107 -- Tektronix 4107 colour terminal
              TEK4662 -- Tektronix 4662 plotter, in series with
                         your terminal
              GIGI   -- a GIGI terminal

       (Support for TEK devices is not available on UQVM.)

       In addition,
             Exit -- exits from the program
             /Help -- types out a short help text

The output device can also contain SWITCHES - again valid switches
are those that are valid to the PLOTIN subroutine. Especially
important is the switch "/TERMINAL". It is used to specify that
the output device is either your terminal or otherwise a device
connected in series with your terminal.

Examples:
              HP7475/terminal
              GIGI/TERMINAL

Because the Tektronix support is provided by the Tektronix TCS
routines, output is always directed to the terminal. Therefore
there is no need to specify the "/TERMINAL" switch for such
devices.

If the user does not enter a valid output device, VUEPLT will
default to produce a plot file for the HP device. If the output is
not to be sent to a terminal, it will be stored in the file,
VUEPLT.PLT.   Additional output devices will be available whenever
PLOTIN understands different devices.

For more details, have a look at the CALF77 documentation file, ie.

on the KL-10:

        .print doc:calf77.doc

on the VAX:

        $ print sys$doc:calf77.doc

C.3.2 WHAT PLOT FILES CAN I PREVIEW?

Naturally the program must be told which plot file you wish to preview or convert. This is entered as a valid file specification. VUEPLT checks the following:

* that the file really does exist; and
* that the plot file format is one that it understands, ie. a HP plot file format. The HP plot file contains HP-GL instructions.

(i) <u>Valid file types</u>

Currently, VUEPLT should be able to process any plots in HP-GL format. It is recommended that files containing multiple plots (e.g. as generated by SAS/GRAPH) should NOT be displayed using VUEPLT. There is no pause between successive plots, which would overlay each other, thus rendering the exercise pointless. Plots generated by SAS/GRAPH should be processed by SASHPCON on UQVM to split the plots into separate files. Type

        INFO SASHPCON

on UQVM for information on SASHPCON.

(ii) <u>Possible error messages</u>

*** filename was not found, so please enter the correct filename

    The file specified could not be opened for reading, so the user should enter the correct file specification.

*** This is not a valid HP plot file

    The plot file specified does not have a valid plot file identification as the first record of the file. Check that the correct filename was entered.

C.3.3 HOW CAN I CHANGE THE SIZE OF THE PLOT FILE OUTPUT?

Normally the plot is output as the real millimetre size of the
initial plot.   However, sometimes the user may want to enlarge or
reduce the size of the plot produced. This can be accomplished by
entering a different value to the factor query. A value for factor
between 0 and 1 REDUCES the size of the plot, whereas a value
greater than 1 enlarges the plot.

C.4 Other goodies

C.4.1 DEFAULTS

VUEPLT can be used to preview as many files as the user wishes. By
typing <cr> to any of the questions asked, the user indicates that
the corresponding previously validated response is required again.
This is very useful when you wish to preview numerous files on the
same terminal, or perhaps to factor the output a number of times.
To any rule there is of course an exception and in this case the
exception is that a request for help (/Help in response to the
output device parameter) does not alter the default value
established - the default reverts to the previous "true" device.

Initially the defaults are established as a combination of user
responses and initial program values.

The defaults are as follows:

  * default output device is established by the user, however if
    the device specified is invalid, the default is a HP plotter.
    For a HP plotter, the default switches (forms, carousel, etc.)
    are the same as the default switches for the device argument of
    the subroutine PLOTIN. These will be overriden (VAX only) by
    switches that were given when the plot file was created - these
    switches are recorded in the plot file. These in turn may be
    overriden (VAX and KL) by switches which the user may append to
    the output device name when prompted by VUEPLT. The resultant
    collection of switches will be recorded in the new plot file.

  * the default plot file is also established by the user. If the
    user enters a file that is not found, the program outputs a
    message to that effect and subsequently requests the user to
    enter a CORRECT filename.

  * default factor is set by the program to be 1.

C.4.2 UNKNOWN PLOT COMMANDS

VUEPLT has been written to conform as closely as possible to the
published HP-GL syntax for HP7475 and HP7585 plotters. If
additional plot commands are encountered, they are merely written
to the file VUEPLT.ERR and then ignored. The unrecognized command
is output together with the line number of the input file on which
the error was found.

When the plot file has been completely processed, the user is
notified of the existence of VUEPLT.ERR. If such a file is
created, please inform the HELP desk at the Prentice Computer
Centre, so that the problem can be investigated.

C.4.3 POSSIBLE PROBLEMS

(i) Terminal characteristics

If the plot file produced on a terminal appears "garbled", check
that the terminal characteristics are appropriate. For example, if
the output is a TEK4014 terminal, the computer must be set to
generate hardware form feeds and should not generate the <cr><lf>
sequence.

So on the KL-10,
     .set tt form no crlf

and on the VAX,
     $ set term /form/nowrap

(ii) Tektronix terminals

Don't forget that if you are using a Tektronix terminal or
Tektronix lookalike, you must press return after the plot has
finished being drawn before any more prompts are sent from VUEPLT.
This allows you to examine the plot on the screen or get a hard
copy of it without the additional messages from VUEPLT.

(iii) Transferring plot files between machines

Normally the plot files produced on either the VAX or the KL-10 can
be copied to the other machine and plotted or processed from its
new location. However, a word of caution ... a few problems have
arisen. For example,

   * transfer a HP plot file from the VAX to the KL-10:

The transfer should not generate fixed-length records, else
VUEPLT may output incorrect values - it expects VARIABLE-length
records. To correct the KL-10 file,
     .COPY =filename/T

C.4.4 EXAMPLES

  1. Run VUEPLT on the KL-10 to preview the HP plot files BOX.PLT
      and ANGLES.PLT on a TEK4662 plotter connected in series with
      your terminal. Since all output is directed essentially to the
      terminal, no VUEPLT.PLT file is produced. As the output is for
      a Tektronix device, the "/TERMINAL" switch does not need to be
      specified.

      .run plo:vueplt

      -- previewing plot file --

          output device, please? :   tek4662
          plot file?   :   box.plt
          factor?   :   <cr>

      (The plot is displayed on the terminal's screen.)

                                      end of plot ...

          *** VUEPLT.ERR contains all the unknown commands found

      -- previewing plot file --

          output device, please? : <cr>
          plot file?   : angles.ptl

          *** angles.ptl was not found, so please enter the
      correct filename

          plot file?   : angles.plt
          factor?   : <cr>

      (The plot is displayed on the terminal's screen.)

                                      end of plot ...

      -- previewing plot file --

          output device, please? : exit
      CPU time 0.87 Elapsed time 42.34

Exit

.

2. Run VUEPLT on the VAX to preview the HP plot files BOX.PLT and ANGLES.PLT on a HP7475A plotter connected in series with your terminal. Since all output is directed essentially to the terminal, no VUEPLT.PLT file is produced.

```
$ run sys$uti:vueplt

-- previewing plot file --

      output device, please? :   hp7475/terminal
      plot file?             :   box.plt
      factor?                :   <cr>
```

(The plot is output to the plotter attached to the terminal.)

                                    end of plot ...

 *** VUEPLT.ERR contains all the unknown commands found

```
-- previewing plot file --

      output device, please? :   <cr>
      plot file?             :   angles.plt
      factor?                :   <cr>
```

(The plot is output to the plotter attached to the terminal - don't forget to put some new paper onto the plotter before answering the last question.)

                                    end of plot ...

```
-- previewing plot file --

      output device, please? :   exit

$
```

3. Run VUEPLT on the KL-10 to convert the HP plot file ANGLES.PLT into a GIGI plot file format, and then output BOX.PLT to a HP plot file as twice its original size. For both functions, the output required is a disk file. The KL-10 will not allow multiple files of the same name (VUEPLT.PLT), so the user must enter an appropriate plot filename.

```
.run plo:vueplt

-- previewing plot file --

     output device, please? :  gigi
       plot file?            :  angles.plt
       factor?               :  <cr>
                                end of plot ...

-- previewing plot file --

     output device, please? :  <cr>
       plot file?            :  box.plt
       factor?               :  2
?FRSNAM OPEN unit 1 DSK:VUEPLT.PLT at HPSTRT+12 (PC 11125)
?FRSOPN can't OPEN file: File already exists
[FRSEFS Enter correct file specs]
*boxby2.plt<esc>
                                end of plot ...

     *** VUEPLT.ERR contains all the unknown commands found


-- previewing plot file --

     output device, please? :  exit
CPU time 0.90  Elapsed time 45.12

Exit


.
```

4. Run VUEPLT on the VAX to convert the HP file ANGLES.PLT
into a GIGI plot file format, and then output BOX.PLT to a HP
plot file as twice its original size. For both functions, the
output required is a disk file. The VAX allows multiple
versions of files (VUEPLT.PLT), however only files with the
three highest version numbers will normally be kept.

```
$ run sys$uti:vueplt

-- previewing plot file --

     output device, please? :  gigi
       plot file?            :  angles.plt
       factor?               :  <cr>
                                end of plot ...

-- previewing plot file --
```

C-9

```
      output device, please? :  hp
      plot file?              :  box.plt
      factor?                 :  2
                                 end of plot ...
```

*** VUEPLT.ERR contains all the unknown commands found

-- previewing plot file --

```
      output device, please? :  exit
```

$

INDEX