

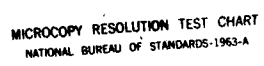
AD-A194 352

THE INTERACTIVE PLOTTING PROGRAM GRAPH - REVISED MANUAL 1/1
(U) MATERIALS RESEARCH LABS ASCOT VALE (AUSTRALIA)
S R KENNETH DEC 87 MRL-CD-0010 DODA-AR-005-192

UNCLASSIFIED

F/G 12/3

NL



DTIC FILE COPY

MRL-GD-0010

AR-005-192



AD-A194 352

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

MATERIALS RESEARCH LABORATORY

MELBOURNE, VICTORIA

GENERAL DOCUMENT

MRL-GD-0010

THE INTERACTIVE PLOTTING PROGRAM GRAPH
- REVISED MANUAL

S.R. Kennett

DTIC
ELECTE
MAY 24 1988
S D

Approved for Public Release

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

DSTO
MELBOURNE

C Commonwealth of Australia
DECEMBER 1987

02 5 22 097

-A-

**DEPARTMENT OF DEFENCE
MATERIALS RESEARCH LABORATORY**

GENERAL DOCUMENT

MRL-GD-0010

D
**THE INTERACTIVE PLOTTING PROGRAM GRAPH
- REVISED MANUAL**

S.R. Kennett

ABSTRACT

This report is a user guide for the interactive graphics program called GRAPH and replaces the earlier Materials Research Laboratories technical note, MRL-TN-485. The suggestions of users and the expansion of the capabilities of the program have led to this revision.

GRAPH is a command driven interactive program which allows the user, with only limited computer experience, to display and modify graphical output. The program can plot data on linear or logarithmic axes. The axis scales can be set by the user or autoscaled by the program. Labels can be added or changed and text can be written to any part of the plot. Data can be plotted as points, histograms or lines.

GRAPH has commands which allow for simple data reduction. Data can be multiplied, divided, added, subtracted by a constant value or by a second set of data. Data can also be exponentiated, integrated, differentiated or smoothed. A least-squares polynomial or a more general function can be fitted to the data.

GRAPH was written in FORTRAN 77 on the VAX 11/780 computer at the Materials Research Laboratories. (Australia)

Approved for Public Release

© Commonwealth of Australia

POSTAL ADDRESS: Director, Materials Research Laboratory
P.O. Box 50, Ascot Vale, Victoria 3032, Australia

CONTENTS

	Page No.
1. INTRODUCTION	1
2. INPUT TO THE PROGRAM	2
2.1 Y Data	2
2.2 XY Data	3
2.3 Titled Files	4
2.4 Multiple Plot Files	4
3. FILE PLOTTING COMMANDS	5
3.1 PLOT and P#	5
3.2 OVERLAY and O#	6
3.3 REDRAW and R#	7
3.4 NEXT	7
3.5 CONTINUE	8
3.6 SAME (DATA)	8
3.7 BOX	8
3.8 HBOX	9
3.9 ERROR BARS and ABSOLUTE	9
4. PLOT TYPE COMMANDS	10
4.1 DASH and LDASH	10
4.2 Logarithmic Axes (Base 10)	11
4.3 MIDPOINT	11
4.4 POINT, TYPE and SIZE	12
4.4.1 TYPE	12
4.4.2 SIZE	13
4.5 AVERAGE	13
4.6 RAPID	14
4.7 GRID	14
4.8 TOP	14
5. SIZE AND SCALE COMMANDS	15
5.1 AXES SIZES	15
5.1.1 SET	15
5.1.2 SETX and SETY	16
5.1.3 XSPACING and YSPACING	16
5.1.4 FREEZE	17
5.1.5 RAXIS and TAXIS	17
5.1.6 RSPACING and TSPACING	18

CONTENTS (continued)

	Page No.
5.1.7 XAXIS and YAXIS	18
5.1.8 TOUT	18
5.1.9 ROUT	18
5.2 LABELS AND TEXT COMMANDS	19
5.2.1 TITLE	19
5.2.2 SUBTITLE	19
5.2.3 XLABEL and YLABEL	19
5.2.4 NNUMBER, YNUMBER and XNUMBER	20
5.2.5 MARK, YMARK, XMARK, RMARK and TMARK	20
5.2.6 LETTER	21
5.2.7 Writting Arbitrary Text	22
5.2.7.1 WRITE	21
5.2.7.2 CENTER	22
5.2.7.3 EWRITE	22
5.2.7.4 SWRITE	23
5.2.7.5 CWRITE	23
5.2.8 SYMBOL	23
5.2.8.1 Direct Data Entry	24
 6. HARD COPY COMMANDS	 25
6.1 Plotter select commands	25
6.1.1 CIT161	26
6.1.2 QUME	26
6.1.3 TAB	26
6.2 PEN	27
6.3 SPEED	27
6.4 FONT	28
6.5 ANGLE	28
6.6 ERASE	28
 7. ANALYSIS COMMANDS	 29
7.1 File manipulation by constant factors	29
7.1.1 DIVX and DIVY	29
7.1.2 MULTX and MULTY	29
7.1.3 SHIFTX and SHIFTY	29
7.1.4 DEXP	30
7.1.5 DLOG	30
7.2 File Manipulation by Other Files	31
7.2.1 FMULTIPLY	31
7.2.2 FDIVIDE	31
7.2.3 FADDITION	32

CONTENTS (continued)

	Page No.
7.3 Integration and Differentiation	32
7.3.1 INTEGRATE	32
7.3.2 DIFFERENTIATE	32
7.4 Smoothing and interpolation	32
7.4.1 SMOOTH	33
7.4.2 INTERPOLATION	33
7.5 Line Fitting	34
7.5.1 PFIT #	34
7.5.1.1 PFIT #*	34
7.5.1.2 PFIT #H	34
7.5.2 FFIT	35
7.6 Auxiliary Manipulation	36
7.6.1 CHANNEL	36
7.6.2 LINE	36
7.6.3 ADDP	37
7.6.4 APPEND	37
7.6.5 INVERT	38
7.6.6 SORT	38
7.6.7 SWAP	38
7.6.8 VALUE	38
7.6.9 YYCOMPARE	39
7.6.10 FUNCTION	39
7.6.11 FOURIER	39
 8 LAYOUT UTILITY	
8.1 MODIFY	40
 9 COMMAND FILES	
9.1 *GO#	41
9.2 *CR#	41
9.3 *HALT	41
9.4 *CONTINUE	41
9.5 *STOP	42
9.6 Example Command Files	42
 10. INFORMATIVE COMMANDS	44
10.1 STATUS	44
10.2 HELP	44
10.3 DIRECTORY	44



Accession For	
NTIS CRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By _____	
Distribution/ _____	
Availability Codes	
Dist	Avail and/or Special
A-1	

CONTENTS (continued)

	Page No.
11. OTHER COMMANDS	44
11.1 EXPAND and E-	44
11.2 CLEAR	45
11.2.1 CLR	45
11.2.2 CLRDAT	45
11.2.3 INITIALIZE	46
11.3 % REPEAT	46
11.4 -AND-	46
11.5 STORE	47
11.6 DEFAULT	47
11.7 REMOVE	47
11.8 RETAIN and RECALL	48
11.9 FINISH	48
11.10 UNKNOWN COMMANDS	48
 12. ERROR MESSAGES	49
12.2 Command file errors	53
12.3 Warnings	54
 APPENDIX 1 - Command List	55
 INDEX	58

The interactive plotting program GRAPH - revised manual

1. INTRODUCTION

GRAPH is a command driven interactive program which allows the user with only limited computer experience to display and modify graphical data. This manual is an updated version of Materials Research Laboratories Technical Note, MRL-TN-485, November 1984.

The need for a general plotting and analysis program that would enable data to be plotted on available graphics terminals and plotters gave the necessary impetus for the creation of GRAPH. The interest and suggestions of users of the program have resulted in a number of improvements in the program since the previous version of the manual was written.

Data input into the program is simple and there are only a small number of restrictions on the input format. Once an appropriate data file is in a suitable form, the user is ready to run the program. On running GRAPH, a title page appears containing the version number and the phone number and name of the author (or person who could help with any difficulty). In the bottom left hand side of the screen appears the command prompt, 'C:'. It appears after every command is completed.

The program can plot data on linear or logarithmic axes. The axis scales can be set by the user or autoscaled by the program. Labels can be added or changed and text can be written to any part of the plot. Data can also be plotted as points, histograms or lines. Simple data reduction is made possible by the inclusion in the program of commands which allow integration, differentiation, polynomial fitting and general function fitting and smoothing. In addition data can be manipulated by multiplication, division, addition and subtraction by a constant or by a second set of data. GRAPH also includes a utility which enables the user to modify the aspect ratio of the plotted data and the position of the title and subtitle.

The commands can be divided into classes, each of which will be discussed in some detail with examples where possible.

- | | |
|----------------------------------|--|
| 1. File plotting commands | : control the input of data and plotting. |
| 2. Plot type commands | : Allow selection for different types of axes and ways of displaying data. |
| 3. Size, scale and text commands | : Control of labels, text and axes scales. |
| 4. Hard copy commands | : Control hard copy plotting variables, for example, pen colour and plotting speed. |
| 5. Analysis commands | : Allow data manipulation such as polynomial fitting, integration, addition and interpolation. |
| 6. Command file commands | : Allow for the handling of a file of commands and responses. |
| 7. Informative commands | : Supply information on the use of the program and status of the program. |
| 8. Other commands | : Cover the storing of manipulated data, expansion of areas of the screen plot and exiting from the program. |

The program GRAPH was written in FORTRAN 77 on a VAX - 11/780 computer. GRAPH uses mainly standard ANSI X3.64 escape sequences for text output control and thus can be used on a variety of graphics terminals and plotters with little or no modification. The graphics terminals available on the system during the writing of GRAPH were mainly Digital VT640s and Visual 550s which are both Tektronix 4010 compatible. The hard copy plotters on the system during the writing of GRAPH were a Tektronix 4662 (option 31), a National plotter, a Graphtec plotter, a Facit plotter (HP compatible), a Zeta plotter and a Houston instruments DM-29 (Bausch and Lomb) plotter. GRAPH was written to handle all these terminals and plotters.

2. INPUT TO THE PROGRAM

In order for GRAPH to read a user's data, the data must be in a form acceptable to the program. GRAPH accepts data from sequential files in either of two formats, called Y data and XY data. (Refer to section 5.2.8.1 for direct data entry.)

2.1 Y Data

This is the simplest file format. The data to be plotted consists of a series of Y values which are to be plotted against their position in the series. In these files:

- (1) The first record in the data file contains the number of points to be plotted (maximum 10000);
- (2) The second record contains a title, of up to 60 characters, which will appear on the top of the plot. (Note that this can be left blank);
- (3) The following record, or records, contain the Y values separated by any acceptable FORTRAN separator (commas, carriage returns, backslashes etc.). Y values can be in either floating point or integer form.

For example:

```
11
title line of test file
1,7,13,18,22,25,27,28,28,27,25
```

or equally valid would be,

```
11
title line of test file
1
7
13
18
22
25
27
28
28
27
25
```

Both files produce the same graph.

The maximum number of points that the program can handle in a plot is 10000.

2.2 XY Data

Most user information to be plotted consists of a series of X,Y pairs. The basic file format remains the same as in the previous example; that is, the number of points is followed by a title line and the data. However, in order to distinguish between the two file types, the number of points is entered as a negative number.

The data in an XY plot is stored as X,Y pairs with the X value preceding the Y value. The data can be separated by any valid FORTRAN separator.

An example file would be ,

```
-8
title line for test XY data file
```

```

13,25
1,25
2,14
6,-.5
-1,-.3
12,12
9,3
12,-8.31

```

2.3 Titled Files

Five extra lines of data can be optionally added to the front of the file to enable the labeling of axes and the inclusion of a subtitle on the plot. The first of these lines contains the X axis label, the third contains the Y axis label and the fifth the subtitle. The second and fourth lines each contain a number which presently is not used in the program and should be set to 1. The rest of the file remains as described before; the number of points, a title line and the the data points.

Thus a Y data plot with title lines could look like,

```

x label
1
y label
1
subtitle
11
title line for test file
1,7,13,18,22,25,27,28,28,27,25

```

Both the axes labels can be up to 35 characters in length and the subtitle can be up to 60 characters long.

2.4 Multiple Plot Files

The user will often require that data associated with one task be stored within the one file. To satisfy this requirement, GRAPH accepts files that consist of multiple sets of data. The five lines of title information in titled files can be included only at the beginning of the file.

For example, the following is a multiple plot file consisting of two plots, one with XY data the other with Y data;

```

-5
DATA CURVE 1
2,11
3,15
4,20
5,26
6,33
6
DATA CURVE 2
8,12,17,23,30,28

```

3. FILE PLOTTING COMMANDS

GRAPH contains a number of commands which control the input of plot data into the program and what happens with that data. These are;

PLOT and P#	OVERLAY and O#	REDRAW and R#
NEXT	CONTINUE	
SAME (DATA)	BOX	HBOX
ERROR	ABSOLUTE	

3.1 PLOT and P#

To plot a data file, type the command PLOT (P) ‡ followed by a carriage return key in response to the "C:" prompt. The program will then request the name of your data file. The specified directory or the user directory from which GRAPH was called will be searched for the file. Files which have the extension name '.DAT', '.KEN' or '.' can be specified without the extension. A version number can be included. GRAPH reads in the data once the file has been located.

A carriage return with no file name, causes the program to plot the last named file. In such a case the following message is given;

DEFAULT - *****.***,

where *****.*** is the file name last used.

The program will scale the plot so that the Y data will fill 90% of the Y-axis and the X data will fill all the X-axis by default. The spacings of the marks along the axes are determined by the data. Between 4 and 10 labeled marks will be displayed on each axis. The title will appear at the top of the plot just above the data. The data points are, by default, connected by straight lines.

For example, to plot the following file called 'test.dat':

```
11
test file
1,7,13,18,22,25,27,28,28,27,25
```

After running GRAPH the following command is issued.

```
C: plot <CR> ‡
ENTER THE FILE NAME : test.dat <CR>
```

‡ The shortest acceptable abbreviation to each command is shown in brackets.

† note:

1. <CR> = carriage return key.
2. The user's commands and responses are kept in lower case and prompts by GRAPH are displayed in upper case for the examples only. In actual use, GRAPH accepts responses in either lower or upper case characters and the prompts by GRAPH are a mixture of both upper and lower case.

The plot shown in figure 1 is drawn on the terminal screen.

To plot the second, or later plot, in a multiple plot file the command PLOT is followed by the numerical position of the plot in the multiple plot file.

For example:

```
C: p 6 <CR>
ENTER THE FILE NAME : test.dat <CR>
```

The sixth plot in the multiple plot file 'test.dat' would then be plotted.

The command can be abbreviated by the inclusion of the file name after the command. Thus

```
C: plot 6 test.dat <CR>
```

or

```
C: p6 test.dat <CR>
```

This will produce the same result in the previous example. Note that the inclusion of spaces is optional. The plot number is also optional. (The plot number defaults to 1.)

3.2 OVERLAY and O#

After plotting a file, a second file of data may be overlaid onto the plot. This is achieved using the command OVERLAY (O). The OVERLAY command is issued followed by a carriage return. The program then requests a file name in exactly the same way as for PLOT. There is no limit on the number of overlays as the data array stored by GRAPH is replaced each time with the overlaid data.

The following is an example of the use of the OVERLAY command:

```
C: plot <CR>
INPUT THE FILE NAME : test.dat <CR>
C: overlay <CR>
INPUT THE FILE NAME : test1.dat <CR>
```

To overlay a given plot in a multiple plot file it is necessary to give the command with the numerical position of the plot in the file appended to it.

For example:

```
C: overlay 3 <CR>
INPUT THE FILE NAME : test2.dat <CR>
```

This would result in the third plot in the multiple plot file 'test2.dat' being overlaid onto the previous plot.

Like the command PLOT, OVERLAY can have the filename included at the end of the command. The previous example could have been written:

```
C: overlay 3 test2.dat <CR>
```

or

```
C: o3 test2.dat <CR>
```

To overlay a plot from the same multiple plot file, the filename DEFAULT can be used.

For example :

```
C: plot test.dat <CR>
C: overlay 3 default <CR>
```

In this example the data of the first plot in test.dat is plotted and the data of the third plot in test.dat is overlaid onto the plot.

3.3 REDRAW and R#

The data last plotted may be redrawn on the same axes by using the command REDRAW (R). This is useful after modification of labels and other parameters. It may be also used to copy the screen data to a plotter.

For example:

```
C: plot <CR>
INPUT THE FILE NAME : test.dat <CR>
C: graph <CR>           — see section 6.1
C: redraw <CR>
```

The above sequence of commands will plot the file 'test.dat' on the screen, turn on the inline† Tektronix 4662 plotter and redraw the plot on the screen to the plotter.

The command REDRAW followed by a number is an extension of redraw that draws the plot specified by that number in a multiple plot file.

For example:

```
C: plot <CR>
INPUT THE FILE NAME : test.dat <CR>
C: r 4 <CR>
```

In this example the data of the first plot in the multiple plot file 'test.dat' is plotted, then the screen is cleared and the data in the fourth plot in the file 'test.dat' is plotted. The scaling of this new plot depends on the new data.

3.4 NEXT

The command NEXT (N) is a request to plot the next plot in a multiple plot file. The data is scaled according to the new data.

For example:

```
C: plot <CR>
INPUT THE FILE NAME : test.dat <CR>
C: next <CR>
```

In this example, the first plot in the multiple plot file 'test.dat' is plotted, then the screen is cleared and the second plot is plotted.

† Inline means that the plotter is connected to the computer and the terminal is connected to the plotter.

3.5 CONTINUE

The command CONTINUE (C) can be used to overlay a sequence of plots without reference to their position within a multiple plot file. The command increments the position to the next plot in the file.

In the example that follows, the first plot in the multiple plot file 'test.dat' is plotted, and the following two plots are overlaid ;

```
C: p3 <CR>
ENTER THE FILE NAME : test.dat <CR>
C: continue <CR>
C: continue <CR>
```

If the command CONTINUE is followed by a number, GRAPH skips that number of plots before overlaying the next one.

For example :

```
C: p test.dat <CR>
C: c 3 <CR>
```

In this example, the fourth plot in the file 'test.dat' is overlaid on a plot of the data of the first plot in the file.

3.6 SAME (DATA)

The data on a plot may need to be plotted separately from the axes. This may be useful, for example where the user wishes to plot the axes in a different color. The command SAME (S) or DATA (DAT) allows the data to be plotted on the same axes as previously plotted. There is one minor difference between the two commands; the command DATA will clear the screen before plotting the data when used to plot to the terminal.

For example:

```
C: plot
INPUT THE FILE NAME : test.dat <CR>
C: point <CR>          — see section 4.4
C: same <CR>
```

The result of this is shown in figure 3.

3.7 BOX

As a complementary function to DATA, the command BOX is used to draw the axes without the data. BOX requires either a previous SET or PLOT command.

For example:

```
C: plot <CR>
INPUT THE FILE NAME : test.dat <CR>
C: graph <CR>          — see section 6.1
C: pen 1 <CR>          — see section 6.2
C: box <CR>
C: pen 2 <CR>
```


C: data <CR>

This command sequence will initially plot the data of 'test.dat' to the terminal screen and then an inline Tektronix 4662 plotter will be activated and the plot will be drawn by the plotter. The box will be in one color and the data in a second color.

3.8 HBOX

The command HBOX acts to toggle between the default axes, where a complete box surrounds the data, to a half box consisting of the left and bottom axes.

For example:

```
C: hbox <CR>
HALF BOX SET
C: hbox <CR>
FULL BOX RESTORED
```

In this example the command is used firstly to set half axes on subsequent plots, so that only the left and bottom axes would be drawn, then the command is used to restore the axes on subsequent plots to the default full box.

3.9 ERROR BARS and ABSOLUTE

The inclusion of error bars into a plot is achieved by using a second file of data which contains only the error information. The command ERROR BAR (ERR) is used with proportional errors † and ABSOLUTE (ABS) is used with absolute errors‡. The commands, ERROR and ABSOLUTE, instruct GRAPH that the data in the plot file specified is to be used for the creation of error bars.

When either command is given the program prompts for the file name of the error information. The file name can be optionally included at the end of the command line. If the data file is in a multiple plot file then the plot number is included after the appropriate command. If the number is absent the number is assumed to be 1.

For error bars which are symmetric about the data points the plot file contains the number of points, a title line and the size of the error bars. When the required error bars are not symmetric, the plot file contains negative number of points, a title line and point pairs of first the negative going error and then the positive going error.

The following example is a multiple plot file called 'test.data.dat' which contains two plots. The first contains the data itself and the second contains the proportional errors for this data:

† Proportional error is defined as the ratio of the uncertainty of the data to the data.

‡ Absolute error is defined as the magnitude of the uncertainty of the data.

```

-5
test data
1.3 , 12.3
2.0 , 14.6
3.3 , 21.3
4.1 , 23.5
4.8 , 27.4
-5
The proportional error associated with the data in plot 1
.055
.078
.067
.040
.088

```

To display this information the following could be used:

```

C: point <CR>
C: set <CR>
LOWER AND UPPER BOUND FOR THE Y-AXIS : 0,30 <CR>
LOWER AND UPPER BOUND FOR THE X-AXIS : 0,5 <CR>
C: p test.data <CR>
C: error 2 test.data <CR>

```

The data of 'test.data.dat' is then plotted and the error bars are overlaid.

4. PLOT TYPE COMMANDS

This section describes commands which alter the way in which the data is drawn and the type of axes used.

4.1 DASH and LDASH

Dashed lines can be used to distinguish between plots drawn on the same axes where there is no choice of color. A toggle between solid lines and dashed line is implemented in GRAPH by the command DASH (DAS) together with an ancillary command LDASH (LDAS). When the command DASH is first given, GRAPH checks to see whether the lengths of the dashes and spaces have been defined. If they have not, then a length value of 10 is assigned to both and the following message is issued ;

DEFAULT - DASH LENGTHS (10).

The command LDASH is used to define the lengths of the dashes and spaces on the plot. On issuing this command, the length of the dashes and the length of the spaces are prompted for. The unit of length is in terms of thousandths of the X-axis length.

For example:

```

C: ldash <CR>

```

```

INPUT THE DASH LENGTH (1,100) : 20 <CR>
INPUT THE SPACE LENGTH (1,100) : 10 <CR>
C: dash <CR>
DASHED LINES SET
C: dash <CR>
SOLID LINES SET

```

This sequence would set the lengths of the dashes to .02 of the screen width and the spaces to .01 of the screen width. The last two commands initially set dashed lines then set solid lines.

4.2 Logarithmic Axes (Base 10)

Three commands allow the user to toggle logarithmic axes on either or both axes. They are LOG, LOGX and LLOG.

The first of these LOG, toggles the Y-axis between logarithmic and linear axes, the second LOGX, toggles the X-axis in the same way and the third LLOG, toggles both axes at the same time. LLOG will give an error if the axes are not of the same type. That is, an error occurs when one axis is linear and the other is logarithmic. GRAPH plots the logarithm of the absolute value of the data. No warning is given with negative numbers.

Zero is plotted as the logarithm of 10^{-99} .

Where a data point is zero and the data is requested to be plotted logarithmically then only four cycles are plotted.

For example:

```

C: plot <CR>
INPUT THE FILE NAME : test.dat <CR>
C: log <CR>
LOG Y-AXIS SET
C: redraw <CR>

```

In this example the data of the file, 'test.dat' is first plotted on linear axes and then is redrawn with a logarithmic Y-axis.

If a previous SET command (section 5.1.1) has been given and a negative or zero bound was given, the effect of the LOG or LOGX command is to remove the set bounds. A warning is given.

4.3 MIDPOINT

MIDPOINT (MID) acts to toggle between successive data points being connected by straight lines and the points being plotted as histograms.

When the histogram mode is entered, a choice is offered between a histogram in which each bar is defined to extend from a position halfway from the last data point to a position halfway to the next data point (center points) and a histogram in which each bar starts at each successive data point.

For data that is not in ascending order it may be necessary to SORT the data before plotting in the histogram mode (see section 7.6.6).

Example:

```

C: plot <CR>
INPUT THE FILE NAME : test.dat <CR>
C: midpoint <CR>
HISTOGRAM PLOTTING SET
Each point at center or start of bar (C or S)? C <CR>
C: redraw <CR>

```

The above sequence of commands will firstly plot the data of 'test.dat' with each point being connected by straight lines and then replot the data as a histogram. The histogram has been chosen so that the data points are at the center of the bars.

4.4 POINT, TYPE and SIZE

The user may wish to plot individual data points as symbols. To do this it is necessary to use the command POINT (POI) which toggles between line and point plotting. GRAPH informs the user of the new state on each use of the command.

For example:

```

C: point <CR>
POINT PLOTTING SET
C: point <CR>
LINE PLOTTING SET

```

When GRAPH receives the command POINT, it checks to see if the size and the type of points are already defined. If not, GRAPH then sets the size to 7 and the type to 1 (see SIZE and TYPE below) and issues the following message;

DEFAULT POINT SIZE = 7

4.4.1 TYPE

There are 9 distinct point types whose size can be set by the user. The point type can be set with the command TYPE (TYP) and their size can be set with the command SIZE (SIZ). The command TYPE followed by the type number is used to specify the point symbol type.

For example:

```

C: type <CR>
INPUT THE POINT TYPE : 2 <CR>

```

The type number also may be entered immediately after the command.

```

C: type 2 <CR>

```

The space between the command and the number is optional.

The available point types are :

Type number	Description
1	square with center mark
2	circle with center mark
3	triangle with center mark
4	upright cross (+)
5	diagonal cross (x)
6	square
7	circle
8	triangle
9	line joining the point to the X-axis

4.4.2 SIZE

The size of the point symbols is set using the command **SIZE** followed by a positive number. The number corresponds to approximately to the number of pixels wide the symbol will appear on the screen.

For example:

```
C: size <CR>
SIZE = 12 <CR>
```

will produce points approximately 12 pixels wide.

The **SIZE** command can be abbreviated to a single line by including the numerical value with the command.

For example:

```
C: size 12 <CR>
```

The space between the command and the number is optional.

4.5 AVERAGE

In many cases, the legibility of the plot can be improved by averaging a number of data points. This is often done with very noisy data to clarify a trend in the data.

The command **AVERAGE (AVE)** can be used to average any number of points in the plot. On the command **AVERAGE**, **GRAPH** prompts for the number of points to be averaged.

For example:

```
C: average <CR>
EVERY HOW MANY POINTS ? 5 <CR>
```

In subsequent plots, every 5 points will be averaged together and the mean value displayed at each of the 5 points.

The command can be abbreviated by the inclusion of the number of points to be averaged together after the command.

For example:

```
C: ave 5 <CR>
```

The space between the command and the number is optional.

4.6 RAPID

When plotting files containing a large number of points, it is often useful to plot every third or fourth point as a means of seeing the trend in the data, with a reduced plotting time. The command RAPID (RAP) allows the user to specify how many points will be skipped before the next point will be plotted. On entering this command GRAPH responds with a prompt for the number of data points between each plotted point. The user can respond with a number between 1 and 99.

For example:

```
C: rapid <CR>  
EVERY HOW MANY POINTS ? 5 <CR>
```

In this example, GRAPH is set up so that the next and subsequent plots are plotted showing every fifth point. The effect of this command can be reversed by repeating the command and setting the spacing to 1.

The command can be abbreviated to RAP ## where the '##' refers to a one or two digit number.

For example:

```
C: rap 3 <CR>
```

4.7 GRID

The command GRID allows the plotting of a grid of lines across the plot. The grid spacing is determined by the spacing of the numbered axis marks. GRID is a command which toggles, i.e. the command is cancelled by repeating it.

For example:

```
C: grid <CR>  
GRID PLOT ON  
C: grid <CR>  
GRID PLOT OFF
```

4.8 TOP

This command effectively reduces the number of data points plotted along the x-axis. It can be used to truncate a plot, for example, at the onset of noisy data.

After being given the command TOP, GRAPH prompts for the last point to be plotted or redrawn. Once set, the top point remains the same until a subsequent TOP command changes the top point. (When the top point is set to zero the following plots will display all of the data.) The following example demonstrates the use of the TOP command.

The user has 500 points of thermocouple data in a file 'therm1.dat'. The data after the 300th point is useless owing to the dislodging of the thermocouple from the specimen. The user wishes to compare this data with a second lot of data, in a file called 'therm2.dat',

where the thermocouple remained attached to the specimen for the 500 points. To do this the following commands are recommended:

```
C: plot <CR>
INPUT THE FILE NAME : therm2.dat <CR>
C: top <CR>
INPUT THE TOP CHANNEL NUMBER : 300 <CR>
C: overlay <CR>
INPUT THE FILE NAME : therm1.dat <CR>
C: top <CR>
INPUT THE TOP CHANNEL NUMBER : 0 <CR>
```

The data of 'therm2.dat' was plotted first and then the top channel was set to 300. The data of 'therm1.dat' was then overlaid onto the plot with only the first 300 points being overlaid. The final command removed the predetermined top channel number for subsequent plots.

5. SIZE AND SCALE COMMANDS

5.1 AXIS BOUNDARIES

5.1.1 SET

The SET command in GRAPH allows the axes limits to be predetermined. After the command SET, the program prompts for the limits for the axes. It expects two numbers to be entered to set up each axis (The lower limit followed by the upper limit.). These numbers can be either real or integer and separated by any valid FORTRAN separator.

For example:

```
C: set <CR>
LOWER AND UPPER BOUNDS FOR THE Y-AXIS : 0 10 <CR>
LOWER AND UPPER BOUNDS FOR THE X-AXIS : 0 100 <CR>
```

In this example the limits of the plot are set such that the Y-axis displays from 0 to 10 and the X-axis displays from 0 to 100. All plots after a set command have these limits until another SET command or a FREEZE command is given. (section 5.1.4)

A warning must be given about the spacing of the marks on the axes. A SET command will cause the mark spacing to remain at the previous spacing. Thus it is advisable to set the spacings of the marks before a PLOT or REDRAW. (see YSPACING and XSPACING, 5.1.3)

5.1.2 SETX and SETY

Often it is necessary to set only one of the axes and allow the other to be autoscaled. The commands SETX and SETY allow each axis to be treated separately. These commands act like toggles and give the status of the particular axis. On giving the command SETX or SETY, the user is prompted for the bounds of that particular axis and informed that the axis bounds are fixed. On giving the same command again, the program simply allows the axis on subsequent plots to be autoscaled and informs the user.

For example:

```
C :setx <CR>
X AXIS FROZEN
LOWER AND UPPER BOUNDS FOR THE X AXIS ? : 0,10 <CR>
C: setx <CR>
X AXIS FREE
```

This example shows the X-axis being set with limits of 0 to 10 and then freed to be autoscaled. If the upper bound is set below the lower bound then the data is ignored and the program returns to the command point.

5.1.3 XSPACING and YSPACING

Occasionally the user will require the ability to fix the spacing of the labeled marks. The commands XSPACING (XSPA) and YSPACING (YSPA) allow the user to predetermine the spacings of the marks along each of the axes. Once the user has used either of these commands the spacing remains fixed until either a FREEZE command is toggled off or until the appropriate command, XSPACING or YSPACING, is given a second time. A limit of twenty five labeled marks is placed on each axis. If this is exceeded while GRAPH is plotting data then a warning is given and that axis is autoscaled.

For example:

```
C: plot <CR>
INPUT THE FILE NAME : test.dat <CR>
C: xspacing <CR>
INPUT THE X-AXIS SPACING : 10 <CR>
C: yspacing <CR>
INPUT THE Y-AXIS SPACING : 20 <CR>
C: redraw <CR>
C: xspacing <CR>
AUTOSCALE X-AXIS
```

In this example the data of 'test.dat' was auto-plotted with the spacing of the axis marks being determined by the program. Then the spacing of the labeled marks on the X-axis was set to 10 units apart and on the Y-axis 20 units apart and the plot redrawn. The second XSPACING command removed the restriction on the next plot of having the X-axis spacing fixed at 10 units apart. The Y-axis on the next plot will have labeled marks 20 units apart.

The numerical spacing can be included after the command.

For example:

```
C: yspacing 10 <CR>
```


In this example the spacing of the marks of the y-axis is set to 10 units.

5.1.4 FREEZE

The autoscaling facility can be effectively removed without using the SET command by using the command FREEZE (FRE). The use of FREEZE allows the autoscaled axes of one plot to be retained for subsequent plots. The effect of this command can be removed by repeating the command.

For example:

```
C: plot <CR>
INPUT THE FILE NAME : test.dat <CR>
C: freeze <CR>
FREEZE FLAG SET
C: plot <CR>
INPUT THE FILE NAME : test1.dat <CR>
C: freeze <CR>
FREEZE REMOVED
```

In this example the data of 'test.dat' was plotted on the screen. The axes limits were retained using the command FREEZE and a second plot of the data of 'test1.dat' was plotted. The second FREEZE command toggles off the fixed axes limits and further plot commands would be autoscaled.

5.1.5 RAXIS and TAXIS

Two commands allow the top and right axes to be optionally labeled and scaled differently from the left and bottom axes. They are TAXIS (TAX) and RAXIS (RAX). These commands act to toggle the respective labeling and scaling on and off. They do not affect the data itself.

When the command RAXIS is first given, a label for the right axis is prompted for. A label of up to 35 characters may be used. After the label has been entered, the functional relationship between the left and right axis is requested. This relationship is entered as a single line polynomial expression. The expression can only contain numbers (with decimal points and exponents), the operations '**', '+', '-', '/', and '*', and the letter 'L' (for left) after the equals sign. 'LL' is interpreted as L*L. The operation '**' is the exponentiation operation as in fortran.

Once the expression has been entered it is translated into terms each having a coefficient and a power of L.

In subsequent plots, the right axis only is rescaled, labeled and approximately numbered.

A second RAXIS command removes the right axis label and scales.

The command TAXIS behaves in the same way except that the expression is now written in terms of 'B'.

For example:

```
C: raxis <CR>
INPUT THE RIGHT AXIS LABEL: test <CR>
INPUT FUNCTIONAL RELATIONSHIP BETWEEN THE RIGHT
AND THE LEFT AXES. THIS IS INPUT AS A SINGLE LINE LIKE
R = L*1000 + 1/L*100 + L**2
```

INPUT FUNCTION: $r = l^2$ <CR>

In this example the right axes is scaled as the square of the left axis and has the label 'test'.

5.1.6 RSPACING and TSPACING

RSPACING (RSPA) and TSPACING (TSPA) set the mark spacing on the right and top axes respectively if the commands RAXIS and TAXIS are in effect. They are used in the same way as XSPACING. (see section 5.1.3).

5.1.7 XAXIS and YAXIS

The x and y axes are included by default when they lie inside the plot region. The commands XAXIS and YAXIS allow the user to switch off these axes when they are not needed. These commands act like toggles and give the status of the axes.

For example:

```
C: xaxis
X AXIS OFF
C: xaxis
X AXIS ON
```

5.1.8 TOUT

The command TOUT acts to toggle off the top axis marks. When used after the TAX command the TOP axis label and numbers are also removed.

For example:

```
C: tout <CR>
C: TOP AXIS MARKS REMOVED
C: tout <CR>
C: TOP AXIS MARKS RESTORED
```

5.1.9 ROUT

The right axes marks can be removed using the command ROUT. This command acts to toggle the right axes marks on and off. When used after the RAX command the right label, and numbers are removed from subsequent plots.

For example:

```
C: rout <CR>
RIGHT AXIS MARKS REMOVED
C: rout <CR>
RIGHT AXIS MARKS RESTORED
```

5.2 LABELS AND TEXT COMMANDS

5.2.1 TITLE

The user can alter the title line on the plot using the command TITLE (TIT). On entering this command, GRAPH responds with a prompt for the new title. A title can be as long as 60 characters. A carriage return after the prompt inserts a blank title.

For example:

```
C: plot <CR>
INPUT THE FILE NAME : test.dat <CR>
C: title <CR>
NEW TITLE : trial #3 , data test <CR>
C: redraw <CR>
```

The original title contained in the file 'test.dat' is replaced with "trial #3 , data test" when the plot is redrawn.

The command can be abbreviated by including the text after the command. In the abbreviated command, blanks before the text are ignored.

For example :

```
C: plot test.dat <CR>
C: title trail #3 , data test <CR>
C: redraw <CR>
```

This would result in the same plot as the previous example.

5.2.2 SUBTITLE

The subtitle is a second line of text directly below the title line. The command SUBTITLE (SUB) allows the user to include or modify a subtitle line of up to 60 characters. The command is used in the same way and has the same formats as the TITLE command.

For example :

```
C: subtitle as of 12/7/87 <CR>
```

The subtitle is set to "as of 12/7/87" for subsequent plots. If no text is included after the command, text is prompted for.

5.2.3 XLABEL and YLABEL

The axes labels can be included in the data file. However it is often more convenient to add or change the axes labels after first plotting the file. The commands YLABEL and XLABEL allow the user to add axes labels of up to 30 characters. These commands are used in the same way as SUBTITLE and TITLE.

5.2.4 NNUMBER, YNUMBER and XNUMBER

Three commands allow the numbers along each of the axes to be toggled off and on, individually or collectively. They are NNUMBER (NNUM), YNUMBER (YNUM) and XNUMBER (XNUM).

The commands XNUMBER and YNUMBER toggle off and on the axes numbers on the X-axis and Y-axis respectively. The command NNUMBER toggles both axes numbers off and on. If the user has only one axis with numbers and the command NNUMBER is given, then GRAPH will turn the axes numbers off.

GRAPH responds to these commands by giving the state of the toggle after the command.

For example:

```
C: nnumber <CR>
NUMBERS OFF
C: nnumber <CR>
NUMBERS ON
```

5.2.5 MARK, YMARK, XMARK, RMARK and TMARK

A number of submarks can be inserted between the numbered marks on each axis using the commands MARK, YMARK, XMARK, RMARK and TMARK. On using any one of these commands the program responds with a prompt asking for the number of submarks to be inserted between each numbered mark. An attempt to enter either a negative number, or a positive number greater than 50, will result in the prompt being given again.

MARK is used when the same number of submarks is to be used on each axis. YMARK and XMARK refer to the Y and X axes respectively. RMARK and TMARK refer to the Right and Top axes respectively and only have an effect if the RAXIS or TAXIS commands have been used. (see section 5.1.5)

For example:

```
C: mark <CR>
INPUT THE NUMBER OF MARKS : 4 <CR>
```

After entering the above command and response the program will, on subsequent plots, show four submarks, effectively dividing each numbered segment on each axis into fifths.

This command can be abbreviated by the inclusion of the number of marks, after the command.

For example:

```
C: mark 4 <CR>
```

This has the same result as the previous example.

5.2.6 LETTER

The VT640 and other terminals that emulate the Tektronix 4010 often have the facility of having a number of text sizes. In the production of overhead projection material larger than normal text sizes are often required. The program GRAPH supports three letter sizes. These are supported on both the terminals and the plotters. The default size is the smallest of the three. Graph alters various positions on the plots to accomodate the larger lettering.

On entering the command LETTER (LET), the users choice of letter size is prompted for.

For example:

```
C: plot <CR>
INPUT THE FILE NAME : test.dat <CR>
C: letter <CR>
SIZE OF THE LETTERS (1,2,3) ? 2 <CR>
C: redraw <CR>
```

In this example, the data of 'test.dat' is drawn in the default letter size. The letter size is increased to size 2 and the data is redrawn with larger letters. An abbreviated form of the command is allowed where the letter size directly follows the command.

For example:

```
C: let 3 <CR>
```

5.2.7 WRITING ARBITRARY TEXT

5.2.7.1 WRITE

The command WRITE (WRI) enables the user to write text anywhere on the plot. When the command WRITE is given, GRAPH will respond differently depending on whether the user is plotting to the screen or a hardcopy device.

When plotting on the screen, GRAPH firstly prompts for the text. Then GRAPH sends a command to the terminal for the terminal to enter its cross hair mode. The cross hairs are used to define the lower left hand corner of the text. (The command CENTER allows the cross hair to define the center of the text.) The cross hairs are positioned using the arrow keys on the terminal. The carriage return key is actuated when the cross hairs are in position. The text then will be written onto the screen and the cross hair position will be displayed for later reference.

For hard copy plots, GRAPH first prompts for the text, then prompts for the text position. The user replies by giving the desired position of the lower left hand corner of the text in terms of the scales along the axes. The X value is entered first followed by the Y value.

After positioning the text on the screen, if the user wishes to copy the text to a hard copy plot, the user need only enter the appropriate hardcopy mode (section 6.1) and give the command WRI. In response to the prompt the user presses the carriage return key and the text is then written to the same coordinate as displayed on the screen.

For example:

```
C: graph <CR>
HARD COPY ON
```

```
C: write <CR>
INPUT THE TEXT : test point <CR>
INPUT THE POSITION (X,Y): 10,20 <CR>
```

In this example, the text 'test point' was written to a hard copy device at the point (10,20). The command can be abbreviated by including the text after the command.

The command sequence can be reduced by including the text after the command. A space is then required before the first letter in the text.

For example:

```
C: write test point <CR>
```

5.2.7.2 CENTER

The command CENTER (CEN) defines the center of the text to be at the position of the cross hairs for subsequent WRITE commands. Repeating the command CENTER resets the cross hairs to the left of the text.

For example:

```
C: center <CR>
CROSSHAIRS AT CENTER OF TEXT
C: center <CR>
CROSSHAIRS AT LEFT OF TEXT
```

5.2.7.3 EWRITE

Any text written to the terminal screen by a WRITE command can be deleted by a EWRITE (EWRI) command. This command rewrites the text of the last WRITE command with appropriate pixels switched to a no-write condition.

It should be noted that a WRITE command followed by no text input will still rewrite the previous text. This means that if the text was only written with the wrong size letters then the text does not need to be reentered.

For example:

```
C: write <CR>
INPUT THE TEXT : test line number one <CR>
'position cross hairs' <CR>
C: ewrite <CR>
C: let 2 <CR>           — see section 5.2.6
C: write <CR>
INPUT THE TEXT : <CR>
```

In this example the text, "test line number one", was written on the terminal screen and the erased. The letter size was altered and the text was rewritten.

5.2.7.4 SWRITE

The command SWRITE (SWRI) is used to create a command file containing the text of a prior WRITE command. Associated PEN and LETTER commands are also written to this command file. The command SWRITE can be used in the production of overhead projection screens. See the example given in section 9.6.

A command file consists of a series of GRAPH commands and their responses and must be named 'graph##.cmd' (## is a two digit number between 1 and 99). Command files and commands associated with their use, are described under COMMAND FILES. The first time SWRITE is used GRAPH asks for a number, ## between 1 and 99, in order to open a new command file, called 'graph##.cmd'. On receiving the number GRAPH writes the following commands to the file;

```
FONT 8
SPEED 3
PEN 1
LETTER
'the letter size being used'
WRITE
'your text from the last write'
'x position' 'y position'
```

The text in quotation marks in this example merely describe the user supplied details. Subsequent SWRI commands append to the file only the last three commands shown in the above example. The command PEN becomes available after the first SWRI command. Thus the pen color can be changed with every SWRITE command, but the first WRITE in the command file written by SWRITE will be with pen 1.

5.2.7.5 CWRITE

The command CWRITE (CWRI) closes the command file opened by the command SWRITE and thus allows the newly created command file to be used, or a new command file to be written by SWRITE.

5.2.8 SYMBOL

The symbols used in point plots can be drawn onto a plot at any position using the command SYMBOL (SYM). GRAPH will prompt for the type of symbol to be drawn. The symbol type can be optionally included after the abbreviated command, SYM. When plotting to the screen, GRAPH enters the terminals cross hairs mode to position the symbol. When plotting to a hardcopy plotter, GRAPH prompts for the X and Y coordinates of the center of the symbol. The size of the symbol is determined by a prior SIZE command or POINT command.

For example:

```
C: size 7 <CR>
C: symbol <CR>
SYMBOL TYPE: 3 <CR>
<CR>
```

Here the size of the symbol is set to 7 and GRAPH draws a triangle after the cross hairs are positioned. An extension of this command allows the user to add data points to the plot; the extension is made to the abbreviated command by adding a '*'

For example:

```
C: size 7 <CR>
C: sym* 3 <CR>
<CR>
```

This will produce the same result as the previous example except that the point will be added to the data of the plot.

A second extension allows the user to specify the use of the position prompt instead of the cross hairs. This extension is made by appending the letter 'H' after the abbreviated command.

For example:

```
C: size 7 <CR>
C: sym*h 3 <CR>
Input the position (x,y) : 2,1 <CR>
```

This is identical to the previous command except that the position of the added points supplied from the keyboard. The letter 'H' must be at the end of the extended command. 'SYM*H' is a valid command but 'SYMH*' is only recognised as 'SYMH'.

5.2.8.1 Direct Data Entry

The command SYMBOL and its extensions allow data to be entered directly into GRAPH. The following example shows GRAPH being set up so that 10 points can be entered and displayed. The command CRLDAT is used to clear the data array. (section 11.2.2).

```
C: clrdat <CR>
DATA CLEARED
C: set <CR>
LOWER AND UPPER BOUND FOR THE Y AXIS ? 0,10 <CR>
LOWER AND UPPER BOUND FOR THE X AXIS ? 0,10 <CR>
C: box <CR>
C: size 10 <CR>
C: %10 sym*h 3 <CR>
INPUT THE POSITION (X,Y) :
```

The first of the points would be entered and the return key pressed. That point would then be plotted and the next point prompted for. The prompting will continue until all ten points have been entered.

The '%' followed by a one or two digit number is a repeat counter (see section 11.3)

After entering the data it can be replotted, manipulated and stored.

6. HARD COPY COMMANDS

These commands enable and control the use of hard copy devices.

6.1 PLOTTER SELECT COMMANDS.

Each plotter is used, by issuing an appropriate command to GRAPH which switches the output from the terminal to the plotter. A list of the commands and the device to which it is appropriate is as follows:

COMMAND ABBREVIATION -TION

GRAPH	GRA	Tektronix 4662 (4010 compatible)
BAUSCH	BAU	Bausch and Lomb (Houston instruments DM-29)
NATIONAL	NAT	National
ZETA8	Z8	Zeta 8 plotter
HP	HP	Facit (HP-GL compatible)
TEC	TEC	Graphtec
LASER	LAS	Digital LN03 - plus

The Tektronix 4662, the Bausch and Lomb, the Zeta 8 and the Facit plotters are assumed to be inline† plotters. The HP, National and Graphtec plotters are assumed to be connected via the terminal's printer port.

Occasionally there will be the need to create a file which can be used later to produce a hard copy of the plot. For instance, when the user does not have direct access to a plotter or when he wants multiple copies of a complex plot. The following command are used to open an output file for later plotting.

COMMAND	ABBREVIATION
GRAPHFILE	GRAPHF
BAUFILE	BAUF
NATFILE	NATF
ZETA8FILE	Z8F
HPFILE	HPF
TECFILE	TECF
LASER	LAS

Once one of these commands is issued, a file name is requested to which the plot will be sent. This file remains open until the same command is given again. This is true for all commands except for the command LASER, (used to create a plot file suitable for sending to a Digital LN03-plus laser printer), in this case following LASER commands act to toggle the output from the terminal to the file. A command LASERFILE closes the output file.

† Inline means that the plotter is connected to the computer and the terminal is connected to the plotter.

The appropriate command, either GRAPH, BAU, NAT, ZETA8 or TEC, is then used to direct data to the file or the screen after the initial GRAPHFILE, BAUFILE NATFILE, ZETA8FILE or TECFILE command.

Example:

```
C: tec <CR>
GRAPHTEC PLOTTER ON
C: plot test.dat <CR>
C: tec <CR>
C: redraw <CR>
```

In this example the sequence firstly directs the output to the plotter and the plots the data of TEST.DAT and then redirects the output to the terminal. The last command then redraws the plot on the screen.

Another example:

```
C: p test.dat <CR>
C: graphfile <CR>
INPUT THE FILE NAME : output_plot.dat <CR>
HARD COPY PLOT FILE ON
C: redraw <CR>
C: graphfile <CR>
HARD COPY PLOT FILE CLOSED
```

In this example the data of 'test.dat' was plotted to the screen. The Tektronix 4662 plotter was selected (output to a file called 'out_plot.dat') and the data of 'test.dat' was redrawn in Tektronix 4010 graphics language to the output file. The output file was then closed. Later, after leaving the program, the file 'out_plot.dat' can be replayed to the plotter.

6.1.1 CIT161

This command allows GRAPH to handle the CIT-161 terminal. If your terminal is of this type it is suggested that this command is given on running GRAPH.

6.1.2 QUME

The command QUME is used to inform the program that the terminal being used is a QUME. This terminal is VT125 and 4010/4014 compatible and has not quite the same capabilities as other terminals. When using a QUME terminal it is advised to give the command QUME on entering the program.

6.1.3 TAB

This command allows GRAPH to handle the TAB132/15-G terminal correctly. If your terminal is of this type, it is suggested that this command is given on running GRAPH.

6.2 PEN

During plotting, the pen color on any of the hardcopy plotters can be changed using the command PEN. GRAPH will prompt the user for the pen number after the command PEN. Any number between 0 and number of pens available for the chosen plotter is a valid response. A response of 0 returns the present pen to the pen holder.

For example:

```
C: graph <CR>
HARD COPY ON
C: pen <CR>
INPUT THE PEN NUMBER 0-8 : 3 <CR>
```

In this example the plotter is switched on and pen number 3 is picked up by the Tektronix 4662 (option 31) plotter. The PEN command can be abbreviated by adding the pen number to the end of the command.

For example:

```
C: graph <CR>
HARD COPY ON
C: pen 3 <CR>
```

Note: the blank after 'pen' is optional.

6.3 SPEED

For the Tektronix 4662 plotter, the Bausch and Lomb plotter and the Facit (HP GL) plotter, the pen speed can be altered using the command SPEED (SPE). GRAPH, on receiving the command SPEED, prompts for the speed setting, a number between 1 and 8. The numbers correspond to increasing speed. Once a speed command has been given the speed of the plotter remains at that speed until a second speed command is given or the plotter is turned off.

In the case of a Tektronix 4662 plotter (option 31) the speed increases as the square of the setting number. For example the number 1 corresponds to a plotting speed of 1 cm/sec and the number 2 corresponds to a speed of 4 cm/sec. For the Bausch and Lomb plotter the speed doubles with each number and for the Facit plotter the speed increases linearly with the number.

For wet ink plotting a speed setting of 2 is recommended and for the felt pens a setting of 4.

For example:

```
C: speed <CR>
INPUT THE PLOTTING SPEED (1-8) : 2 <CR>
```

This command can be abbreviated by the addition of speed number directly after the command.

For example:

```
C: speed 3 <CR>
```

The space after the command is optional and may be left out.

6.4 FONT

The Tektronix 4662 plotter and the Facit (HP GL) plotter a number of slightly different fonts are available. These can be obtained in the plotter's programmer's reference guide.

For the tektronix 4662 plotter, font number 8 has the advantage that it has no slashes across the zeros. The default font is font 7. On giving the command FONT the program prompts for the users choice of font. There are nine available numbered 0 to 8. (number 7 and 0 are the same). Once set, the font type remains the same until a second FONT command is given or until the plotter is switched off.

For example:

```
C: font <CR>
INPUT THE FONT TYPE : 8 <CR>
```

This can be abbreviated to

```
C: font 8 <CR>
```

The inclusion of the blank after the command is optional.

6.5 ANGLE

The command ANGLE (ANG) allows the text of subsequent WRITE commands to be written vertically on the hardcopy plotters. The command acts to toggle the text from vertical to horizontal.

For example:

```
C: angle <CR>
TEXT ANGLE SET TO 90 DEGREES
C: angle <CR>
TEXT ANGLE SET TO 0 DEGREES
```

6.6 ERASE

The command ERASE (ERA) is used to suppress the clear screen instruction used by GRAPH when a new plot is to be drawn. This instruction allows the display to mimic a hard copy plotter and thus is able to display inserts.

For example :

```
C: era <CR>
various commands
C: plot <CR>
ENTER THE FILE NAME : aaa.dat <CR>
C: mod <CR>          — see section 8.1
various commands
C: plot <CR>
ENTER THE FILE NAME : bbb.dat <CR>
```

The effect is to display both plots on the screen at once.

7. ANALYSIS COMMANDS

7.1 FILE MANIPULATION BY CONSTANT FACTORS

A number of commands allow the user to manipulate the data in a particular plot with simple arithmetic processes. These commands are DIVX, DIVY, MULTX, MULTY, SHIFTX, SHIFTY, DEXPX, DEXPY, DLOGX and DLOGY.

For all of these commands, where a constant is needed, the constant value may be included after the command.

7.1.1 DIVX and DIVY

DIVX and DIVY are used when the data along the X-axis or Y-axis is to be divided by a constant value. The screen is erased and the rescaled data is replotted.

7.1.2 MULTX and MULTY

MULTX and MULTY are used when the data along the X-axis or Y-axis is to be multiplied by a constant value. The screen is erased and the rescaled data is replotted.

7.1.3 SHIFTX and SHIFTY

SHIFTX and SHIFTY are used when a constant value is to be added to the data along the X-axis or Y-axis.

For example:

The file, 'data.dat' contains information in non-metric units along both axes. The temperature is in degrees Fahrenheit along the Y-axis and distance in inches is along the X-axis. The file needs to be converted into metric units.

i.e. Fahrenheit to Celsius conversion

$$C = (F - 32) * 5/9$$

and Inches to centimetres conversion

$$cm = ins * 2.54.$$

To convert the plot the following procedure is followed.

```
C: plot <CR>
INPUT THE FILE NAME : data.dat <CR>
C: shifty <CR>
SHIFT OF ? -32 <CR>
C: divy <CR>
INPUT THE FACTOR ? 9 <CR>
C: multy <CR>
INPUT THE FACTOR ? 5 <CR>
C: multx <CR>
INPUT THE FACTOR ? 2.54 <CR>
C: store <CR> — section 11.5
INPUT THE FILE : converted.dat <CR>
```

This example may be abbreviated to :

```
C: plot data.dat <CR>
C: shifty -32 <CR>
C: divy 9 <CR>
C: multy 5 <CR>
C: multx 2.54 <CR>
C: store converted.dat <CR>
```

— see section 11.5

7.1.4 DEXP

The commands, DEXPY and DEXPX are used when the data along the Y-axis or X-axis is to be altered so that the new data in that direction is now ten to the power of the old data.

That is,

$$\text{new data} = 10^{\text{old data}}$$

The screen is erased and the new data is replotted.

7.1.5 DLOG

DLOGX and DLOGY are the inverse functions of DEXPX and DEXPY. Using the commands DLOGX and DLOGY, the user can plot the logarithm, base 10, of the data. In each of these commands the data in the direction of the axis included in the command is converted to its logarithm base 10.

For example:

```
C: DLOG <CR>
```

In this case the data in the Y direction is converted to its logarithm base 10, the screen is erased and the data is replotted.

$$\text{new data} = \log_{10}(\text{old data})$$

Together the commands DEXP and DLOG allow the user to take the square root (or third, or any other power) of the data.

For example:

```
C: plot test.dat <CR>
C: dlogy <CR>
C: div 2 <CR>
C: multy 2 <CR>
```

In the resultant plot, each y value of the data, is the square root of the original y value.

7.2 FILE MANIPULATION BY OTHER FILES

Three commands allow the user to perform file operations with the data of two plots. These commands are FMULTIPLY (FMULT), FDIVISION (FDIV) and FADDITION (FADD).

7.2.1 FMULTIPLY

The command FMULTIPLY (FMULT) allows the user to multiply files together. The Y values of the first plot are multiplied with the corresponding Y values of the second plot and the product is plotted. If the X values of the two plots are not the same GRAPH interpolates the data of the first plot so as to match the X values of the second plot. The method of interpolation is described under INTERPOLATE and is of second order.

The first of the two files to be multiplied together should be plotted before the command FMULTIPLY is given. GRAPH then prompts for the second file name.

Example:

```
C: plot <CR>
INPUT THE FILE NAME : test.dat <CR>
C: fmultiply <CR>
INPUT THE FILE NAME : test2.dat <CR>
```

In this example the data of 'test.dat' is multiplied by the data of 'test2.dat' and the result plotted. Plot files in multiple plot files are accessed by appending their position within the plot file to the command.

For example:

```
C: plot <CR>
INPUT THE FILE NAME : test.dat <CR>
C: fmult 4 <CR>
INPUT THE FILE NAME : test2.dat <CR>
```

In this example the data of 'test.dat' is multiplied by the data of the fourth plot in the file 'test2.dat' and the result plotted.

The command can be further abbreviated by appending the file name after the command.

For example:

```
C: fmultiply 4 test2.dat <CR>
```

7.2.2 FDIVIDE

FDIVIDE (FDIV) allows the user to divide the Y values of one plot by the Y values of another. If the X values of the two plots are not the same, GRAPH interpolates the data of the first plot so that the interpolated data has X values that match the X values of the second plot.

The syntax and use of the command is the same as FMULTIPLY.

7.2.3 FADDITION

FADDITION (FADD) allows the user to add the Y values of two plots together. The result of the addition is then plotted. If the X values of the two plots are not the same, GRAPH interpolates the data of the first plot so that the interpolated data has X values that match the X values of the second plot.

The syntax and use of the command is the same as FMULTIPLY.

7.3 Integration and Differentiation

The data of a plot may be differentiated or integrated using the commands DIFFER (DIF) and INTEGRATE (INT). Second or higher differentials are obtained by repeated application of DIFFER.

7.3.1 INTEGRATE

The command INTEGRATE (INT) causes the data of a plot to be integrated and redrawn. The data should be sorted so that the X values are in ascending order. This can be done with the command SORT (section 7.6.6) before using INTEGRATE. The integration is carried out using Simpsons rule. Each point on the plot is the integral of the data from lower limit of the X values up to that point.

7.3.2 DIFFERENTIATE

The data can be differentiated using the command DIFFER (DIF). Two differentiation techniques are used to differentiate the data depending on if the data has equally spaced X values or not.

In the case of evenly spaced X values the differentiation is carried out using a parabolic least squares technique based on three points either side of the point being evaluated as described by Savitzky et al. in *Analytical Chemistry* **36**, 1627 (1964). The number of points involved in differentiation can be altered by using the abbreviated command DIF and appending after the command the number of points either side of the point of interest to be considered. (between 1 and 9 see below).

When the data has unevenly spaced a simple three point differentiation technique is used. This method consists of fitting a parabola at each X value and differentiating that parabola. The parabolas are fitted to three data points only; that is, the point being evaluated and the two points on either side of it.

7.4 SMOOTHING AND INTERPOLATION

The data of a plot can be both interpolated and smoothed using the command INTERPOLATE and SMOOTH

7.4.1 SMOOTH

Smoothing of data is achieved using the command SMOOTH (SM). This command uses the least squares smoothing technique described by Savitzky et al. in Analytical Chemistry 36, 1627 (1964). involving 3 points either side of the point of interest. (seven point in all) The smoothed curve is plotted over the original data.

As an extension to this command, the number of points either side of the point of interest can altered by appending a number to the abbreviated command. The number, between 1 and 9, corresponds to the number of points on each side of the point of interest.

For example:

```
C: SM4 <CR>
```

In this example the plotted data will be smoothed using 11 point smoothing, 5 points on either side of the point of interest.

As a further extension a repeat count, '#', (between 1 and 9) can be added to the end of the command. This repeats the smoothing operation # times before plotting the smoothed data. A '*' must be used between the command and the repeat count.

For example:

```
C: SM4*5 <CR>
```

This instructs GRAPH to smooth the data using an 11 point smoothing technique five times before overlaying the smoothed data.

7.4.2 INTERPOLATE

To create a smooth curve where the data in a plot is sparse may require the data to be interpolated. The command INTERPOLATE (INTERP) is used to interpolate the data displayed in the plot and produce a curve consisting of equally spaced points across the width of the plot. This is overlaid on the existing plot. The number of new points is requested after the command is issued. Prior sorting of non-sequential X values using the SORT command is recommended.

The method used to interpolate each point is as follows;

1. Parabolas are fitted to every three sequential points in the original data.
2. The two parabolas formed from the three points whose central points are on either side of the interpolated point are evaluated at the interpolated point.
3. A weighted average of the two values is then the interpolated value. The weighting is determined by the distances from the two data points either side of the point being interpolated.

Example:

```
C: plot <CR>  
INPUT THE FILE NAME : test.dat <CR>  
C: interpolate <CR>  
HOW MANY NEW POINTS ? 100 <CR>
```

Here the data of the file 'test.dat' is plotted and a curve consisting of 100 interpolated points is overlaid on the plot.

The command can be abbreviated by including the number of new points required after the command. The above example would then become:

```
C: plot test.dat <CR>
C: interp 100 <CR>
```

7.5 LINE FITTING

A Least squares polynomial of order up to ten can be fitted to the data in a plot with the command PFIT and a more general function can be fitted with the command FFIT.

7.5.1 PFIT

This command requests that a least-squares-fitted polynomial be fitted to the plot data. The order of the polynomial is given by the number '#' and is restricted to be in the range 1 to 10. GRAPH responds to this command by first prompting for the domain over which the fit is to be made. A 200 point line is then drawn on the plot over the region specified and the coefficients, normal errors and the correlation coefficient are displayed on the screen. A good fit to the data will have small normal errors and a correlation coefficient close to 1.

The least squares fitting is made by the Gauss-Jordan method of matrix inversion.

In the following example the data of 'test.dat' is fitted with a polynomial over the domain 3 to 8.

```
C: plot <CR>
INPUT THE FILE NAME : test.dat <CR>
C: pfit 2 <CR>
INPUT THE LIMITS : 3,8 <CR>
FITTED POLYNOMIAL :
coef. X**2 : -0.5000 +/- 0.0
coef. X**1 : 8.5000 +/- 0.0
coef. X**0 : -8.0000 +/- 0.0
THE CORRELATION COEFF. is = 1.0000000E+00
```

7.5.1.1 PFIT #*

The inclusion of a '*' after the order of the polynomial tells GRAPH to retain the polynomial points as the new data points. Most other commands (e.g. SMOOTH, INTERPOLATE) automatically modify the data set.

7.5.1.2 PFIT #H

The inclusion of an 'H' after the order of the polynomial toggles on a flag which will tell GRAPH that the user wishes to create a file containing the information displayed on the screen (coefficients etc). The first time the 'H' is included in the command, a file name is requested and the displayed information is stored in that file. A default name, 'COEF.DAT' is used if no file name is offered after the prompt. Subsequent PFIT#H commands add to the file. A second PFIT#H command closes the file.

7.5.2 FFIT

The command FFIT is used to fit an algebraic function to previously plotted data. When this command is given, a domain over which a function is to be fitted requested. After entering the domain, the program prompts for an algebraic function.

The algebraic function (similar to a line of FORTRAN or BASIC) can include :

Functions

SIN()	- sine of the angle
COS()	- cosine of the angle
TAN()	- tangent of the angle
LOG()	- logarithm base 10
EXP()	- exponent base e
ABS()	- absolute magnitude
SQR()	- square root
INT()	- integer truncation

Variables - A to F, which are to optimised

Numeric constants - numbers

Algebraic operations + - / * ** (or ^)

Brackets ()

The inclusion of brackets is essential as the program treats all algebraic operations as having the same priority.

The function is expressed in terms of x, the horizontal axis. Some example functions acceptable to GRAPH are :

```
log( x ) * sin( a*x ) + b
a*exp(-((x-b)^ 2/c)) + (b*(exp(-(x-e)^ 2/d))
a*x*x + (b*x) + c
```

As GRAPH treats the operations as having the same priority, the bracket included in the third example is necessary. If it were left out the function would be translated as :

$$(a \times x^2 + b) \times x + c.$$

Once the function has been entered the initial values of each of the variables and the maximum number of iterations are prompted for.

The method used to fit the function involves the minimisation of the root mean square error (RMS)(effectively the sum of the geometric distances of the data points from the function) by varying each variable in turn finding a minimum RMS error, then rotating the order of the variables and repeating the process. Naturally with a large number of variables there could be a number of local minima in the RMS error and the minimization technique may proceed towards one of these. Therefore a careful and an appropriate choice of the starting values of the variables should be made to ensure the best fit.

The iteration number, the RMS error and the current state of the fitted function are displayed between iterations. The rate of reduction of the RMS error between iterations will indicate whether it is necessary to continue with more iterations.

When the number of iterations is exceeded then the program allows the fitting process to be continued by adding more iterations, started again with new starting values for the variables or stopped and the fitted function plotted. The fitted function is plotted as 300 points thus; if point mode has been previously selected the program will ask if you wish

to continue. If the selection of point mode is not required then answer no and remove the selection of point mode using the command POINT (section 4.4) and then plot the function using the command SAME (section 3.6).

7.6 AUXILIARY MANIPULATION

7.6.1 CHANNEL

The command, CHANNEL (CH) allows the user to use the cross hair mode of the terminal to position the cross hair over a feature of interest and return the co-ordinate of the cross hair.

The procedure is as follows;

```
C: channel <CR>
(arrow keys)
<CR>
```

The terminal is put into cross hairs mode. The cross hairs are positioned using the arrow keys. Once in position the carriage return key is pressed and the co-ordinate of the cross hair is displayed. The co-ordinate of the closest data point to the cross hair is also displayed.

The result will appear like this;

```
X = 1.00000E3 Y = 2.95000E1
( closest point x = 980.0 , y = 29.5 )
```

7.6.2 LINE

An arbitrary line can be placed anywhere on the terminal screen using the command LINE (L). If the command LINE is given while plotting to the terminal screen, the terminal is put into the cross hairs mode. The arrow keys are used to position the cross hairs over any point on the proposed line and the return key is pressed when the cross hairs are in position.

After the first point has been chosen the terminal is again put into the cross hairs mode so that in the same way a second point on the proposed line can be defined. After the return key is pressed the line is drawn and the slope and intercept of the line are shown in the top right hand corner of the screen.

To use this command on a hard copy device the line is defined by answering the prompts for the slope and intercept. The line is then drawn on the plot. Unlike the screen version of LINE the slope and intercept are not included on the plot.

7.6.3 ADDP

The addition of a polynomial to the data of a plot can be accomplished using the command ADDP. On the command ADDP, the order of the polynomial is requested followed by the polynomial coefficients. After entering all the coefficients the polynomial is evaluated at the each of the X values of the plotted data, and the result added to the original data. The plot is then redrawn.

For example:

```
C: plot <CR>
INPUT THE FILE NAME : test.dat <CR>
C: addp <CR>
INPUT THE ORDER OF THE POLYNOMIAL : 2 <CR>
COEFF. FOR X**2 : 2.5 <CR>
COEFF. FOR X**1 : -1 <CR>
COEFF. FOR X**0 : 5 <CR>
```

In this example the parabola $2.5X^2 - X + 5$ would be added to the data points of the file 'test.dat'.

7.6.4 APPEND

The command APPEND adds the data of the present plot to the end of the data of a second plot. On being given the command APPEND (APP), the file name of the second plot is requested. On receiving the file name, the screen is cleared and the combined data are plotted.

When the data of a multiple plot file is to be appended, the position of the plot is included after the command APP.

For example:

```
C: plot <CR>
INPUT THE FILE NAME : test.dat <CR>
C: app 3 <CR>
INPUT THE FILE NAME : test2.dat <CR>
```

In this example the data of 'test.dat' are plotted then the data of the third plot in the multiple plot file 'test2.dat' are plotted with the data of 'test.dat' appended to it. The command sequence can be abbreviated by including the plot file name, to be appended, after the command.

For example :

```
C: plot test.dat <CR>
C: app 3 test2.dat <CR>
```

This would produce the same result as in the previous example.

7.6.5 INVERT

The user can obtain the reciprocal of the data points by using either the command INVERT (INV) or INVERTX (INVX). The command INVERT results in the reciprocal the y axis values being plotted. The x axis data is inverted using INVERTX.

Division by zero is avoided by setting the result to 10^{20} .

7.6.6 SORT

If the data is fed into the plot file in a random order, the data may require sorting before plotting. The command SORT causes the data to be sorted into increasing X values using a simple bubble sort. Sorting before interpolation is essential.

For example:

```
C: plot <CR>
INPUT THE FILE NAME : test.dat <cr>
C: sort <CR>
C: redraw <CR>
```

In this example the data of 'test.dat' was first plotted then sorted and redrawn.

7.6.7 SWAP

Occasionally the need to exchange the X and Y axes will arise. The command SWAP exchanges all the X data with the appropriate Y data. This command can be useful when the X data of one plot is to be compared with Y data of a second plot.

For example:

```
C: plot <CR>
INPUT THE FILE NAME: test.dat <CR>
C: swap <CR>
C: yycompare <CR> — section 7.6.9
INPUT THE FILE NAME: test1.dat <CR>
```

In this example the Y values of 'test.dat' are exchanged with the X values. The Y values of 'test1.dat' are then plotted against the original X values of 'test.dat'.

7.6.8 VALUE

The X and Y values associated with the #th data point within a plot file can be ascertained using the command V#. Only positions between 1 and 999 are allowed.

```
C: plot <CR>
INPUT THE FILE NAME : test.dat <CR>
C: v5 <CR>
```

VALUE = -0.30000 AT -1.00000

7.6.9 YYCOMPARE

The user may wish to compare the data of two different plots by plotting the Y values of one plot against the Y values of a second plot. To do this the user plots one of the two plots on the screen using a PLOT or OVERLAY command, whichever is appropriate. Then the user issues the command YYCOMPARE (YYC). The file name of the second file is then prompted for. The Y values of the second plot are then plotted against the appropriate Y values of the first plot.

Where the X values are different in the two plots the Y values of the first plot are interpolated to give Y values at the X values of the second plot. Multiple valued X points are averaged before being interpolated.

A multiple plot file can be plotted by using the abbreviated form of the command, YYC, with the number corresponding to the position of the plot in the multiple plot file.

For example if the user wished to compare Y values of the second plot in the multiple plot file 'test2.dat' with the Y values of the file 'test.dat' the following command sequence would suffice;

```
C: plot <CR>
INPUT THE FILE NAME : test.dat <CR>
C: yyc2 <CR>
INPUT THE FILE NAME : test2.dat <CR>
```

7.6.10 FUNCTION

The command FUNCTION (FUN) allows the user to plot simple algebraic functions. On receiving the command FUNCTION, a domain is prompted for over which the function is to be plotted. Once a suitable domain has been entered, the function is prompted for as an algebraic expression in terms of x.

For example:

```
C: function <CR>
INPUT THE DOMAIN OF THE FUNCTION: 0,10 <CR>
INPUT THE FUNCTION (y = F(x))
e.g. y = 5*x + 100*x**2 - 1/x
y = 4*x*x - 3*x - 2
C:
```

In this example the function $y = 4x^2 - 3x - 2$ was plotted over the domain 0 to 10. 300 points were generated. The form and scope of the allowed functions are set out in section 7.5.2

7.6.11 FOURIER

This command allows the user to perform a discrete fourier transform on the present plot data. The data is replaced by the coefficients.

The coefficients are ordered, a2, a1, a0, b0, b1, b2..... where the b coefficients are for the cos terms and the a coefficients are for the sin terms. Repeating the command inverts the transform. The number of fourier terms required can be included on the end of the command. The default number is 20 or half the number of points (which ever is smaller).

When used in conjunction with the command REMOVE (section 11.7), fixed frequencies may be removed from the data.

8. LAYOUT UTILITY COMMAND

The layout utility command, MODIFY, is included in GRAPH to allow the user to change the layout of the plot.

8.1 MODIFY

This is a utility built into GRAPH to allow the user some freedom to alter the layout of the plots. For example, aspect ratio, title and subtitle position can be altered.

On the command MODIFY (MOD), GRAPH clears the screen and displays the positional data relevant to the present plot settings. This will vary depending on what the user is doing. Each plotter and letter size has slightly different positional layout.

The display after the command, MODIFY, will look like ;

C: modify <CR>

***** MODIFICATION *****

- | | |
|------------------------|---------------------------|
| 1. LETTER SIZE = 1 | 2. PLOT TYPE : SCREEN |
| 3. XMIN = 140. | 4. XMAX = 1023. |
| 5. YMIN = 50. | 6. YMAX = 779. |
| 7. TITLE HEIGHT = 742. | 8. SUBTITLE HEIGHT = 705. |
| 9. DEFAULT SETTINGS | 0. RETURN |

ENTER NUMBER :

The user indicates which variable is to be altered by entering a number between 0 and 9. A response of '1' indicates the letter size is to be altered. GRAPH then prompts the user for the desired size. (1,2 or 3). On responding GRAPH then corrects the modification table for that letter size and plot type.

A response of '2' allows the user to change the plot type from that shown to any one of four. GRAPH prompts for the number corresponding to the plot type required. After the choice is entered the table is altered to the table for that particular letter size and plot size.

A response of any number between '3' and '8' indicates the user wishes to modify that particular parameter for that plot type and letter size. Graph then prompts for the new value to be inserted into the table. XMAX and XMIN have the range 0 to 1023. All the others have the range 0 to 779.

9. COMMAND FILES

The repetition of a large number of commands in the setting up of particular plot can be avoided by the use of command files. Command files, which consist of lists of GRAPH commands and their responses, are defined by the user. Each command or response should be in separate lines except when -AND- is used (See section 11.4). A comment can be included after the command or any numerical response and is indicated by an exclamation mark. Do not write a comment on a line with a text response as GRAPH could include part of it in the response. Up to 99 of these files may be created, called 'GRAPH1.CMD' to 'GRAPH99.CMD'. Command files can be created either while running GRAPH and using the command *CR# or by using EDT (or any other text editor).

A number of commands are included in GRAPH which enable the use of these files. All of these commands start with an asterisk. They are;

- *GO
- *CREATE
- *STOP
- *CONTINUE
- *HALT

9.1 *GO#

This command causes GRAPH to read and execute the commands and responses contained in the file 'graph#.cmd'. The '#' is a number between 1 and 99. If the number is not included in the command GRAPH reads the file 'graph1.cmd'.

9.2 *CR#

The command files can be created from within GRAPH using the command *CR#. With this command, commands and their responses are written sequentially to the file 'graph#.cmd'. No editing facilities are included so care must be taken in writing each command and its response. To exit from this command type END or EOF.

9.3 *HALT

The command *HALT (*H), when included in a command file, causes GRAPH to temporarily leave the command file and allow the user to enter commands and responses interactively. This command can only be included in a command file after all required responses to the previous command have been given, except when GRAPH requires a file name. In that case GRAPH will enter interactive mode to get the file name.

9.4 *CONTINUE

After the command *HALT has been given in a command file, GRAPH remains interactive until a *CONTINUE (*C) is given. GRAPH returns to the command file at the command directly after the previous *HALT command.

9.5 *STOP

To abort a command file cleanly after a *HALT so that the user can use a second command file the user can use the command *STOP which closes the file and removes any flags set.

9.6 Example Command Files

Two examples of command files are given. In both examples a command file is shown that could be used by a user several times.

In the first example, a user wants to compare a number of sets of experimental data with a theoretical curve. A command file, shown below, called 'graph21.cmd' was set up to aid with the comparison.

The command file sets up the axes, sets point plotting on and enters interactive mode so the user can enter a file name. After entering the file name and the command *CONTINUE, the command file sets dashed line plotting and overlays the file 'theory.dat'.

SET	! set the X and Y axis
0,10	! Y axis 0 to 10
0,15	! X axis 0 to 15
XSPACING 2	! X axis marker spacing
YSPACING 2	! Y axis marker spacing
POINT	! point mode set
PLOT	! plot file
*HALT	! go interactive for file name
POINT	! line mode set
DASH	! dashed lines
OVERLAY THEORY.DAT	! overlay file

The inclusion of comments in a command file is allowed and is indicated by the exclamation mark. All characters after the exclamation mark are regarded as being in the comment.

To run this command file the user would issue the following commands:

```
C: *go21 <CR>
INPUT THE FILE NAME : expt1.dat <CR>
C: *cont<CR>
```

The second example command file can be used to create a fifteen lined screen for use in the production of overhead projections (VU-graphs). The limits on the usable area of the screen are removed using the MODIFY utility command for letter sizes 1 and 2 and for both the screen and the Tektronix 4662 plotter. The numbers are removed off each axis and the axis labels are cleared. Grid plotting is set. The X-axis spacing is set to equal to full width of the screen so that no vertical lines will appear in the box.

set	!set up page size
0 15	!15 lines
0 10	!arbitrary width
modify	!modify parameters of screen plot
3	!xmin set
1	! to 1
5	!ymin set
1	! to 1
1	! change letter size
2	! to 2
3	!xmin set
1	! to 1
5	!ymin set
1	! to 1
2	! change plot type
2	! to plotter
3	!xmin set
1	! to 1
5	!ymin set
1	! to 1
1	!change letter size
1	! to 1
3	!xmin set
1	! to 1
5	!ymin set
1	! to 1
2	! change plot type
1	! to screen
0	! return
nnumber	! switch off numbers
xlabel	! blank xlabel
ylabel	! blank ylabel
grid	!draw lines on
xspacing	!show no vertical lines
10	
yspacing	!show 15 horizontal lines
1	
box	!draw it

If this command file is stored as 'graph1.cmd'. The user could use this command file in the following way. On running GRAPH the user would type *go1. GRAPH would then execute the command file and finish by displaying a screen with fifteen horizontal lines. The user would now choose between letter size 1 and 2 for the first line of text, using the command LETTER and then issue the command WRITE. After positioning the text the user issues the command SWRITE to create a command file.(see section 6.2.7) The user may now choose the pen number for the second line of text using the command PEN. (The pen number for the first line of text is always set to 1). Having done this the user repeats the sequence of WRITE and SWRITE commands until all the text of the VU-graph is on the

screen. The commands PEN and LETTER can be used at the appropriate time to change pens or letter size.

When finished, close the command file with the CWRITE command.

To view the VU-graph use the command DATA to clear the screen (no data has been entered and then issue the *GO command with the appropriate number appended to it.

At a later session when reentering GRAPH the previously developed VU-graph can be drawn by firstly using the command *GO1 to set up the scales and page sizes. Then the command *GO# is issued to write the text. (# is the command file number of the VU-graph.)

10. INFORMATIVE COMMANDS

10.1 STATUS

This command (abbreviation STA) results in the program displaying on the screen information concerning the type of plot set, the size of the plot, the file name last used and which plot in the file was last displayed. The state of a number of the toggles in GRAPH are also shown.

10.2 HELP

HELP (H) is a command that provides the user with a list of the commands and their uses. The command results in a single screen of information being displayed at a time. GRAPH prompts for a continuation before displaying a new screen.

10.3 DIRECTORY

DIR is a command which allows the user to examine the contents of a directory without leaving the program. The command is like the DCL command. A directory search for a particular file name can be done by including the file name after the command. For example:

```
C: dir [fred]test.dat
TEST.DAT;1
1 FILE IN DIRECTORY [FRED]
```

Wild cards and directory specifications can also be included optionally. If no file is specified after the command, all the files from the default directory with the extension '.dat' will be listed.

11. OTHER COMMANDS

11.1 EXPAND and E-

After plotting data on the screen it is possible to expand a section of the plot using the command EXPAND (E). The region to be expanded is defined by the positioning of cross hairs. When the command is issued, the terminal enters its internal cross hair mode and displays the cross hairs. The arrow keys are used to position the cross hairs to define any corner of the region to be expanded. The carriage return key is pressed when the cross hairs

are in position. At this point two lines will be drawn over the cross hairs. After the lines have been drawn, the cross hairs are moved to define the diagonally opposite corner of the region to be expanded. When the carriage return entered this time, the data is redrawn showing only the expanded region.

The EXPAND command has a number of short cuts. If the cross hairs are positioned to define a corner of the expansion region and the return is hit twice, the region that will be displayed is the region defined by the bottom left hand corner of the plot and the cross hairs shown.

If, after defining the first position and pressing the carriage return key, only one of the cross hairs is subsequently moved, the region redrawn is defined by the moved cross hair and the two drawn lines and the left or bottom axis. (whichever is appropriate.)

E- is the command to undo the EXPAND.

11.2 CLEAR

11.2.1 CLR

The screen can be cleared at any time using the command CLR. This command does not clear the data array.

11.2.2 CLRDAT

The data array in GRAPH can be cleared using the command CLRDAT. This command zeros the Y data array and sets the number of points to zero. The X data array is set up so that the array contains integers in ascending order.

Doing this to the X array allows the program to display polynomials. A sequence of commands similar to the following commands can be used :

```
C: clrdat <CR>
C: top <CR>
INPUT THE TOP CHANNEL NUMBER : 100 <CR>
C: divx <CR>
INPUT THE FACTOR : 10 <CR>
ERROR 5 — YMAX IS SMALLER OR EQUAL TO YMIN
C: addp <CR>
INPUT THE ORDER OF THE POLYNOMIAL : 2 <CR>
COEFF> FOR X**2 : 2.5 <CR>
COEFF> FOR X**1 : -1 <CR>
COEFF> FOR X**0 : 5 <CR>
```

In this example the data arrays are cleared of previous data. The top channel is set to 100. The integer values assigned by the CRLDAT command to the X array are divided by 10. This reduces the domain over which the polynomial is going to be plotted from 1 to 100 down to 0.1 to 10. The error message that occurs can be ignored; it arises because the Y data is all zero and no scale can be given to the Y data, and can be avoided using a prior SET command. The final command and responses add the polynomial $2.5X^2 - X + 5$ to the Y data. A plot is then drawn displaying the polynomial over the domain 0.1 to 10.

11.2.3 INITIALIZE

The command INITIALIZE (INIT) allows the user to restore the plot settings back to their default values. This command enables command files to be used without the problem of having to reset all the switches. It can be included in a command file as the first command so that the same initial condition exists whenever the command file is used.

```
C: init  
INITIALIZED
```

The screen is cleared and the data of the previous plot is removed

11.3 % command repeat

The user may wish to have a number of plots on the same axis all of which are in the same file. This would require the command CONTINUE to be typed a number of times. To avoid this a repeat counter is included in the program. The repeat count is indicated by a percent sign preceding the command to be repeated. A repeat count of between 1 and 99 is included after the percent sign.

For example:

```
C: plot <CR>  
INPUT THE FILE : test.dat <CR>  
C: %6continue <CR>
```

In this example the data in the first plot in the multiple plot file 'test.dat' would be plotted onto the screen. Then the next six plots in the file would be overlaid.

11.4 -AND-

Compound commands, consisting of a number of graph commands can be formed using the -AND- connecting word. A number of GRAPH commands would then occupy one line. The main use of compound commands arises when it is not necessary to display intermediate plots. For example:

```
C: shiftx 8 -and- multy 20.5 -and- divx 3
```

In this example the data is shifted 8 units to the right and then the y values multiplied by 20.5. Lastly the x values are divided by 3. It is only after this last process that the data is replotted.

Note: This could have been written as ;

```
C: shiftx -and- multy -and- divx
```

The required responses are prompted for before the plot is drawn.

11.5 STORE

The user can store the data and axes labels of a plot using the command STORE (STO). On receiving this command GRAPH prompts for a file name. A response of a carriage return will either create a file called 'gstore.dat' or append the data of the plot to the previously named file. GRAPH remembers the file name used in response to the last STORE command.

For example :

```
C: store <CR>
INPUT THE FILE NAME [my_directory.sub_directory]temp.dat <CR>
```

The file name can be included after the command. The above example then becomes :

```
C: store [my_directory.sub_directory]temp.dat <CR>
```

11.6 DEFAULT

When using a command that requires a data file name and the user simply wants to use the previous filename. The filename can be replaced by the word DEFAULT (or DEF) and the previous filename will be used.

For example:

```
C: p test <CR>
C: clr <CR>
C: p default <CR>
```

In this example the data of test.dat will be plotted, the screen cleared and the data of test.dat replotted. This is particularly useful in command files.

11.7 REMOVE

The command REMOVE allows the user to remove a single point from a plot. The position of the point in the plot (the point number) is expected after the command. If the point number is not included after the command, GRAPH displays the terminal's cross hairs. The user can then position the crosshairs on to the point he or she wishes to remove, using the arrow keys. Once in position any other key may be hit and the point GRAPH finds closest to the position of the crosshairs will be removed.

Example:

```
C: remove 10 <CR>
POINT ( X,Y ) REMOVED
```

In this example the 10th point in the plot is removed. The values of X and Y are displayed.

11.8 RETAIN and RECALL

These commands allow the data which GRAPH has in memory to be copied into temporary store and recalled at a later time during the current GRAPH session.

For example:

```
C: plot test.dat <CR>
C: overlay test1.dat <CR>
C: retain <CR>
C: plot test2.dat <CR>
C: recall <CR>
DATA RECALLED
C: same <CR>
```

11.9 FINISH

The command FINISH (F) is provided to stop the program and clear the screen. If a plotter is connected the program replaces the pen before finishing. To clear the screen when exiting from the program, the abbreviated command 'F' is used. Conversely, to retain the screen image, the full command is issued. All open files are also closed on exiting.

11.10 UNKNOWN COMMANDS

Rather than give an error immediately after receiving a command that it does not understand, GRAPH, gives a warning and creates a sub-process to try and execute a DEC command line (or DCL command). DEC command lines can be selected directly, where there is no conflict with GRAPH commands, and where there is conflict a space before the command forces the use of the DCL command.

Thus it is possible to run a second program or execute a DCL program from a GRAPH command file or interactively whilst executing GRAPH.

example:

```
C: run program <CR>
WARNING — TRYING DCL COMMAND
```

This will execute the program 'PROGRAM.EXE' whilst still in GRAPH.

12. ERROR MESSAGES

READ ERROR 1 — DURING INPUT OF DATA

This error will occur if GRAPH receives alphanumeric characters when expecting a number during the reading of a plot file.

ERROR 2 — END OF FILE ENCOUNTERED DURING READ

This occurs when there is no data left in the file and an attempt is made to read.

ERROR 3 — FILE LENGTH ERROR — ONE POINT

NUMBER OF POINTS WANTED - ***

TITLE - *****

XLABEL - *****

YLABEL - *****

The asterisks represent numbers which correspond to the data translated. This message normally appears when there is only one point to plot, since the axes cannot be scaled.

ERROR 4 - FILE NOT FOUND - ATTEMPTED FILE NAME WAS ***** ***

This message is given when there is an error in the file name or the file does not exist in that directory. The correct file name can then be entered. After three attempts GRAPH defaults to the command entry point.

ERROR 5 - YMAX IS SMALLER OR EQUAL TO YMIN

This normally occurs when the data file has all values equal to zero.

ERROR 6 - XMAX IS SMALLER OR EQUAL TO XMIN

This will occur if all data points in a file have the same X values and the plot is plotted without a prior SET command

ERROR 7 IN FILE - LAST PLOT FOUND WAS # ***

TITLE ***** ***

HIT < return > TO CONTINUE

An attempt to overlay a plot from a multiple plot file was made beyond the last plot in that file.

ERROR 8 — NO DATA TO DRAW

There is no data to plot. The number of points requested to be plotted is zero.

ERROR 9 — NO PREVIOUS PLOT OR SET COMMAND

Use of OVERLAY requires either setting up the axes using the SET command or a prior PLOT command.

ERROR 10 — NO FILE ATTACHED

Use of NEXT requires a previous PLOT or OVERLAY command.

ERROR 11 — ATTEMPT TO REDRAW WITH NO DATA FILE

An attempt has been made to redraw a plot with no data. A previous PLOT or OVERLAY command is required.

ERROR 12 — NO DATA

A previous PLOT or OVERLAY command is required.

ERROR 13 — PLOTTER NOT ON

The FONT command is not supported by all plotters. It is supported on the Tektronix 4662 plotter and the Facit plotter (HP). The appropriate plotter must be selected before the FONT command is given.

ERROR 14 — INVALID PLOT TYPE

The command SPEED is not available when plotting to the screen or the national plotter.

ERROR 15 — NO PLOT

The command requires a prior SET or PLOT command.

ERROR 16 — FILE GRAPH#.CMD COULD NOT BE OPENED

When using the command SWRI the program opens a new file GRAPH#.CMD, where '#' represents a number between 1 and 99. If there is a protected version of this file and three other versions of the file GRAPH will fail to open the file and this error will occur.

ERROR 17 — ATTEMPT TO FREEZE WITH NO SCALE

The FREEZE command requires a previous PLOT or SET command. GRAPH looks for a non-zero scale before setting the axes.

ERROR 18 — NO COMMAND FILE ON

An attempt was made to continue a command file that is not attached.

ERROR 19 — USE EITHER *GO, *CREATE, *STOP

The command started with '*' but was not a valid command.

ERROR 20 — LOOP COUNT ERROR

There is no loop count number directly after the '%' sign.

ERROR 21 — NO HARD COPY - NO PEN TO CHANGE

Before a pen can be selected the relevant plotter toggle must be active. See chapter 6.

ERROR 22 — IN OPENING FILE

GRAPH could not open the file for the plotter. The file name may not be compatible with the operating system or the file protection may be stopping GRAPH from creating a new version.

ERROR 23 — NO DATA POINTS

The number of points in the plot is zero. The command requires data on which to act.

ERROR 24 — FILE NOT OPEN - CANNOT REWIND

GRAPH cannot find the correct file to rewind.

ERROR 25 — NO DATA TO SHIFT

An attempt was made to add a constant to nonexistent data.

ERROR 26 — FAILED TO OPEN FILE

GRAPH could not open the file for the plotter. Check the validity of the file name and its protection if it already exists.

ERROR 28 — TOO SMALL OR NEGATIVE (SET TO 10)

The lengths of both the dashes and spaces must be positive numbers greater than 1 otherwise GRAPH sets the lengths to 10.

ERROR 30 — LOWER LIMIT GREATER THAN UPPER LIMIT

The values of XMAX and YMAX must be larger than the values of XMIN and YMIN respectively.

ERROR 31 — NO DATA AVAILABLE

The command requires a prior PLOT, OVERLAY or FUNCTION command to provide data on which to act.

ERROR 32 — ATTEMPT TO EXPAND WITH NO DATA

GRAPH requires existing scales to preform the expansion.

ERROR 33 — ATTEMPT TO INTERPOLATE WITH NO DATA

The command requires a prior PLOT, OVERLAY or FUNCTION command to provide data on which to act.

ERROR 34 — NO DATA TO INTEGRATE

The command requires a prior PLOT, OVERLAY or FUNCTION command to provide data on which to act.

ERROR 35 — NO PLOT TO CONTINUE

An attempt to continue plotting a multiple plot file without having first called the file using a PLOT or OVERLAY command.

ERROR 37 — OUT OF DATA — (END OF FILE ENCOUNTERED)

An attempt was made to use either the commands CONTINUE, NEXT or REDRAW beyond the last plot in the file or alternatively the file is empty. (No data).

ERROR 40 — IN OPENING FILE - FOR STORE

The file may be protected or may have an invalid VAX file name structure.

ERROR 41 — MISSING PSIZE.DAT — FAILED TO OPEN FILE

PSIZE.DAT, the file containing the default positional data for each of the plotters is either missing or wrongly protected. Contact the computer center or author.

ERROR 42 — MUST BE BETWEEN 2 AND 10000

The maximum number of points handled by the program in a single plot is 10000. The minimum number of points for a sensible plot is 2. Thus for interpolation the number of points lies within the range 2 to 10000.

ERROR 43 — LESS THAN TWO POINTS IN THE CHOSEN RANGE

The minimum number of points required to fit a line is two. The chosen range contains either 1 or 0 points.

ERROR 44 — INSUFFICIENT POINTS TO FIT POLYNOMIAL

The minimum number of points required to fit a polynomial of order n is $n + 1$. The chosen range contains less than $n + 1$ points.

ERROR 45 — IN OPEN STATEMENT — CANNOT OPEN FILE

This error can occur if an attempt is made to use a file name unsuitable for the VAX or an attempt is made to open a fifth version of a protected file.

ERROR 47 — OVERFLOW ERROR AT $X = *$**

The Y value at this point in the fitted polynomial will be outside the range 10^{-10} to 10^{10} .

ERROR 48 — NOT SUPPORTED BY SCREEN TEXT

The text on the terminals, that GRAPH was designed for (VT100 terminals with Retrographics and Visual 500 terminals), cannot be rotated.

ERROR 49 — NEGATIVE BOUND ON A LOG PLOT

An attempt was made to plot data on logarithmic axes with negative bounds fixed.

ERROR 50 — X VALUE OUT OF RANGE

The X value specified in the WRITE command is outside the plotted area. This can be over ridden by attempting the WRITE command again and using the default previous text and position.

ERROR 51 — Y VALUE OUT OF RANGE

The Y value specified in the WRITE command is outside the plotted area. This can be over ridden by attempting the WRITE command again and using the default previous text and position.

ERROR 52 — WRITE FILE NOT OPEN

The command CWRITE was given without a prior SWRITE command.

ERROR 53 — INVALID PEN NUMBER

The pen number chosen is outside the range 0-8

ERROR 54 — NO SIZE SET

Using the command SYMBOL requires a previous use of the SIZE or POINT command.

ERROR 55 — NO DATA TO RETAIN

The command RETAIN was used when the number of data points in memory is 0.

ERROR 56 — NO DATA HAS BEEN RETAINED

No data has been retained using the RETAIN command.

12.1 MODIFICATION ERRORS

ERROR M1 — MUST BE BETWEEN 1 AND 3

The letter size can be set to only one of three sizes.

ERROR M2 — MUST BE 1 - 8

Only 8 types of plotter are supported.

ERROR M3 — CANNOT OPEN PSIZE.DAT

Either PSIZE.DAT does not exist or the file has the incorrect protection. Contact the person mentioned when the program is first started.

ERROR M4 — DATA ERROR IN PSIZE.DAT

The version of PSIZE.DAT, which is being accessed by the program, has a data type error in it. PSIZE.DAT contains the default positional values.

ERROR M5 — MUST BE LESS THAN 1023 AND GREATER THAN 0

The number of points across the screen is 1024 (for a Tektronix 4010), numbered 0 to 1023, thus any number outside that range is off the screen.

ERROR M6 — MUST BE LESS THAN 779 AND GREATER THAN 0

The number of points vertically across the screen is 780, They are numbered 0 to 779, thus any number outside that range is off the screen.

ERROR M7 — XMAX OR YMAX TOO SMALL

This error occurs when an attempt is made to leave MODIFY and either or both of the upper limits are smaller than the corresponding lower limits.

12.2 COMMAND FILE ERRORS

ERROR C1 — CANNOT OPEN THE FILE

The file does not exist or cannot be created.

ERROR C2 — ERROR IN COMMAND FILE

Data type error. Wrong file type.

ERROR C3 — NO COMMAND FILE RUNNING

There is no command file attached to continue executing.

ERROR C4 — MUST BE 1 - 99

A command file number is in the range 1 to 99. An attempt was made to use a number outside this range.

ERROR C8 — END OF FILE ENCOUNTERED

An attempt was made to read information beyond the end of the command file.

12.3 WARNINGS

WARNING — DATA MAY BE INCOMPATIBLE WITH PROGRAM

When GRAPH attempts to plot files that it cannot readily read this warning is given. This occurs when the program detects an incompatible set of initial lines in the data. However the program ignores these and assumes a file of up to 10000 points.

WARNING — TOO MANY Y MARKS

The number of marks exceeds the limit of twenty five. GRAPH resets the toggle, the spacing and autoscales the Y-axis.

WARNING — TOO MANY X MARKS

The number of marks exceeds the limit of twenty five. GRAPH resets the toggle, the spacing and autoscales the X-axis.

WARNING — MAY BE GREATER THAN 100 POINTS

When plotting in the point mode. GRAPH checks to see how many points will be plotted. With files that are monotonically increasing with X, GRAPH will only give this warning when the number of points to be plotted actually exceeds 100. For other files GRAPH only checks to see how many points are in the file.

APPENDIX 1

COMMAND LIST

COMMAND	ABBREVIATION	SECTION	BRIEF DESCRIPTION
ABSOLUTE	ABS#	3.9	plot absolute error bars
ADDPOLY	ADDP	7.6.3	add a polynomial to plot data
-AND-	-AND-	11.4	allows the construction of compound commands
ANGLE	ANG	6.5	change from horizontal to vertical text
APPEND	APP#	7.6.4	append more data to the present plot
AVERAGE	AVE	4.5	average points together
BAUSCH	BAU	6.1	switch for Bausch and Lomb plotter
BOX	BOX	3.7	draw plot axes and titles
CENTER	CENTER	5.2.7.2	cross hairs at center of text
CHANNEL	CH	7.6.1	show the cross hair position
CIT161	CIT161	6.	correct output for CIT161 terminal
CLR	CLR	11.2.1	clear screen
CLRDAT	CLRD	11.2.2	clear plot data array
CONTINUE	C#	3.5	overlay next plot
*CONTINUE	*CO	9.4	continue command file
*CREATE	*CR#	9.2	create command file
*GO	*GO#	9.1	execute command file
*HALT	*HA	9.3	halt command file
*STOP	*ST	9.5	disconnect command file
CWRITE	CWRI	5.2.7.5	close write command file
DASH	DASH	4.1	choose dash line
DEFAULT	DEF	11.6	choose last file name
DEXP	DEXP	7.1.4	exponentiate the data
DIFFERENTIATE	DIFF	7.3.2	differentiate the plot data
DIR	DIR	10.3	show file directory
DIVX	DIVX	7.1.1	divide all X values by a constant
DIVY	DIVY	7.1.1	divide all Y values by a constant
DLOG	DLOG	7.1.5	take the logarithm of the data
EDT	EDT	11.10	edit a file using the EDT editor
E-	E-	11.1	redraw the plot with default axes
ERASE	ERA	6.6	inhibit GRAPH clearing the screen
ERRORBARS	ERR#	3.9	plot proportional error bars
EWRI	EWRI	7.7.2.3	erase last written text
EXPAND	E	11.1	expand an area of the plot
FADDITION	FADD#	7.2.3	add the plot data to a second plot
FDIVIDE	FDIV#	7.2.2	divide the plot by new data
FFIT	FFIT	7.5.2	fit a function to the data
FINISH	F	11.9	stop the program
FMULTIPLY	FMULT	7.2.1	multiply the plot data by new data
FONT	FONT	6.4	change the font of the plotter
FREEZE	FREE	5.1.4	switch to retain axes scales
FUNCTION	FUN	7.6.10	enable the plotting of an expression
GRAPH	GRA	6.1	switch for 4662 plotter.

COMMAND	ABREVIATION	SECTION	BRIEF DESCRIPTION
GRID	GRID	4.1	switch for axes grid
HELP	H	10.2	on line help
HP	HP	6.1	switch for Facit (HP) plotter
INTEGRATE	INT	7.3.1	integrate the plot data
INTERPOLATE	INTERP	7.4.2	interpolate and extrapolate
INVERT	INV	7.6.5	invert the data
LASER	LAS	6.1	switch for laser printer
LDASH	LDAS	4.1	length of dashes and spaces
LETTER	LET	5.2.6	letter size
LINE	L	7.6.2	draw a line by giving two points
LLOG	LLOG	4.2	switch for logarithmic axes
LOG	LOG	4.2	switch for logarithmic Y axis
LOGX	LOGX	4.2	switch for logarithmic X axis
MARK	MARK	5.2.5	insert submarks between labeled marks
MIDPOINT	MID	4.3	switch from midpoint to histogram
MODIFY	MOD	8.1	enter modify utility
MULTX	MULTX	7.1.2	multiply X data by a constant
MULTY	MULT	7.1.2	multiply Y data by a constant
NATCLOSE	NATC	6.1	close file for the National plotter
NATIONAL	NAT	6.1	switch for National plotter
NEXT	N	3.4	plot the next plot in file
NNUMBER	NNUM	5.2.4	switch for axes numbers
OVERLAY	O#	3.2	overlay new data onto the plot
PEN	PEN	6.2	choose pen number
PFIT	PFIT#	7.5.1	least squares fit of a polynomial
PLOT	P#	3.1	plot data file
POINT	POI	4.4	switch for point and line plotting
RAXIS	RAX	5.1.5	switch for the right axes labeling
RECALL	REC	11.8	recall data that was temporarily stored
REMOVE	REM	11.7	remove a point using cross hairs
REPEAT	%	11.3	loop counter
RETAIN	RET	11.8	temporarily store plot data
RMARK	RMARK	5.2.5	set submarks on the right axis
ROUT	ROUT	5.1.9	remove right axis marks
RSPACING	RSPA	5.1.6	set spacing of marks on right axis
RAPID	RAP	4.6	plot every other point
REDRAW	R#	3.3	redraw the last plot
SAME	SA	3.6	redraw the data
SET	SET	5.1.1	preset the axes ranges
SETX	SETX	5.1.2	preset the X axis range

COMMAND	ABREVIATION	SECTION	BRIEF DESCRIPTION
SETY	SETY	5.1.2	preset the Y axis range
SHIFTX	SHIFTX	7.1.3	add a constant to each X value
SHIFTY	SHIFT	7.1.3	add a constant to each Y value
SIZE	SIZE	4.4.2	set the size of the points
SMOOTH	SM#	7.4	smooth the data
SORT	SORT	7.6.5	sort into ascending X values
SPEED	SPE	6.3	set the pen speed
STATUS	STA	10.1	show information about the plot
STORE	STO	11.4	store the plot data
SUBTITLE	SUB	5.2.2	change the subtitle
SWAP	SWA	7.6.6	swap the X and Y values
SWRITE	SWRI	5.2.7.4	store the last written text
SYMBOL	SYM#	5.2.8	draw a symbol
TAXIS	TAX	5.1.5	switch for the top axes labeling
TEC	TEC	6.1	switch for Graphtec plotter
TITLE	TIT	5.2.1	change the plot title
TMARK	TMARK	5.2.5	set submarks on the top axis
TOP	TOP	4.8	set the number of points in the plot
TOUT	TUOT	5.1.8	remove the top axis marks
TSPACING	TSPA	5.1.6	set spacing of marks on top axis
TYPE	TYPE	4.4.1	set the point symbol to be used
VALUE	V#	7.6.7	display the value in the #th point
WRITE	WRI	5.2.7.1	write text
XAXIS	XAXIS	5.1.7	switch to display X axis
XLABEL	XLA	5.2.3	change the X axis label
XMARK	XMARK	5.2.5	set submarks on the X axis
XNUMBER	XNUM	5.2.4	switch for the X axis numbers
XSPACING	XSPA	5.1.3	set the mark spacing on the X axis
YAXIS	YAXIS	5.1.7	switch to display Y axis
YLABEL	YLA	5.2.3	change the Y axis label
YMARK	YMARK	5.2.5	set submarks on the Y axis
YNUMBER	YNUM	5.2.4	switch for the Y axis numbers
YSPACING	YSPA	5.1.3	set the mark spacing on the Y axis
YYCOMPARE	YYC	7.6.8	compare the Y values of two plots
ZETA8	Z8	6.1	switch for zeta plotter

INDEX

- ABSOLUTE, 9
 - absolute errors, 9
- ADDP, 37
- AND-, 46
- ANGLE, 28
- APPEND
 - appending plot data, 37
- arbitrary line — see LINE
- AVERAGE, 13
- axes
 - control see HBOX
 - numbers, 20
 - removal, 18
 - switches, 20
 - submarks, 20
- BAUSCH, 25
- Bausch and Lomb plotter, 25
 - pen change, 27
 - speed, 27
- BOX, 8
 - description, 9
- bubble sort — see SORT
- CENTER, 22
- CHANNEL, 36
- CIT161, 26
- clear
 - data array, 45
 - screen, 45
- CLR, 45
- CLRDAT, 45
- command files, 41
 - *CONTINUE, 41
 - *CREATE, 41
 - *GO, 41
 - *HALT, 41
 - *STOP, 42
- example command files, 42
 - comments, 42
 - running, 42
 - VU — graphs, 42
- CWRITE, 23
- SWRITE, 23
- command prompt, 1
- comparing Y data — see YYCOMPARE
- compound commands, 46
- CONTINUE, 5, 8
 - description, 8
- cross hair position, 36
- CWRITE, 23
 - VU-graph, 42
- DASH, 10, 42
 - LDASH, 10
 - dash lengths — see LDASH
- DATA
 - description, 8
- data analysis, 29
 - arithmetic operations, 29
 - set operations, 31
- data types
 - see Y data or XY data
- DCL commands, 48
- DEFAULT, 47
 - default file name, 5, 47
 - default file extensions, 5
- deletion of points — see REMOVE
- DEXP, 30
- DIFFERENTIATE, 32
- differentiation
 - equally spaced points, 32
 - unequally spaced points, 32
- direct data entry, 24
- DIRECTORY, 44
- division
 - by a constant, 29
 - one file by another file, 31
- DIVX, 29
- DIVY, 29
- dotted lines
 - see DASH
- DLOG, 30
- E-, 44
- ERASE, 28
- ERROR bars, 9
- error messages, 49
- EWRITE, 22
- exiting the program
 - see FINISH
- EXPAND, 44
- exponentiate data, see DEXP
- Facit plotter, 1, 25-26
- FADDITION, 32
- FDIVIDE, 31
- FFIT, 35
- FINISH, 48
- fitting a function

INDEX

polynomial - see PFIT
 arbitrary - see FFIT
 FONT, 28, 23
 FOURIER, 39
 fourier transform (discrete), 39
 FMULTIPLY, 31
 FREEZE
 description, 17
 FUNCTION, 39

 GRAPH, 7-8, 25
 Graphtec plotter, 1, 25-26
 GRID, 14

 hard copy plotting, 25
 HBOX, 9
 help
 HELP, 44
 STATUS, 44
 commands, 44
 Hewlett Packard plotter, 1, 25-26
 histogram plotting
 see MIDPOINT
 HP plotter, 1, 25-26
 Huston Instruments plotter
 see BAUSCH

 INTEGRATE, 32
 interchange axes, 38
 INTERPOLATE, 33
 method, 33
 INITIALIZE, 46
 INVERT, 38

 labeling axes
 see titled files
 see XLABEL and YLABEL
 LASER, 25-26
 layout utility, 40
 LDASH, 10
 least squares polynomial
 see PFIT
 LETTER, 21, 22
 LINE
 arbitrary, 35
 line fitting, 34
 LLOG, 11
 logarithm of data, 30
 logarithmic axes, 11
 see LOGX, LOGY and LLOG

treatment of negative numbers, 11
 treatment of zero, 11
 LOGX, 11
 LOGY, 11

 MARK, 20
 MIDPOINT, 11
 MODIFY, 40
 multiple plot file, 4
 multiplication,
 by a constant, 29
 one file by another file, 31
 MULTX, 29
 MULTY, 29

 NATIONAL, 25
 NATCLOSE, 25
 National plotter, 25
 pen change, 27
 NEXT, 5, 7
 description, 7
 NNUMBER, 20

 OVERLAY, 5, 6-7
 description, 6
 O#, 6

 P# - see PLOT
 PEN, 8, 23, 27
 description, 27
 SWRITE, 23
 PFIT
 coefficients file, 34
 polynomial fitting, 34
 retaining polynomial, 34
 PLOT, 5-8, 11, 15-17, 19, 21, 25, 28-31,
 33-34, 37-39, 46-48
 description, 5
 plotting alternate points, 14
 POINT, 10, 12
 symbol size, 13
 symbol types, 12
 polynomial addition, 37

 proportional error bars, 9

 QUME, 26

 R# - see REDRAW
 RAPID, 14

INDEX

RAXIS, 17
 RECALL, 48
 REDRAW, 5, 7, 11, 16
 description, 7
 REMOVE, 40, 47
 description, 46
 repeat counter (%), 24, 46
 repeat counter SMOOTH, 32-33
 reset initial conditions, 46
 RETAIN, 48
 right axis label, 17
 RMARK, 20
 ROUT, 18
 RSPACING, 18

 SAME, 5, 8
 description, 8
 screen erase control, 28
 SET, 15
 setting axes
 see SET, SETX or SETY
 setting R mark spacing, 18
 setting T mark spacing, 18
 setting X mark spacing, 16
 setting Y mark spacing, 16
 SETX, 16
 SETY, 16
 SHIFTX, 29
 SHIFTY, 29
 SIZE, 12-13
 description, 13
 see MODIFY
 SMOOTH, 33
 smoothing data, 32-33
 SORT, 11, 38
 description, 38
 SPEED, 27
 squared paper, 14
 STATUS, 44
 STORE, 47
 storing modified data, 47
 storing data in memory - see RETAIN
 submarks, 20
 SUBTITLE, 4, 19
 summing points together, 13
 SWAP, 38
 SWRITE, 23, 43
 VU-graph, 42-44
 SYMBOL, 23
 symbol type, 12

TAXIS, 17
 TEC, 25
 TECFILE, 25
 Tektronix 4010, 1, 21, 25
 Tektronix 4662, 1, 2, 9, 25
 font type selection, 28
 pen change, 27
 terminal control
 CIT161, 26
 QUME, 26
 TAB, 26
 text
 arbitrary, 21
 SUBTITLE, 19
 TITLE, 19
 YLABEL, XLABEL, 19
 text size - see LETTER
 TITLE, 19
 title page, 1
 titled files, 4
 TMARK, 20
 toggle on-off autoscaling, 17
 TOP, 14
 top axis label, 17
 TOUT, 18
 TSPACING, 18
 truncating plot data, 14
 TYPE, 12
 description, 12

 Unknown commands
 treatment of, 48

 VALUE, 38
 Visual 550, 1
 VT640, 25, 1
 VU-graph, 42-44

 WRITE
 angled text, 28
 description, 21
 store, 23
 close file, 23

 XAXIS, 18
 XLABEL, 19
 XMARK, 20
 XNUMBER, 20
 XSPACING, 16
 XY Data, 3

INDEX

description, 3

YAXIS, 18

Y Data, 3

description, 3

YLABEL, 19

YMARK, 20

YNUMBER, 20

YSPACING, 16

YYCOMPARE, 39

Zeta 8 plotter, 1

ZETA8, 25-26

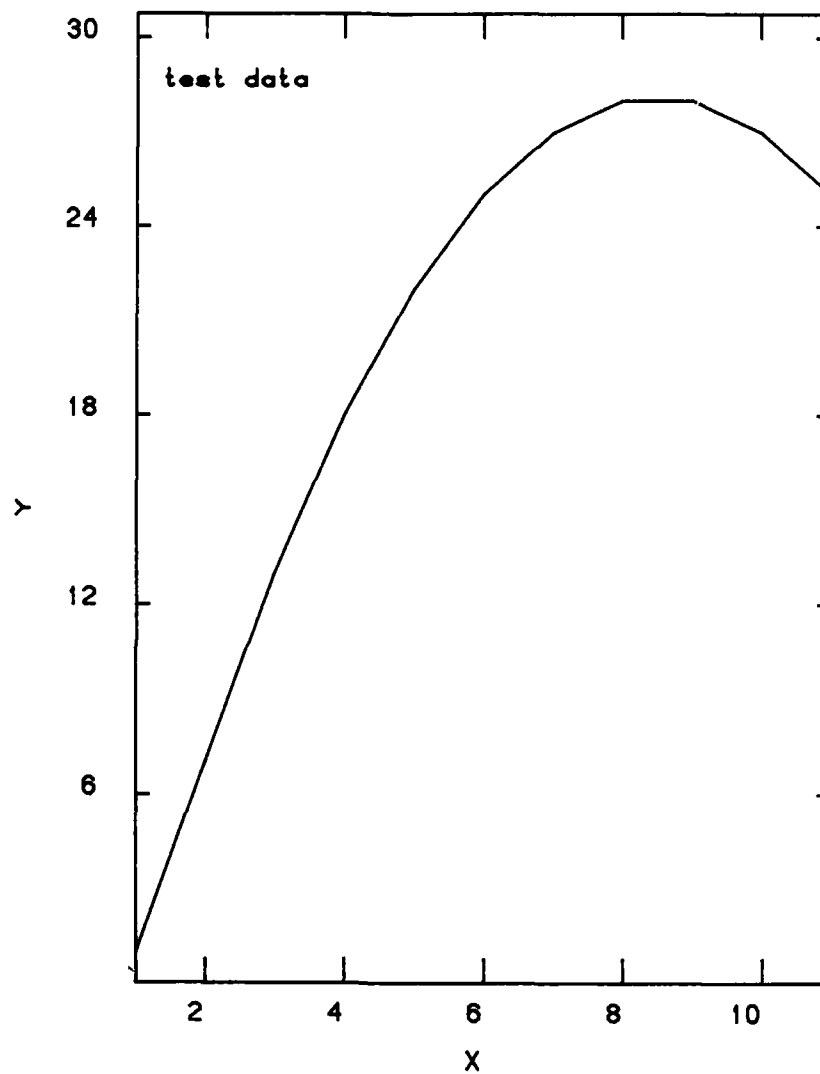


FIGURE 1. The plot file 'test.dat' of section 3.1 will produce the above display on the terminal screen after the command, PLOT, has been entered and the response 'test.dat' has been given to the request for a file name.

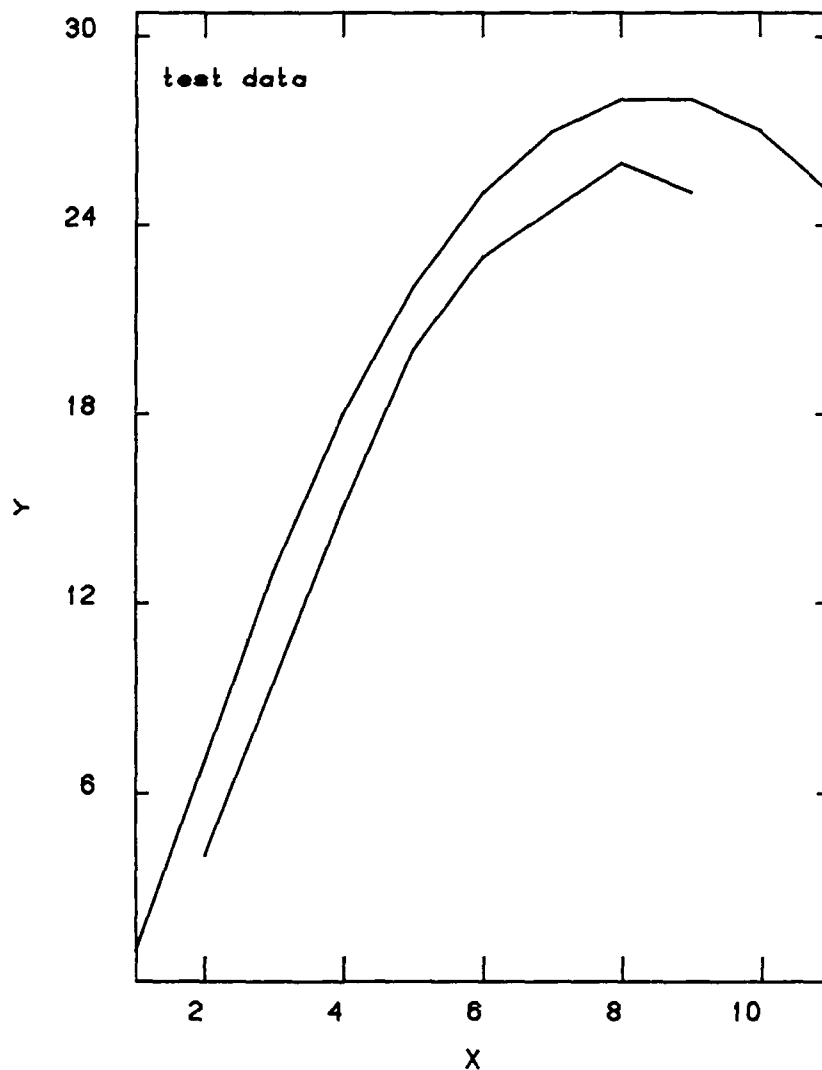


FIGURE 2

The OVERLAY command allows the user to display the data of a plot file onto the preceding plot. See section 3.2. This figure shows the result of the data of file 'test.dat' being overlaid onto the plot of the data of file 'test1.dat'. The contents of the file 'test.dat' is given in section 3.1 and the contents of 'test1.dat' is given in section 3.2.

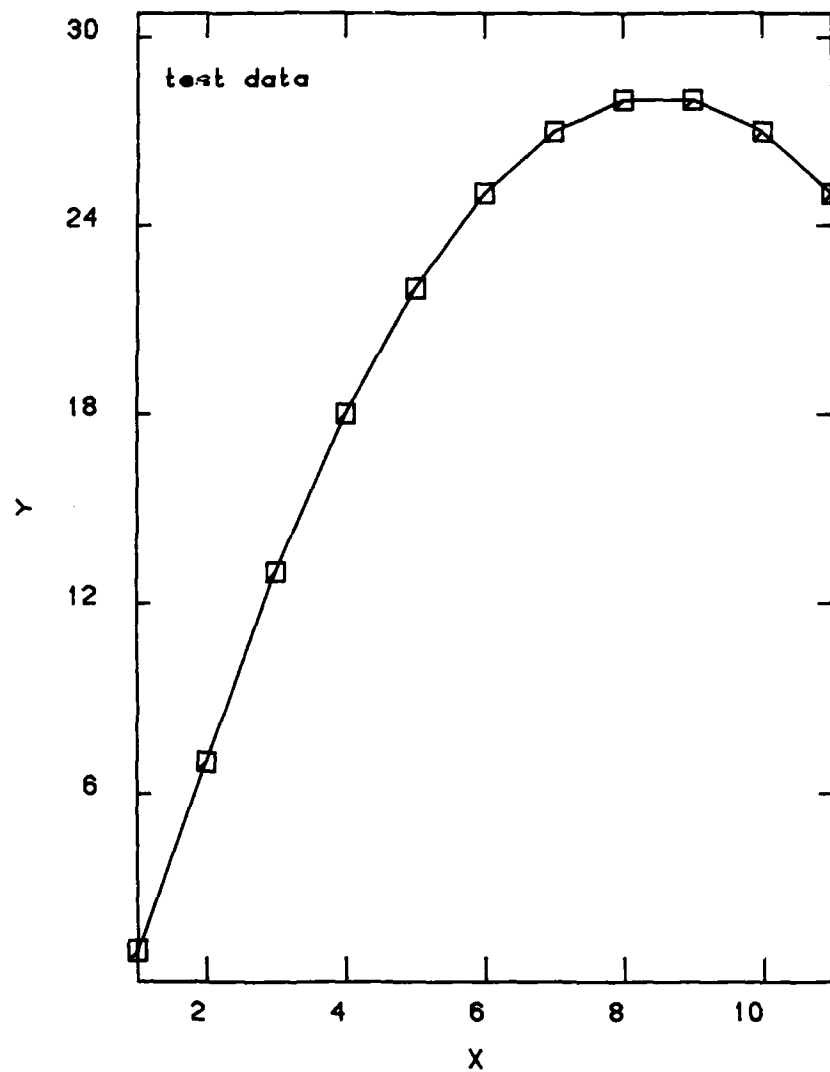


FIGURE 3 The data of the plot file 'test.dat' of section 3.1, has been plotted using the command PLOT. Point plotting was then set (section 3.1) and the data was redrawn onto the plot (section 3.6).

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

DOCUMENT CONTROL DATA SHEET

REPORT NO.
MRL-GD-0010AR NO.
AR-005-192REPORT SECURITY CLASSIFICATION
Unclassified

TITLE

The interactive plotting program GRAPH - revised manual

AUTHOR(S)

S.R. Kennett

CORPORATE AUTHOR

Materials Research Laboratory
PO Box 50,
Ascot Vale, Victoria 3032REPORT DATE
December 1987TASK NO.
DST 85/117SPONSOR
DSTOFILE NO.
G6/4/8-3429REFERENCES
5PAGES
72

CLASSIFICATION/LIMITATION REVIEW DATE

CLASSIFICATION/RELEASE AUTHORITY
Chief of Division,
Materials, MRL

SECONDARY DISTRIBUTION

Approved for Public Release

ANNOUNCEMENT

Announcement of this report is unlimited

KEYWORDS

Computer Graphics

Plotting

COSATI GROUPS 0062B

ABSTRACT

This report is a user guide for the interactive graphics program called GRAPH and replaces the earlier Materials Research Laboratories technical note, MRL-TN-485. The suggestions of users and the expansion of the capabilities of the program have led to this revision.

GRAPH is a command driven interactive program which allows the user, with only limited computer experience, to display and modify graphical output. The program can plot data on linear or logarithmic axes. The axis scales can be set by the user or autoscaled by the program. Labels can be set by the user or any part of the plot. Data can be plotted as points, histograms or lines.

GRAPH has commands which allow for simple data reduction. Data can be multiplied, divided, added, subtracted by a constant value or by a second set of data. Data can also be exponentiated, integrated, differentiated or smoothed. A least squares polynomial or a more general function can be fitted to the data.

GRAPH was written in FORTRAN 77 on the VAX 11/780 computer at the Materials Research Laboratory.

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

END

DATE
FILMED

88R