

UNCLASSIFIED



**Australian Government**  
**Department of Defence**  
Defence Science and  
Technology Organisation

## KGRAPH – Interactive Plotting and Analysis

*S.R. Kennett*

**Maritime Platforms Division**  
Defence Science and Technology Organisation

DSTO-GD-0718

### **ABSTRACT**

This report forms a user guide for the Windows based interactive graphics plotting and analysis program called KGRAPH. The program is based on an earlier program which was the subject of an earlier Materials Research Laboratories general document MRL-GD-010.

### **RELEASE LIMITATION**

*Approved for public release*

UNCLASSIFIED

UNCLASSIFIED

*Published by*

*Maritime Platforms Division  
DSTO Defence Science and Technology Organisation  
506 Lorimer St  
Fishermans Bend, Victoria 3207 Australia*

*Telephone: (03) 9626 7000  
Fax: (03) 9626 7999*

*© Commonwealth of Australia 2013  
AR-015-484  
January 2013*

**APPROVED FOR PUBLIC RELEASE**

UNCLASSIFIED

**UNCLASSIFIED**

# KGRAPH – Interactive Plotting and Analysis

## Executive Summary

This report details the operation and behaviour of the commands, functions and macros of KGRAPH and is intended to form a user guide for the current version (0.54) of the software. This software came out of a need to visually present and extract information from large data files generated by instrumentation that exceed the capability of available programs. KGRAPH based on an earlier plotting program has been designed to both manipulate the data visually and to be able to undertake the essential data analysis for processing, presentation and reduction.

**UNCLASSIFIED**

UNCLASSIFIED

*This page is intentionally blank*

UNCLASSIFIED

## Author

### **S.R. Kennett**

#### **Maritime Platforms Division**



*Stephen Kennett graduated from the University of Melbourne with a B.Sc (Hons) in 1976, an M.Sc (Nuclear Physics) in 1978 and a Ph.D. (Nuclear Astrophysics) in 1981. He commenced work with the Department of Defence, Materials Research Laboratory, Defence Science and Technology Organisation (DSTO) in 1981 and has undertaken research in interaction of lasers with materials, atmospheric optical propagation, laser eye protection. From 1989 to 2004 he lead a team undertaking research into the effects of fire on life safety and asset protection on RAN vessels. In 2004 as part of a DSTO major strategy, he was asked to investigate the cost drivers of Blackhawk Helicopters which lead to the development a web-based tool to analyse maintenance data. From 2008 to 2010 he applied the techniques developed to RAN ships maintenance data which resulted in a web-based tool to analyse and evaluate maintenance data. Dr Kennett was elected a fellow of the Royal Institution of Naval Architects in 2010.*

---

---

UNCLASSIFIED

*This page is intentionally blank*

UNCLASSIFIED

# Contents

## ABBREVIATIONS

1. INTRODUCTION.....	1
2. SYSTEM REQUIREMENTS.....	1
3. LAUNCHING THE PROGRAM.....	1
4. QUICK RUN THROUGH .....	3
5. INPUT DATA AND FILE TYPES .....	6
5.1 Y DATA.....	8
5.2 X Y DATA.....	8
5.3 X Y Z DATA .....	9
5.4 Comma separated values files (.csv).....	9
5.5 Data files with error bars.....	11
5.5.1 X EVEN ERROR BARS .....	11
5.5.2 Y EVEN ERROR BARS .....	12
5.5.3 X UNEVEN ERROR BARS.....	13
5.5.4 Y UNEVEN ERROR BARS.....	14
5.5.5 XY EVEN ERROR DATA .....	15
5.5.6 XY UNEVEN ERROR DATA.....	16
5.6 TITLE LIST FILES .....	17
6. COMPOUND COMMANDS.....	18
6.1 Repeat Count # <#> or # [<#><..<#>>].....	18
7. START-UP FILE .....	19
8. ALPHABETIC COMMAND DESCRIPTIONS.....	20
8.1 ABOVE (abo) <#> <#> <#>.....	20
8.2 APPEND (app) <#> <filename>.....	22
8.3 ARROW (arr) <h/v> <#> <#> <#> <#>.....	22
8.4 AVERAGE (ave) <#> <#> <#> <#> .....	23
8.5 BACKGROUND (bac) <filename> .....	24
8.6 BOX.....	25
8.7 CAPTURE (cap) <#> <#>.....	26
8.8 C3D - Converting point aerial density to 3-D.....	28
8.9 CLEAR, (clr or cle) .....	30
8.10 COMBINE (com) <x y z> <#> <x y z> <#> <x y z> <#>.....	30
8.11 CONTINUE (c) <#> or c [<#><..<#>>] .....	30
8.12 CONTOUR (con).....	31

8.13	COPY (cop) <[#<..#>] #> <or cop # #>.....	33
8.14	CURSOR (cur) <x/y> .....	34
8.15	DASH (das) .....	34
8.16	DELETE (del) <[#<..#>]> .....	34
8.17	DEGREE (deg) .....	35
8.18	DISTANCE (dis) <#> <#> <#> <#>.....	35
8.19	DIFFERENTIATE (dif) <#> <[#<..#>]> .....	36
8.20	ELEVATION (ELE) <#> .....	37
8.21	Error bars .....	37
8.22	EXPORT (exp) <[#<..#>]> <filename> .....	39
8.23	FFIT (ffi).....	39
8.24	FFT (fft) <#> .....	41
8.25	FILL (fil) .....	41
8.26	FLOWOVER (flo) <#> .....	43
8.27	FONT (fon) <#> .....	44
8.28	FOURIER (fou) <#> <[#<..#>]> .....	45
8.29	Free axes commands .....	45
8.30	FUNCTION (fun).....	46
8.31	GRID (gri).....	48
8.32	G3D (g3di) <#> <#> for 3-D plots.....	48
8.33	HELP (hel) (or ?) .....	49
8.34	HIGHLIGHT (hig) .....	49
8.35	IF ELSE ENDIF .....	50
8.36	INCREMENT (inc) <#> .....	51
8.37	INTEGRATE (int) <[#<..#>]> .....	51
8.38	INTERPOLATE (interp) <#> .....	52
8.39	INFORMATION (info) .....	53
8.40	LDASH (lda) .....	54
8.41	LEGEND (leg) <num> .....	55
8.42	LIST (lis) <[#<..#>]> file <filename> .....	56
8.43	LOAD (loa, l) <#> <filename> .....	57
8.44	MACRO (mac) {go   stop   quit   record   #   filename}.....	59
8.44.1	Example Macros .....	61
8.45	MARK (mar) <x/y/z> <#> .....	62
8.46	MIDPOINT (mid) .....	63
8.47	MOVE (mov) <[#<..#>] #> <or mov # #> .....	64
8.48	NEW.....	65
8.49	NEXT (n) .....	65
8.50	NOISE (noi) <#> .....	66
8.51	OCCURRENCE (occ) <#> <#> <#> .....	67
8.52	OUTPUT <filename>   quit .....	67
8.53	OVERLAY (o) <#> <filename> .....	68
8.54	PADDITION (pad) <#> <#> <#> .....	69
8.55	PAPPEND (pap) <#> <#> .....	70
8.56	PDIVIDE (pdi) <#> <#> <#> .....	70
8.57	PEAK (pea) <#> <#> .....	71
8.58	PEN (pen) < box/ grid/ fill/ value/ weight> <#> .....	72
8.59	PFIT <#> <#> <#> .....	74



8.60	PLOT (p) <#> .....	76
8.61	PMULTIPLY (pmu) <#> <#> <#> .....	77
8.62	POINT (poi) .....	78
8.63	POLAR (pol).....	78
8.64	PREVIOUS (pre) .....	79
8.65	PSUBTRACT (psu) <#> <#> <#>.....	79
8.66	Peak and valley analysis (PVA) <#> .....	80
8.67	POWER .....	81
8.68	QUIT (qui).....	84
8.69	RADIAN (rad) .....	84
8.70	RAINFLOW (rai) .....	84
8.71	RAPID (rap) <#> .....	85
8.72	REMOVE (rem) <#> .....	86
8.73	RESET (res).....	87
8.74	ROLL (rol) <#> .....	87
8.75	ROTATE (rot) <#> .....	88
8.76	SAME (sam) .....	89
8.77	SAVE (sav) < *   &   +> <[#<..#>]> <filename> .....	90
8.78	SET (set) <#> <#> <#> <#> .....	90
8.79	SIZE (siz) <#> .....	91
8.80	SLICE (sli) <[#..#]> <#> <#> <#>.....	92
8.81	SMOOTH (smo) <[#<..#>]> <#> <#>.....	93
8.82	SORT (sor) <[#<..#>]> .....	95
8.83	SUBTITLE (sub) <title> .....	96
8.84	SWAP (swa) <[#<..#>]> or swa <#> .....	96
8.85	TEXT (tex) <text to be written> .....	97
8.86	TITLE (tit) <title> .....	97
8.87	TYPE (typ) <#> .....	98
8.88	VALUE (val) <#> .....	100
8.89	VARIABLE (var) <# #> .....	101
8.90	VELOCITY (vel) <#> .....	101
8.91	WINDOW (win) <[#<..#>]> <#> <#>.....	102
8.92	Exponentiation of data .....	102
8.93	Division of data by a constant .....	103
8.94	Reciprocal of data.....	104
8.95	Axes label text.....	104
8.96	Logarithm of data.....	105
8.97	Logarithmic Axes .....	105
8.98	Multiplying data by a constant .....	106
8.99	Setting axes limits .....	106
8.100	Shifting plot along axes .....	107
8.101	Axes Mark Spacings .....	108
8.102	YYCOMPARE (yyc) <#> <#> <#> .....	109
8.103	ZOOM (z) .....	111
9.	MENU ITEMS.....	113
9.1	File menu .....	113
9.2	Edit Menu .....	114

9.3	Function Menu .....	114
9.4	Plot Menu .....	115
9.5	Plot function Menu.....	115
9.6	Information Menu .....	117
10. REFERENCES .....		118
APPENDIX A: PLOT AND FILE CONTROL COMMAND .....		119
APPENDIX B: AXES CONTROL COMMANDS .....		120
APPENDIX C: FUNCTION COMMANDS .....		122
APPENDIX D: SCREEN DISPLAY AND PRINTING COMMANDS.....		124
INDEX .....		125

## Abbreviations

The following table lists abbreviation and conventions used.

<CR>	Carriage Return key press.
<ESC>	Escape key press.
Double click	Move the cursor over the item of interest by sliding the mouse and press the left mouse button twice in rapid succession.
Click	Move the cursor over the item of interest by sliding the mouse and press the left mouse button.
Dragging	Move the cursor over the item of interest by sliding the mouse and then press and hold down the left mouse button. Move the cursor to the new position and release the mouse button.
< # >	This is used to indicate an optional number can be added to a command.
[ # .. #]	Square brackets indicate a range of plots. The two dots are used to separate the lower and upper limits.
()	This is used in section headings to indicate a short form of the command.

UNCLASSIFIED

DSTO-GD-0718

*This page is intentionally blank*

UNCLASSIFIED

## 1. Introduction

The analysis of large data files collected by instruments and the necessity to extract, process and present such data in a graphical form has driven the development of KGRAPH. KGRAPH is a WINDOWS based program that provides a graphical interface to data.

The program is based on an earlier version of the program which was written in FORTRAN 77 for a VAX 11/780 computer and was the subject of an earlier publication by Kennett [1]. Significant sections have been rewritten and expanded on. The current version is written in FORTRAN 95 with a graphical interface supported by library function of Winteracter<sup>1</sup>.

The program is designed to manipulate data graphically rather than as lists of numbers and can handle 20 million points consisting of up to 1000 individual curves. Output can be presented in the form of plot and text options. Macro and compound commands provide a powerful and flexible means of analysis.

This manual is intended to give the reader a brief introduction to the program and a description of the functions available within the program environment.

## 2. System Requirements

The program is able to be run on all current Microsoft Windows based systems at the time of publication.

Microsoft windows environment:	Windows 2000, XP or Windows 7
Memory	> 1GB
CPU	Pentium 4 (M) or better
Graphics resolution	At least 1024 by 768
Mouse	Two button mouse with wheel

## 3. Launching the Program

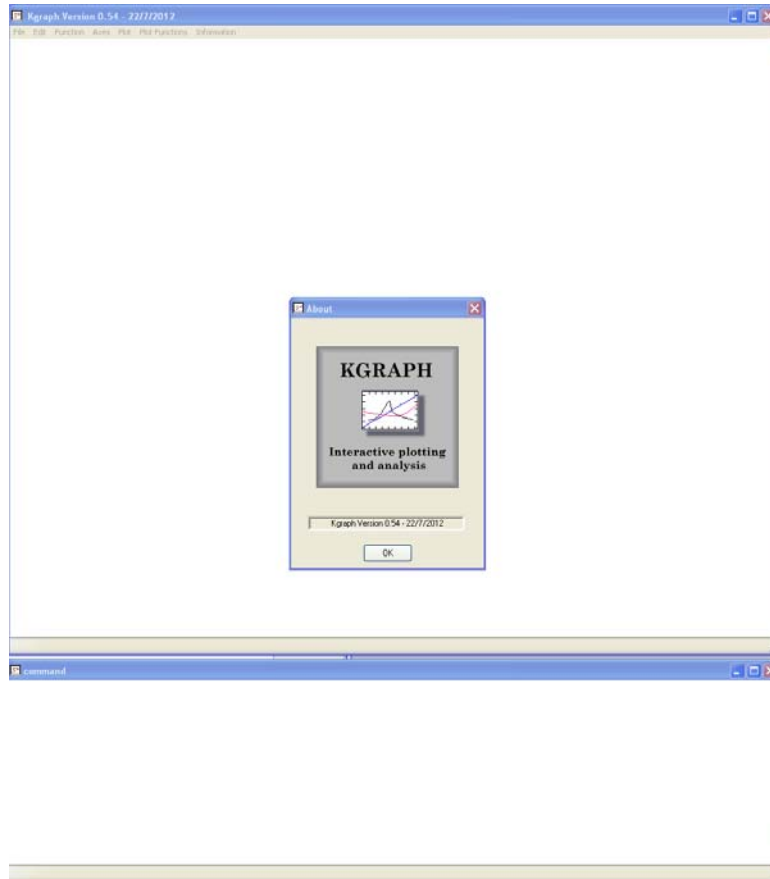
The program KGRAPH can be launched from the windows environment (Windows 2000, XP or Windows 7) by double clicking on the program icon. When first launched, the program will display two windows and a splash screen as shown in Figure 1. The two windows will occupy 90% of the available screen. The upper larger window is used to display the plotted data, the

---

<sup>1</sup> Winteracter Version 9.0, Interactive Software Services Ltd, Westwood House, Littleton Drive, Huntington, United Kingdom.

plot window, and the lower smaller window provides a command line interface, the command window. The window title of the plot window displays the KGRAPH version number. The current version is 0.54.

The user can resize and relocate the two main windows. When KGRAPH is closed the current size and positions of the windows is stored and these will be restored when the program is relaunched. This can be across two physical screens.



*Figure 1 The initial arrangement of the windows and splash screen for KGRAPH is shown. Each of the windows can be resized and arranged to suit the user preferences.*

Clicking on the 'OK' button will close the splash screen and launch the Start-up file if it exists, see Section 7.

Commands can be entered via the command line or alternatively by using dropdown menus provided on the plot window. When the dropdown menus are used the corresponding command is displayed in the command window.

When first launched the dropdown menus are restricted to loading a data file, getting help or exiting the program.

Multiple sessions of the program can be launched concurrently. The current file name is displayed at the bottom of the windows, in the window's status line, to help distinguish between the different sessions.

## 4. Quick Run Through

In order to examine the features of the program, it is necessary to have some data to manipulate. KGRAPH can generate data using the FUNCTION command, see Section 8.30, or by loading a file which has been created by Notepad<sup>2</sup>. Run Notepad and enter the following data and then save the file as 'c:\temp.dat'.

```
%xydata
0, 1
1, 2
2, 4
3, 8
4, 16
5, 32
6, 64
%end
```

This can be loaded into KGRAPH by selecting the load option from the file dropdown menu on the plot window and using the standard windows file selection tool. Alternatively, the command *load c:\temp.dat* can be typed into the command window. Both methods should result in the plot shown in Figure 2.

As in most conventional axes systems, the x-axis refers to the horizontal axis and the y-axis refers to the vertical axis. When a z-axis is being considered it will form a right hand axis system and will come out of the screen.

By default the plot is automatically scaled with the y-axis 10% larger than the largest y value and the x-axis scaled to match the domain of the data and the plot is drawn as a series of connected lines. The axes marks are also scaled automatically so that at least 4 (and no more than 8) marks are shown on each axis.

Multiple plots can be kept in memory and thus the command line prompt contains the number of the current plot of focus.

The choice of dropdown menu items is opened up when data is entered into KGRAPH. These are self explanatory and can be explored by the user. It is recommended that the user become familiar with the drop down menu items.

---

<sup>2</sup> Notepad comes with the Microsoft operation systems and is a basic text editor.

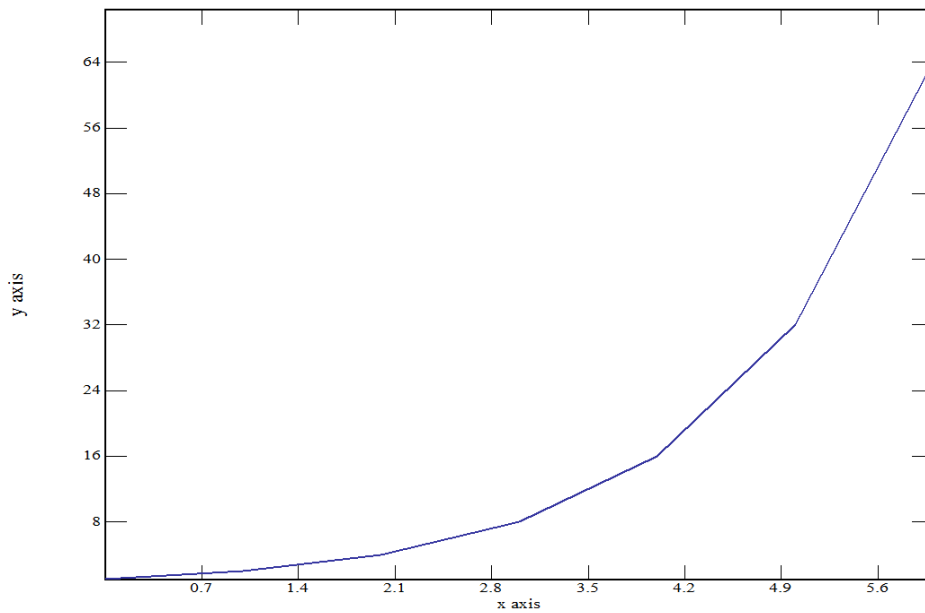


Figure 2 The default plot of the example data contained in 'temp.dat' as described in the above text is shown.

Clicking on and dragging the graph will move the plotted data and alter the axes appropriately. This acts to move the view of the data. If the data is dragged a small distance to the right and up, the ranges for x and y axes are altered as shown in Figure 3.

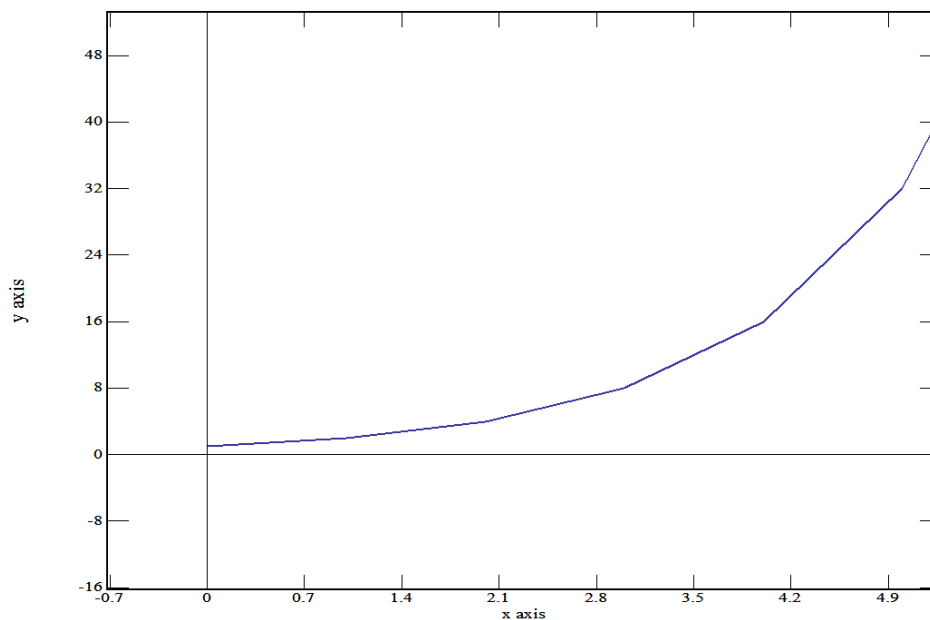
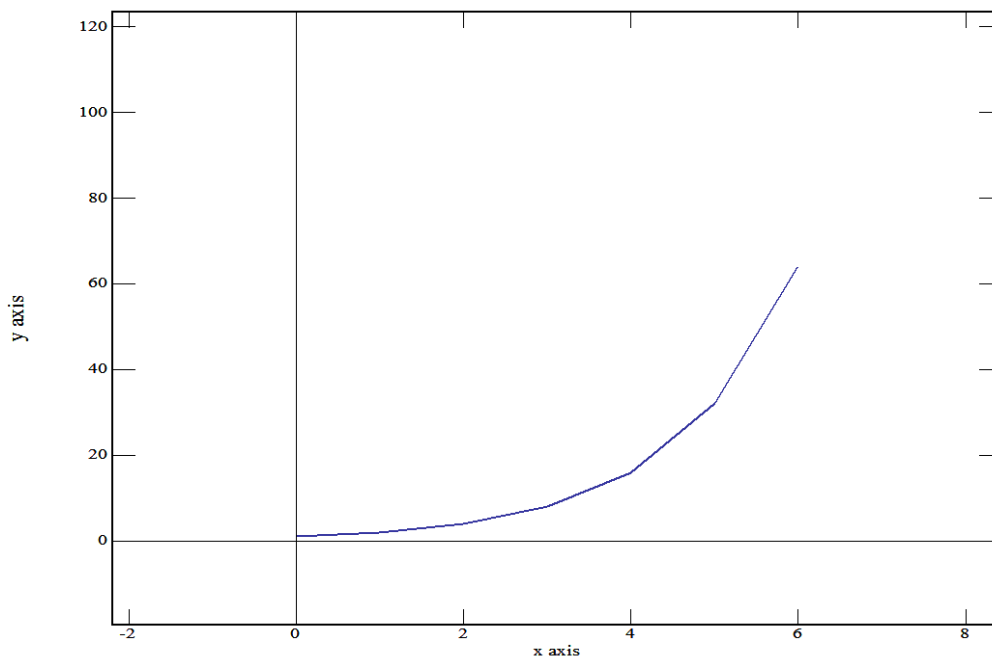


Figure 3 The relative position of the data within the plot can be moved by positioning the mouse on the plotted area, pressing the left mouse button, holding the press and dragging the mouse to a new position. When the mouse button is released the plot is redrawn at the new position.



The mouse wheel is used to rescale the plot when the cursor is located in the plot window. Rotation of the mouse wheel in one direction reduces the scale on the x-axis and the other increases the scale on the x-axis. The rescaling is centred on the mouse position so that the x value at that location remains stationary. To rescale the y-axis the mouse wheel is used in the same way with the addition of pressing the left shift key while rotating the mouse wheel.

The result of a change in scale in both directions is shown in Figure 4. The automatic scaling of the mark spacing has resulted in a vertical mark spacing of 20 and a horizontal mark spacing of 2.



*Figure 4 The axes scales and plotted information can be rescaled by using the mouse wheel and the left shift key. The mouse wheel is used to alter the x scale and the mouse wheel together with the left shift key is used to alter the y scale.*

Additionally by pressing the shift key and arrow keys the plot scale is stepped in the indicated direction of the arrow by the range in that direction. Alternatively, the x-axis scale can be shifted to the right and left by the current x range amount by pressing the key pad 6 and 4 keys, respectively.

Similarly, the y-axis scale can be shifted to up and down by the current y range amount by pressing the key pad 8 and 6 keys, respectively.

Information and settings that can alter and define the current plot can be accessed using the command INFO from the command window or from the general selection of the information menu on the plot window. When the command INFO is given, KGRAPH shows a dialog window with information about the current plot, see Figure 5.

In the dialog window, the values and text information can be altered to modify the plot. Text can be copied and pasted into the text boxes for Title, Subtitle, x label, y label and legend from other windows applications such as notepad or from one plot to another.

Check boxes allow and restrict the way in which the axes and data are displayed.

Navigation between plots is done via the next and previous buttons at the bottom right of the dialog or by entering the required plot number in the box located at the top left corner of the dialog window, see Figure 5.

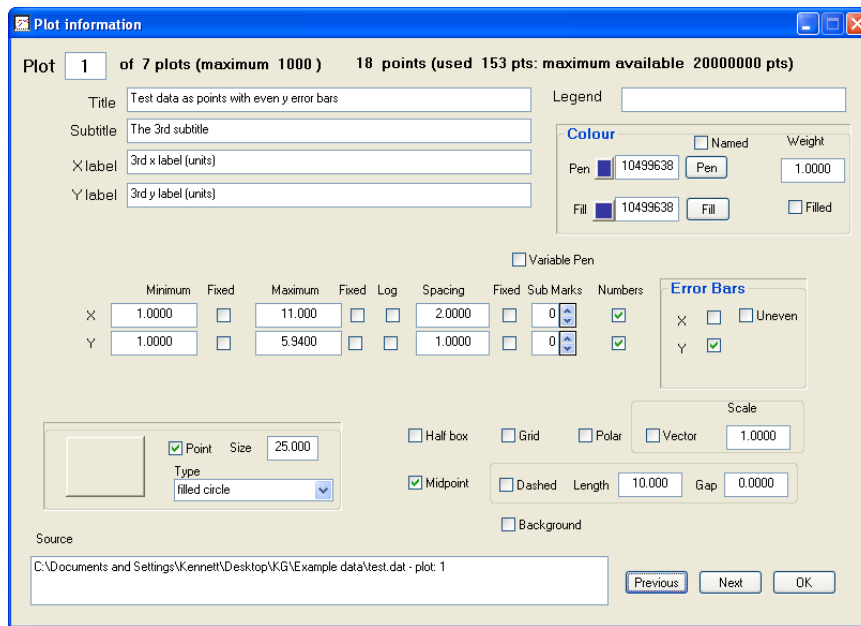


Figure 5 The information dialog window provides the state of the plot options and text information of the current plot.

More detailed information and examples on the available functions can be found in the following sections.

## 5. Input data and file types

Most data files consist of lists of ascii character values and information concerning titles and axes. The exception is the internal save command which saves the data in a compressed binary form. The following section describes the input that can be read and modified from a text editor.

Titles and axes can be altered within the data file by including imbedded data commands. The commands that can be included in input data files are listed in Table 1.

*Table 1 List of commands that can be included in input data files.*

<b>Command</b>	<b>Abbreviation</b>	<b>Description</b>
%grid	%gri	!turns on grid
%ylog	%ylo	!selects logarithmic y axes
%xlog	%xlo	!selects logarithmic x axes
%log	%log	!selects logarithmic axes
%subtitle 'text'	%sub	!adds a subtitle to the plot
%title 'text'	%tit	!adds a title to the plot
%ylabel 'text'	%yla	!adds a new ylabel
%xlabel 'text'	%xla	!adds a new xlabel
%end	%end	!marks the end of a plot in the file
%ydata	%yda	!indicates following plots contain y data
%xydata	%xyd	!indicates following plots contain x, y data
%xyzdata	%xyz	!indicates following plots contain x, y, z data
!	!	!characters after '!' in input are comments

The first group alters the conditions for subsequent plotting. The second group alters the labeling of the plot being read. The third group defines the nature of the data of the plot.

Data can be separated by commas, spaces or tab characters.

The following subsections provide information on:

- Data consisting of y values
- Data consisting of x and y pairs
- Data consisting of x, y, z triplets
- Title list files
- Data taker input files.
- Comma separated values files (.csv) exported data from Microsoft Excel<sup>3</sup> or some other program.
- Data with error bars.

---

<sup>3</sup> Excel is part of the Microsoft Office suit and is under copyright by Microsoft

## 5.1 Y DATA

Data can be entered as a list of y values. These are preceded by the data file command %ydata which indicates to the program that the subsequent data has only y values. The %end indicates the end of the current plot data and can be used to separate multiple sets of data within the same file.

For example:

```
%ydata
1
2
4
8
16
32
64
%end
```

## 5.2 X Y DATA

Data can be entered as a list of x, y value pairs. These may be preceded by the data file command %xydata which indicates to the program that the subsequent data is in the form of xy paired values. The x y data entry form is the default setting and the data file command is unnecessary if no preceding %ydata or %xyzdata is included in the data file.

For example:

```
%xydata    !optional comment
0, 1
1, 2
2, 4
3, 8
4, 16
5, 32
6, 64
%end
```

This format may be repeated any number of times within a file.

### 5.3 X Y Z DATA

Data can be entered as a list of x, y, z value triplets separated with commas. These should be preceded by the data file command %xyzdata which indicates to the program that the subsequent data is in the form of x, y, z triplets (Always in threes). The xyz data is used to define a 3D surface and thus the ordering of the points is not used.

For example:

```
%xyzdata    !optional comment
0, 1, 1
1, 2, 4
2, 4, 16
3, 8, 64
4, 16, 256
5, 32, 1024
6, 64, 4096
%end
```

This format may be repeated any number of times within a file.

### 5.4 Comma separated values files (.csv)

Microsoft Excel or a variety of other programs can export data in Comma separated values format (.csv). Files containing data in the .csv format and with the .csv extension can be imported into KGRAPH.

A typical .csv file consists of a table of values of text and numbers separated by commas. The first line often contains a set of column titles. If a text value contains a comma then it is delimited using quotation marks or double quotation marks. An example file could be as follows;

```
choice,1st choice,2nd choice,3rd choice,4th choice,5th choice
1,1,2,3,4,5
2,2,4,6,8,10
3,3,6,9,12,15
4,4,8,12,16,20
5,5,10,15,20,25
6,6,12,18,24,30
7,7,14,21,28,35
8,8,16,24,32,40
```

9,9,18,27,36,45

Using the load command with this data, KGRAPH initially loads the data into a data grid in a dialog window, see Figure 6. Values can be altered in the dialog box. Each column of data is a potential plot. The domain variable, the x-axis information, can be set using the drop down in the top part of the dialog box. When selected that column changes colour indicating that the column will be used as the domain for all plots.

The checkbox under each title indicates that the column will be plotted. Removing the tick will indicate to KGRAPH to ignore the column. Text columns will not be checked. Time and date and data in hexadecimal formatted columns can be plotted but by default these are not checked. Time and dates are converted to seconds since midnight on the day of the first date in the plot. This may result in negative time values if the data is not in time increasing order.

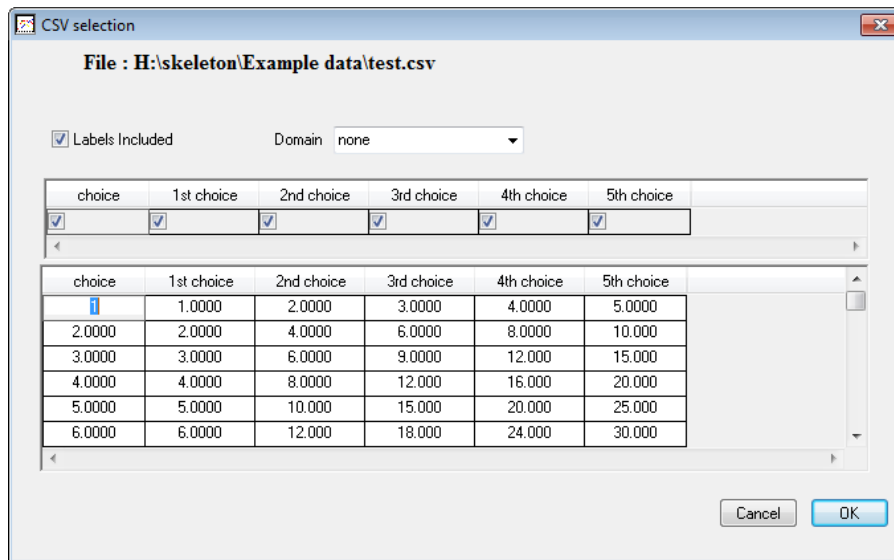


Figure 6 A dialog window of the incoming information in a .csv file is displayed. It is assumed that the first line contains column headings. The columns that contain numerical data to be plotted are selected by the tick in the check box. The column containing the domain can be selected from the dropdown menu.

The inclusion of an additional line at the beginning of the file can remove the necessity of selecting the domain. This line consists of the words 'array data' followed by the column number of the domain values. For the example shown in Figure 6 to load this data with column 3 as the x-axis the file would then become;

```
%array data 3
choice,1st choice,2nd choice,3rd choice,4th choice,5th choice
1,1,2,3,4,5
...
8,8,16,24,32,40
```

9,9,18,27,36,45

In this case the data grid dialog is considered not necessary and the data is all loaded. In addition the file extension '.csv' can be replaced with any characters.

It should be noted that random text in these files can produce unexpected results.

## 5.5 Data files with error bars

Error bars are used to indicate uncertainty associated with a data point. The usual practice is to associate a line of length two times the standard deviation centered on the data point in the direction of the uncertainty. Occasionally the uncertainty is not symmetric about the data point being larger in one direction over the other, creating uneven error bars. The following subsections provide a method of loading uncertainty information into KGRAPH.

### 5.5.1 X EVEN ERROR BARS

Data can be entered as a list of x, y paired values with associated x-axis uncertainties. The coordinate data is entered first, in the usual way, followed by the data format command **%xerror** which indicates that the subsequent data has uncertainty in the x-axis which is entered as a list of uncertainties.

The following is an example data file which defines a set of points with error bars in the x direction.

```
%title Test data as points with even x error bars
%xlabel X label (units)
%ylabel Y label (units)
%xydata
1.050 0.900 2.000 2.500 3.000 4.000 5.000 5.500 6.000 3.500 7.540 1.890 7.160
1.600 11.600 1.1310 10.10 2.5700
%xerror ! horizontal error bars equal plus and minus dx1, dx2, dx3, dx4, dx5 ....
0.25 0.3 0.14 0.123 0.1 .15 .25 .31 .10
%end
```

This format may be repeated any number of times within a file. A plot of this data is shown in Figure 7.

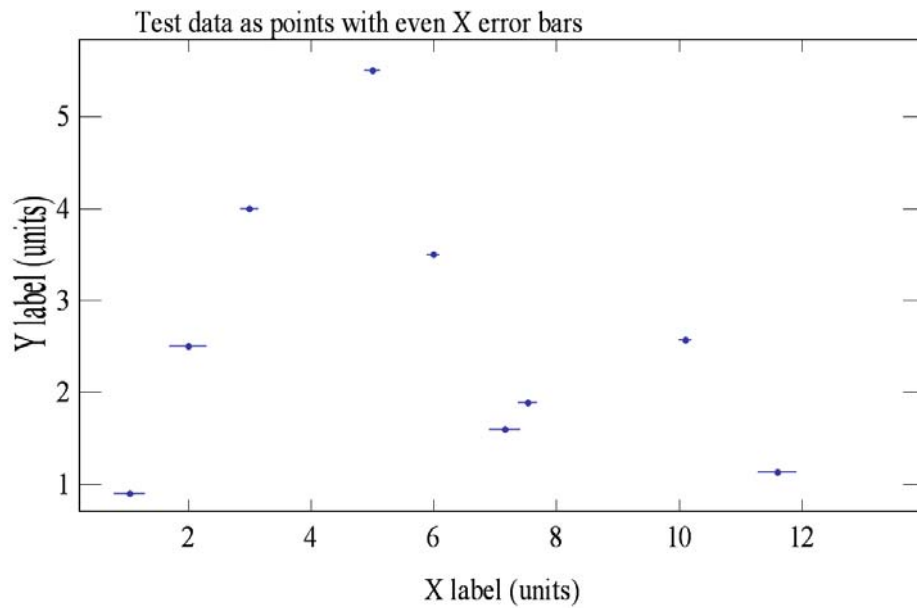


Figure 7 Plot window for the data of the above example is presented with even error bars in the x direction.

### 5.5.2 Y EVEN ERROR BARS

Data can be entered as a list of x, y paired values with associated y-axis uncertainties. The coordinate data is entered first, in the usual way, followed by the data format command **%yerror** this indicates that the subsequent data has uncertainty in the y-axis which is entered a list of uncertainties.

The following is an example data file which defines a set of points with error bars in the y direction.

```
%title Test data as points with even y error bars
%xlabel X label (units)
%ylabel Y label (units)
%xydata
1.000 1.400 1.900 2.550 3.300 4.600 5.500 5.400 6.100 3.600 7.500 2.700 8.100
1.100 10.00 1.100 11.00 2.500
%yerror ! vertical error bars equal plus and minus dy1, dy2, dy3, dy4, dy5 ....
0.25 0.3 0.14 0.213 0.1 .15 .125 .13 .1
%end
```

This format may be repeated any number of times within a file. A plot of this data is shown in Figure 8.



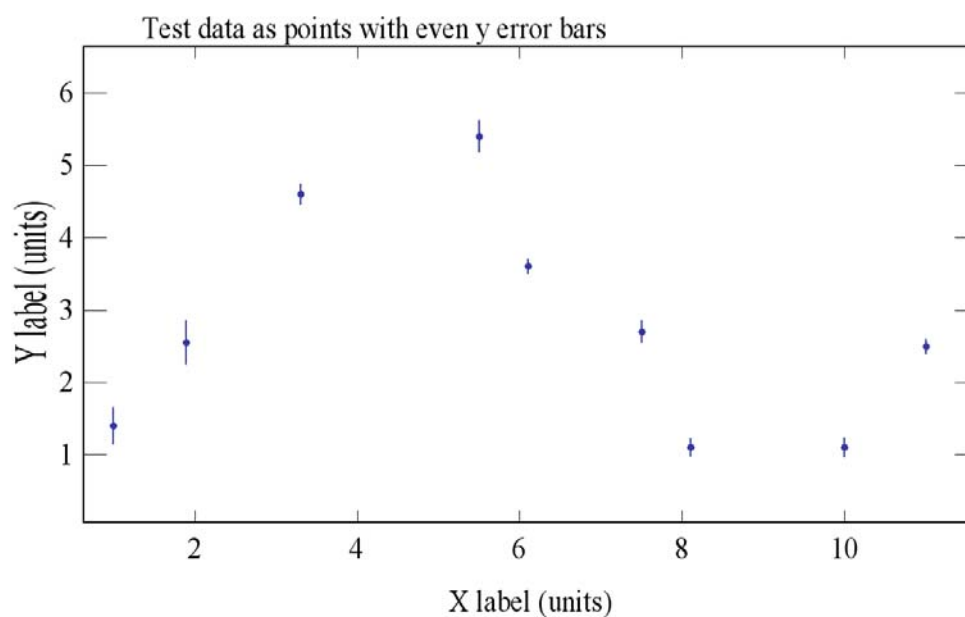


Figure 8 Plot window for the data of the above example is presented with even error bars in the y direction.

### 5.5.3 X UNEVEN ERROR BARS

Data can be entered as a list of x, y paired values with associated x-axis uncertainties. The coordinate data is entered first, in the usual way, followed by the data format command **%uxerror** which indicates that the subsequent data has uneven uncertainty in the x-axis which is entered as a list of 'upper bound' and 'lower bound' uncertainties.

The following is an example data file which defines a set of points with uneven error bars in the x direction.

```
%title Test data as points with even x error bars
%xlabel X label (units)
%ylabel Y label (units)
%xydata
.000 1.500 2.000 3.500 3.000 4.200 5.100 5.500 6.000 3.500 7.050 2.300 8.100
1.600 10.70 1.100 12.00 2.500
%uxerror ! horizontal error bars unequal plus and minus dx1U, dx1L, dx2U,
dx2L, dx3U .....%
0.25 0.15 0.3 0.15 0.4 0.154 0.23 0.123 0.12 0.21 .125 .315 .215 .325 .13 .43 .1
.51
%end
```

This format may be repeated any number of times within a file. A plot of this data is shown in Figure 9.

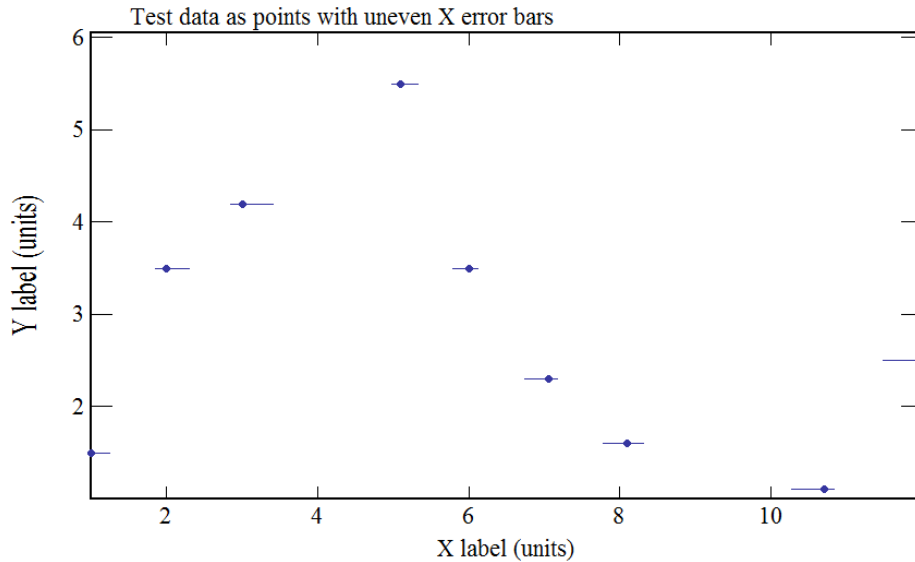


Figure 9 Plot window for the data of the above example is presented with uneven error bars in the x direction.

#### 5.5.4 Y UNEVEN ERROR BARS

Data can be entered as a list of x, y paired values with associated y-axis uncertainties. The coordinate data is entered first, in the usual way, followed by the data format command **%uyerror** which indicates that the subsequent data has uneven uncertainty in the y-axis which is entered as pairs of 'upper bound' and 'lower bound' uncertainties with upper uncertainties followed by lower uncertainties.

The following is an example data file which defines a set of points with uneven error bars in the y direction.

```
%title Test data as points with even y error bars
%xlabel X label (units)
%ylabel Y label (units)
%xydata
1.000 1.400 1.900 2.550 3.300 4.600 5.500 5.400 6.100 3.600 7.500 2.700 8.100
1.100 10.00 1.100 11.00 2.500
%uyerror ! vertical error bars unequal plus and minus dy1U, dy1L, dy2U, dy2L,
dy3U .....
0.25 0.15 0.3 0.23 0.4 0.14 0.23 0.123 0.1 0.21 .15 .315 .25 .325 .23 .43 .1 .51
%end
```

This format may be repeated any number of times within a file. A plot of this data is shown in Figure 10.

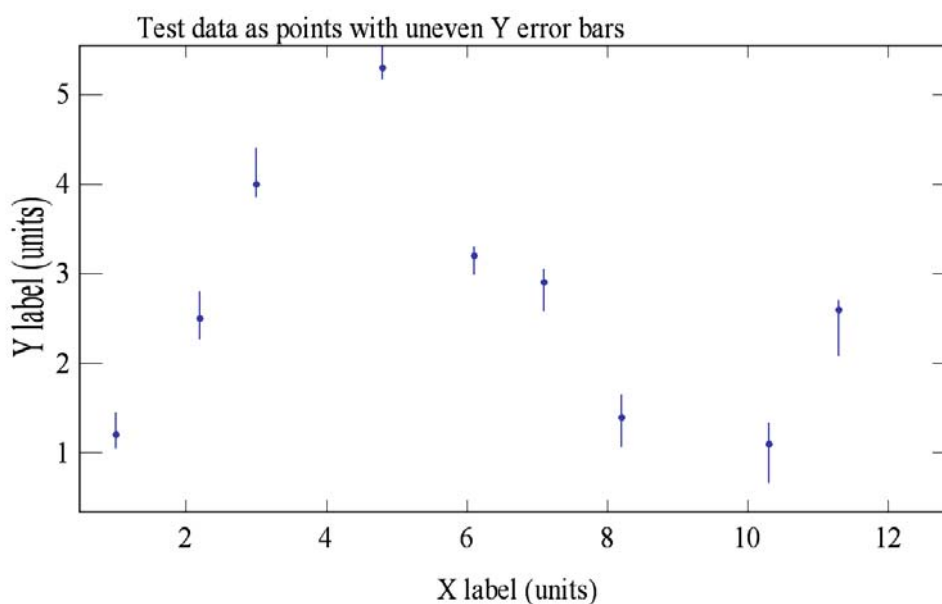


Figure 10 Plot window for the data of the above example is presented with uneven error bars in the y direction.

### 5.5.5 XY EVEN ERROR DATA

Data can be entered as a list of x, y paired values with associated uncertainties. The coordinate data is entered first, in the usual way, followed by the data format command **%xyerror** which indicates that the subsequent data is uncertainty data which is entered as pairs of uncertainty in the x value followed by the uncertainty in the y value.

The following is an example data file which defines a set of points with even error bars in x and y directions.

```
%xydata
%title Test data as points with even x and y error bars
%xlabel X label (units)
%ylabel Y label (units)
1.050 1.0500 2.200 2.500 3.000 4.200 4.510 6.500 6.300 3.250 7.300 2.800 8.100
1.000 10.30 1.140 11.20 2.5300
%xyerror ! both horizontal and vertical error bars equal plus and minus dx1, dy1, dx2,
dy2, dx3 .....
0.25 0.25 0.3 0.3 0.2 0.4 0.23 0.23 0.3 0.1 .15 .14 .25 .25 .3 .3 .1 .1
%end
```

This format may be repeated any number of times within a file. A plot of this data is shown in Figure 11.

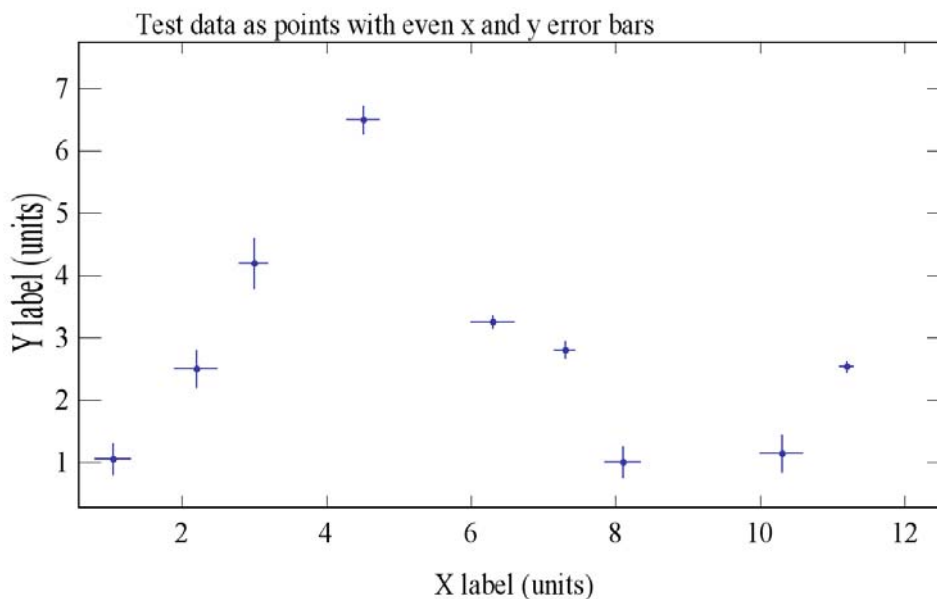


Figure 11 Plot window for the data of the above example is presented with even error bars in both the x and y directions

### 5.5.6 XY UNEVEN ERROR DATA

Data can be entered as a list of x, y paired values with associated uncertainties. The coordinate data is entered first, in the usual way, followed by the data format command **%uxyerror** this indicates that the subsequent data is error data unequal in both directions which is entered as pairs of the upper uncertainty in the x value followed by the upper uncertainty in the y value followed by the lower uncertainty of the x value followed by the lower uncertainty of the y value.

The following is an example data file which defines a set of points with uneven error bars in both x and y directions.

```
%title Test data as points with even x and y error bars
%xlabel X label (units)
%ylabel Y label (units)
%xydata
1.050 1.0500 2.200 2.500 3.000 4.200 4.510 6.500 6.300 3.250 7.300 2.800 8.100
1.000 10.30 1.140 11.20 2.5300
%uxyerror ! horizontal error bars unequal plus then minus dx1U, dy1U, dx1L, dy1L,
dx2U, dy2U, dx2L, dy2L, .....
0.3 0.13 0.23 0.13 0.25 0.15 0.25 0.15 0.4 0.54 0.4 0.14 0.23 0.123 0.13 0.53 0.1 0.21
```

```
0.18 0.21 .15 .315 .45 .315 .25 .325 .25 .325 .3 .43 .3 .43 .1 .51 .1 .51
%end
```

This format may be repeated any number of times within a file. A plot of this data is shown below. A plot of this data is shown in Figure 12.

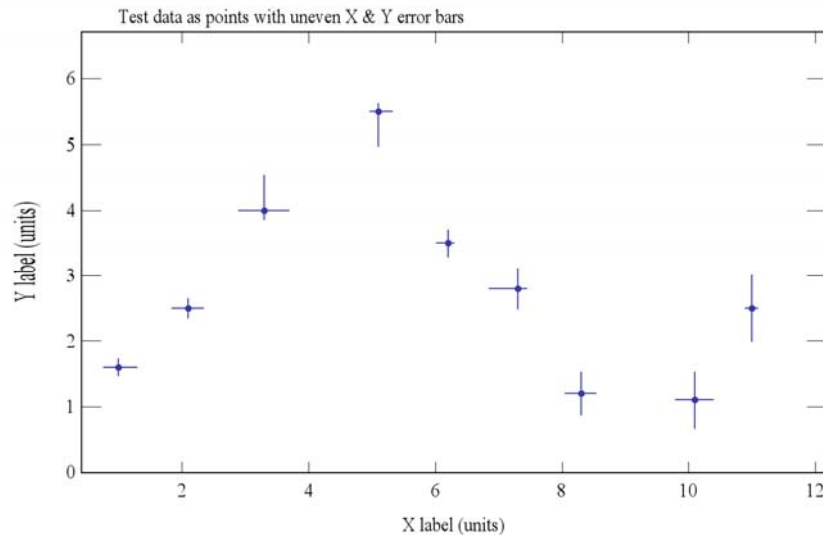


Figure 12 Plot window for the data of the above example is presented with uneven error bars in both the *x* and *y* directions.

## 5.6 TITLE LIST FILES

Title list files can be used to add title and label information to the plots in memory. These files consist of the command `%list` (can be abbreviated to `%l`) with from one up to four options (in any order) after it. The options are title, subtitle, xlabel and ylabel.

For example:

```
%list title subtitle
```

enables title and subtitle information to be specified.

The command is followed on subsequent lines by a list of the appropriate label information. Where more than one option has been used the data is separated by the '|' character. A list is terminated by the `%end` command. For example if 5 plots are assumed to be stored in memory and it is required to change the title and x-axis labels. If this was to be a single occasion the use of `TITLE` and `XLABEL` commands would have been sufficient. However the same process will be repeated on frequent occasions. It is possible to create a title file, called for example *'titles.file'* which contains information and

```
%list title xlabel
title one | x label one
title two | x label two
title three | x label three
title four | x label four
title five | x label five
%end
```

This file is used by the command APPEND;

```
1> app titles.file
1>
```

## 6. Compound Commands

The program allows for the inclusion of more than one command per line by using the '|' as a command separator. There is no checking of the validity of the individual commands and any error will be ignored.

For example:

```
1> ymult 2 | same
```

In this example the y data of the current plot is multiplied by 2 and the data is then plotted over the existing plot.

### 6.1 Repeat Count # <#> or # [<#><..<#>>]

The user may wish to modify the data of a number of plots which would require the repetition of a command or a compound command. To avoid the repetition of the commands the user may use a repeat count. To do this the user includes the hash symbol #, followed by a repeat count or range specification immediately before the command or compound command.

Compound commands become most useful when used in conjunction with the repeat count.

For example:

```
1> # 30 ave 1 200 | yshift %y- | ymult -1 | n <CR>
1>
```

In this example, first the average of the y values for x values between 1 and 200 is determined, then this average y value is subtracted from all the y data of the current plot, followed by the y data is multiplied by -1 and the next plot displayed. This process is repeated 30 times.

Further two examples are as follows :

```
1> # 20 n | ydiv 10 | xmult 2.54 <CR>
```

or

```
1> # [2..21] ydiv 10 | xmult 2.54 <CR>
```

In the first of these, the compound command increments the plot being acted upon by using the next command. In the second example, the plot incrementation is implied and is interpreted as acting on plot 2 to 21 inclusive. Care must be taken when using commands that increment the plot as they may produce unexpected results.

## 7. Start-up file

An optional startup file is included to allow a number of normally fixed parameters of the display to be altered at the startup of the program. The file is an ascii text file which contains, on separate lines, text indicating the parameter to be altered and a number. The file is called *kgraph.ini* and is stored in the directory with the main program. A summary of the attributes that can be altered by inclusion in the start-up file, *kgraph.ini* is given in Table 2.

Table 2 KGRAPH options for inclusion in the start-up file 'kgraph.ini'.

Command	Default	Description
Box	1	Box pen colour number
Wbox	1	Box pen weight
Grid	1	Grid pen weight
Overflow	99999.9	Overflow value
Plot	1000	Plot number limit
Points	20000000	Point number limit
Subtitle	90	Subtitle vertical position in pixels from the top of the screen.
Top	60	Top of box vertical position in pixels from the top of the screen.
Title	80	Title vertical position in pixels from the top of the screen.
Window	0	Window pen colour number corresponds to black.
Directory	C:\	Default directory for output files.
Help	C:\guide\	Directory location of the help files.
Xlegend	100	Width of legend
Ylegend	10	Top of legend vertical position in pixels from the top of the screen

The first three characters of each line are interpreted as the command. The parameter value is included on the same line. This allows a flexible and descriptive file to be formulated.

An example start-up file could be:

Overflow = 9999.9

Title = 40

Subtitle = 50

Top = 60

The effect of this file would be to personalize the plot appearance for the user by ;

- changing the overflow value from 99999.9 to 9999.9,
- moving the title to the left from 80 to 40 pixels from the window edge,
- Moving the subtitle to 50 pixels from the window top and
- Moving the plot box down below the subtitle.

## 8. Alphabetic command descriptions

This section presents the command descriptions in alphabetical order. Each command has an abbreviated form of three letters in most cases. The abbreviated version of the command is included in the sub-section headings in brackets. However KGRAPH will understand and accept any fraction command name from the abbreviated version to the full name.

For example the command ABOVE can be abbreviated to ABO, however ABOV and ABOVE will produce the same result.

All commands are case insensitive.

A summary of the commands are presented in Appendices A, B, C and D.

### 8.1 ABOVE (abo) <#> <#> <#>

The command, ABOVE, is used to return the proportion of the domain of the current plot than exceeds a given value.

The values after the command are:

1. the y value of interest,
2. the minimum x value,
3. the maximum x value,

A value is prompted for if the user does not specify the required value.



If the minimum x value is not defined then the currently displayed minimum and maximum x values of the plot are assumed.

If the maximum x value is not defined then the maximum displayed x value is assumed to be the maximum x value.

1> above 30 <CR>

Above 30 for 688.83 (31.9%) and 110 times.

1>

In this example the threshold value was set to 30 and the program returned the quantity of x value for which the y value was above the threshold value. In brackets is the percentage of the domain that this represents and the number of times the threshold was exceeded. An extension of the command allows the x domain to be included. Two numbers are included after the y value of interest which defines the minimum x and maximum x values of the domain, as shown in the following;

1> above 30, 0, 2159 <CR>

Above 30 for 688.83 (31.9%) and 110 times.

1>

The example data depicted in Figure 13 shows the concept behind the application of the command ABOVE. The dashed line is shown at the value required. The crossings are labelled  $x_{n,m}$  where n is the nth crossing of the value and m refers to the direction of the crossing. (1 from lower to higher y values and 2 from higher to lower y values.)

Expressing this mathematically the fraction above the line for the domain  $x_{\min}$  to  $x_{\max}$ , is given by;

$$above = \frac{\sum_{j=1}^k (x_{j,2} - x_{j,1})}{(x_{\max} - x_{\min})}$$

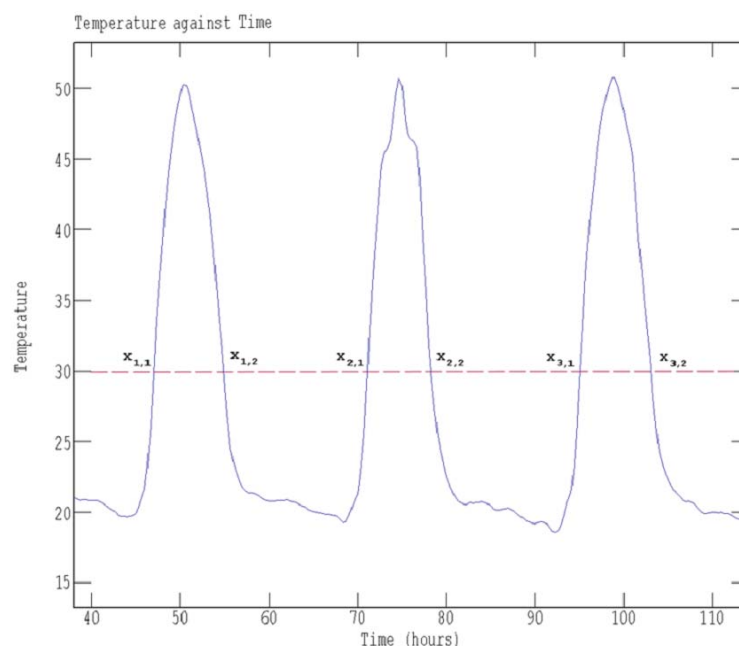


Figure 13 The concept behind the application of the command ABOVE is shown. The dashed line is shown at the value required. The crossing are labelled  $x_{n,m}$  where  $n$  is the  $n$ th crossing of the value and  $m$  refers to the direction of the crossing. (1 from lower to higher  $y$  values and 2 from higher to lower  $y$  values.)

## 8.2 APPEND (app) <#> <filename>

The command APPEND allows the user to append additional plot data to the end of the existing data in memory. After the data is read from the file, the current plot of focus is moved to the first plot in the appended information. They can be changed by inclusion of a plot number within the plot being appended immediately after the command.

A filename can be optionally included in the command line. If no filename is included the standard file requester is invoked to request a file.

The format of the appended data can be of any type of data able to be loaded.

This command can be invoked from the file dropdown menu of the plot window.

## 8.3 ARROW (arr) <h/v> <#> <#> <#> <#>

The command ARROW provides the capability of displaying an arrow on the plot. When the command ARROW is given the user is requested to indicate the position of the arrow using the cursor. The cursor is then moved to the desired position of the tail of the arrow and the left mouse button is pressed and held down. The mouse is then positioned at the point of the arrow and the left mouse button is released. The arrow is then drawn.

The command has five optional values that can be included after the command.

The first of these is an 'h' or 'v' corresponding to horizontal or vertical arrow. These cause the arrow to be drawn accurately as a horizontal or vertical arrow. The remaining four options specify the coordinates of the start and end (arrowhead) points of the arrow.

An example output where an arrow is drawn from the text 'arrow' to a point on the plotted curve is shown in Figure 14.

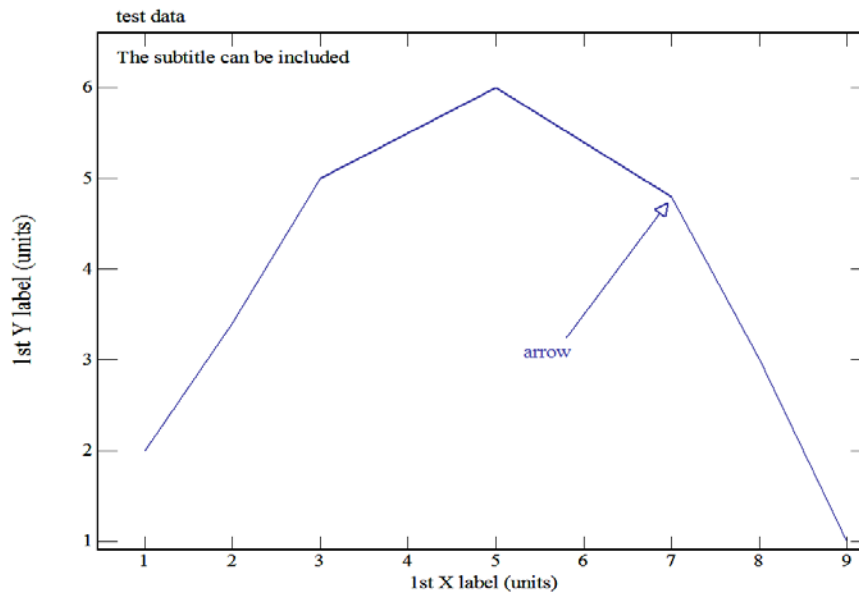


Figure 14 The user may highlight a feature on the current plot using the command, ARROW. In this plot an arrow is drawn between the text 'arrow' and a point on the plotted curve.

#### 8.4 AVERAGE (ave) <#> <#> <#> <#>

The command AVERAGE allows the user to select a range of data for the current plot and calculate the average of the data within the selected region. The command has four optional values that can be included after the command.

The values after the command are:

1. minimum x value,
2. maximum x value,
3. minimum y value, and
4. maximum y value of the section.

If the user does not supply any of these, then the program assumes that the user will select a range of values using the mouse pointer.

If the user does not specify the fourth value (i.e. 3 values after the command), the maximum y value of the displayed plot is assumed.

If the minimum y value is not defined (i.e. 2 values after the command) then the minimum y value of the displayed plot is assumed.

If the maximum x value is not defined (i.e. 1 value after the command) then the maximum x value displayed is assumed.

The number of data points in the selected region and resultant x and y values together with their associated standard deviation is displayed in the command window.

The resultant x and y averages are stored in the variables %x and %y respectively and can be used later.

For example:

```
>1 ave <CR>
----- Select area with mouse pointer
----- 20 points in window (1531.4, 852.05)..( 2059.5 , 1361.7 )
Average x value = 1766.54041 +- 104.34259
Average y value = 1123.55969 +- 124.367821
>1
```

or

```
>1 ave 1531.4 2059.5 852.05 1361.7 <CR>
----- 20 points in window (1531.4, 852.05)..( 2059.5 , 1361.7 )
Average x value = 1766.54041 +- 104.34259
Average y value = 1123.55969 +- 124.367821
>1
```

It should be noted that the optional numbers specify the domain and range of the data to be averaged. The output differs from this by presenting the two limiting points of the bounding box over which the points within are averaged.

## 8.5 BACKGROUND (bac) <filename>

The plotting area can have a background which consists as an image either in the paste buffer or in a file. KGRAPH checks whether the content of the copy buffer contains a picture before

including it. Non image information is ignored and no background will be included if there is no image in the paste buffer.

If a filename is included after the command, then the image contained in that file is loaded and scaled to fit the plot area, see Figure 15.

For example:

```
1> background <CR>
```

```
1> plot <CR>
```

```
1>
```

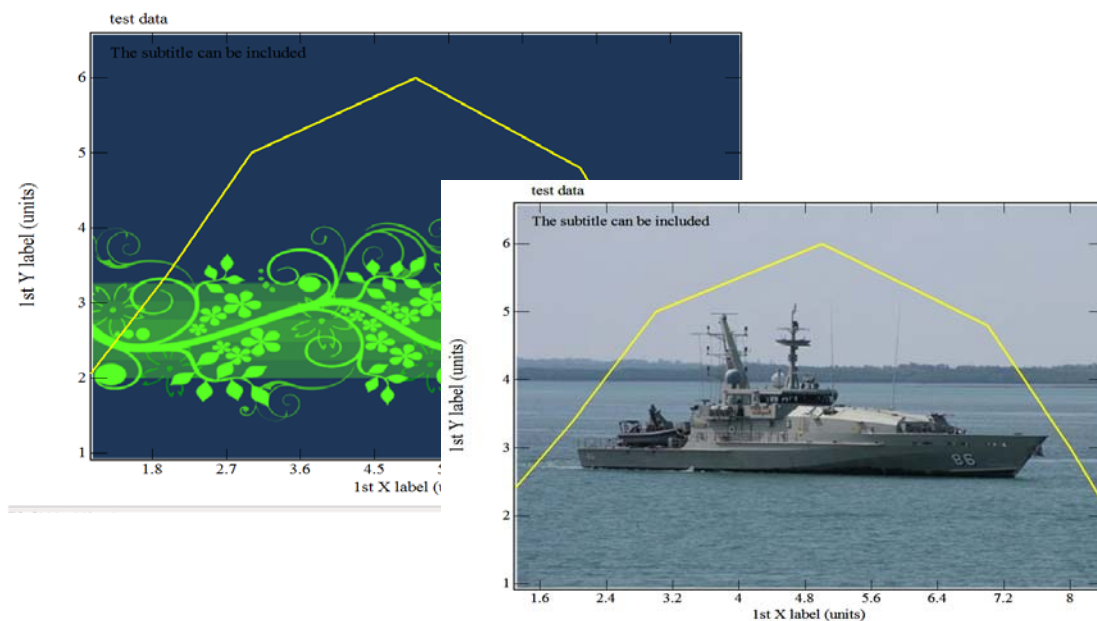


Figure 15 A background image can be added to a plot using the command BACKGROUND. The last image put into the WINDOWS paste buffer is used as the background if no filename is appended after the command.

## 8.6 BOX

The command BOX is used to draw the axes without the data. BOX requires either a previous SET or LOAD command which defines the box. The command BOX has a complementary function to the command SAME which will be defined later.

For example:

```

0> load test.dat <CR>

1> clr <CR>

1> pen box red <CR>

----- Box pen colour set to 255

1> box <CR>

1> pen blue <CR>

----- Pen colour set to 6711680

1> same <CR>

```

This command sequence will initially load the data from the file '*test.dat*' and then clear the screen and re-plot the box and data using different colours.

## 8.7 CAPTURE (cap) <#> <#>

The command, CAPTURE, allows the user to add individual points to an existing plot. This command has a number of options. The user may optionally add the point coordinates after the command (either (x, y) or (x, y, z)) or use crosshairs to locate the point to be added. The following example adds a point to the existing plot. The coordinates are written with the x value first, followed by the y value. For example:

```

1> capture 3, 5 <CR>
--- point (3, 5 ) added.
1>

or

1> capture<CR>
--- Select point position with mouse pointer.
--- point ( 3.4869,3.5166) added.
1>

```

The command, CAPTURE, can be used to digitise a graph image. To do this the following procedure can be followed:

1. Place the graph image in the paste buffer. Using for example, Print Screen button<sup>4</sup> to capture a screen image and a paste command (edit menu - paste) to place the image into Microsoft PAINT<sup>5</sup>. The image should then be trimmed to the area of interest and copied into the paste buffer (edit menu - copy).
2. In KGRAPH, use the background command to display the graph image behind the current plot.
3. Set the plot limits to match the scale of the graph image. This may be a logarithmic or linear scale.
4. Request a new blank plot.
5. Use the capture command to capture points on the curve. Repeat the process to capture the characteristics of the whole curve.
6. Save the plot.  
For example:

```
0> background <CR>
----- Background file on
0> point <CR>
----- Point plotting set
0> set 0 10 0 10 <CR>
----- Fixed axes
0> new <CR>
----- New empty plot created.
1> #100 capture | same<CR>
----- Select point position with mouse pointer.
....
1>
```

The use of the command SAME after the command CAPTURE causes the program to re-display the current data. Thus exposing the newly added points, see Figure 16.

---

<sup>4</sup> The Print Screen key is often abbreviated to 'Prt Scr' or 'Print Scrn' and is often found on the top row of the keyboard.

<sup>5</sup> Microsoft Paint is a basic image editing program which comes with the Microsoft Windows PC operating system.

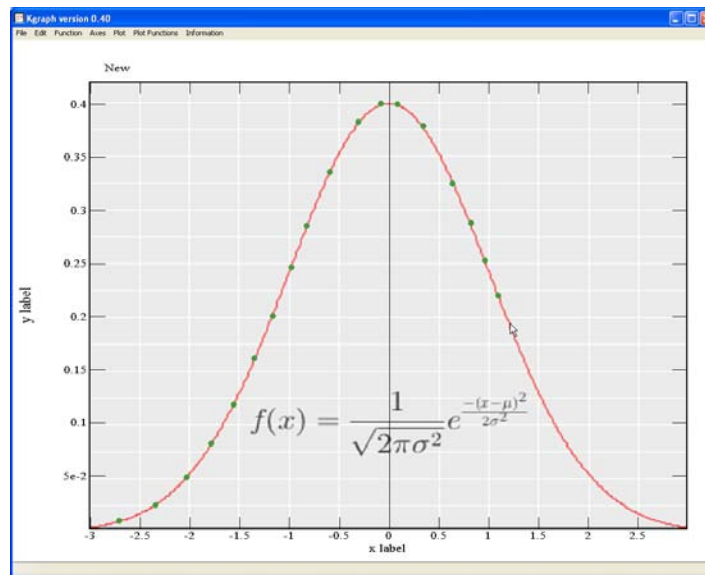


Figure 16 This plot was captured during the digitisation of an image using the method described above. The green dots are the captured points.

## 8.8 C3D - Converting point aerial density to 3-D

The density of points in a plot may provide an insight into the data that would otherwise be hidden. The command C3D can be used to convert the current plot into 3 dimensional data that can be plotted as an isometric projection or a contour plot.

The command GRID can be used to control the fineness of the grid of the rendered 3D surface.

In Figure 17 is shown the variation of pitch angle against the roll angle of a boat for a period of one hour in head seas. As expected a two dimensional plot is unable to provide information on the relative density; that is the relative time spent at any particular roll and pitch combination. However by using the command C3D, a 3-D plot showing the aerial density of the readings can be produced as shown in Figure 18. The use of the C3D command is demonstrated in the following example:

```
0> load pitch_roll.dat <CR>
1> C3D<CR>
----- Distribution store in plot 2<CR>
1> plot 2<CR>
1>
```



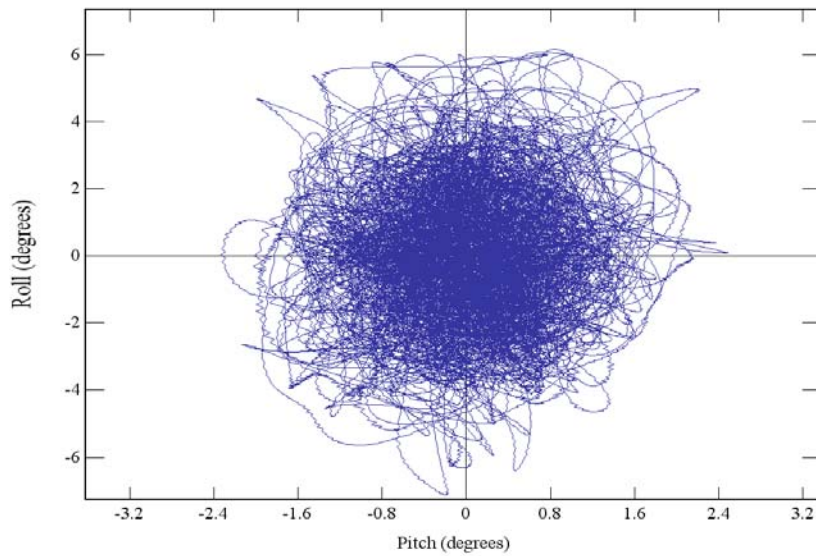


Figure 17 A plot of pitch and roll angle for a boat is displayed for a period of an hour. The time spent at any particular roll and pitch angle is clearly biased toward the origin. However, the shape of the distribution is not clear.

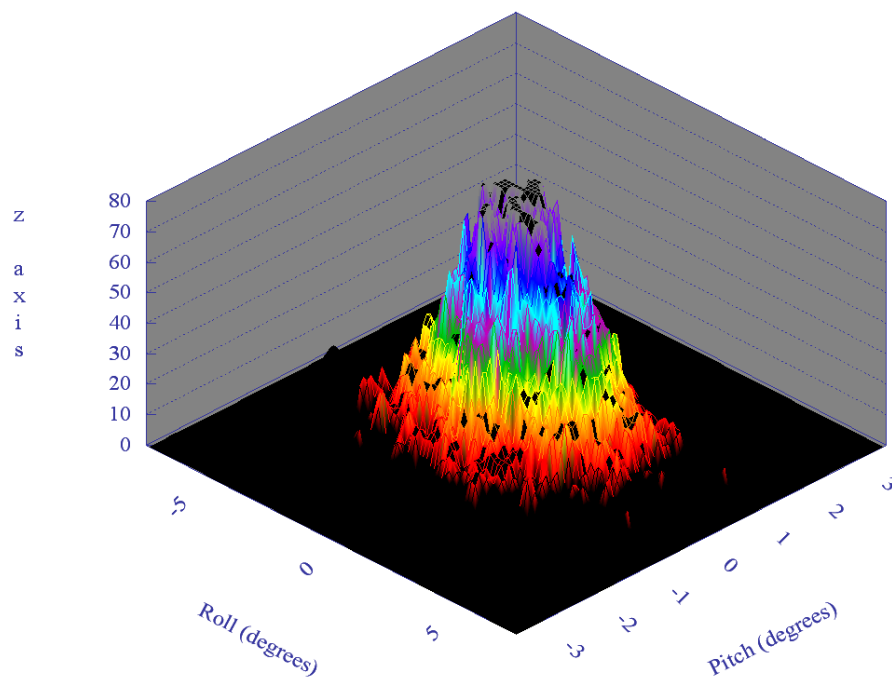


Figure 18 The relative time spent at any particular roll and pitch angle can be seen in this plot which was created by using the command C3D from the data shown in Figure 17.

## 8.9 CLEAR, (clr or cle)

The command CLEAR, when issued, will result in the plot window being cleared. This command does not clear or affect the data in memory.

The following is an example of the use of the command, CLEAR:

```
0> load test.dat <CR>
1> clr <CR>
1> box <CR>
1> pen green <CR>
----- Pen colour set to 65280
1> same <CR>
1>
```

In this example the data of '*test.dat*' is plotted and the screen cleared. The box is redrawn, the pen is changed to a green colour and the data is redrawn in the new colour.

## 8.10 COMBINE (com) <x|y|z> <#> <x|y|z> <#> <x|y|z> <#>

The command COMBINE allows the user to combine the data of different plots to produce a new plot. The number of points in each plot must be the same. No interpolation is employed.

An example of the command COMBINE is given in the following:

```
0> load test.dat
1> com x 1 y 2
----- x data of plot 1 is combined with the y data of plot 2 stored at plot 3.
1> p 3
3>
```

In this example the x values of the first plot in the file '*test.dat*' is combined with the y values of the second plot to produce a third plot which is appended at the end of the plots as a new plot, in this case as plot 3. Plot 3 is then displayed.

## 8.11 CONTINUE (c) <#> or c <[#<..#>]>

The command, CONTINUE, allows the user to plot further data onto the same axes displayed and scaled by the settings for the current plot. A single plot can be selected by appending the required plot number directly after the command or a range of plots can be selected by

specifying a range after the command. The command, without specifying a number or range, will overlay the data of the plot after the current plot.

This command can be abbreviated to the single letter 'c'.

An example using the command, CONTINUE, is given in the following:

```
0> load test.dat
1> poi
----- Point plotting set
1> same
1> c [2..3]
1>
```

In this example, the data of the first plot in '*test.dat*' is plotted, point plotting mode is selected and the same data is overlaid. The command CONTINUE is then used to overlay the second and third plots in the file.

Note: The point plotting mode is associated with the first plot only.

## 8.12 CONTOUR (con)

The command CONTOUR acts to toggle between the default projection and a colour density map of the values. The default 3D plot displays the data as a coloured isometric projection, such as that shown in Figure 19.

The data displayed in Figure 19 is a typical performance curve of a marine diesel engine and represents the fuel usage rate in the z direction against power output in the y direction and engine rpm (revolutions per minute) in the x direction. In the contour plot of Figure 20, the fraction of the range in the z dimension determines the density of the plot at a particular x and y value. The data is smoothed to provide a value at grid points.

In both Figure 19 and Figure 20 the data is only defined sparsely and is interpolated to produce a surface. However where there is insufficient data to adequately carry out the interpolation, the surface is left blank.

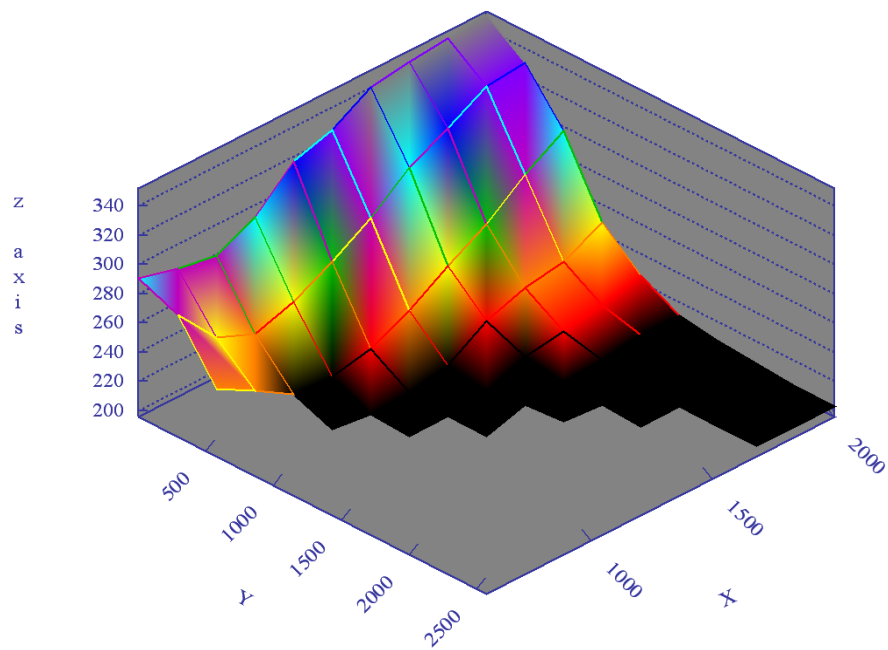


Figure 19 Shown is 3-D contour plot of a typical performance curve for a marine diesel engine. The fuel usage rate plotted in the z direction against power output in the y direction and rpm in the x direction.

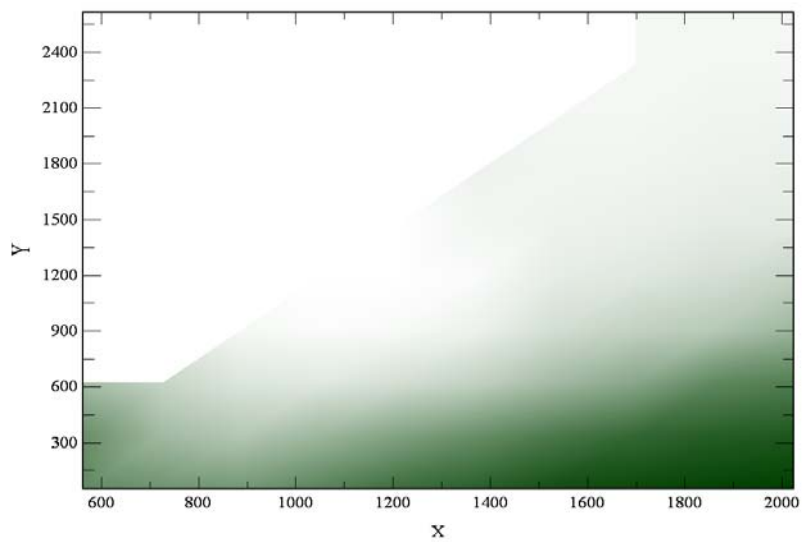


Figure 20 A colour density contour map of the data displayed in Figure 19 is produced when contour plotting is active.

### 8.13 COPY (cop) <[#<..#>] #> <or cop # #>

The command COPY allows the user to store additional copies of the plot data within the computer memory. A range of plots can be copied by issuing the command followed by a range then a destination plot number. The destination plot number cannot be specified within the range.

For example:

```
0> load test.dat <CR>
1> list <CR>

# 1 pts: 7 title: test data
Subtitle: The subtitle can be included
Source: C:\ test.dat - plot: 1

# 2 pts: 5 title: test data
Subtitle: The 2nd subtitle
Source: C:\ test.dat plot: 2

# 3 pts: 9 title: added Data
Subtitle: The 3rd subtitle
Source: C:\skeleton\old\test.dat plot: 3
1> cop [1..2] 3 <CR>
----- Copied range [1..2] to [3..4]

1> list <CR>

# 1 pts: 7 title: test data
Subtitle: The subtitle can be included
Source: C:\ test.dat - plot: 1

# 2 pts: 5 title: test data
Subtitle: The 2nd subtitle
Source: C:\ test.dat plot: 2

# 1 pts: 7 title: test data
Subtitle: The subtitle can be included
Source: C:\ test.dat - plot: 1

# 2 pts: 5 title: test data
Subtitle: The 2nd subtitle
Source: C:\ test.dat plot: 2

# 5 pts: 9 title: added Data
Subtitle: The 3rd subtitle
```

Source: C:\ test.dat plot: 3

1>

This sequence of commands copied the data of the first two plots to a position before the third plot. The destination is by default after the last plot currently in the computer memory and the default plot to be copied is the current plot.

## 8.14 CURSOR (cur) <x/y>

Often the user requires an approximate location of a feature on the plot. A cross hair cursor can be used to do this.

When the command CURSOR is issued by the user, the program waits for the left mouse button to be pressed. While the mouse button remains pressed down the cross hairs are visible and the location of the cross hairs is displayed in the text window.

As the cross hairs are moved the x and y values are updated. When the mouse button is released the cross-hairs are removed and the location is updated and control returns to the command line.

Optionally the user can include the letter x or y after the command indicating to the program that only part of the cross-hair is required. In this case only the relevant x or y coordinate is displayed.

## 8.15 DASH (das)

Dashed lines can be used to distinguish between plots drawn on the same axes complementary to the use of colour. A toggle between solid and dashed lines is implemented in KGRAPH by the command DASH together with the ancillary command LDASH.

If the command DASH is given before the command LDASH a default length of 5 for the dashes and 10 for the spaces is used.

## 8.16 DELETE (del) <[#<..#>]>

The DELETE command allows the user to delete a plot or a series of plots from those loaded in memory. To avoid unintended deletions a confirmation of the deletion is requested when the command is given from the command window.

The DELETE command by default will delete the current plot.

The user can specify a range of plots to be deleted by the inclusion of a range specification after the command. Before deleting the plots, the program asks for confirmation of the action.

For example:

```
0> 1 test.dat <CR>
1> del 3 <CR>
Delete # 3. Are you sure ?y <CR>
1> del [1..2]
Delete # 1.. 2. Are you sure ?n <CR>
1>
```

In this example the third plot of the file, '*test.dat*', in memory is deleted. Then a command to delete the first two plots is aborted.

### 8.17 DEGREE (deg)

The conversion of plot range information which consists of angles in radians to the equivalent in degrees is achieved using the command, DEGREE. The inverse operation is undertaken using the command RADIAN.

The conversion is on the y values only. To convert the x values use the command SWAP before and after DEGREE.

### 8.18 DISTANCE (dis) <#> <#> <#> <#>

The distance between two points on the earth's surface can be approximated by the use of the following formula for small to moderate distances. (< 1000 km)

The distance  $d$ , i.e. the arc length for a sphere of radius  $r$  and  $\Delta\hat{\sigma}$  given in radians, is then:

$$d = r \Delta\hat{\sigma}.$$

The average radius for a spherical approximation to the geoid, the Earth's shape, is approximately 6371.01 km. For a change in longitude,  $\Delta\phi$ , from  $\phi_s$  to  $\phi_f$  and an associated change in latitude,  $\Delta\lambda$ , a good approximation of the distance is given by;

$$d = r \Delta\hat{\sigma} = 2r \cdot \arcsin \left( \sqrt{\sin^2 \left( \frac{\Delta\phi}{2} \right) + \cos \phi_s \cos \phi_f \sin^2 \left( \frac{\Delta\lambda}{2} \right)} \right).$$

This formula has been implemented in KGRAPH to enable distances to be measured from a plot of GPS position. For simplicity, latitude and longitude are assumed to be in degrees and it is assumed that the current plot is in that format.

For example:

```
1> distance <CR>
```

```
---- Use the mouse pointer to indicate distance to measure.
```

```
---- distance from [123.06,-12.09931] to [123.445, -12.4463] = 56.934 km ( 30.7286 nm)
```

```
1>
```

When the command DISTANCE is given, the program prompts the user to position the mouse at the desired starting point. The user presses and holds the left mouse button while dragging the mouse pointer to the desired end point. While the mouse is in motion a line is drawn between the start point and the current mouse position. When the mouse button is released at the end point, the coordinates of the start and end point and the calculated distance is displayed.

Optionally coordinates can be entered after the command. In this case the latitude and longitude of the two points is included after the command in the order latitude 1, longitude 1, latitude 2 and longitude 2.

For example:

```
1> distance 123.06 -12.0993 123.445 -12.4463<CR>
```

```
---- distance from [123.06,-12.0993] to [123.445, -12.4463] = 56.909 km ( 30.7286 nm)
```

```
1>
```

## 8.19 DIFFERENTIATE (dif) <#> <[#<..#>]>

The data of a plot or a range of plots can be numerically differentiated using the command DIFFERENTIATE. Two differentiation techniques are used to differentiate the data depending on whether the data has equally spaced points or not.

In the case of equally 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 Golay and Savitzky [2].

The number of points involved in the differentiation can be altered by appending after the command the number of points to be considered either side of the point of interest (Between 1 and 9 points).

A range of data plots can be differentiated at once by appending a range after the command. If the number of points is included it must precede the range specification.



When the data has unevenly spaced points a simple three point differentiation technique is used. This method consists of fitting a parabola at each X value using a single point either side and differentiating that parabola to determine the differentiated value.

By default the data of the present plot is differentiated using three points either side of the point being evaluated.

## 8.20 ELEVATION (ELE) <#>

For 3-D data plotted as isometric projections the angle of elevation of the view can be changed using the command ELEVATION. The command sets the angle of elevation of the view from 0 to 45 degrees. The command can include the elevation angle as a number after the command. If the elevation is not included it is prompted for.

The effect of increasing the angle of elevation from 15 to 30 degrees in Figure 21 is displayed.

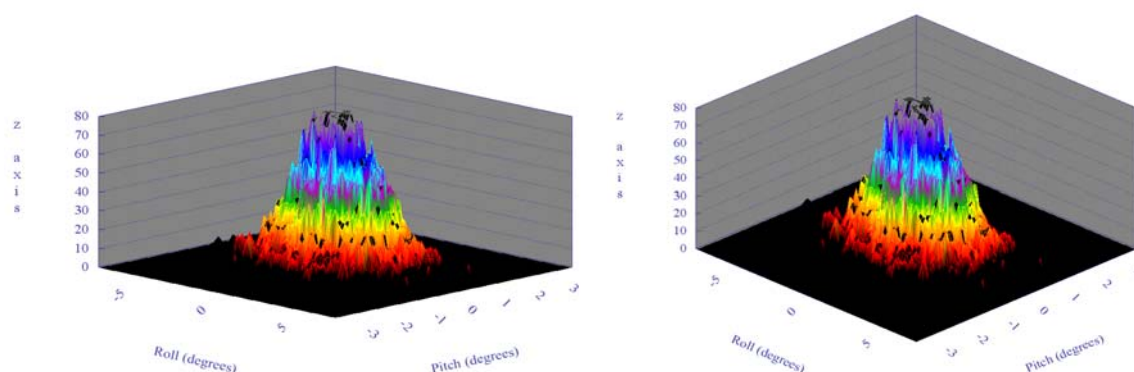


Figure 21 Two angles of elevation are shown. On the left is an elevation of 15 degrees and on the right is an elevation of 30 degrees.

## 8.21 Error bars

Uncertainty in the values plotted can arise from a number of sources and arise as a combination of random variation and systematic errors. The command ERROR allows the user to assign uncertainties to the data. When the command ERROR is executed either from the edit dropdown menu of the plot window or from the command line, a dialog box appears as shown in Figure 22. This allows for the creation or removal of uncertainty information on the current plot.

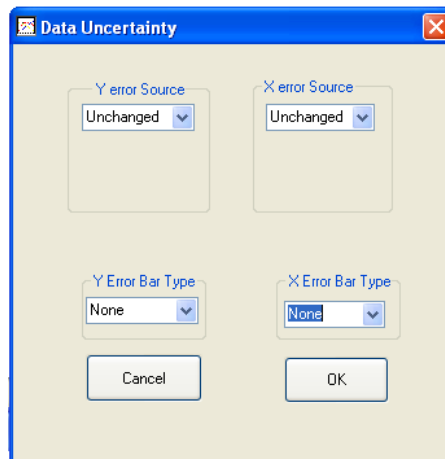


Figure 22 The error bar dialog box for generation of error bars within KGRAPH is shown.

There are four dropdown menus on the dialog. The left two refer to the y-axis and the right two to the x-axis. The two top dropdown lists define the type of error bar to be applied.

The options are presented in Table 3.

Table 3 Options for the source of uncertainty to be applied to the data.

Unchanged	No action is to be applied to this axis
Constant	All points have a constant uncertainty above and below the point value.
Percent	All points have a percentage uncertainty above and below the point value.
Fraction	All points have a fractional uncertainty above and below the point value.
Normal	All points have an uncertainty proportional to the square root of the value at that point.
In plot	The uncertainty is contained in another plot.

The two bottom dropdown lists; present options for error bars type; even, uneven or no error bars in the associated axes.

For example:

```

1>load test.dat<CR>
---- 3 plots in memory
1> Error
----- Error bars
----- Error bars altered
1>

```

It should be noted that error bar information can also be entered as part of the input data, see Section 5.5.

## 8.22 EXPORT (exp) <[#<..#>]> <filename>

Loaded plots or a range of plots can be exported in an ASCII text form suitable for input into a number of spread sheet packages using the command EXPORT. The command can include an optional plot range which determines the plot to be stored and can optionally include the file name into which the data will be stored.

If the range of plots is not included in the command line the program assumes that all plots will be stored. If the file name is not included the program will use the system file selection dialog box to prompt the user for the file command.

## 8.23 FFIT (ffi)

The command FFIT is used to fit an algebraic function to previously plotted data. When the command is entered, a function domain is asked for. When this has been entered, the program prompts for an algebraic function to be entered. The algebraic function can include:

*functions* such as sin, cos, log, abs, exp  
*variables* (A to F), which are to be optimised  
*algebraic operations* + - / \* \*\* (or ^)  
*brackets* ()

The function is expressed in terms of x, the horizontal axis. The initial value of each of the variables together with the maximum number of iterations is requested. Lastly, the maximum required size of the root mean square (RMS) error of the data from the fitted function is prompted for.

In the method used, a functional form is fitted by minimising the sum of root mean square of the difference between the data points and the fitted function. This is done by varying each variable in turn then rotating the variables and repeating the process. Naturally with a large number of variables there are a number of local minima in the summed differences. So, care must be taken when selecting the initial values of the variables to ensure a good fit.

When the required accuracy is achieved or 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 at 300 points.

For polynomials or functions that can be transformed to polynomials the command PFIT will produce a quicker result.

In Figure 23 an example where the two approaches to fitting a function to the data are given. Velocity data for a transit of a boat in a seaway is fitted by a Gaussian curve;

$$f(x) = ae^{-b(x+c)^2},$$

using the command FFIT. This is then compared to the result of fitting a polynomial of order 2 to the logarithm of the data using the command PFIT, see Section 8.59:

$$f(x) = e^{a'x^2+b'x+c'},$$

where the variables of the two equations are related by  $a' = -b$ ,  $b' = -2bc$  and  $c' = -bc^2 + \log_e(a)$ .

The two fitted curves are seen to be in good agreement.

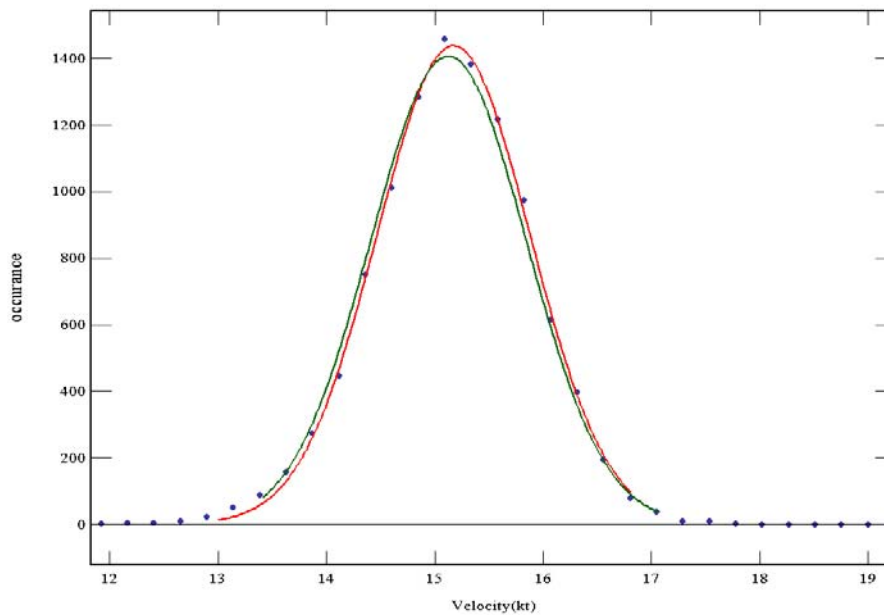


Figure 23 A comparison is shown of the fitted curves using FFIT and PFIT to the data representing the observed velocity of a boat at sea using GPS data. The red curve is the result of using FFIT to fit a Gaussian distribution and the green curve was created by plotting the data on a logarithmic scale and using PFIT to fit a polynomial of order 2 using the least squares method.

## 8.24 FFT (fft) <#>

The command FFT is used to apply a Fast Fourier Transform to the plotted data. The algorithm used to calculate the FFT is the Cooley-Tukey FFT algorithm [3]. The result of the transform is held in a new plot appended after the last plot.

The optional input is used to restrict the number of coefficients to be calculated. This is a number between 1 and half the number of points in the current plot.

The FFT algorithm requires the number of points involved to be a power of 2. In the example shown in Figure 24, 32768 points were used.

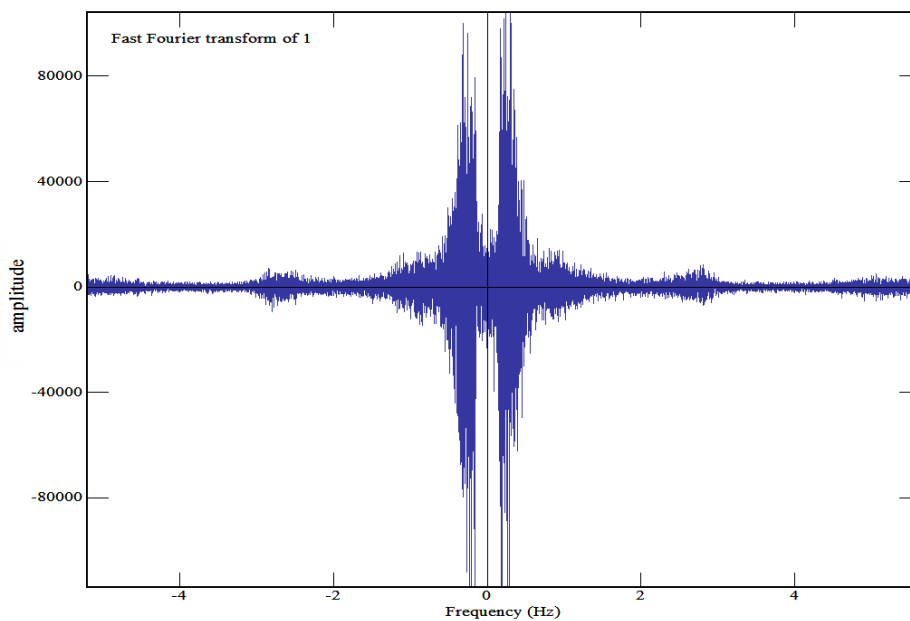


Figure 24 An FFT plot of the data presented in Figure 56. The sine terms are shown along the positive  $x$  direction and the cosine terms along the negative  $x$  direction.

## 8.25 FILL (fil)

The command FILL acts to toggle between fill and no fill for the area under the plot with the currently selected fill colour. By default the fill colour is the same as the original pen colour.

This command is available from the axes dropdown menu of the plot window.

For example:

```
0> load test.dat <CR>
1> fill <CR>
----- Filling on<CR>
1> pen fill
Fill pen colour set to 5300919<CR>
1>plot
1>
```

In this example the first plot of the file, 'test.dat', loaded in to memory. Area fill is toggled on, the fill pen colour is selected using the colour selection tool and the plot is redrawn. The result is shown in Figure 25.

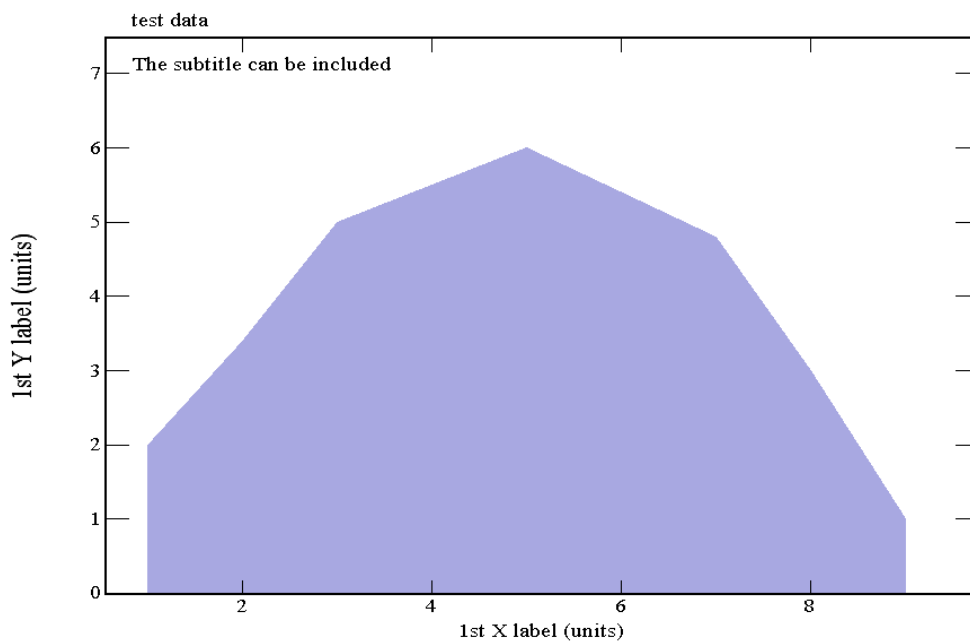


Figure 25 The result of the FILL command on the displayed data of 'test.dat' is shown.

More complex data can require some planning to get the desired effect using the FILL command. In the example shown in Figure 26 the data has differing x values and thus the histogram displayed does not align.

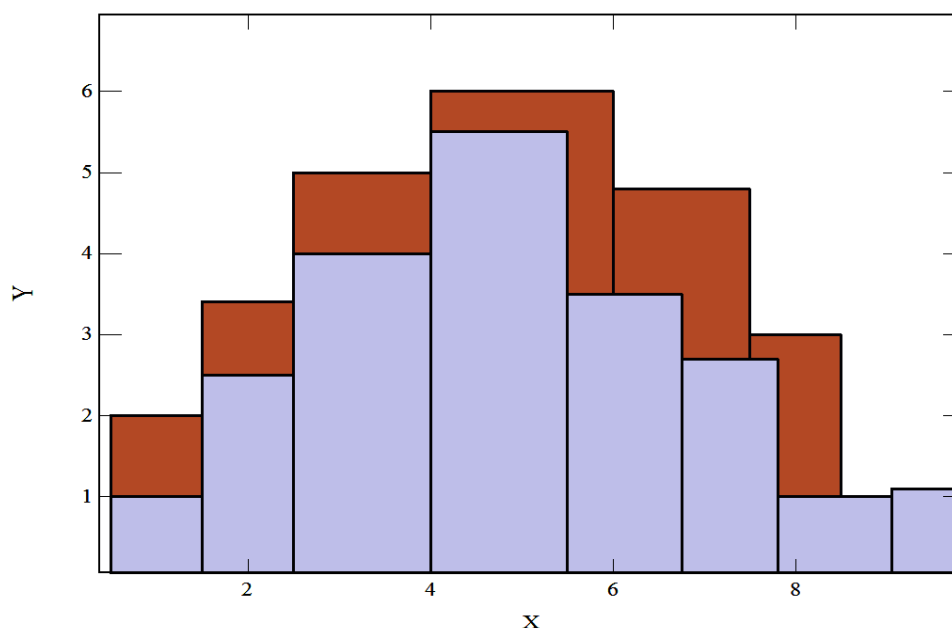


Figure 26 Histogram plotting combined with FILL will allow the user to display histogram data where the information has differing domains.

## 8.26 FLOWOVER (flo) <#>

Often data taken in experimental setups can have data associated with an over flow value. For instance, this could indicate that the input voltage exceeded the range of possible readings of a digital voltmeter. The use of this command effectively masks these data values from being plotted. The command FLOWOVER sets this overflow value. The default value is set to 99999.9. The numerical value can optionally be included directly after the command.

When KGRAPH has a value equal to the out of range value the point is not displayed and for a line plot the connecting lines are not shown, see Figure 27. Displayed is the result of the inclusion of an overflow entry at an x value of 3. The line plot is discontinued from the prior point to the following point.

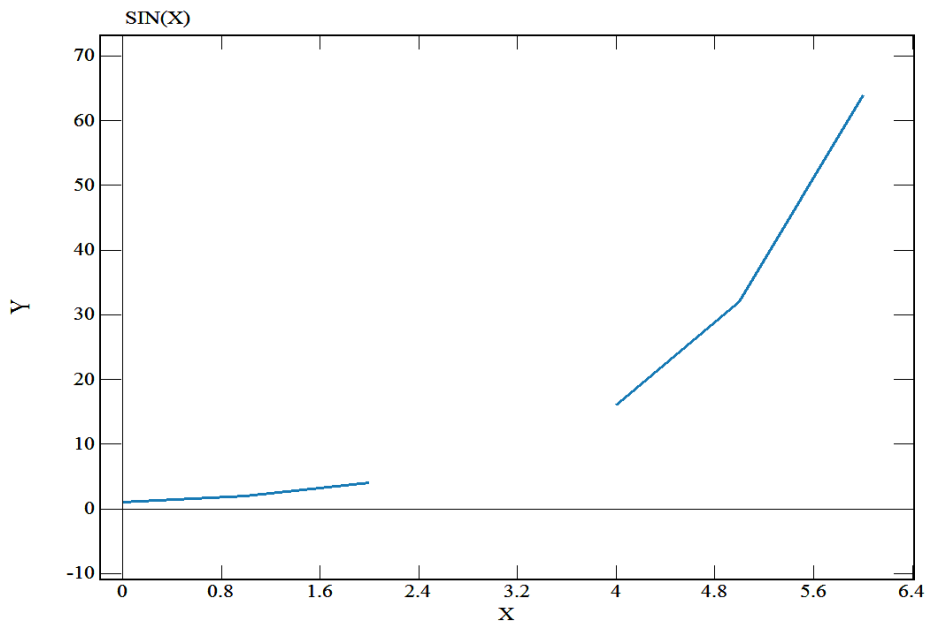


Figure 27 The result of the inclusion of an overflow entry at an x value of 3 is displayed. The line plot is discontinued at the prior point and recommenced at the following point.

## 8.27 FONT (fon) <#>

The command FONT allows the user to alter the size of the font being plotted. The size of the text is set using the command FONT followed by a positive number. The number corresponds to a scaling of the default font size, which is set to have a font width of 1/100<sup>th</sup> the width of the plot area.

For example:

```
1> FONT 1.5<CR>
Character size set to 1.5 <CR>
```

This will produce a text size 1.5 times the size of the default size.  
The default size is restored when the command is given without the scaling factor.

For example:

```
1> FONT<CR>
Character size set to 1.0 <CR>
```



## 8.28 FOURIER (fou) <#> <[#<..#>]>

The command FOURIER is used to apply a discrete Fourier transform to data. A Fourier transform is an integral of the form;

$$F(u) = \int_0^{\infty} f(x) e^{-i2\pi ux} dx$$

The transform operates in complex domain. The imaginary exponent can be written as:

$$e^{i\phi} = \cos(\phi) + i \sin(\phi)$$

For a sampled function, the continuous transform turns into a discrete one;

$$F_n = \sum_{k=0}^{N-1} f_k e^{-i \frac{2\pi}{N} kn},$$

Where N is the number of data points, k refers to the contribution of each of the contributing frequencies.

When the command is entered without options, the currently displayed data is transformed to produce a new plot of coefficients. Coefficients with negative value are used for the sine coefficients and coefficients with a positive value are the cosine coefficients.

The discrete Fourier transform is numerically intensive and will take some time before the result is stored as a plot appended after the last plot in memory. For an Intel core duo processor running at 3.1GHz it took 20 seconds to undertake a discrete Fourier transform of 72000 points.

## 8.29 Free axes commands

**FREE (fre)**  
**XFREE (xfr)**  
**YFREE (yfr)**  
**ZFREE (zfr)**

The command FREE is used to remove the fixed axes dimensions resulting from the use of the commands XSP, YSP, ZSP, SET, XSET, YSET, ZSET and ZOOM.

Note that the commands SET, XSET, YSET and ZSET can be used to remove the fixed axes dimensions by repeating the command without a data range.

The commands XFREE, YFREE, ZFREE apply to the x, y and z-axis respectively. An example of its use could be;

`0> set 0 10 0 10 <CR>`

```

0> load test.dat <CR>
1> free <CR>
1> plot <CR>
1>

```

### 8.30 FUNCTION (fun)

The command FUNCTION (FUN) allows the plotting of simple algebraic functions. On receiving the command FUNCTION, a domain over which the function is to be plotted is prompted for. Once a suitable domain has been entered, the function is entered as an algebraic expression. The algebraic function can include;

<i>functions</i>	sin()	Sine angle in radians	
	cos()	Cosine angle in radians	
	abs()	Absolute value	
	exp()	Exponentiation of the number	
	asn()	Inverse sine function returns the angle in radians	
	acs()	Inverse cosine function returns the angle in radians	
	atn()	Inverse tangent function returns the angle in radians	
	log()	Logarithm of the number base 10	
	lgn()	Logarithm of the number base e	
	int()	Integer component of the number	
	sqr()	Square root of the number	
	erf()	Error function evaluated at the number	
<i>variables</i>	A to F	Numbers which are to be entered later	
<i>algebraic operations</i>	+ - / * ** (or ^)		Standard mathematical operations
<i>brackets</i>	()		Brackets which set the order of operations explicitly. These can be 10 deep.

The function is expressed in terms of x, the horizontal axis. All angles are expressed in radians.

For example:

```
0> fun <CR>
Input the domain of the function (minimum, maximum):0.1 10 <CR>
Input the number of points (20 or more): 100 <CR>
-----
The input function line is an expression involving X
that can contain; up to 10 brackets.
functions: sin, cos, tan, asn, acs, atn
log (base 10), lgn (base e)
abs, int, sqr , erf
operators: +, -, /, *, ** (or ^)
variables: a,b,c,d,e,f,g
-----
Enter a function line
:y = 0.4*x^2 -3*x-2 <CR>
1>
```

In the example, the function  $y = 0.4x^2 - 3x - 2$  was plotted over the range 0.1 to 10 and evaluated at 300 points. This is shown in Figure 28.

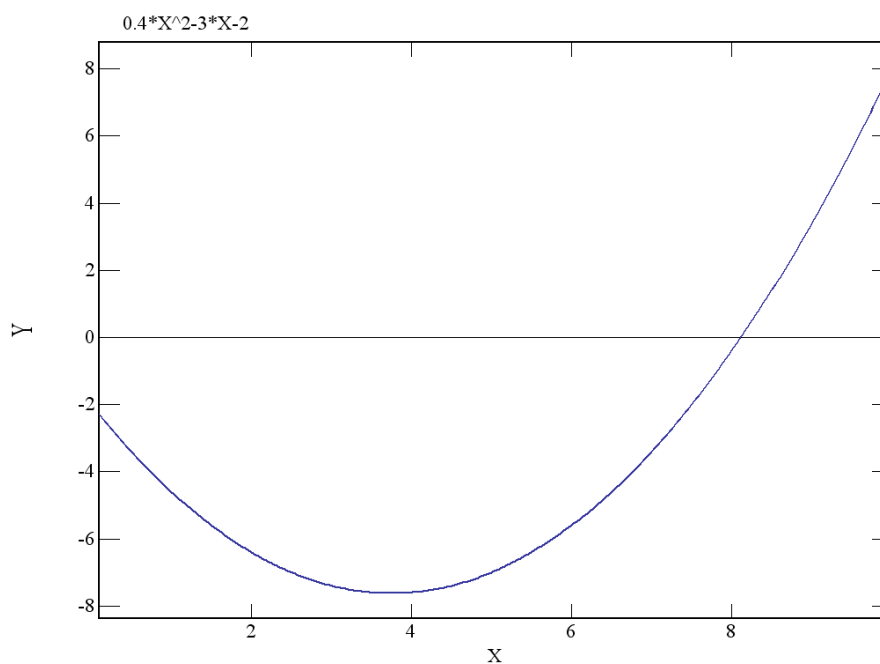


Figure 28 The resultant plot for the generated function,  $y = 0.4x^2 - 3x - 2$  over the domain 0.1 to 10 using the command FUNCTION.

### 8.31 GRID (gri)

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, see Figure 29. GRID is a command which toggles, i.e. the command is cancelled by repeating it.

The pen colour of the grid can be changed using the PEN GRID <#> command, see Section 8.58.

For example:

```
1> grid <CR>
1> ---- Grid on
1> grid <CR>
1> ---- Grid off
1>
```

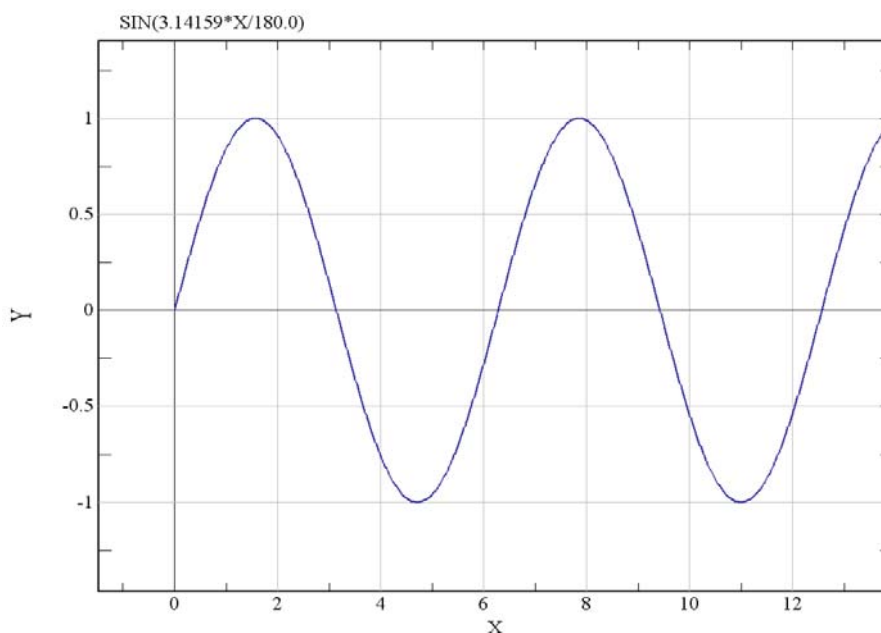


Figure 29 An example plot with grid lines is shown. The grid is applied to the major marks. In this example a sub-mark is shown between the numbered marks on the graticule.

### 8.32 G3D (g3di) <#> <#> for 3-D plots

The 3-D plotting uses the available points to create a surface which is what is displayed. This is based on a grid derived by default from the spacing of the data. This grid size can be changed using the command G3D. Two optional numbers can be inserted in the command line corresponding to the number of grid points in the x and y directions respectively. If only

one number is included it is assumed that the number of grid points is the same in both directions. If no number is included the number of grid points is prompted for.

The effect of grid spacing on the result is shown in Figure 30. The plot shown in Figure 30a was created with a grid of 40 by 40 and the plot shown in Figure 30b with a grid of 80 by 80. If the local data distribution is not well determined then panels are plotted black. Reducing the grid number creates larger panels which will be more likely to contain data but the resolution is correspondingly reduced.

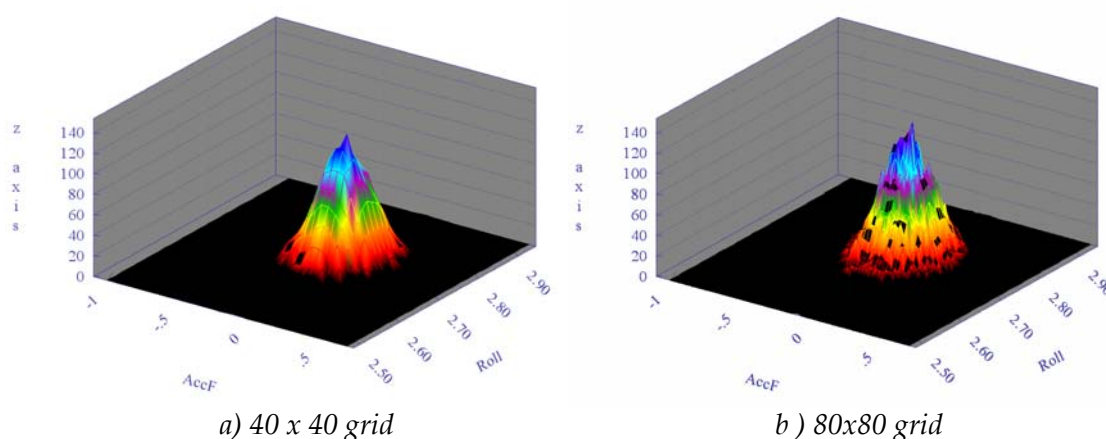


Figure 30 The effect of grid spacing on the result is shown.

### 8.33 HELP (hel) (or ?)

This guide help file can be initiated from within KGRAPH using the command HELP or the “?” key and can be used for in line help.

### 8.34 HIGHLIGHT (hig)

When examining data such as a GPS track of a boat, it may be desirable to ascertain the direction of motion on a particular stretch of the track. The command HIGHLIGHT produces an arrow head on the curve in the direction of motion at the highlighted position.

For example:

```
0> load track.dat <CR>
1> pen vary 4 5 15<CR>
1> plot <CR>
1>highlight
----- Select point on curve. Click right button or a key to exit
```

In this example the file 'track.dat' contained a GPS track in plot 1 and a corresponding speed record in plot 4. A small section of the plot was displayed. The command HIGHLIGHT was used to highlight the direction of motion at a number of points along the track. The points are selected using crosshairs. An arrow head is placed on the curve at the point closest to the crosshair as the left mouse button is clicked. This is repeated until the right mouse button or a key is pressed. The result can be seen in Figure 31.

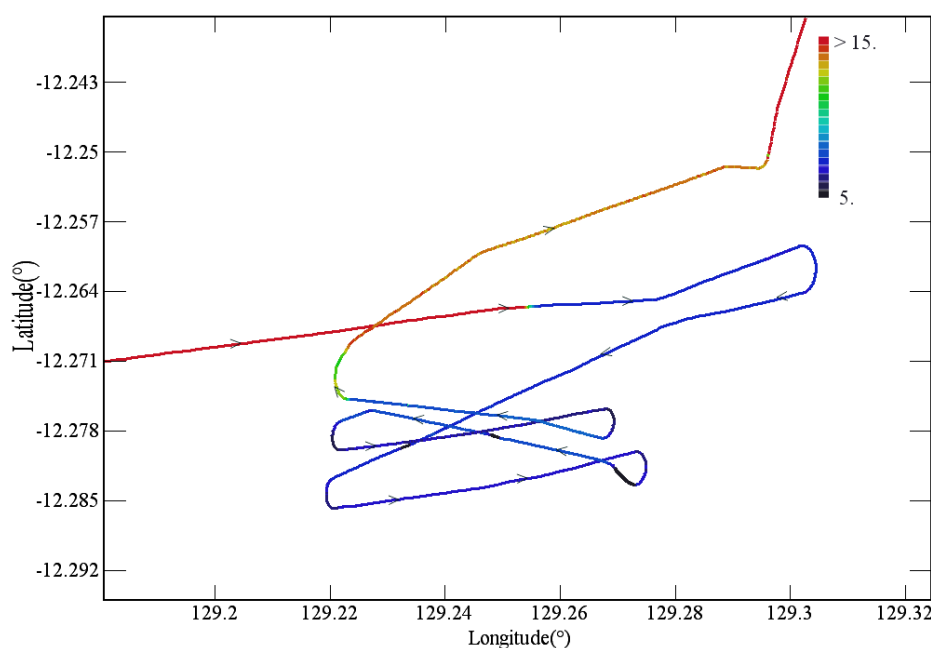


Figure 31 A segment of a boat's voyage is shown. The result of using the command HIGHLIGHT can be seen as arrow heads indicating the order of plotting or the direction of motion.

### 8.35 IF ELSE ENDIF

Conditional statements to control the flow of a macro are available in KGRAPH. The IF command contains a logical comparison that causes a macro to either execute the commands prior to the END IF command when true or to skip over them when false.

An ELSE command between the IF command and the END IF command will alter this flow. Commands prior to the ELSE command will be executed and those after the ELSE and prior to the END IF command will be ignored when the IF command logical comparison is true. If the logical comparison is false then reverse occurs; the commands after the ELSE and before the END IF are executed and those between the IF and ELSE are ignored.

### 8.36 INCREMENT (inc) <#>

The command INCREMENT (inc) allows the user to alter the plot pointer so that subsequent actions are on the new plot. When the command INCREMENT is given the pointer is incremented by 1. If an integer is appended after the command, then the pointer is incremented by the integer. The command is most useful in compound commands where the user may wish to skip plots in a routine fashion.

For example:

```
1> plot | inc 8 | pen + | c | inc 8 | pen + | c <CR>
```

The result of this is that the user is plotting the 1st, 9th and 17th plots in different pens on the same axes.

### 8.37 INTEGRATE (int) <[#<..#>]>

The command INTEGRATE causes the data of a plot or a range of plots 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 before using the command INTEGRATE.

The integration is carried out using Simpson's rule. Each point on the plot is the integral of the data from the lower limit of the x values up to that point.

For example:

```
0> load test.dat <CR>
1> int [1..2] <CR>
1> plot <CR>
1>
```

In this example the data of the first plot in the file '*test.dat*' is plotted. The data of the first two plots are integrated and the integrated data of the first plot is displayed, see Figure 32.

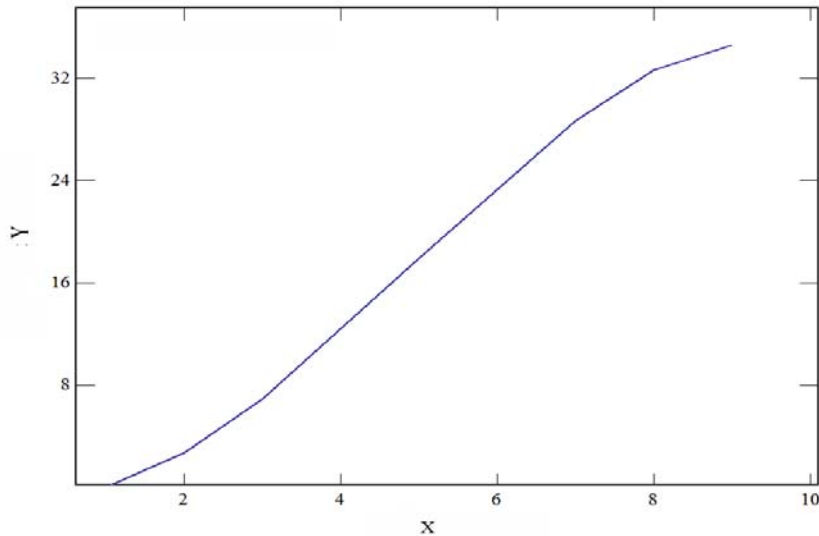


Figure 32 This plot is the result of INTEGRATION of the data of plot shown in Figure 14.

### 8.38 INTERPOLATE (interp) <#>

To create a smooth curve where the data in a plot is sparse may require the data to be interpolated. The command INTERPOLATE is used to interpolate the data displayed in the plot and produce a curve consisting of equally spaced points across the width of the displayed domain. If the data is not sorted into sequentially increasing x values a poor result can be expected. Use of the SORT command prior to interpolation 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 considered the interpolated value. The weighting is determined by the distances from the two data points either side of the point being interpolated.

For example:

```
0> load test.dat <CR>
1> interpolate <CR>
Input the number of points : 100 <CR>
----- Data interpolated at 101 points over 100 intervals.
1> same <CR>
1>
```



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

The interpolated data using this interpolation method is continuous and differentiable. Overshoot is kept to a reasonable level.

Interpolation of data with error bars is done by interpolation of the data and the error bars. Interpolation is not able to be made easily for data with uneven error bars and is currently not allowed.

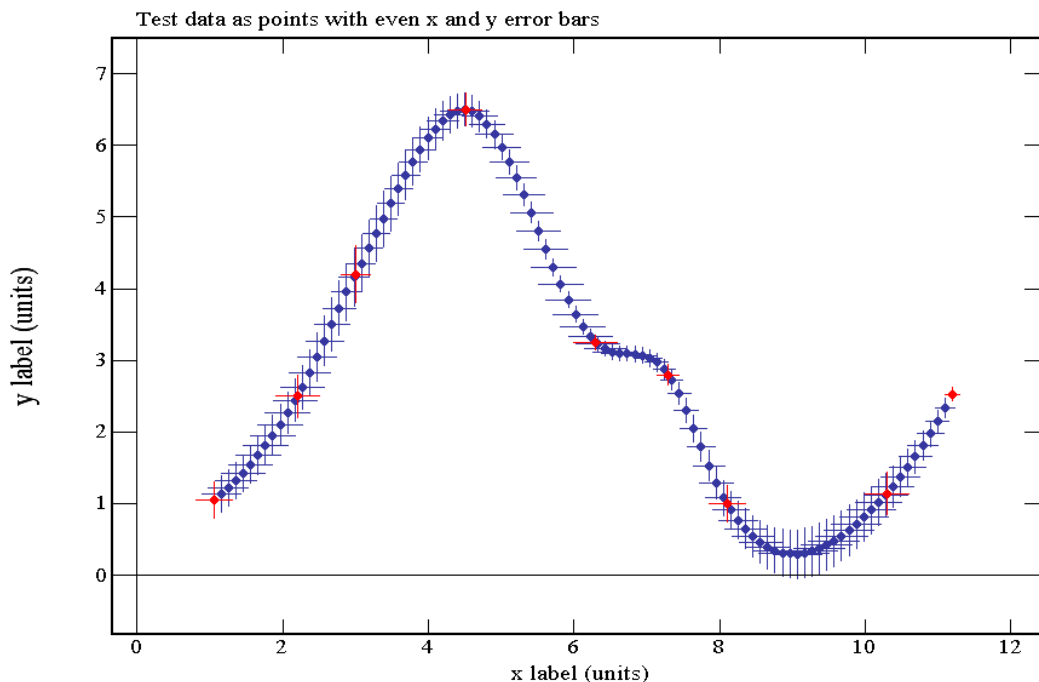


Figure 33 The effect of interpolating points with error bars in both directions is shown. The red points are the original data points and the blue points are the result of an interpolation with 100 intervals.

### 8.39 INFORMATION (info)

The command INFORMATION results in the program displaying a dialog window as shown in Figure 34. This dialog window initially displays information about the currently displayed plot and includes plot title and subtitle, x and y axis labels, pen selection, x and y-axis scale and marks. The user can interact with the dialog to alter individual setting and values. Navigation within the dialog allows the setting of values for any plot in memory.

Clicking on the 'pen' or 'fill' button will open the colour selection tool, see Figure 44. The colour of the pen used to draw the plot data or the pen used as a fill colour is updated when the 'OK' button is selected.

Selecting the named checkbox will replace the colour selection textbox and button with a dropdown list of 50 standard named colours selected from the HTML standard named colours [4].

For example:

2> info  
>2

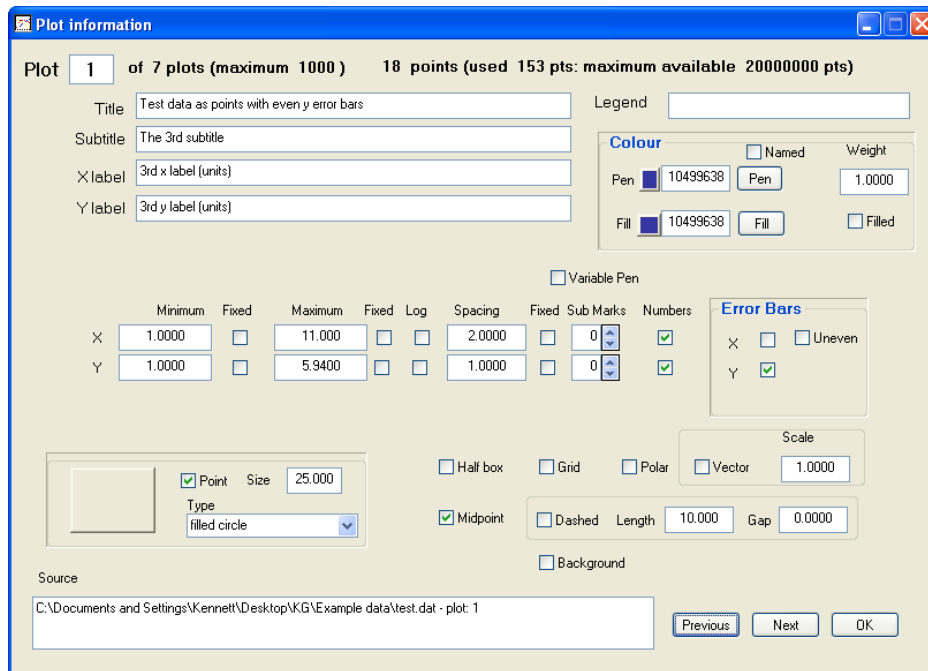


Figure 34 An information dialog window is shown. Text can be cut and pasted into all text boxes. Fixed checkboxes restrict the plot variables for this plot when checked.

## 8.40 LDASH (lda)

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. It should be noted that the unit of length used here is one thousandths of the x-axis length.

For example:

```
1> ldash <CR>
INPUT THE DASH LENGTH (1,100) : 20 <CR>
INPUT THE SPACE LENGTH (1,100) : 10 <CR>
1> dash <CR>
```

```
---- dashed lines set  
1> dash <CR>  
---- solid lines set  
1>
```

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

## 8.41 LEGEND (leg) <num>

The command LEGEND can be used to toggle on and off a legend. The first 15 characters of each plot title are used to fill the legend entry immediately after the command is given. If no title exists the program uses the plot number in memory. The present version does not update the legend if more plots are added subsequent to the LEGEND command. To avoid problems it is recommended to use the command to toggle the legend off and on again after appending more files.

The contents of the legend text can be altered using the command INFORMATION and entering or copying into the provided legend text box, see Section 8.39.

Optionally, if the plot number is considered adequate, the inclusion of 'num' will use the plot number in the legend. An example plot displaying legends is shown in Figure 35.

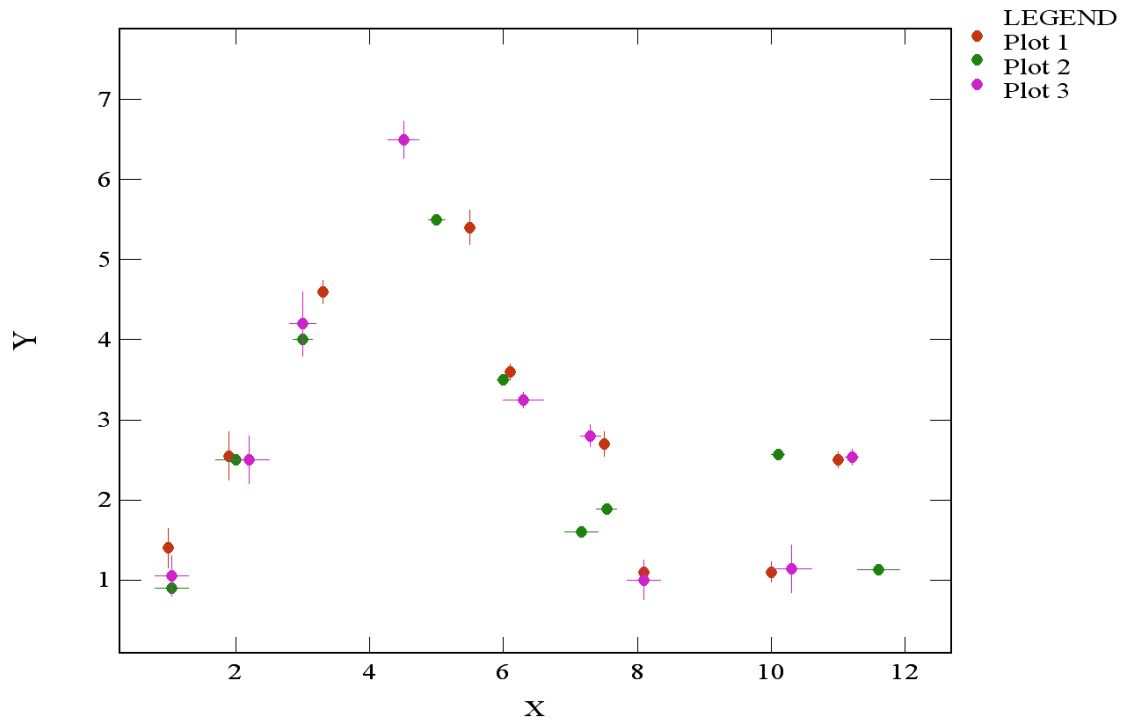


Figure 35 Test data with a variety of error bars is used to show the position of the legend. The text in the legend has been altered using the legend textbox in the information dialog window.

#### 8.42 LIST (lis) <[#<..#>]> file <filename>

A list of information on the plots currently in memory is displayed using the command LIST. A range of plots can be included after the command to restrict the information to the plots of immediate interest.

In addition, the output can be directed to a file by including an optional file name after the command. To enable the selection of a file name using the file requester, the keyword 'file' (which can be abbreviated as 'f') can be included after the command to indicate the intention to direct a copy of the output to a file.

For example:

```
0> new <CR>
1> load test.dat <CR>
1> list <CR>
```

```
# 1 pts: 18 title: Test data as points with even y error bars  
Subtitle: The 3rd subtitle  
Source: C:\test.dat plot: 1
```

```
# 2 pts: 18 title: Test data as points with even x error bars  
Subtitle: The 4th subtitle  
Source: C:\ test.dat plot: 2
```

```
.  
.
```

```
# 7 pts: 27 title: Test data as points with uneven x error bars  
Subtitle: The 8th subtitle  
Source: C:\test.dat plot: 7
```

### 8.43 LOAD (loa, l) <#> <filename>

To load a data file for plotting, type the command LOAD followed by a carriage return key in response to the '#>' prompt. The program will then use the standard system requester to allow the user to choose a data file. When the user selects the data file, the selected file is read into memory and displayed.

An extended version of the command provides for the inclusion of a plot number to be displayed and a filename after the command (see example below). If no plot number is provided then the first plot in the file is displayed. If no filename is given the standard system file requester is displayed to allow for interactive selection of the data file.

For example:

```
0> load 23 test.dat <CR>  
23>
```

The data of '*test.dat*' is loaded into memory and the 23rd plot is displayed. An example of the result of the use of the LOAD command can be seen in plot window in Figure 36 and in the command window in Figure 37.

When the loaded data file has a first plot consisting of x, y, z data an isometric projection of the data is plotted. An example is shown in Figure 38. The default output is scaled to a grid covering the data.

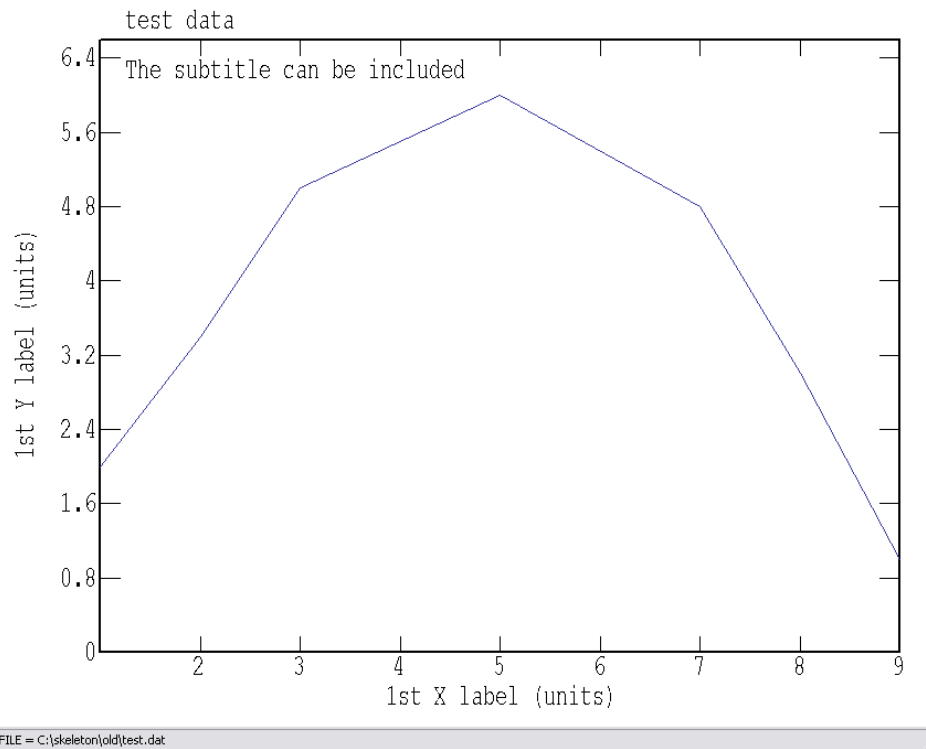


Figure 36 Shown is the plot window following the LOAD command for test.dat. The information contained is loaded and plotted.

```

Using Startup file
Top of box set to 690
Box pen weight set to 2
Default directory set to d:\temp\

0>load
  Reading plot data
  3 plots in memory
1>

```

FILE = C:\skeleton\old\test.dat

Figure 37 The command window reports the status of the data reading process as well as listing the commands. The prompt shows the current plot number of focus. A plot number of 0 indicates that no data is loaded into memory.

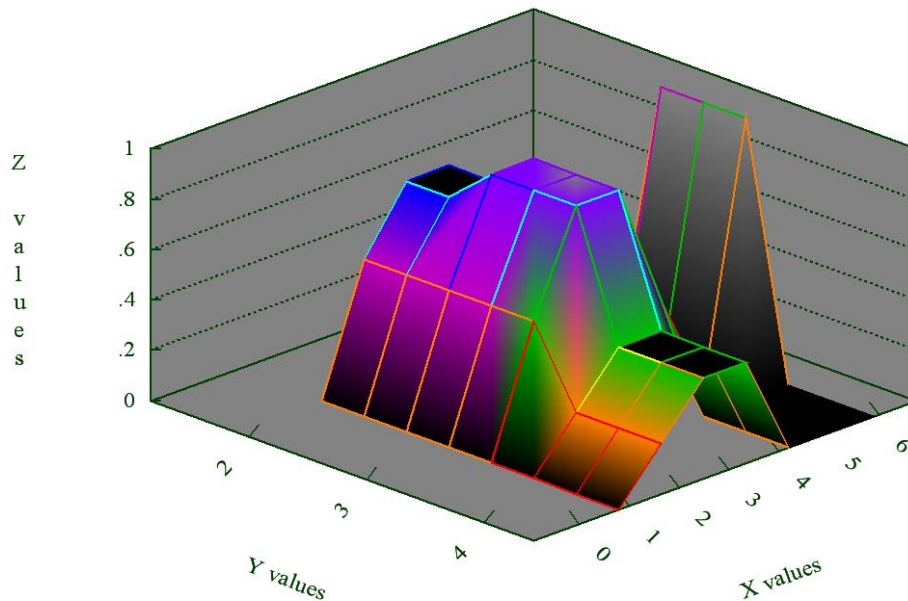


Figure 38 When the loaded data file has a first plot consisting of x, y, z data an isometric projection of the data is plotted. The default output is scaled to a grid covering the data.

#### 8.44 MACRO (mac) {go | stop | quit | record | # | filename}

Macro files are sequences of commands that can be initiated using the MACRO command.

KGRAPH supports the use of macro files which consist of KGRAPH commands.

Commands which require a response by the user will still require user input. The macro sequence is halted while the user input is given.

Command	Abbreviation	description
Macro	mac	Toggle a macro file on and off. If no macro file is running, a file name is prompted for using the standard file requester
Macro Go	mac g	Continues the existing Macro file after a macro stop
Macro Stop	mac s	Halts the execution of a Macro file. This enables commands to be manually added into a macro sequence during execution. During recording this command quits the macro recording and closes the file.
Macro quit	mac q	This command quits the execution of a macro file and closes the macro file.
Macro record	mac r	This command allows the user to record KGRAPH commands in a file. The file name is prompted for using the standard file requester. If a macro file is open when the

		command is given the file is closed and a new file opened.
Macro #	mac #	Starts the execution of the macro file 'kgraph.macro.#'. If an existing macro file is running, the file is closed and the new one is started. This allows the chaining of macro files.
Macro <fname>	mac <fname>	This command has the same effect as macro # except that the file <fname> is used.
Macro slow	Mac sl	This command acts to toggle the plotting of data during the execution of a macro. This will slow down the macro process significantly for large looping macros with large datasets.

There are a number of symbols which are useful in writing macros and are used in place of a specific plot number or parameter. These are:

%ld	The value of the last destination plot of the last command producing a new plot;
%p	The position of the current plot;
%n	The number of points in the current plot;
%min	The minimum x value of the current display;
%max	The maximum x value of the current display;
%ymin	The minimum y value of the current display;
%ymax	The maximum y value of the current display;
%dmax	The maximum y value of the data associated with the current plot;
%dmin	The minimum y value of the data associated with the current plot;
%x	The last displayed x value from one of the following commands VALUE, AVERAGE, PEAK;
%y	The last displayed y value from one of the following commands VALUE, AVERAGE, PEAK;
%l	The position of the last plot;
%r	The last numeric result;
%lnz	The last point with a non zero y value in the current plot;
%fnz	The first point with a non zero y value in the current plot.

The commands Macro # and Macro <fname> have two optional inputs which allow limited user input to a macro. These are five variables which are accessed within the macro using the symbols %a, %b, %c, %d and %e. For example:

There exists the macro file '*test.macro*' which contains,

```

reset
load test.dat
legend

```



```
ylog
pdiv %a %b
p %ld
```

which can be used to plot the result of dividing the y values of plot 2 and 3 on log-linear axes with legends on using the command:

```
1> mac test.macro 2 3 <CR>
1>
```

A second series of variables %v1 to %v99 can also be used as an array of results.

Conditional execution of macro commands is implemented by the inclusion of IF and ENDIF commands.

Looping within macros is provided by the macro commands LOOP and END LOOP. The command LOOP has two choices:

- a number after the command which indicates how many times the loop will be repeated; or
- a logical condition which if true allows looping to continue.

Loops may be nested up to 10 deep.

#### 8.44.1 Example Macros

The following macro is an example which combined the data of a number of plots into a single plot from a number of files. Each file has a number of plots and the plots of interest are quite large requiring the use of interpolation to reduce the point numbers. It provides an example of the use of looping and variables. The variable, a (%a), is used as part of the file name and as a number.

```
!macro to join data of a number of plots into a single plot.
!%a contains the number of the plot.
!assume that the plots have been brought into memory in increasing time
%a = 0
load datafile%a.csv
mov 4 1
delete [2..%l]      !delete unwanted plots
p 1
%c = %max           ! Set variable c to the maximum of the current domain
interp 1000         ! Interpolate data to reduce the number of points
loop while %a < 1715
  inc %a 1
  app datafile%a.csv !append the next file
  mov 5 2           !move the plot of interest
```

```

DEL [3..%l]           !delete unwanted plots
p 2
interp 1000           !reduce the count per hour to 1000
xshift %c             !shift plot to align current minimum with domain maximum
papp 2 1              !append the new data to the end of the old
p 1
%c = %max             ! Set variable c to the maximum of the current domain
inc %c 3.6            !add an assumed time difference for the next point (3600/1000)
loop end

```

This second macro completes a similar task of joining plots. Two file sequences each containing a series of plots are brought together to form a single series of plots. In this case the variable %d contains the last file number and the variable %a contains the initial file number. The variable %e contains the number of points and should be set to zero prior to launching the macro.

```

!macro to join a number of plots into a single plot.
!%a contains the number of distinct plots.
!assume that the plots have been brought into memory in increasing time
app datafile%a.csv
app data2file%a.csv
#[29..38] xshift %e      !move the x values up by the previous maximum
%b = 1
#19 papp 20 %b | inc %b 1
if %a < %d then
    inc %a 1
    c 10
    %e = %n
    mac ../join.mac
end if

```

This example shows the use of an IF statement and the reentrant use of the macro command which calls itself. It should be noted that the macro and the data files are not in the same directory. The macro file is in the parent directory indicated by the '../'.

#### 8.45 MARK (mar) <x/y/z> <#>

The command MARK allows the user to specify the number of sub-marks between the major marks on the axes. The command has a number of options.

The number of sub-marks on all axes can be individually altered by the including the axes to which the command is to be applied, directly after the command.

For example:

```
1> mark x 5 <CR>
```

Number of x sub-marks set to 5

1>

If the axes are not specified then the command is applied to both axes.

1> mark 5 <CR>

Number of x sub-marks set to 5

Number of y sub-marks set to 5

1>

The use of the command MARK is demonstrated with the example plot shown in Figure 39.

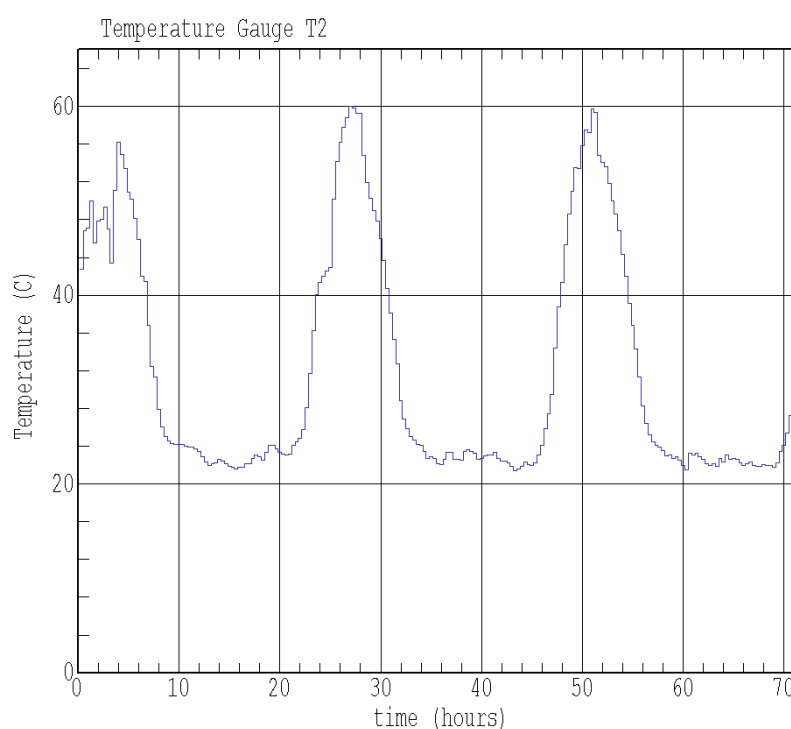


Figure 39 A plot demonstrating the use of sub-marks.

## 8.46 MIDPOINT (mid)

The command MIDPOINT acts to toggle between successive data points being connected by straight lines and the points being plotted as histograms.

For data that is not in ascending order it may be necessary to execute the command SORT with the data before plotting in the histogram mode.

For example:

```
0> load test.dat <CR>
1> mid<CR>
----- Histogram plotting set
1> mid <CR>
----- Midpoint plotting set
```

#### 8.47 MOVE (mov) <[#<..#>] #> <or mov # #>

The command MOVE allows the user to reorder the existing plots in memory. This is probably best used with the list command. A range of plots can be moved by issuing the command followed by a range then a destination plot number. The destination plot number cannot be specified within the range.

For example:

```
0> l test.dat <CR>
1> list<CR>
-----
#      Points      Title
-----
1         7    test data
2         5    test data
3        10   added Data

1> mov [1..2] 3 <CR>
1> lis <CR>
-----
#      Points      Title
-----
1        10   added Data
2         7    test data
3         5    test data

1>
```

This sequence of commands moved the first two plots to the third plot. Of course an equivalent command could have been;

```
1> mov 3 1 <CR>
```

or

```
1> mov [3] 1 <CR>
```

which moves only the third plot to the first position.

## 8.48 NEW

The command NEW is a request for an empty plot in memory and is used with the capture command when entering data directly into KGRAPH.

For example:

```
0> new <CR>
1> cap 1,2 <CR>
1> cap 2,4 <CR>
1> cap 3,8 <CR>
1> plot <CR>
```

In this example, a new plot is created, three data points are added and the data is plotted. The result is shown in Figure 40.

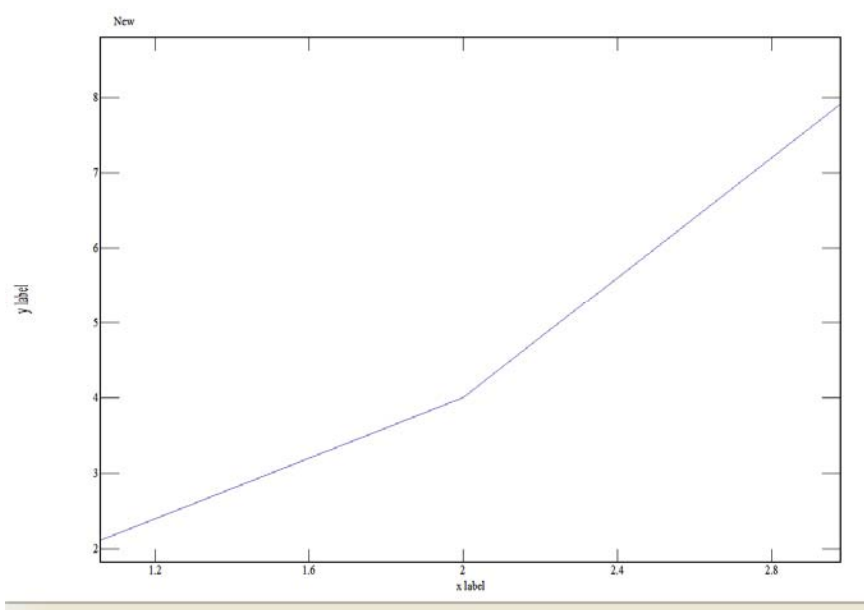


Figure 40 The resultant plot is shown for a plot created using the command CAPTURE to add three data points to a new plot.

## 8.49 NEXT (n)

The command NEXT is a request to plot the next plot in memory when plotting a multiple plot file. The screen is cleared and the axes and data are drawn. If there is no following plot data an error message is given.

For example:

```
0>load test.dat <CR>
1> next <CR>
2>
```

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.

### 8.50 NOISE (noi) <#>

Digital signal data often has associated with it noise spikes created by short random electrical signals. These detract from the signal under examination. The command NOISE is a method to filter short period spikes from the plot data. By default a signal is considered noise if the signal both rises and falls over 3 data points where the difference between the points is 0.1 of the displayed range. An optional number after the command specifies the fraction of the visible range that a noise spike will exceed.

For example:

```
0> load test.dat <CR>
----- 3 plots in memory
1> noise 0.1 <CR>
1> pen red <CR>
1> same <CR>
```

In this example, the first plot file of '*test.dat*' is plotted, then data is filtered with the noise filter function, the pen is changed and the filtered plot is overlaid. In Figure 41 is shown the result of such an application of the command NOISE to a set of noisy data. The red spikes in the original data are replaced by the mean of the surrounding points as seen in the green line.

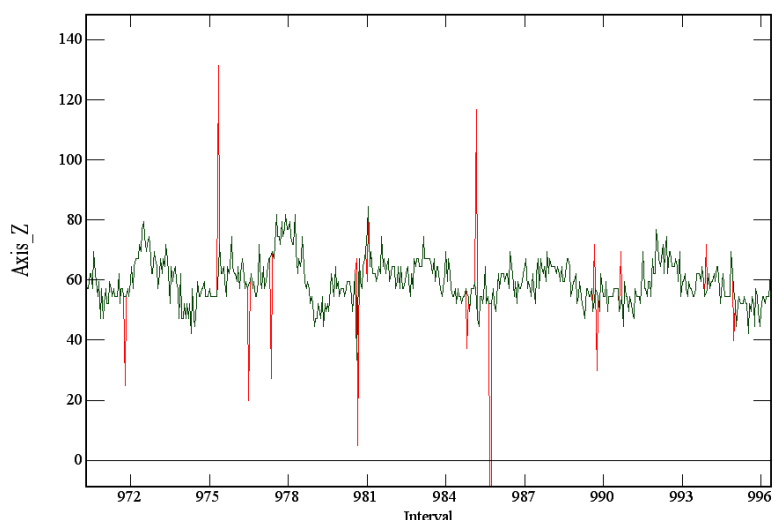


Figure 41 The effect of the noise spike removal algorithm is shown. The original data was plotted in red and the same data set after applying the NOISE command is overlaid in green.

### 8.51 OCCURRENCE (occ) <#> <#> <#>

The command OCCURRENCE enables the creation of a histogram of the occurrences of y values in a plot. For each point pair in the plot, the y value is assigned to its appropriate bin and the bin count is increased by 1. By default the number of equally sized bins is set to 100 with the minimum and maximum bin set to match minimum and maximum y values. Three optional numbers can be included after the command. The first is the bin width, the second is the minimum y value and the third is the maximum y value.

For example:

```
0> load test.dat <CR>
```

```
1> occ 1 0 25 <CR>
```

In this example, the first plot in the multiple plot file 'test.dat' is plotted, and then a histogram of the occurrences of y values between 0 and 25 is made with a bin width of 1.

### 8.52 OUTPUT <filename> | quit

The command OUTPUT allows the user to save the text output to a file for later use. A filename included after the command OUTPUT is checked for when the command is first used. If none is included the program will prompt for a filename. Subsequent OUTPUT commands act to toggle the output file on and off. This allows the user to control the contents of the file.

If a new filename is included after the command the current output file is closed and the new file is opened for output.

The extension to the command, 'OUTPUT QUIT' allows the saving of text output to be stopped and the output file closed.

For example:

```
0> load test.dat <CR>
---- Wait, Reading
... Read 100% of data.
Done.
1> output outfile.txt <CR>

---- Output text file to outfile.txt

---- text output file on

1> ave <CR>

---- 4 points in window ( 1.795, 3.300)..( 7.452, 6.043)
Average y value = 4.800 +- .9274
Average x value = 4.250 +- 1.920

1> output quit
1>
```

The file *outfile.txt* now contains the text

```
---- 4 points in window ( 1.795, 3.300)..( 7.452, 6.043)
Average y value = 4.800 +- .9274
Average x value = 4.250 +- 1.920
```

### 8.53 OVERLAY (o) <#> <filename>

After plotting data from a file, a second set of data from another file may be overlaid onto the plot. This is achieved by using the command OVERLAY. The OVERLAY command is issued followed by a carriage return. The program then requests a file name in exactly the same way as for load. If no filename is given the standard system file requester is displayed to allow for interactive selection of the data file.

This differs from the APPEND command in that the plot is not cleared when the new data is brought in.

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



```
0> load test.dat <CR>
1> overlay 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:

```
1> overlay 3 test2.dat <CR>
```

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

### 8.54 PADDITION (pad) <#> <#> <#>

The y values of two plots can be added together using the command PADDITION. The command is optionally issued with three numbers immediately after the command. The first number is the plot number to which the data will be added, the second number is the plot number of the data being added and the third is the plot number where the result will be placed.

If the domains of the two plots differ the data of the second plot is sorted and interpolated so that the domain values of the two plots match before being added.

If the third number is not included the summed plot is appended to the end of the existing plots.

An example of the command PADDITION is shown in Figure 42. Plotted is the result of the application of PADDITION to Plot 1 and Plot 2 to produce Plot 1 + Plot 2.

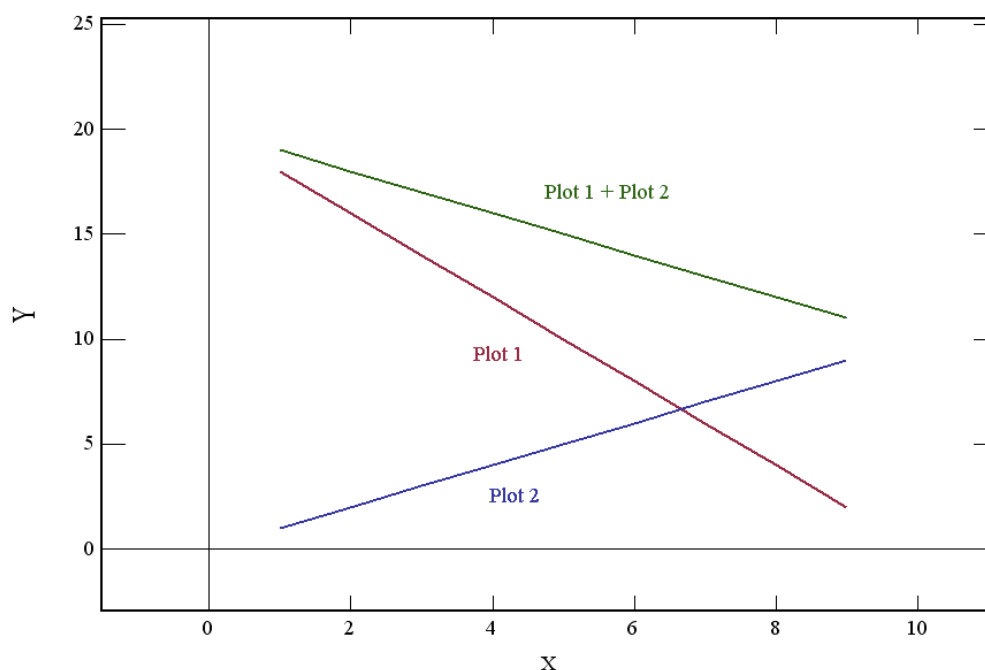


Figure 42 Plotted is the result of the application of PADDITION to Plot 1 and Plot 2 to produce Plot 1 + Plot 2. For each value along the x-axis, the y value of the two source plots is added to produce a plot of the sum.

### 8.55 PAPPEND (pap) <#> <#>

This command allows the user to combine the data of two plots into a single plot. Two optional inputs can be included specifying the two plots to be combined.

If only one is included the other is prompted for.

If neither is included then both plots are prompted for.

Note that neither plot is retained in its pre-appended form. The user should first copy the plots if either is required.

### 8.56 PDIVIDE (pdi) <#> <#> <#>

The y values of a plot can be divided by the y values of a second plot using the command PDIVIDE. The command is optionally issued with three numbers immediately after the command:

- The first number is the plot number of the data being divided;
- the second number is the plot number of the data forming the divisor; and
- the third is the plot number where the result will be placed.

If the domain values of the two plots differ the data of the second plot is sorted and interpolated so that the domain values of the two plots match before being divided.

If the third number is not included the resultant plot is appended to the end of the existing plots.

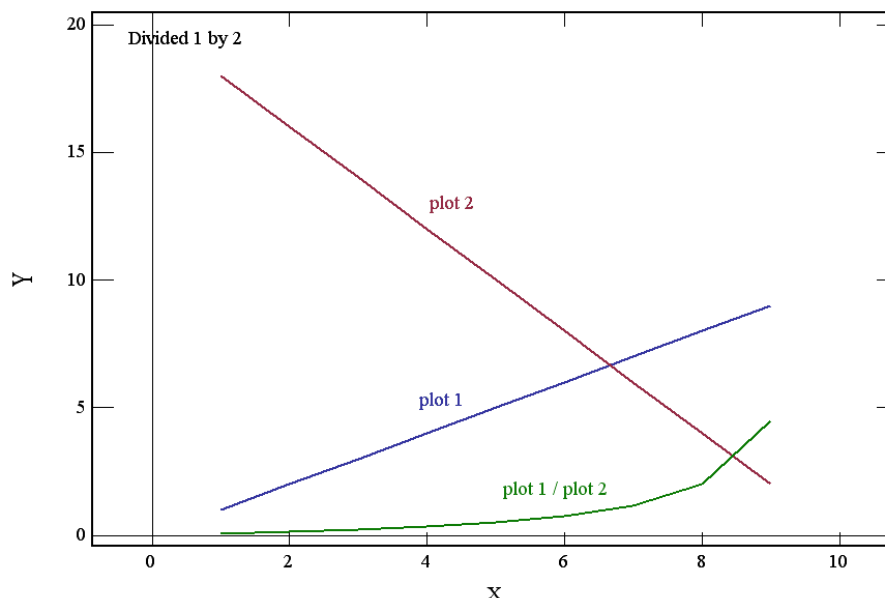


Figure 43 The result of the command PDIVIDE 1 2 is shown. For each x value in plot 1, the y value is divided by the corresponding y value of plot 2.

### 8.57 PEAK (pea) <#> <#>

The command PEAK allows the user to obtain information about selected peaks (or valleys) in plot data. Two sets of peak shape information are displayed.

The user selects a domain in which a peak of interest exists. The selected domain must include sufficient points on either side of the peak for a linear background fit to be made. The domain can be entered by including the lower and upper limits after the command. If none are included the program assumes the limits of the current plot window.

The program uses 10 points at each end of the selected domain to determine a linear fit to the background. The program then scans the selected region for the largest peak (or valley). The amplitude of the peak is then determined and the 10%, 50% and 90% points calculated. The program draws lines on the plot, at the 10%, 90% and peak values and draws the linear fit to the background as well as a line at the Full Width at Half Height (FWHH). The program also types out information in text form giving: Peak location, peak height, area, background linear fit, x values at 10% 50% and 90% and the FWHH.

The program also uses the 10% and 90% values to produce a triangular peak approximation. The equation to a line between the 10% and 90% points on each side of the peak is found. The intersection of the two lines gives the peak position and the intersection of these lines with the background gives a start and end value to the peak approximation.

### 8.58 PEN (pen) < box/ grid/ fill/ value/ weight> <#>

During a session, the pen colour on the screen can be changed using the command PEN. The user either includes the pen colour as a number after the command PEN or the user will be able to use a colour selection window to select a colour. Additionally the user can include + or - signs after the command to increment or decrement the pen colour number.

The colour number is formed as 24 bit colour with red, green and blue being an 8 bit number (0-255) with increasing power.

$$\text{colour} = \text{red} + \text{green} * 256 + \text{blue} * 65536$$

The pen colour number can be replaced by a named colour [4] from the following list:

black	red	orange	yellow	green
grey	blue	cyan	magenta	crimson
pink	hotpink	raspberry	purple	indigo
slateblue	navy	steelblue	skyblue	lightblue
turquoise	springgreen	emeraldgreen	cobaltgreen	palegreen
limegreen	darkgreen	lawngreen	olive	olivedrab
banana	goldenrod	darkgoldenrod	wheat	tan
brick	cadmiumyellow	carrot	chocolate	sienna
coral	salmon	brown	indianred	firebrick
maroon	lightgrey	silver	gold	khaki

Both of the following examples can select green for subsequent output.  
For example:

```
1> pen 65280 <CR>
Pen 65280 selected
```

or

```
1> pen green <CR>
Pen 65280 selected
```

or

```
1> pen <CR>
```

and then use the colour selector as shown in Figure 44 to visually select the colour.

From the colour selector tool, a colour is selected by moving over a colour patch or over the rainbow patch and clicking on the desired colour. Click on the 'OK' button to confirm the selection when the large colour patch has changed to the desired colour. The slider bar at the right allows the saturation to be varied and numerical values can be entered into the text boxes in the lower right to specify the RGB value or HSL value directly.

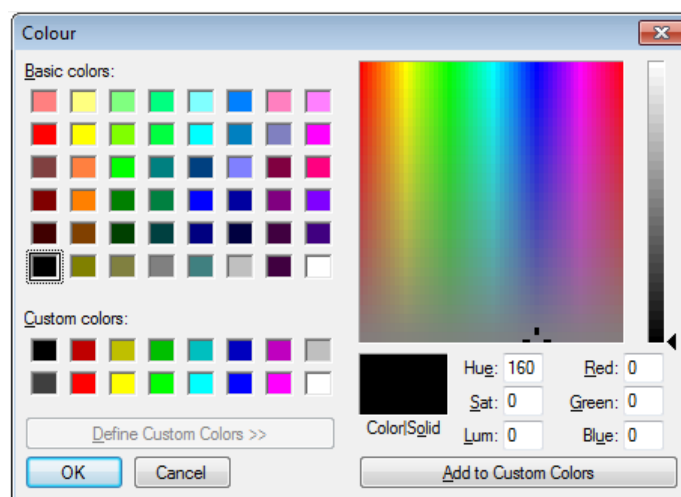


Figure 44 Display of the colour selector tool.

The inclusion of 'box' (can be abbreviated to 'b') after the command PEN alters the box pen colour.

The inclusion of 'grid' (can be abbreviated to 'g') after the command PEN alters the grid pen colour.

The inclusion of 'fill' (can be abbreviated to 'f') after the command PEN alters the fill pen colour.

The inclusion of 'weight' (can be abbreviated to 'w') after the command PEN or after PEN BOX alters the line width for the plot or axes lines respectively. The line width can be any weight from 1 to 16. A value of 1 corresponds to a width of 1 pixel on the screen.

The inclusion of 'vary' (can be abbreviated to 'v') after the command PEN allows the user to vary the colour of the plotted curve using the values of another plot. Three optional values are included after the command. The first of these is the plot number which is going to determine the colour values. The second and third numbers determine the range of values to be considered. Leaving off these two numbers allows KGRAPH to use the maximum and minimum values in the colouring plot for the colour range. There are 20 distinct colours in the spectrum displayed.

An example plot where the variation of speed of a vessel together with location information displayed using different colours is shown in Figure 45.

For example:

```
>1 pen vary 4 0 25 <CR>
```

Variable pen colour dependent on plot 4 selected with range 0 to 25.

```
>1
```

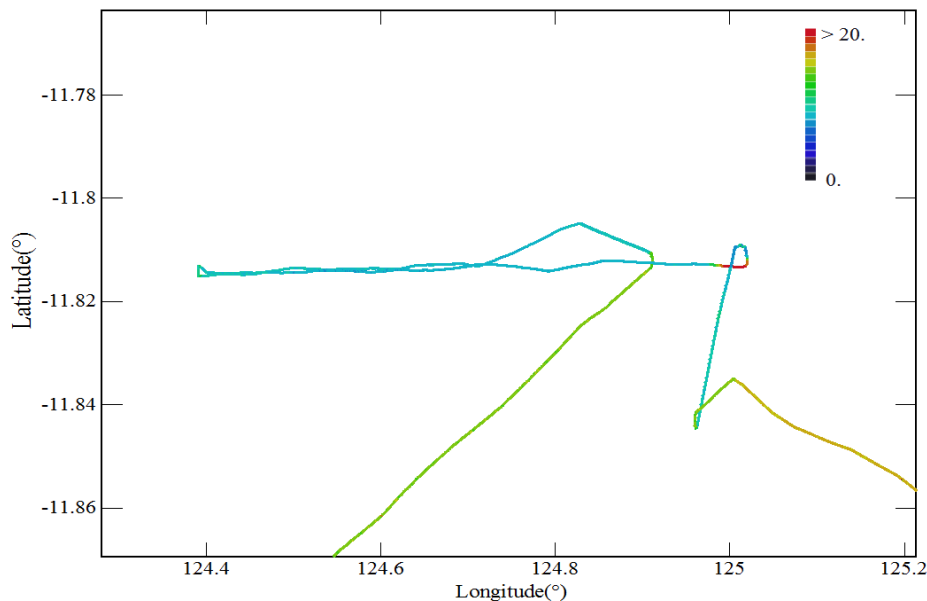


Figure 45 The colour of a plot data can be altered by the values in a second plot. In this example the velocity of a boat is used to colour the positional plot.

### 8.59 PFIT <#> <#> <#>

This command requests that a least-squares-fitted polynomial be fitted to the plot data. Three optional numbers can be included after the command.

1. The order of the polynomial is given by the first number and is restricted to the range 1 to 9.
2. The lower limit of the domain to be fitted.
3. The upper limit of the domain to be fitted.

If no optional numbers are included, KGRAPH responds to this command by prompting for the order of the polynomial to be fitted and the domain of the fit then defaults to the span of the displayed plot. 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 using the Gauss-Jordan method of matrix inversion (See for example Atkinson, 1989 [5]).

In the following example the data of the file '*test.dat*' is fitted with a polynomial of order 2 over the domain 1 to 9 and a comparison of the original data and the resultant fit is shown in Figure 46.

For example:

```
0> load test.dat <CR>
----- 3 plots in memory
1> pfit 2 1 9<CR>
-----
      Fitted Polynomial
-----
power    coefficients

  2 -0.284  +/- 1.371E-02
  1  2.74   +/- 0.140
  0 -0.662  +/- 0.277

correlation coeff. = 0.995501637
-----
      Fit added as plot # 4
-----
1>
```

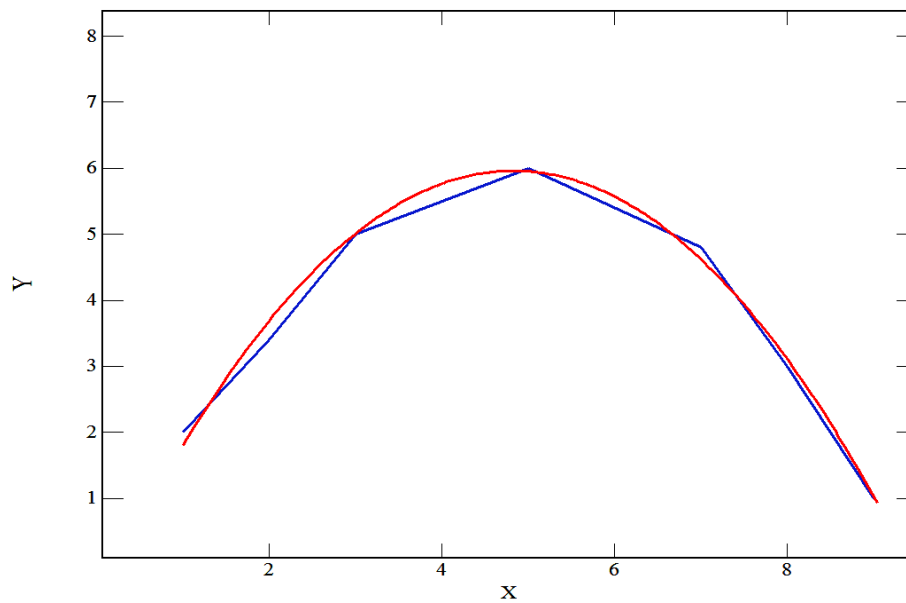


Figure 46 A 2<sup>nd</sup> order polynomial (red) is fitted to the data of 'test.dat' overlaid on the data (blue) as described in the text.

## 8.60 PLOT (p) <#>

Data of the selected plot number is drawn or redrawn using the command PLOT (which can be abbreviated to p). This is useful after modification of labels and other parameters.

For example:

```
0> load test.dat <CR>
1> pen red
1> plot <CR>
```

The above sequence of commands will plot the data of the first plot in the file 'test.dat' on the screen file and redraw the plot on the screen with a changed pen colour.

Often the user will require a different plot than the one currently displayed. The inclusion of the optional number after the command provides a means of determining which of the data sets in memory will be plotted.

The included plot number can be replaced with a global variable.

```
%1 The last plot in memory
```



## 8.61 PMULTIPLY (pmu) <#> <#> <#>

The y values of a plot can be multiplied by the y values of a second plot using the command PMULTIPLY. The command is optionally issued with three numbers immediately after the command. The first number is the plot number of the data being multiplied, the second number is the plot number of the data and the third is the plot number where the result will be placed.

If the domain values of the two plots differ, the data of the first plot is sorted and interpolated so that the domain values of the two plots match before being multiplied.

If the third number is not included the resultant plot is appended as a new plot at the end of the existing plots.

The result of the command PMULTIPLY is demonstrated in Figure 47. The command PMU 1 2 was used. In the command process, the values of plot 1 were interpolated so as to match the x values of Plot 2. Then the interpolated values were multiplied by the corresponding values to produce the product plot at the x values of the second plot.

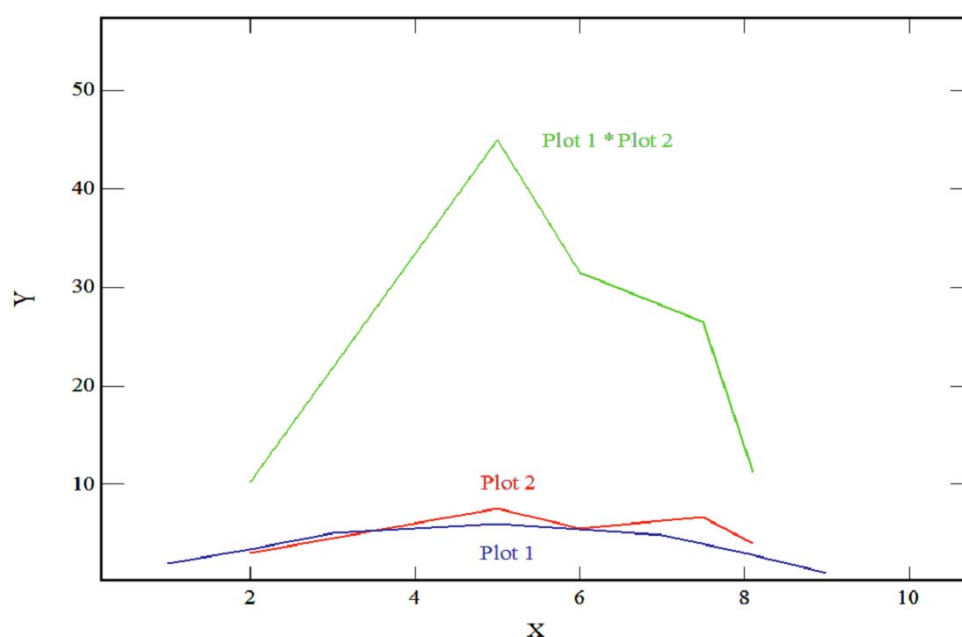


Figure 47 The result of the command PMULTIPLY 1 2 is shown. The values of plot 1 are interpolated so as to match the x values of Plot 2. Then the interpolated values are multiplied by the corresponding values to produce the product plot at the x values of the second plot.

## 8.62 POINT (poi)

The command POINT acts to toggle between point and line plotting. The default setting is line plotting thus when first used it will switch subsequent plotting of the current plot to point plotting.

For example:

```
1> poi <CR>
----- Point plotting set
1> poi <CR>
----- Line plotting set
```

## 8.63 POLAR (pol)

Occasionally there will be a need to plot data in polar coordinates. Plots can be plotted on a polar coordinate system using the command POLAR which acts to toggle from Cartesian to polar plotting. In Figure 48 a simple sine curve is shown and in Figure 49 is shown the polar plot of the same function.

In a polar plot the circumference is always 360 degrees. The radial scale can be changed using the shift key and mouse wheel.

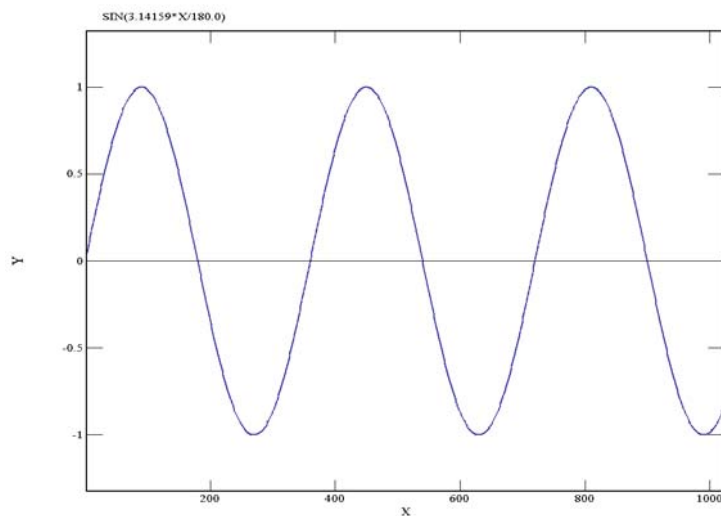


Figure 48 The Cartesian plot of a sine function is shown.

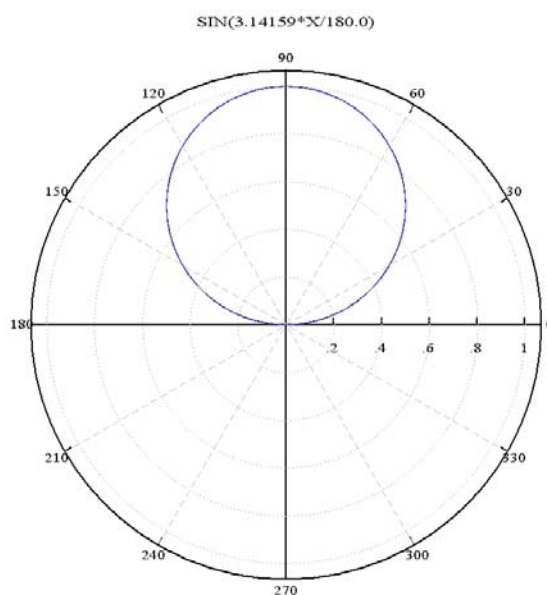


Figure 49 The polar plot of a sine function is shown.

## 8.64 PREVIOUS (pre)

When the PREVIOUS command is used the screen is cleared and the plot preceding the current displayed plot in memory is drawn. If the current plot is the first plot in memory an error message is given.

For example:

```
0> load 5 test.dat <CR>
5> pre <CR>
4>
```

The fifth data set in the file 'test.dat' is plotted and then the screen is cleared and the fourth set of data is plotted.

## 8.65 PSUBTRACT (psu) <#> <#> <#>

The y values of the data of two plots can be subtracted from each other using the command PSUBTRACT. The command is optionally issued with three numbers immediately after the command. The first number is the plot number of the data from which the data of the second plot will be subtracted and the third is the plot number where the result will be placed.

If the domain values of the two plots differ, the data of the second plot is sorted and interpolated so that the domain values of the two plots match before being subtracted.

If the third number is not included the resultant plot is appended to end of the existing plots.

The result of the application of PSUBTRACTION to Plot 1 and Plot 2 to produce Plot 1 - Plot 2 of test data plotted is shown in Figure 50. For each value along the x-axis, the y value of the two source plots is subtracted to produce a plot of the differences.

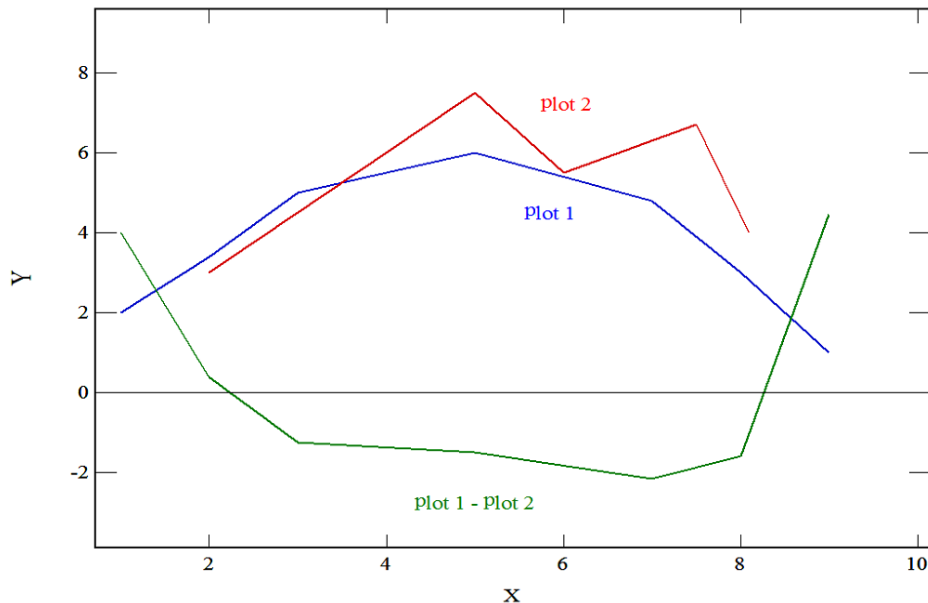


Figure 50 The result of the application of PSUBTRACTION to Plot 1 and Plot 2 to produce Plot 1 - Plot 2 is shown.

## 8.66 Peak and valley analysis (PVA) <#>

The command PVA allows the user to determine all the peaks and valleys of a signal. A measure of the size of the smallest peak to valley reasonably expected in the data is entered after the command. If the estimated size is not included a value will be prompted for.

The results are stored as a new plot. This can be plotted over the existing data to confirm the quality of the result.

The result of the peak and valley analysis of a sample of strain data is displayed in Figure 51. The data has some signal noise thus an estimated minimum value of 15 was included in the command. The red dots are the positions of resulting peaks and valleys.

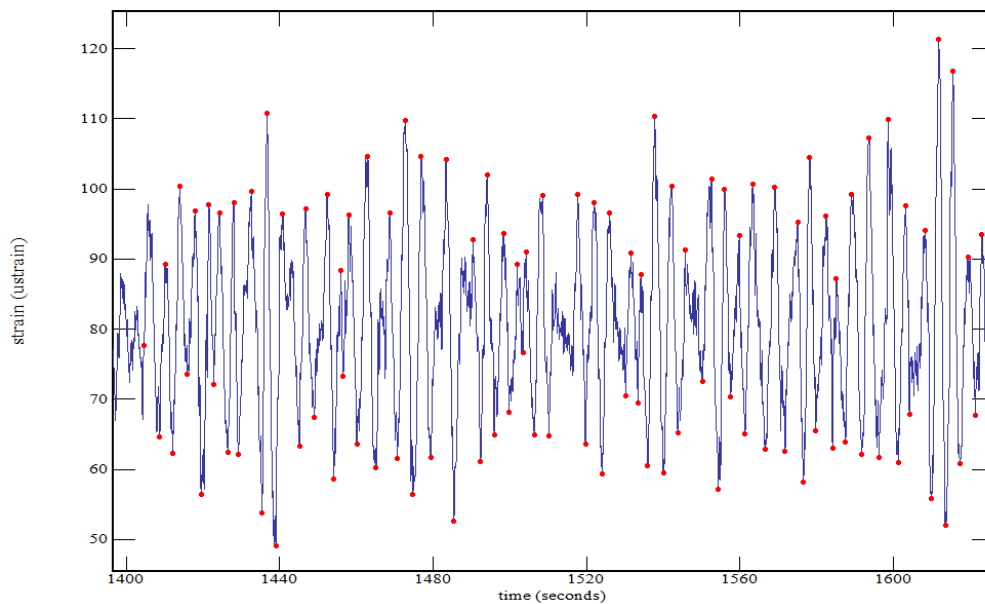


Figure 51 The result of the peak and valley analysis of a sample of strain data is displayed. The data has some signal noise thus an estimated minimum value of 15 was included in the command. The red dots are the positions of resulting peaks and valleys.

## 8.67 POWER

A power spectrum is created from the sum of the squared sine and cosine coefficients of the Fourier transform of a data set. The command POWER will create a power spectrum of the current plot. The Fourier transform uses the Cooley-Tukey FFT algorithm [3] for plots of more than 4095 points and the discrete Fourier transform for smaller plots.

The result of a power spectrum is dependent on the number of points under consideration and the frequency domain of interest. A sampling rate that is near the frequencies of interest and duration of sampling that is as long as possible will produce the best result. However the sample rate will need to be greater than twice the maximum frequency of interest to avoid aliasing errors.

In Figure 52, the result of a power spectrum for the function,

$$f(t) = \sin(2t) + \cos(10t/6) + \cos(t/2) + \sin(2t/3),$$

is shown for different sample rates with the restriction of 32768 sample points. The blue curve is sampled at 400 Hz, the green curve is sampled at 40 Hz and the red curve is sampled at 4 Hz. The results clearly show the improvement with increasing duration and slower sampling rate. The finite length of the sampling period and the limited accuracy of the digital numbers can result in artefacts.

Significant improvement can be obtained by convoluting a function such as;

$$g(x) = 2 \sin^2(ax)$$

where  $a = \pi / d$  and  $d$  is the duration of the data sample, shown in Figure 53. This function will start and end at zero and will integrate to the duration.

The result of the power spectrum of the generated data of the function convoluted with the function of Figure 53 is shown in blue in Figure 54. The green curve is the power spectrum of the data of the function. There is clearly a much reduced background.

When a power spectrum is requested of data that do not have well defined frequencies such as the strain gauge data of Figure 56 there is little advantage in applying a convolution function. The resultant power spectrum of that data is shown in Figure 55.

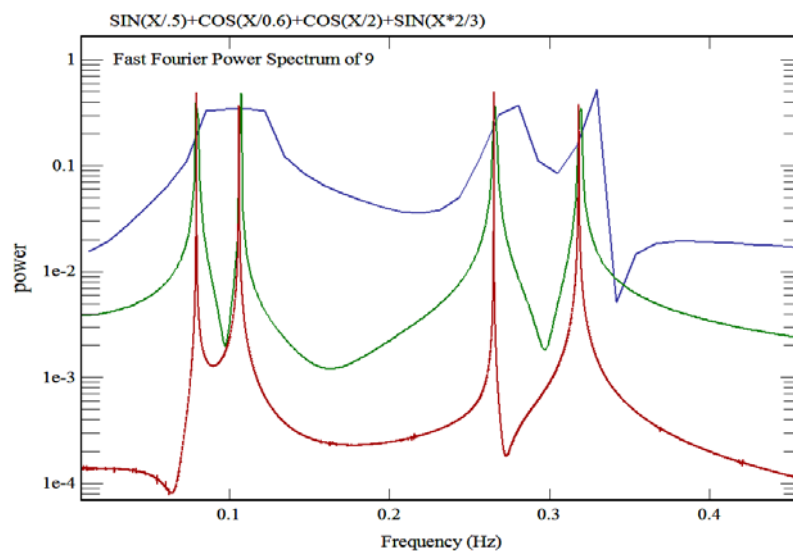


Figure 52 The result of a power spectrum for  $f(t) = \sin(2t) + \cos(10t/6) + \cos(t/2) + \sin(2t/3)$  for different sample rates. The blue curve is sampled at 400 Hz, the green curve is sampled at 40 Hz and the red curve is sampled at 4 Hz. Each is sampled at 32768 points.

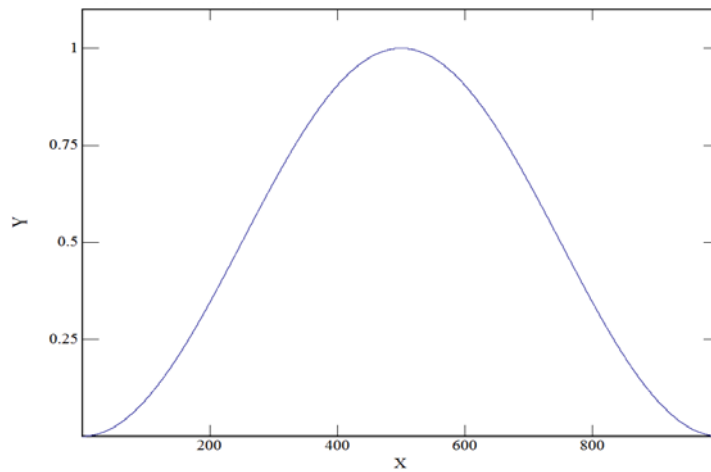


Figure 53 Functions such as the one shown,  $f(t) = 2 \sin^2(\pi t/d)$  where  $d$  is the sample duration, with the property of being two at the centre and diminishing to zero at start and end can be convoluted with a signal prior to extracting a power spectrum to reduce the effect of the finite length, see Figure 54.

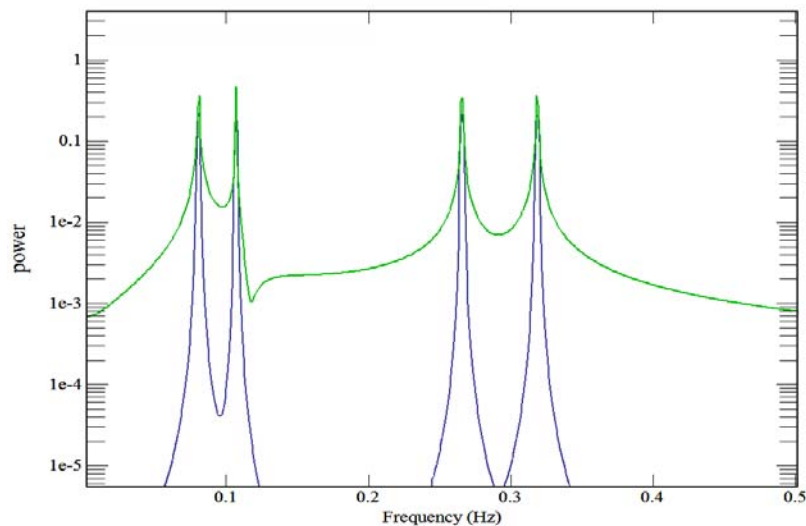


Figure 54 The result of a power spectrum for  $f(t) = \sin(2t) + \cos(10t/6) + \cos(t/2) + \sin(2t/3)$  for a sample taken at 40Hz for 1000 seconds ( green) . The blue curve is the result of the same data being convoluted with the function of Figure 53.

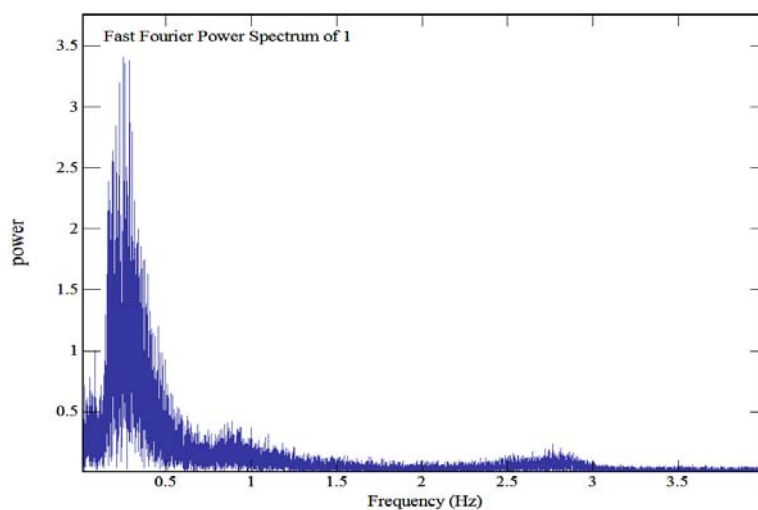


Figure 55 A fast Fourier power spectrum of the strain gauge data of Figure 56.

## 8.68 QUIT (qui)

The command QUIT terminates the execution of the program. The plot screen is closed, open data files are closed, memory is returned to the system.

Prior to exiting, the positions of the windows are stored in a temporary file '*c:/temp/kg.dat*' for use in the next session. If this file is deleted the default initial positions are restored.

## 8.69 RADIANT (rad)

The conversion of plot range information which consists of angles in degrees to the equivalent in radians is achieved using the command, RADIANT. The inverse operation is undertaken using the command DEGREE.

The conversion is on the y values only. The conversion of x values can be achieved by using the command SWAP before and after the command RADIANT.

## 8.70 RAINFLOW (rai)

The analysis of strain gauge data where varying loads are applied to a structure to determine fatigue life is often done using rainflow cycle counting. This command invokes a rainflow count for the displayed plot. The rainflow algorithm is as described by Matsuishi and Endo [6].

In Figure 56, a sample of strain gauge data from a boat at sea during rough weather is shown. Rainflow counting will enable the number of cycles at a given strain level to be determined.



This can then be used to predict fatigue life. The result of application of rainflow counting to this data is shown in Figure 57. Note the lowest possible count will be a half cycle.

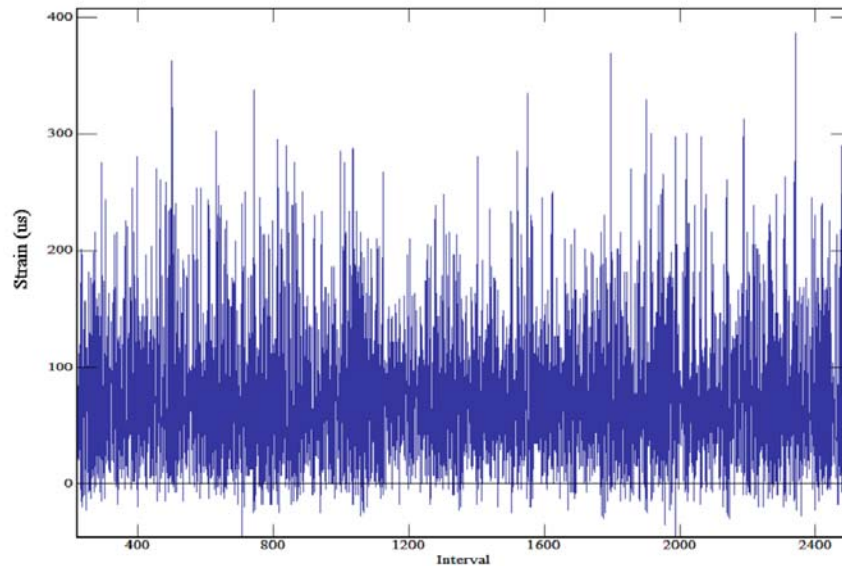


Figure 56 Shown is a sample of strain gauge data from a boat at sea during rough weather.

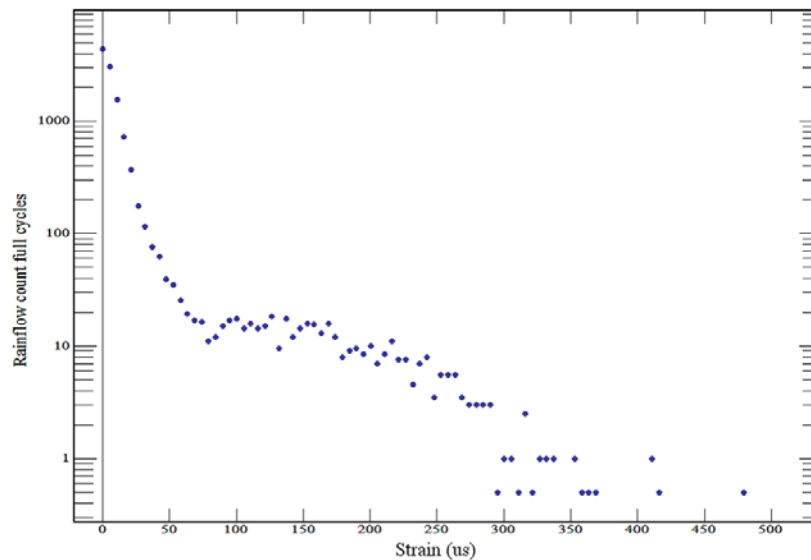


Figure 57 The result of application of rainflow counting to the data of Figure 56 is shown. Note the lowest possible count will be a half cycle.

## 8.71 RAPID (rap) <#>

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 for

improved file manipulation. The command RAPID allows the user to specify how many points will be skipped before the next point will be plotted. On entering this command KGRAPH 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:

```
0> rapid 5 <CR>
0> load test.dat <CR>
1>
```

In this example every 5th point of the first plot in the file '*test.dat*' is plotted.

## 8.72 REMOVE (rem) <#>

The command REMOVE allows the user to remove individual data points or groups of data points from the current plot. To remove an individual point the user can either provide a point number immediately after the command or select a point using the mouse pointer to draw a box around the point to be removed.

If the user wishes to select a number of points, this is done by issuing the command and then using the mouse pointer to draw a box around the points.

If no point lies within the box the program gives a warning.

For example:

```
1> l test.dat <CR>
1> rem 2 <CR>
---- Point 2 removed
1> rem <CR>
---- Point 5 removed
---- Point 4 removed
1>
```

In this example, the data of the first plot in the file '*test.dat*' was plotted. The 2nd point of the plot was removed. Then the remove command was issued and the mouse used to draw a box around points to be removed.

### 8.73 RESET (res)

On occasion the user may wish to reset the program to the start-up condition. Deleting all data and resetting all flags. The command reset initializes all the settings and uses the start-up file if it exists.

### 8.74 ROLL (rol) <#>

The command ROLL allows the user to apply a rolling average to the data of the current plot or optionally to a selected plot or range of plots. The optional input after the command determines the number of points either side of the current point included in the rolling average. If no number is included after the command then a rolling average of the current plot is undertaken with 3 points either side of the point under consideration.

For example:

```
0> load test.dat <CR>
1> roll 5 <CR>
----- 11 point rolling average applied to data of plot 1 .
1> roll 6 2<CR>
----- 13 point rolling average applied to data of plot 2 .
1> roll 4 [3..6] <CR>
----- 9 point rolling average applied to data of plots [3..6].
```

The result of applying a rolling average is shown in Figure 58. A raw plot is shown in blue and the result of a rolling average with the application of five points on either side of a point is shown in red.

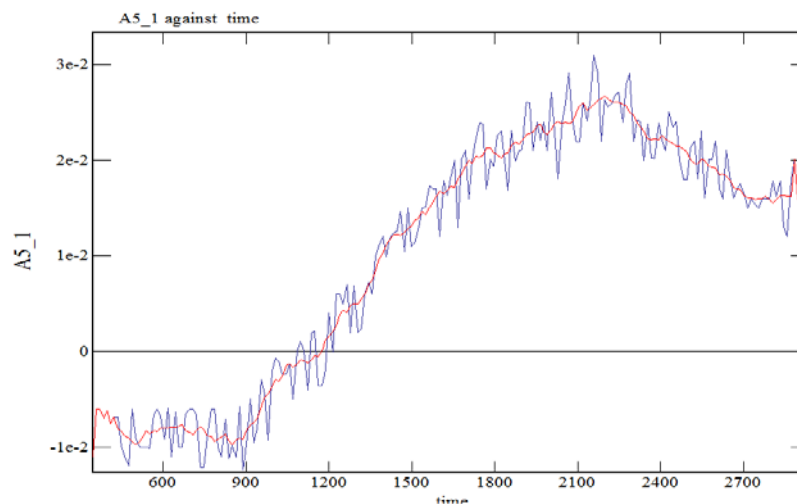


Figure 58 The result of applying a rolling average to data is shown. The blue curve is the raw data and the red curve is the result of an 11 point rolling average.

## 8.75 ROTATE (rot) <#>

The data can be rotated about the origin using the command ROTATE. The angle of rotation, expressed in degrees, can be included after the command.

For example:

```
1> rotate <CR>
Rotate through an angle of (degrees)? 30 <CR>
----- Data rotated by 30. degrees
1> plot <CR>
1> Pen red
1> same
```

The result is demonstrated in Figure 59 for a rotation of 30 degrees about the origin. The original data is plotted in blue and the rotated data is plotted in red.

```
1> rot 30 <CR>
----- Data rotated by 30.00 degrees
1> plot <CR>
```

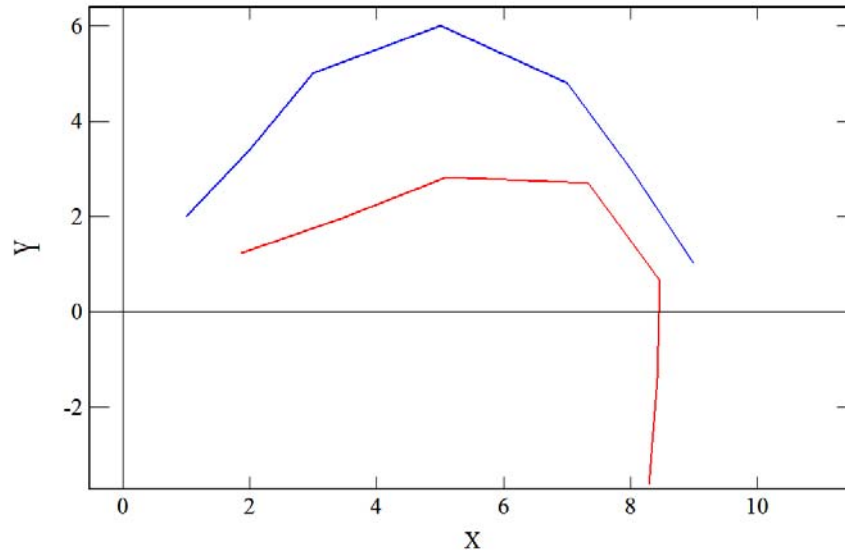


Figure 59 The result of using the ROTATE command is shown for a rotation of 30 degrees about the origin. In this example the original data is plotted in blue and the rotated data is plotted in red.

## 8.76 SAME (sam)

The SAME command allows the data of the current plot to be redrawn onto the same plot without clearing the screen. This is useful when a symbol needs to be drawn at each data point on a line connecting the points.

For example:

```
0> load test.dat <CR>
1> type 14
1> poi <CR>
---- Point plotting set
1> pen red <CR>
1> same <CR>
1>
```

In this example the data of the first plot in the file, 'test.dat', is plotted in the default line plotting mode, point plotting is selected, the pen colour is changed to red, and the data re-plotted onto the plot, see Figure 60.

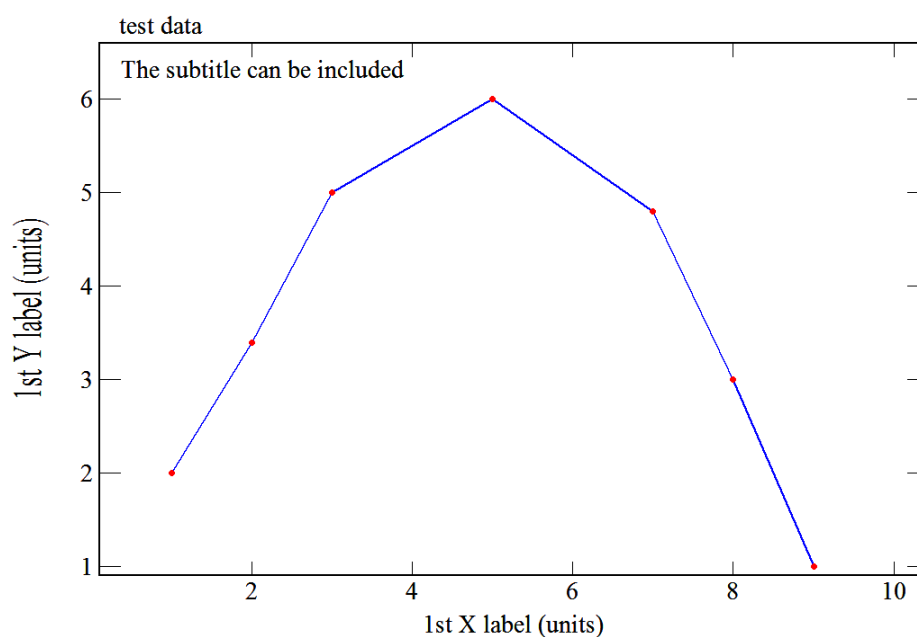


Figure 60 The command SAME allows data to be redrawn after changes have been made onto the displayed plot. In this plot the colour has been changed to red and the type of plot has been changed to point plotting.

### 8.77 SAVE (sav) <\* | & | +> <[#<..#>]> <filename>

The user can save the plot data and axes labels to a file using the command SAVE. On receiving this command, KGRAPH prompts for a file name using the standard file requester. A response of a carriage return at this point will create a file called '*plot.file*' in the current directory.

KGRAPH checks the existence of the file name. If the file name exists then the user is asked to confirm that the file is to be overwritten. If not the standard file requester is used again. The command has a number of optional extensions.

1. The first of these is one of the following symbols \*, & and + which respectively either store the data in an uncompressed form, store the data in a compressed but ascii form or save the data to the end of an existing file. When appending the data the existence of the file is checked.
2. The second is the inclusion of a plot range for saving only a subset of the plots.
3. The third option is the inclusion of the file name to which the data will be saved.

This command can be invoked from the file dropdown menus of the plot window.

Storing in ascii character formats allows the user to edit the stored file in a text editor such as Notepad. However it is considerably faster to load a compressed file and requires less storage space.

### 8.78 SET (set) <#> <#> <#> <#>

The SET command in KGRAPH allows limits of the axes to be predetermined. After the command SET, the program prompts for the limits for the axes. It normally expects two numbers to be entered to set up each axis. The first of these numbers is the lower limit and the second number is the upper limit. These numbers can be either real or integer.

For example:

```
1> set <CR>
Input lower x value :0 <CR>
Input upper x value :10 <CR>
Input lower y value :0 <CR>
Input upper y value :100 <CR>
----- Fixed axes
1>
```

In this example the limits of the plot are set such that the x-axis displays from 0 to 10 and the y-axis displays from 0 to 100. All plots after a set command have these limits until another SET

command or a FREE command is given. An abbreviated form of this command allows the inclusion of the axes limits.

set <x low> <x high> <y low> <y high>

For example:

```
1> set 0 10 0 100 <CR>
----- Fixed axes
1>
```

In this example the SET command has defined the plot area to have a domain of 0 to 10 and a range of 0 to 100. The numbers can be separated by comma, space or tab characters.

If only a lower or upper limit on either or both axis is required, then the user may use a less than or greater than sign to indicate whether the data will be above or below the limit.

For example:

```
1> set > 2 10 100 <CR>
----- Fixed axes
1>
```

In this example the lower limit of the x-axis is set to 2 and the upper limit is allowed to be scaled by the data. The y-axis is set with a lower limit of 10 and an upper limit of 100.

This command can be invoked from the axes menu of the dropdown menus of the plot window and required values are entered from dialog boxes.

## 8.79 SIZE (siz) <#>

This command allows the user to alter the size of the points being plotted. The size of the point symbols is set using the command SIZE followed by a positive number. The number corresponds to the number of pixels wide the symbol will appear on the screen.

For example:

```
1> size <CR>
Input the size : 12 <CR>
```

This will produce symbol 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:

```
1> size 12 <CR>
1>
```

This command can be invoked from the size menu selection under the point selection of the axes menu of the plot window. A choice of five point sizes ranging from very small to very large (8 to 50) is offered.

### 8.80 SLICE (sli) <[#..#]> <#> <#> <#>

The command SLICE provides an ability to collect together the y values, from a series of plots, at particular x values. The command is provided with a number of optional parameters:

- The first is the range of plots over which the slice is collected.
- The second is the lower x value at which the data will be collected.
- The third is the upper x value at which the data will be collected.
- The fourth is the x spacing at which the data will be collected between the lower and upper x values.

The results of the command are appended to the end of the plots in memory.

The SLICE command would be meaningless if the range specified contains only one plot. This condition results in a error message being generated. If the user specifies only the range then an x value is prompted for and a single slice over the plot range is taken.

Figure 62 shows the results of using the SLICE command with a set of curves shown in Figure 61. The command SLICE [1..7] 2 10 2, produces a set of 5 curves for x=2 to x=10 with 7 points in each curve corresponding to the number of plots in the plot range shown in Figure 61.



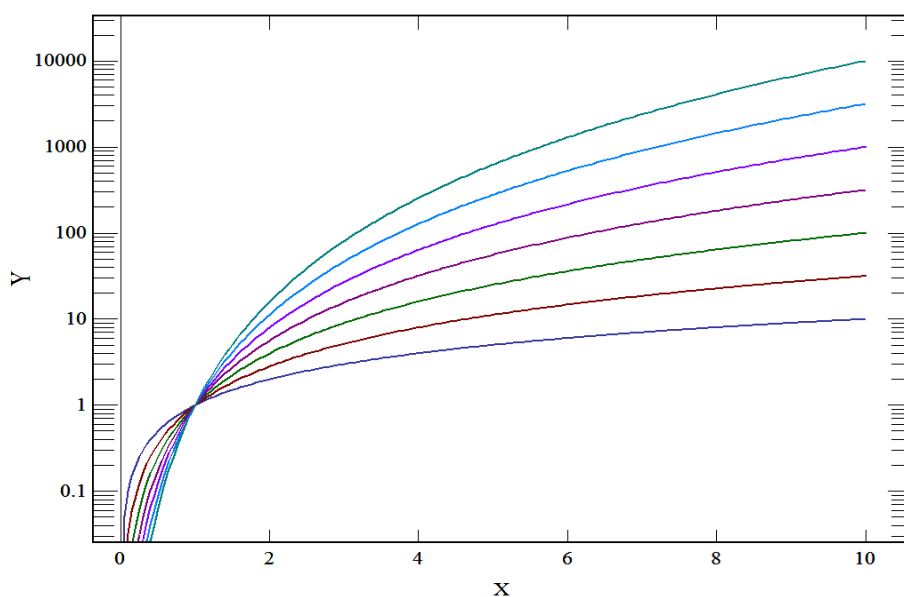


Figure 61 A range of plots is displayed. These are  $f(y) = x^{n/2}$  where  $n$  is an integer between 1 and 7.

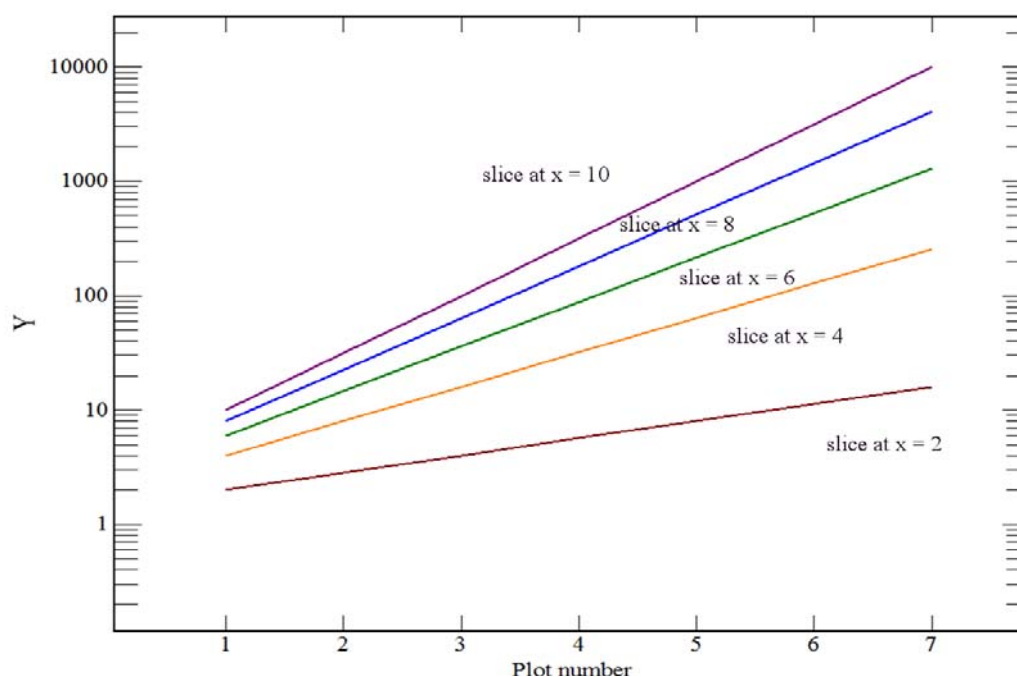


Figure 62 The results are shown for the command `SLICE [1..7] 2 10 2`, which produced slices at  $x=2$ , 4, 6, 8 and 10 through the data curves of Figure 61.

### 8.81 SMOOTH (smo) <[#<..#>]> <#> <#>

Smoothing of data is achieved using the command SMOOTH. This command uses the least squares smoothing technique described by Golay and Savitzky [2] involving 3 points either

side of the point of interest (seven points in all). The method locally fits a polynomial to the points. The smoothing by default is applied to the current plot however a range of plots can be selected by appending a range immediately after the command.

As an extension to this command, the number of points either side of the point of interest can be 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:

```
1> smo 4 <CR>
----- Data plot 1 smoothed 1 times 9 point smoothing.
1>
```

In this example the plotted data will be smoothed using 9 point smoothing, 4 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.

For example:

```
1> smo 4 5 <CR>
----- Data plot 1 smoothed 5 times 9 point smoothing.
1>
```

This instructs KGRAPH to smooth the data using a 9 point smoothing technique repeated five times before overlaying the smoothed data.

A range of plots can be smoothed at once by including a plot range immediately after the command. The previous example can be extended to apply to plots 3 through 11 using the following command.

```
1> smo [3..11] 4 5 <CR>
----- Data plot [3 .. 11 ] smoothed 5 times 9 point smoothing.
1>
```

The result of using a 9 point smoothing 5 times is shown in Figure 63. The data plotted in blue is the raw data and the smoothed data is plotted in red. The effectiveness of the smoothing will depend on the ratio of the number of points in the plot to the number of points involved in the smoothing operation and the noise of the signal.

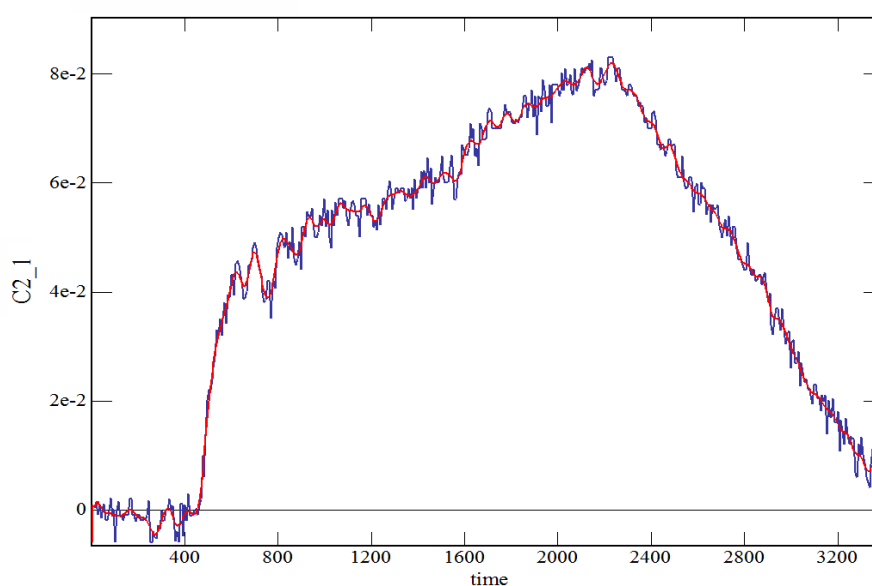


Figure 63 The effect of using a 9 point smoothing repeated 5 times on a data set is shown. The data plotted in blue is the raw data and the smoothed data is plotted in red.

## 8.82 SORT (sor) <[#<..#>]>

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 heap sort. A heap sort is at worst a  $N\log_2 N$  process where N is the number of points, and is thus a fast sorting technique [7].

Sorting before interpolation is essential.

For example:

```
0> load test.dat <CR>
1> sort <CR>
----- Data of plot 1 sorted
1> plot <CR>
```

In this example the data of 'test.dat' was first plotted then sorted and redrawn. In the following example, the data of the first to tenth plots are sorted so that each of the plots has ascending x values.

```
1> sort [1..10]
----- Data of plot [ 1 .. 10 ] sorted.
1>
```

### 8.83 SUBTITLE (sub) <title>

The subtitle is a second line of text directly below the title line. The command SUBTITLE allows the user to include or modify a subtitle line. (see also TITLE). On entering the SUBTITLE command, KGRAPH responds with a prompt for the new title. A subtitle can be as long as 60 characters. A carriage return after the prompt inserts a blank subtitle.

For example:

```
0> load <CR>
Input the file name : test.dat <CR>
1> sub <CR>
New subtitle : trial #3 , data test <CR>
1> p <CR>
1>
```

The original subtitle contained in the first plot of the file '*test.dat*' is replaced with "trial #3 , data test" when the plot is redrawn. An extension of this command allows the user to include the new subtitle in the command line. So that the above example could be entered as:

```
0> l test.dat <CR>
1> sub trial #3 , data test <CR>
1> p <CR>
1>
```

The subtitle text can be altered in the subtitle textbox in the information dialog window, see Section 8.39.

### 8.84 SWAP (swa) <[#<..#>]> or swa <#>

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 of a range of files or by default the current plot.

For example:

```
0> load test.dat <CR>
1> swap <CR>
----- x and y data swapped in plot 1
1> swap 2 <CR>
----- x and y data swapped in plot 2
1> swap [5..10] <CR>
----- x and y data swapped in plot range [ 5 .. 10 ]
```

1>

In this example all three forms of the SWAP command have been given. The first defaulted to swapping x and y values of the current plot, the second swapped the x and y values of the second plot in memory and the third swapped the x and y values of the 5th to 10th plots inclusively.

## 8.85 TEXT (tex) <text to be written>

Text can be positioned anywhere on the plot using the command TEXT. The text can be optionally included after the command. If this option is not used the text will be prompted for. The text is positioned on the plot using the mouse. The text will be located such that the mouse pointer will be at the bottom left of the text when the mouse is clicked.

For example:

```
0> load test.dat <CR>
1> text test one two three <CR>
---- Select text position with pointer
1> text <CR>
---- Input text: test one two three <CR>
---- Select text position with pointer
1>
```

In this example, the text '*test one two three*' is firstly included in the command line and secondly is prompted for.

## 8.86 TITLE (tit) <title>

The user can alter the title line on the plot using the command TITLE (tit). On entering this command, KGRAPH 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:

```
0> load test.dat <CR>
1> title <CR>
New title: trial #3 , data test <CR>
1> plot <CR>
1>
```

The original title contained in the file '*test.dat*' is replaced with "trial #3 , data test" when the plot is redrawn. An extension of this command allows the user to include the new title in the command line. So that the above example could be entered as:

```
0> load test.dat <CR>
1> tit trial #3 , data test <CR>
1> plot <CR>
1>
```

The title text can also be altered in the title textbox in the information dialog window, see Section 8.39.

### 8.87 TYPE (typ) <#>

There are 20 distinct point types whose size can be set by the user. The point type can be set with the command TYPE and their size can be set with the command SIZE. The command TYPE followed by the type number is used to specify the point symbol type.












For example:



























```
1> type <CR>
Input the point type (1 to 20) : 2 <CR>
```




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

```
1> type 2 <CR>
```

The space between the command and the number is optional. The available point types are:

Type number	Description	symbol
1	upright cross (+)	
2	diagonal cross (x)	
3	square	
4	triangle (point up)	
5	triangle (point down)	
6	square with upright cross (+)	
7	triangle (point up) with upright cross (+)	
8	Triangle (point down) with upright cross (+)	
9	Square with diagonal cross (x)	
10	Triangles (up and down)	
11	Tetrahedron	

Type number	Description	symbol
12	Cube	
13	Filled square	
14	Filled circle	
15	Filled triangle (point up)	
16	Filled triangle (point down)	
17	Filled triangle (point right)	
18	Filled triangle (point left)	
19	Two filled triangles (vertical)	
20	Two filled triangles (horizontal)	
21	Filled Diamond	
22	Filled Hexagon	
23	Filled right bottom corner	
24	Filled left top corner	
25	Filled left bottom corner	
26	Filled right top corner	
27	Star	
28	Star and square	
29	Two triangles (vertical)	
30	Two triangles (horizontal)	
31	Triangle (point right)	
32	Triangle (point left)	
33	Diamond	
34	Hexagon	
35	Right bottom corner	
36	Left top corner	
37	Left bottom corner	

Type number	Description	symbol
38	Right top corner	
39	Vertical crossed diamond	
40	Vertical crossed hexagon	

### 8.88 VALUE (val) <#>

The command VALUE allows the user to examine the values of individual data points or groups of data points. When selecting individual points the user can either provide a point number immediately after the command or select a point using the mouse pointer to draw a box around the point of interest.

If the user wishes to select a number of points, this is done by issuing the command then using the mouse pointer to draw a box around the points of interest.

If no point lies within the box the program finds the point closest to the centre of the box selected.

For example:

```
0> load test.dat <CR>
1> val 2 <CR>
value of 2nd point is (2. , 3.4)
1> val <CR>
# 3 ( 3., 5.)
# 4 (5., 6.)
# 5 (7., 4.8)
1>
```

In this example, the data of the first plot in the file '*test.dat*' was plotted. The value of the 2nd point was ascertained and the values of three points were examined using the command VALUE alone and using the mouse to select points.

Points with error bars are displayed as:

Coordinate display	Example	Description
$(x \pm \Delta_x, y)$	$(7.5 \pm 0.15, 1.89)$	xy pair with uncertainty in the x direction
$(x, y \pm \Delta_y)$	$(7.5, 1.89 \pm 0.15)$	xy pair with uncertainty in the y direction
$(x \pm \Delta_x, y \pm \Delta_y)$	$(7.5 \pm 0.15, 1.89 \pm 0.15)$	xy pair with uncertainty in the both directions
$(x [\pm \Delta_{x1} \pm \Delta_{x2}], y)$	$(7.5 [ +.20 -0.15], 1.89)$	xy pair with unequal uncertainties in



$(x, y [\pm \Delta_{y1} \pm \Delta_{y2}])$	$(7.5, 1.89 [ +.20 -0.15])$	the x direction
		xy pair with unequal uncertainties in the y direction
$(x [\pm \Delta_{x1} \pm \Delta_{x2}], y [\pm \Delta_{y1} \pm \Delta_{y2}])$	$(7.5 [ +.20 -0.15], 1.89 [ +.20 -0.15])$	xy pair with unequal uncertainties in both directions

### 8.89 VARIABLE (var) <# #>

The variables, %v1 to %v99, allow the user to store up to 100 values for later output. The command VARIABLE allows the user to list the contents of the variables. All variables are listed up until the last non-zero variable.

For example:

```
1> %v5 = 5 <CR>
---- variable 5 = 5.
1> var<CR>
0.,0.,0.,0.,5.,
1> %v8 = 12.5 <CR>
---- variable 8 = 12.5
1> var<CR>
0.,0.,0.,0.,5.,0.,0.,12.5,
1>
```

### 8.90 VELOCITY (vel) <#>

The command VELOCITY allows the user to extract the average velocity of a selection of points in a GPS position plot where the velocity information is in another plot, the number of which is included after the command.

To select a number of points, the command is issued then using the mouse pointer draw a box around the points of interest. The average velocity associated with the points is returned.

If no point lies within the box the program finds the point closest to the centre of the box selected.

For example:

```
0> load test.dat <CR>
1> velocity 4 <CR>
---- Select area with mouse pointer
----- 606 points in window [130.80,-12.460]..[130.91, -12.455]
```

Average velocity = 1.0234 +- 0.093312

1>

In this example, the data of the first plot in the file '*test.dat*' was plotted. The 4th plot in the file contained the velocity-time plot. An area of GPS position data was selected and the average velocity was displayed.

### 8.91 WINDOW (win) <[#<..#>]> <#> <#>

The command WINDOW is provided to remove data from memory that lies outside a selected window. The command window has an optional plot range included immediately after the command which is followed by a minimum x value and a maximum x value. If the plot range option is not included in the command line the window command is assumed to apply only to the current plot. If the minimum x value or maximum x value is not included in the command line, the values are prompted for.

For example:

```
0> load test.dat <CR>
1> win <CR>
Input the minimum x value :2 <CR>
Input the maximum x value :8 <CR>
WINDOW 2.000 to 8.000
Data window applied to plot [ 1 ]
Plot # 1 : 5 Pts
1> win [1..3] 5 8 <CR>
WINDOW 5.000 to 8.000
Data window applied to plots [ 1 .. 3 ]
Plot # 1 : 3 Pts
Plot # 2 : 3 Pts
Plot # 3 : 3 Pts
Removing empty plots.
1>
```

### 8.92 Exponentiation of data

XDEXP (xde) <[#<..#>]>

YDEXP (yde) <[#<..#>]>

ZDEXP (zde) <[#<..#>]>

The data in a plot or a range of plots can be converted to  $e$  raised to the power of the data using the commands XDE, YDE and ZDE. These commands are applied to the data of the x, y or z-axis respectively.

For example:

```
0> l test.dat <CR>
1> xde <CR>
----- Data now exp(X data) for plot 1
1> yde <CR>
----- Data now exp(Y data) for plot 1
1> plot <CR>
1>
```

In this example the data of the 1st plot in the file '*test.dat*' is converted to exponentiation and re-plotted.

### 8.93 Division of data by a constant

```
XDIVIDE (xdi) <[#<..#>]> <#>
YDIVIDE (ydi) <[#<..#>]> <#>
ZDIVIDE (zdi) <[#<..#>]> <#>
```

The commands XDIVIDE, YDIVIDE and ZDIVIDE are used when the data along the x-axis, y-axis or z-axis, respectively, is to be divided by a constant value. The command is optionally followed by a range specifier for selecting multiple plots and a constant. If no range specifier is given, KGRAPH defaults to using the current plot. If no constant is given the program prompts for a value.

For example:

```
0> l test.dat <CR>
1> xdi 10.5 <CR>
----- x data of plot 1 divided by 10.5
1> ydi [2..5] 100 <CR>
----- y data of plots [2..5] divided by 100
1>
```

In this example the data of the first plot in the file '*test.dat*' is plotted. The x-axis values of the current plot are divided by 10.5 and the y values of the second to fifth plots in memory are divided by 100.

An error is given when the user attempts to divide by zero.

## 8.94 Reciprocal of data

**XINVERT (xin) <[#<..#>]>**

**YINVERT (yin) <[#<..#>]>**

**ZINVERT (zin) <[#<..#>]>**

Plot data may be inverted using the commands XINVERT, YINVERT or ZINVERT. When one of these commands is given the reciprocal of each data point value replaces the current value in the indicated dimension. Division by zero is avoided by setting the result to overflow value. (By default the overflow value is set to 99999.9)

For example:

```
0> load test.dat <CR>
1> yin <CR>
----- Y data inverted for plot 1
1> xin <CR>
----- X data of plot 1 inverted
1> zin [2..3] <CR>
----- Z data of plots [2.. 3] inverted
1> p <CR>
1>
```

## 8.95 Axes label text

**XLABEL (xla) <x label>**

**YLABEL (yla) <y label>**

**ZLABEL (zla) <y label>**

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 XLABEL, YLABEL and ZLABEL allow the user to add axes labels of up to 60 characters. A carriage return after the prompt inserts a blank label.

For example:

```
0> load test.dat <CR>
1> xla <CR>
New x label : temperature (C) <CR>
1> yla time (seconds) <CR>
1> p <CR>
1>
```

The original x label contained in the data of the first plot of the file '*test.dat*' is replaced with "temperature (C)" and the y label is replaced with "time (seconds)". The YLABEL command employed the extension of these commands which allows the user to include the new label in the plot.

## 8.96 Logarithm of data

XDL (xdl) <[#<..#>]>

YDL (ydl) <[#<..#>]>

ZDL (zdl) <[#<..#>]>

The data in a plot or a range of plots can be converted to the logarithm base 10 of the data using the commands XDL, YDL and ZDL. These commands are applied to the data of the x, y or z-axis respectively.

For example:

```
0> load test.dat <CR>
1> xdl <CR>
----- Data now log10(X data) for plot 1
1> ydl <CR>
----- Data now log10(Y data) for plot 1
1> p<CR>
1>
```

In this example the data of the 1st plot in the file '*test.dat*' is converted to logarithms and re-plotted.

## 8.97 Logarithmic Axes

XLOG (xlo)

YLOG (ylo)

ZLOG (zlo)

Three commands allow the user to toggle logarithmic axes on particular axes. They are XLOG, YLOG and ZLOG.

The first of these XLOG, toggles the x-axis between logarithmic and linear axes, the second YLOG, toggles the Y-axis in the same way as does ZLOG for the z-axis. It should be noted that KGRAPH plots the logarithm of the absolute value of the data.

Zero is plotted as the logarithm of  $10^{-20}$ . When zero is present only four cycles are plotted.

For example:

```
0> load test.dat <CR>
1> ylo <CR>
----- Logarithmic Y axis
1>
```

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.

## 8.98 Multiplying data by a constant

```
XMULTIPLY (xmu) <[#<..#>]> <#>
YMULTIPLY (ymu) <[#<..#>]> <#>
ZMULTIPLY (zmu) <[#<..#>]> <#>
```

XMULTIPLY, YMULTIPLY and ZMULTIPLY are used when the data of a plot or a range of plots along the x-axis, y-axis or z-axis is to be multiplied by a constant value. The data of the plot indicated is rescaled data is re-plotted.

For example:

```
0> load test.dat <CR>
1> xmu10.5 <CR>
1> ymu [2..5] 100 <CR>
1>
```

In this example the data of the first plot in the file '*test.dat*' is plotted. The X axis values of the current plot are multiplied by 10.5 and the Y values of the second to fifth plots in memory are multiplied by 100.

## 8.99 Setting axes limits

```
XSET (xse) <#> <#>
YSET (yse) <#> <#>
ZSET (zse) <#> <#>
```

It is often necessary to set only one of the axes and allow the other to be auto-scaled. The commands XSET, YSET and ZSET allow each axis to be treated separately. These commands act as toggles and give the status of the particular axis. On giving the command XSET, YSET or ZSET, 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 auto-scaled and informs the user.

For example:

```
1> xse <CR>
Input lower x value :0 <CR>
Input upper x value :10 <CR>
----- Fixed X axis
1> xset<CR>
----- Free X axis
1>
```

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

An abbreviated form of these commands allows the inclusion of the limits after the command. For instance using this previous example would become:

```
1> xse 0 10 <CR>
----- Fixed X axis
1>
```

A further extension to this command allows for the possibility of setting only one limit using the less than and greater than symbols.

```
1> xse > 0 <CR>
----- Fixed X axis minimum
1> yse < 10 <CR>
----- Fixed Y axis maximum
1>
```

## 8.100 Shifting plot along axes

```
XSHIFT (xsh) <[#<..#>]>,<#> <#> <#>
YSHIFT (ysh) <[#<..#>]> <#> <#> <#>
ZSHIFT (zsh) <[#<..#>]> <#> <#> <#>
```

The commands XSHIFT, YSHIFT and ZSHIFT are used when a constant value is to be added to the data along the x, y and z axes, respectively. The command can be applied to a range of plots by optionally including a plot range immediately after the command. The size of the shift can be optionally included after the command and after the range specification if included. When the shift size is included two optional numbers specifying a domain over which the command will be applied may be included.

XSHIFT (range specification) (Shift size) (Lower X value) (Upper X value)  
 YSHIFT (range specification) (Shift size) (Lower Y value) (Upper Y value)  
 ZSHIFT (range specification) (Shift size) (Lower Z value) (Upper Z value)

For example:

The file, '*data.dat*' contains information in non-metric units along both axes. 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.

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 can be followed.

```
0> load data.dat <CR>
1> yshift <CR>
Input y shift : -32 <CR>
1> ydi <CR>
Input factor : 9 <CR>
1> ymu <CR>
Input factor : 5 <CR>
1> xmu <CR>
Input factor : 2.54 <CR>
```

## 8.101 Axes Mark Spacings

XSPACING (xsp) <#>  
 YSPACING (ysp) <#>  
 ZSPACING (zsp) <#>

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



For example:

```
0> load test.dat <CR>
1> xspacing <CR>
Input the x spacing : 10 <CR>
1> yspacing <CR>
Input the y spacing : 20 <CR>
1> plot <CR>
1> xspacing <CR>
---- Autoscale x spacing
1>
```

In this example the data of '*test.dat*' was 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.

### 8.102 YYCOMPARE (yyc) <#> <#> <#>

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. The command YYCOMPARE provides the facility for doing this.

The YYCOMPARE command has three optional numbers after the command. The first two of these are the plot numbers of the plots being compared and the third is the plot number into which the comparison will be placed.

If the third of the optional numbers is not included the program will append the comparison plot after the last plot.

If the first or second of the optional numbers are not included the program will prompt the user for the appropriate values.

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

For example if the user wishes to compare y values of the first and second plots. The following command sequence would display the required result:

```
0> fun
Input function domain (minimum, maximum): 0 30 <CR>
```

Input the number of points (20 or more):1000 <CR>

---

The input function line is an expression involving X

That can contain; up to 10 brackets.

Functions: sin, cos, tan, asn, acs, atn

Log (base 10), Lgn (base e), erf

Abs, int, sqr

Operators: +, -, /, \*, \*\* (or ^)

Variables: a, b, c, d, e, f, g

---

Enter the function :

: sin(3\*x) <CR>

1>fun <CR>

Input function domain (minimum, maximum): 0 30 <CR>

1> Input the number of points (20 or more): 1000 <CR>

---

The input function line is an expression involving X

That can contain; up to 10 brackets.

Functions: sin, cos, tan, asn, acs, atn

Log (base 10), Lgn (base e), erf

Abs, int, sqr

Operators: +, -, /, \*, \*\* (or ^)

Variables: a, b, c, d, e, f, g

---

Enter the function :

: cos(5\*x) <CR>

2>yyc 1 2 <CR>

Comparison of Y values for 1 & 2 stored at plot 3

2> p 3 <CR>

---

A plot of the two generated functions are shown in Figure 64,  $y = \sin(3x)$  and  $y = \cos(5x)$ . A comparison of the y values generated using the YYC command for these two functions is shown in Figure 65.

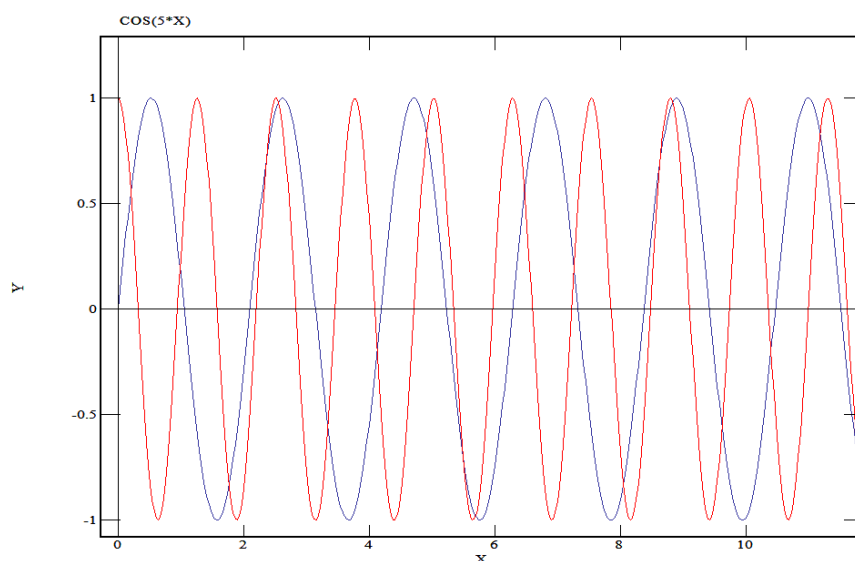


Figure 64 Displayed is a plot of the two functions  $y = \sin(3x)$  (in blue) and  $y = \cos(5x)$  (in red).

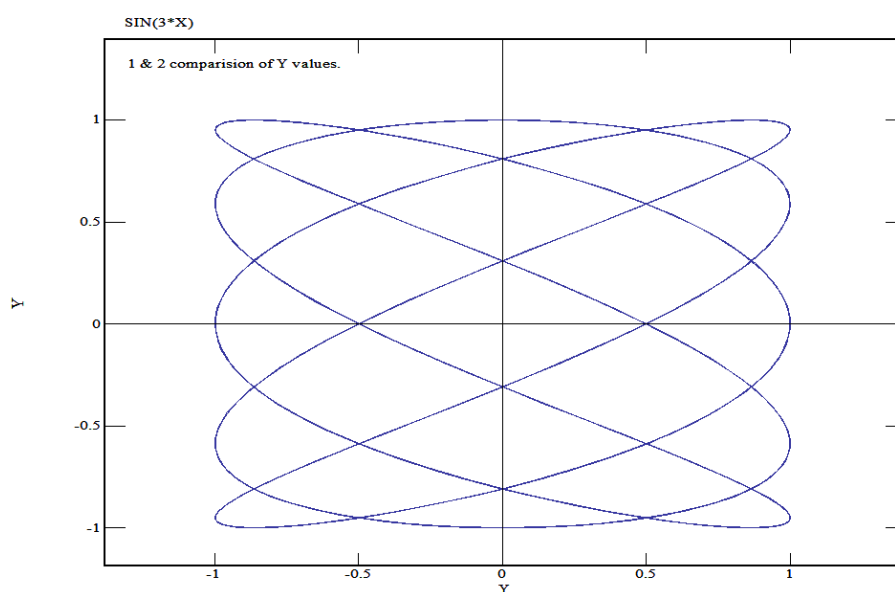


Figure 65 The y values of the two plots of Figure 64 are plotted against each other using the command `YYCOMPARE`. The result is a Lissajous curve [8].

### 8.103 ZOOM (z)

The command `ZOOM` allows the user to expand a section of the current plot so that the selected region occupies the whole of the plot area. After the command is received the program waits for the user to select the region of interest using the mouse. The user places the mouse pointer over the corner of the region of interest and presses and holds the left mouse button down. While continuing to press the button, the user moves the mouse pointer to the

opposite corner of the region of interest. A box will be drawn and will expand with the mouse movement to highlight the selected region of interest (See Figure 66). Release the mouse button and the region will be redrawn as shown in Figure 67.

The result of the zoom command is identical to an appropriate SET and PLOT combination and will leave the axes at the setting until a further SET, ZOOM or FREE command is executed.

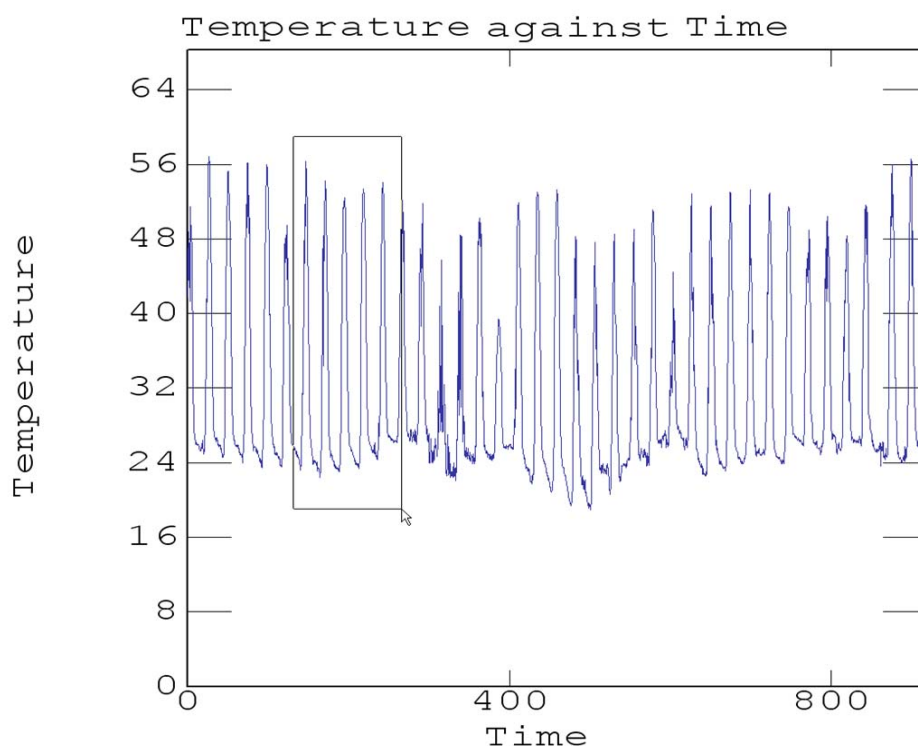


Figure 66 The ZOOM command allows the KGRAPH user to focus in on a region of the plot. Shown is a plot of temperature time data on which the user has issued the ZOOM command and is in the process of selecting a region by clicking and dragging the mouse over the region of interest. The result of this process is shown in Figure 67.

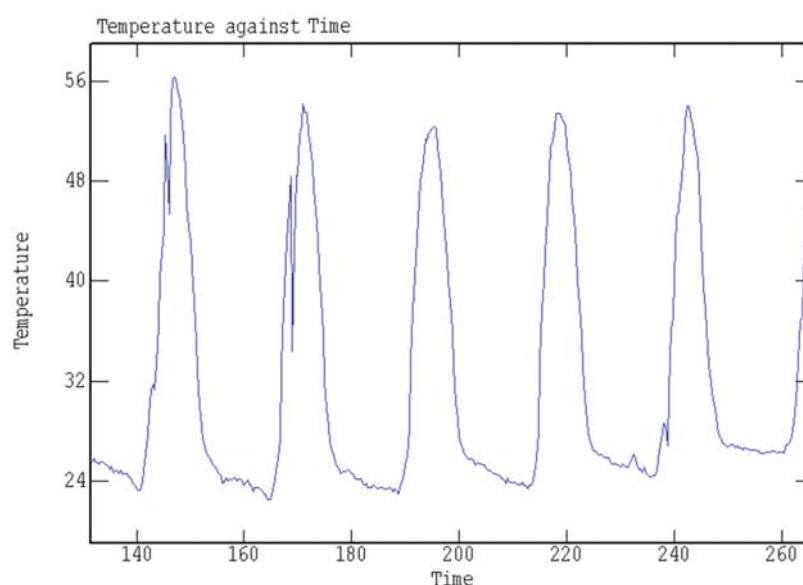


Figure 67 The resultant plot following the use of the ZOOM command on the plot of Figure 66 is shown.

## 9. MENU Items

Dropdown menus provide a way of accessing most of the functions of KGRAPH using the mouse only. These menus are found at the top of the plot window.

There are seven menus. The following subsections describe the uses of these menus.

### 9.1 File menu

The file menu items are concerned with transfer of information from computer memory to file storage and output devices. Each item within this menu acts in the same manner as the command, see Figure 68.

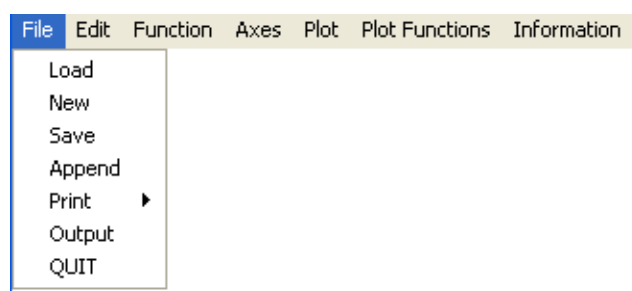


Figure 68 A snapshot of the KGRAPH menu for file transfers and output.

## 9.2 Edit Menu

The edit menu provides access to commands which alter the content of the plots in memory (Figure 69).

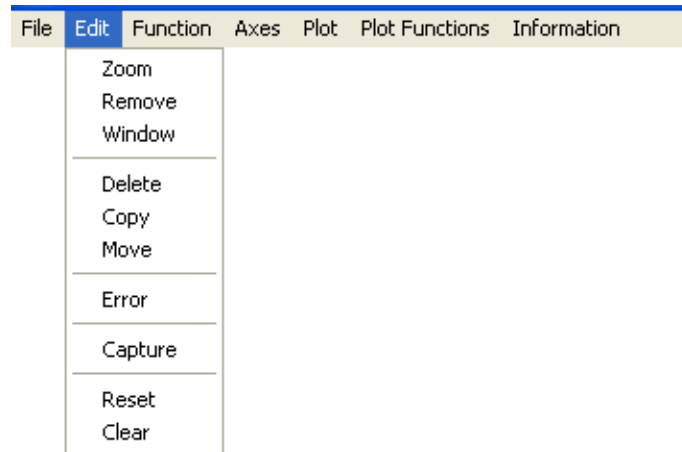


Figure 69 The KGRAPH edit menu items for altering plots within the memory.

If numeric input is required a dialog window is opened into which the appropriate numbers can be entered, for example see Figure 70 . The up and down arrows to the right of the numeric fields allow the numbers to be increased or decreased, respectively, using the mouse. Pressing the tab key will cycle through the fields. The cancel button will close the dialog box without changing the data.

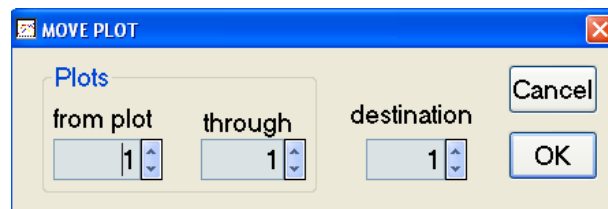


Figure 70 The menu selections 'copy' and 'move' require information as the range of plots and the destination location. The dialog used to input the range and destination is shown.

## 9.3 Function Menu

The function menu provides the user access to functions which act on the data of the current plot (see Figure 71). A full list of function commands are provided in Appendix C.

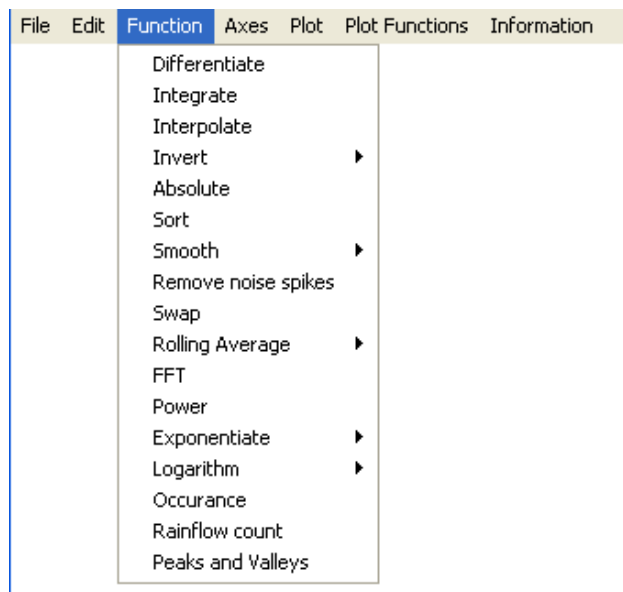


Figure 71 The KGRAPH function menu items for altering the points in the current plot or produce a second plot that is a function of the data in the current plot.

## 9.4 Plot Menu

This menu provides access to commands which allow navigation between plots in memory (see Figure 72).

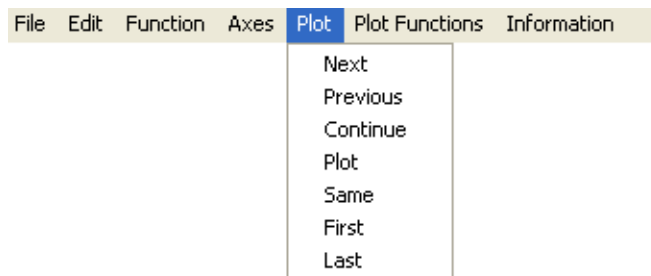


Figure 72 The KGRAPH plot menu commands for navigating around the plots in memory.

## 9.5 Plot function Menu

The plot function menu provides access to plot manipulation functions. The top items manipulate plots with other plots and the bottom items manipulate the current plot (See Figure 73).

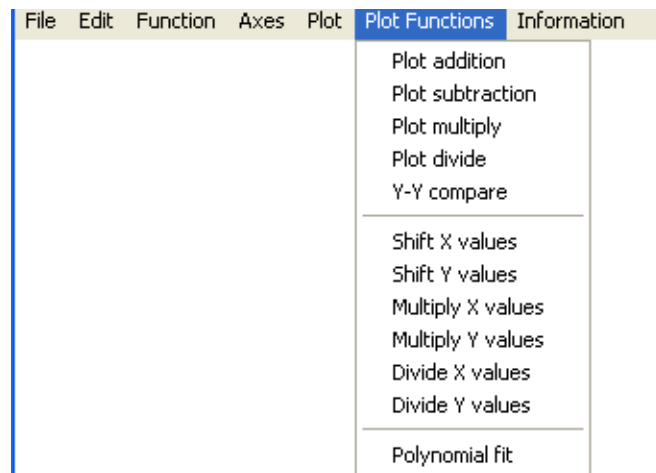


Figure 73 The KGRAPH plot function menu for commands to manipulate plots. The top items manipulate plots with other plots and the bottom items manipulate the current plot.

A dialog box is shown to allow the user to select input using the mouse where input is required to undertake a function. Figure 74 shows a typical dialog box used to input source plots and destination plot for the functions requiring two input plots.

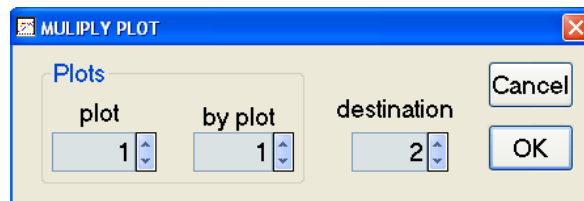


Figure 74 A dialog box used to input source plots and destination plot for the functions requiring two input plots is shown.

A dialog box where a function involves a constant that can be applied to a number of plots over a defined range is shown in Figure 75.

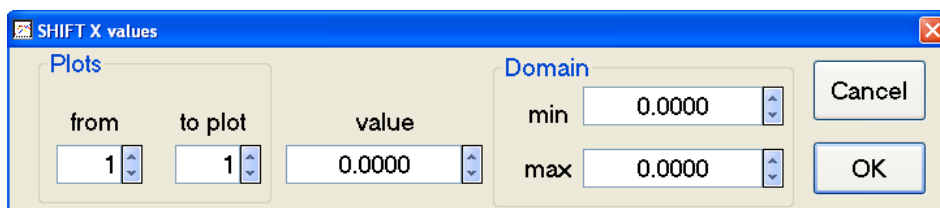
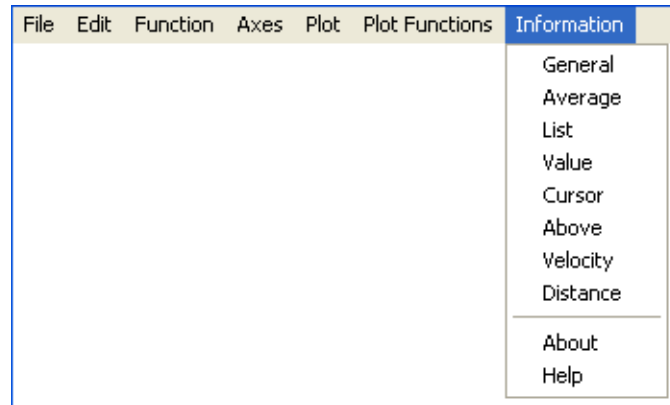


Figure 75 A dialog box is shown where a function involves a constant that can be applied to a number of plots over a defined range.



## 9.6 Information Menu

This menu provides information or derived information on the current plot. The first item 'General' invokes the INFORMATION command. The other commands invoke the command as described in Alphabetic command descriptions in Chapter 8 (See Figure 76).



*Figure 76 Shown is the information menu. This menu contains commands that provide information about the current plot.*

## 10. References

1. Kennett, S. R. (1987) *The interactive plotting program graph – revised manual*. MRL-GD-0010, Materials Research Laboratory
2. Golay, M. J. E. and Savitzky, A. (1964) *Analytical Chemistry* **36** 1964 16-27
3. [http://en.wikipedia.org/wiki/Cooley%E2%80%93Tukey\\_FFT\\_algorithm](http://en.wikipedia.org/wiki/Cooley%E2%80%93Tukey_FFT_algorithm), January 2012
4. [http://www.w3schools.com/html/html\\_colornames.asp](http://www.w3schools.com/html/html_colornames.asp), January 2012
5. Atkinson, K. E., (1989) *An introduction to Numerical Analysis*, John Wiley and Sons
6. Matsuishi, M. and Endo, T. (1968) *Fatigue of metals Subjected to Varying Stress*. Japan Society of Mechanical Engineers, March 1968
7. Press, W. H., (1988) *Numerical Recipes in C, The art of Scientific Computing*, Cambridge University Press
8. [http://en.wikipedia.org/wiki/Lissajous\\_curve](http://en.wikipedia.org/wiki/Lissajous_curve), January 2012

## Appendix A: Plot and File Control Command

The commands that deal with the input and output of data to the screen and to files are listed as follows:

command	abbreviation	description
append <#> <filename>	app	Append plot file data to existing data.
continue <[#<..#>]> or <#>	c	Continues to overlay next data.
copy <[#<..#>]> #	cop	Copy range of plots to a new location.
delete <[#<..#>]>	del	Deletes a plot or range of plots from memory.
export <[#<..#>]> <filename>	exp	Export data as ASCII for other programs.
flowover <#>	flo	Set the overflow value.
increment <#>	inc	Increment the plot pointer.
load <#> <filename>	l	Load data from file and plot data.
new	new	Create a new empty plot in memory.
next <#>	n	Plot next plot in memory.
move <[#<..#>]> <#>	mov	Move a range of plots to a new location.
plot <#>	p	Redraw current plot or plot #.
previous	pre	Plot previous plot from memory.
same	sa	Overlay the same data again.
save <[#<..#>]> <filename>	sav	Save data to a file.

## Appendix B: Axes Control Commands

In the following list the commands that alter the appearance of the plot are presented.

command	abbreviation	description
arrow	arr	Allows an Arrow to be placed on a plot.
background <Filename>	bac	Toggles on a background picture.
box	box	Toggles between bottom half axes and full box axes.
elevation	ele	Change the elevation of the viewing angle (3D)
pen <#>	pen	Select the line pen colour.
Pen width <#>	pen w	Select the width of the line.
Pen vary <#> <#> <#>	pen v	Vary the colour of the line using the values in a second plot.
Pen fill <#>	pen fil	Select the fill pen colour.
Pen grid <#>	pen gri	Select the grid pen colour.
Pen grid width <#>	pen gri w	Select the grid pen width.
Pen box <#>	pen box	Select the box pen colour.
Pen box width <#>	pen box w	Select the box pen width.
Text <text>	tex	Add text to the plot.
Dash	das	Toggles between dashed and continuous lines.
Fill	fil	Toggles between areas filled and line plotting.
Font	fon	Scales the default font size.
Free	fre	Removes the effect of set axes.
Grid	gri	Toggles between tick marks and a full grid.
grid <#> <#>	gri	(3D plot only) Alter the size of the surface grid.
Ldash	lda	Alters the lengths of the spaces and dashes.
Legend	leg	Toggles a plot legend on and off.
Midpoint	mid	Toggles between midpoint and histogram line plotting.
Mark <x/y> <#>	mar	Sets the number of sub marks on the x and y axes.
point	poi	Toggles between point and line plotting.
Rapid <#>	rap	Plots every #th point. Speeds up plotting.
Reset	res	Resets the axes and flags to the start-up values.
Set <#> <#> <#> <#>	set	Predetermines the axes spans of subsequent plots.
Size <#>	siz	Sets the size of the symbols used in point plotting.
Subtitle <subtitle line>	sub	Replaces the subtitle in the current plot.
Title <title line>	tit	Replaces the title in the current plot.
Type <#>	typ	Sets the type of symbol used in point plotting.
Xfree	xfr	Allows auto-scaling of the x axis.

Xlabel <x axis label>	xla	Replaces the x axis label of the current plot.
Xlog	xlo	Toggles between logarithmic and linear x axis.
Xset <#> <#>	xse	Sets the x axis span of the subsequent plots.
Xspacing <#>	xsp	Sets the spacing of the x axis marks.
Yfree	yfr	Allows auto-scaling of the y-axis.
Ylabel <y-axis label>	yla	Replaces the y-axis label of the current plot.
Ylog	ylo	Toggles between logarithmic and linear y-axis.
Yspacing <#>	ysp	Sets the spacing of the y axis marks.
Yset <#> <#>	yse	Sets the y-axis span of the subsequent plots.
Zfree	zfr	Allows auto-scaling of the z-axis.
Zlabel <z-axis label>	zla	Replaces the z-axis label of the current plot.
Zlog	zlo	Toggles between logarithmic and linear z-axis.
Zspacing <#>	zsp	Sets the spacing of the z axis marks.
Zset <#> <#>	zse	Sets the z-axis span of the subsequent plots.
Zoom	z	Uses the mouse to select a section of current plot.

## Appendix C: Function Commands

A list of functions that can be used to alter the content of a plot or examine some characteristics of a plot is presented in the following.

Command	abbreviation	description
above <#> <#> <#>	abo	Display the proportion of data above a given y value.
Average	ave	Display the average of the data in a zoom box .
cursor <x/y>	cur	Move a mouse directed cursor and return its location
C3d	c3d	Convert point aerial density to 3D plot.
Combine <x/y/z> <#> <x/y/z> <#> <x/y/z> <#>	com	Combine the data of different axes and plots to form a new plot.
Degree	deg	Convert the y values from radian to degrees.
Differentiate <#> <[#<..#>]>	dif	Differentiate the data with respect to x.
distance <#> <#> <#> <#>	dis	Measure the distance between points on a plot of latitude and longitude (degrees).
ffit	ffit	Fit a function to the data of the current plot.
Fft <#>	fft	Fast Fourier transform of the displayed plot.
function	fun	Create data from a function.
fourier <#> <[#<..#>]>	four	Create a plot of the □coefficients of the Fourier transform of the current plot.
highlight	hig	Place arrow heads on curve indicating plot order.
Integrate <[#<..#>]>	int	Integrate the data with respect to x.
interpolate <#>	interp	Create # interpolated points.
invfourier <#> <[#<..#>]>	inv	Create a plot based on the displayed Fourier transform coefficients.
noise <#>	noi	Remove noise spikes of fractional plot height # from the current plot data.
Occurance <#> <#> <#>	occ	Create a histogram of the occurrences of y values in the current plot.
Padd <#> <#> <#>	pad	Add the data of two plots.
papp <#> <#>	pap	Append the data of two plots.
pdivide <#> <#> <#>	pdi	Divide the data of the first plot by the second.
peak <#> <#>	pea	Determine area, 10%, 50%, 90%, FWHH of a peak.
pmultiply <#> <#> <#>	pmu	Multiply the data of the first plot by the second.
power <#> <[#<..#>]>	pow	Create a plot of the Fourier power spectrum of the selected plot or plots.
psubtract <#> <#> <#>	psu	Subtract the data of the first plot by the second.
pfit <#> <#> <#>	pfit	Fit a polynomial of order #.

<b>Command</b>	<b>abbreviation</b>	<b>description</b>
radian	rad	Convert the y values from degrees to radian.
Rotate <#>	rot	Rotate the data around origin by # degrees.
slice <[#<..#>]> <#> <#> <#>	sli	Make a new plot from slices through a set of plots.
sort <[#<..#>]>	sor	Sort the data into increasing x values.
smooth <[#<..#>]> <#> <#>	smo	Smooth the data using # points either side.
swap <[#<..#>]>	swa	Swap the x and y values of a range of plots.
value <#>	val	Determine the value/s of a point or selection of points.
velocity <#>	vel	Determine the average velocity at a selection of positional points along a GPS track.
Xdivide <[#<..#>]> <#>	xdi	Divide the x values of the current plot by #.
xdexp <[#<..#>]> <#>	xde	Replace the x data with the exponentiation of the y data.
xdlog <[#<..#>]> <#>	xdl	Replace the x data with the log10 of the y data.
xinvert <[#<..#>]>	xin	Invert the x values of the data of a range of plots.
xshift <[#<..#>]> <#> <#> <#>	xsh	Add a constant # to the x values of the current plot.
xmult <[#<..#>]> <#>	xmu	Multiply the x values of the current plot by #.
ydivide <[#<..#>]> <#>	ydi	Divide the y values of the current plot by #.
ydexp <[#<..#>]> <#>	yde	Replace the y data with the exponentiation of the y data.
ydlog <[#<..#>]> <#>	ydl	Replace the y data with the log10 of the y data.
yinvert <[#<..#>]>	yin	Invert the y values of the data of a range of plots.
yshift <[#<..#>]> <#> <#> <#>	ysh	Add a constant # to the y values of the current plot.
ymult <[#<..#>]> <#>	ymu	Multiply the y values of the current plot by #.
yyc <#> <#> <#>	yyc	Y values with y values comparison of two plots.
win <[#<..#>]> <#> <#>	win	Delete data of a range of files outside a window # #.

## Appendix D: Screen Display and Printing Commands

A list of screen display and printing commands is provided in the following:

<b>command</b>	<b>abbreviation</b>	<b>description</b>
clear	clr	Clear the display.
pen <#> or <+> or <->	pen	Select the colour of the line.
Pen Vary <#> <#> <#>	pen var	Select the pen colour to vary with the values in a plot.
Pen fill <#>	pen fill	Select the fill pen colour.
Pen box <#>	pen box	Select the box pen colour.
Pen grid <#>	pen gri	Select the grid pen colour.
Pen weight <#>	pen wei	Select the width of the line.
Text <text>	tex	Add text to the plot.
Print <filename>	pri	Send a copy of the screen to the printer or file.
wmf	wmf	Toggle duplicate output plot to a windows metafile.



## Index

%uxerror	20	empty plot	<i>See</i> NEW
%uxyerror	23	ENDIF	68
%uyerror	21	error bars	17
%xerror	18	Error bars	<i>See</i>
%xyerror	22	exchange x and y data	<i>See</i> SWAP
%yerror	19	EXPORT	46
.csv	<i>See</i> Comma separated values	Fast Fourier Transform	<i>See</i> FFT
ABOVE	27	FFIT	46
angle of elevation	<i>See</i> ELEVATION	FFT	48
APPEND	29	FILL	48
approximate location	<i>See</i> CURSOR	fill pen colour	80
available point types	105	filling the area under the plot	48
AVERAGE	30	fit an algebraic function	46
background	34	fix the mark spacing	115
BOX	33	FLOWOVER	50
C3D	35	FONT	51
CAPTURE	33	FOURIER	51
colour selector	79	FREE	52
COMBINE	37	Full width at Half Height	<i>See</i> FWHH
combine the data of two plot	77	FUNCTION	53
Comma separated values	15	function menu	121
command line prompt	9	FWHH	78
compare y data of two plots	116	G3D	55
compound commands	25, 26	Gaussian	47
CONTINUE	37	Gauss-Jordan	82
CONTOUR	38	GPS position	42
Coordinate display	107	GRID	35, 55
COPY	39	grid pen colour	80
copy buffer	32	HIGHLIGHT	57
CURSOR	41	histogram	71
DASH	41	IF57, 68, 69	
Dashed line	41	increment	58
data exponentiation	109	INFO	60, 123
DELETE	41	Input data and file types	12
Delete points	<i>See</i> REMOVE	INTEGRATE	58
density of points	<i>See</i> C3D	interpolation method	59
DIFFERENTIATE	43	interpolation of data with error bars	60
digitise a graph image	34	KGRAPH variables	67
direction of motion	<i>See</i> HIGHLIGHT	LDASH	41, 61
discreet Fourier transform	51	least-squares-fitted polynomial	<i>See</i> PFIT
distance between two points	42	LEGEND	62
Dropdown menus	120	line plotting	85
edit menu	120	line width	80
ELEVATION	44	LIST	40, 63

## UNCLASSIFIED

DSTO-GD-0718

LOAD	64	SAME	33, 95
logarithmic axes	112	SET	33
LOOP	68	SORT	59
MACRO	66	startup file	26
macro files	66	SUBTITLE	40, 102
MARK	69	SWAP	103
MIDPOINT	70	System Requirements	6
mouse wheel	11	TEXT	103
MOVE	71	text output	74
multiply by a constant	112	TITLE	104
named colours	79	title and label information	24
NEW	34, 72	TYPE	104
NEXT	73	Uncertainty	44
NOISE	73	user variables	67
noise spikes removal	73	VALUE	106
OCCURRENCE	74	VELOCITY	108
OUTPUT	74	WINDOW	108
over flow value	50	x, y value pairs	14
OVERLAY	75	x, y, z value triplets	15
PADDITION	76	XDEXP	109
paste buffer	32, 34	XDIVIDE	109
PDIVIDE	77	XDL	111
PEAK	78	XFREE	52
peaks and valleys	See PVA	XINVERT	110
PEN	79	XLABEL	111
pen colour	79	XLOG	112
PFIT	81	XMULTIPLY	112
PLOT	83	XSET	113
plot function menu	122	XSHIFT	114
PMULTIPLY	84	XSPACING	115
POINT	85	y values	14
point type	See TYPE	YDEXP	109
POLAR	85	YDIVIDE	109
POWER	88	YDL	111
power spectrum	See POWER	YFREE	52
PREVIOUS	86	YINVERT	110
PSUBTRACT	86	YLABEL	111
PVA	87	YLOG	112
Quick Run Through	9	YMULTIPLY	112
QUIT	91	YSET	113
RAINFLOW	91	YSHIFT	114
RAPID	92	YSPACING	115
REMOVE	93	YYCOMPARE	116
Repeat Count	25	ZDEXP	109
RESET	93	ZDIVIDE	110
ROLL	94	ZDL	111
ROTATE	94	ZFREE	52

UNCLASSIFIED

ZINVERT	110	ZOOM	118
ZLABEL	111	ZSET	113
ZLOG	112	ZSHIFT	114
ZMULTIPLY	112	ZSPACING	115

<b>DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION DOCUMENT CONTROL DATA</b>					
				1. PRIVACY MARKING/CAVEAT (OF DOCUMENT)	
2. TITLE  KGRAPH – Interactive Plotting and Analysis			3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION)  Document (U) Title (U) Abstract (U)		
4. AUTHOR(S)  S.R. Kennett			5. CORPORATE AUTHOR  DSTO Defence Science and Technology Organisation 506 Lorimer St Fishermans Bend Victoria 3207 Australia		
6a. DSTO NUMBER DSTO-GD-0718		6b. AR NUMBER AR-015-484		7. DOCUMENT DATE January 2013	
8. FILE NUMBER 2012/10156791/1		9. TASK NUMBER NAV 07/359		10. TASK SPONSOR CMPD	
				11. NO. OF PAGES 128	
				12. NO. OF REFERENCES 7	
13. DSTO Publications Repository  <a href="http://dspace.dsto.defence.gov.au/dspace/">http://dspace.dsto.defence.gov.au/dspace/</a>			14. RELEASE AUTHORITY  Chief, Maritime Platforms Division		
15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT  <p style="text-align: center;"><i>Approved for public release</i></p>					
OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SA 5111					
16. DELIBERATE ANNOUNCEMENT  No Limitations					
17. CITATION IN OTHER DOCUMENTS Yes					
18. DSTO RESEARCH LIBRARY THESAURUS  Interactive plotting, graphical analysis					
19. ABSTRACT This report forms a user guide for the Windows based interactive graphics plotting and analysis program called KGRAPH The program is based on an earlier program which was the subject of an earlier Materials Research Laboratories general document MRL-GD-010					