# ACE/gr User's Manual

Graphics for exploratory data analysis

***Paul J. Turner***
***Center for Coastal and Land-Margin Research***
***Oregon Graduate Institue of Science and Technology***
***19600 NW von Neumann Dr., Beaverton, Oregon, 97006-1999***

Software Documentation Series, SDS3, 91- 3
Copyright 1991-1993 Paul J Turner

Copyright 1991-1993 by Paul J Turner

All rights reserved.

ACE/GR IS PROVIDED “AS IS” AND WITHOUT ANY WARRANTY EXPRESS OR IMPLIED. THE USER ASSUMES ALL RISKS OF USING ACE/GR. THERE IS NO CLAIM OF THE MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

YOU MAY MAKE COPIES OF ACE/GR FOR YOUR OWN USE, AND MODIFY THOSE COPIES. YOU MAY NOT DISTRIBUTE ANY MODIFIED SOURCE CODE OR DOCUMENTATION TO USERS AT ANY SITES OTHER THAN YOUR OWN.

FrameMaker and Frame Technology are registered trademarks of Frame Technology Corporation.
Adobe and PostScript are registered trademarks of Adobe Systems Incorporated.
Sun Microsystem, Sun Workstation, XView, and OpenWindows are trademarks of Sun Microsystems.
Open Look is a trademark of AT&T.
UNIX is a registered trademark of AT&T Bell Laboratories.
X Window System is a trademark of the Massachusetts Institute of Technology.
OSF/Motif and Motif are trademarks of the Open Software Foundation.

v2.10

CHAPTER 1 Overview 5

1.1 Introduction 5
1.1.1 About ACE/gr 5
1.1.2 Terminology 5
1.2 Execution 6
1.3 Command line parameters 6
1.3.1 SYNOPSIS 6
1.4 Setting ACE/gr defaults 11
1.5 Environment variables 13
1.6 Batch printing 14
1.7 Context sensitive help 14
1.8 Contacting the author 14

CHAPTER 2 The Calculus of ACE/gr 15

2.1 Using ACE/gr 15

CHAPTER 3 Guide to Menus and Popups 24

3.1 Overview 24
3.2 FILE 25
3.2.1 Read sets... 25
3.2.2 Read parameters... 27
3.2.3 Read block data 27
3.2.4 Write sets... 28
3.2.5 Write parameters... 28
3.2.6 Save all... 28
3.2.7 Clear all... 29
3.2.8 Print 29
3.2.9 Printer setup... 29
3.2.10 Command interpreter... 29
3.2.11 Status... 29
3.2.12 Results 30
3.2.13 About... 30
3.2.14 Exit 30
3.3 VIEW 31
3.3.1 Graphs 31
3.3.2 Define world 33
3.3.3 Define view 33
3.3.4 Autoscale 34
3.3.5 Draw options 34
3.3.6 Title/subtitle 34
3.3.7 Ticks/tick labels 34
3.3.8 Frame 37
3.3.9 Symbols 37
3.3.10 Error bars 39
3.3.11 Legends 39
3.3.12 Strings & Things 40
3.3.13 Flip X-Y 42
3.3.14 Invert X 42

3.3.15 Invert Y 42
3.3.16 Locator 42
3.4 EDIT 43
3.4.1 Transformations 43
3.4.2 Set operations 50
3.4.3 Region operations 55
3.4.4 Edit sets 56
3.4.5 Point operations 57
3.4.6 Block data 58
3.5 Main panel items 58
3.5.1 Draw 58
3.5.2 Locator 58
3.5.3 Calc 58
3.5.4 Scrolling and scaling buttons 59
3.6 Shortcuts 59

CHAPTER 4 Command interpreter reference and parameter file format 61

4.1 Introduction 61
4.2 Reference 61
4.2.1 Exit 63
4.2.2 Devices 63
4.2.3 Display 64
4.2.4 Set operations 65
4.2.5 Block data 66
4.2.6 Graph operations 67
4.2.7 Transformations 67
4.2.8 Autoscaling 68
4.2.9 Graph focus 68
4.2.10 Locator 69
4.2.11 IO 69
4.2.12 Boxes 69
4.2.13 Lines 71
4.2.14 Strings 72
4.2.15 World 73
4.2.16 World stack 73
4.2.17 Viewport 74
4.2.18 Title and subtitle 74
4.2.19 Sets 75
4.2.20 Legend 77
4.2.21 Graph frame 79
4.2.22 Graph axes 79
4.2.23 tick marks. 80
4.2.24 Axes tick mark labels 82
4.2.25 Axes label strings. 84
4.2.26 Graphs 85

CHAPTER 5 Reference 87

5.1 Hot keys 87
5.2 Fonts 88
5.2.1 Changing fonts within a string 88

CHAPTER 1

# Overview

## 1.1 Introduction

### 1.1.1 About ACE/gr

ACE/gr is an XY plotting tool for workstations or X-terminals using X. A few of its features are:

- Polynomial regression, splines, running averages, DFT/FFT, cross/auto-correlation.
- Plots up to 10 graphs with 30 data sets per graph.
- User defined scaling, tick marks, labels, symbols, line styles, colors.
- Batch mode for unattended plotting.
- Read and write parameters used during a session.
- Hardcopy support for PostScript, HP-GL, and FrameMaker .mif format.

While ACE/gr has a convenient point-and-click interface, most parameter settings and operations are available through a command line interface (found in Files/Commands).

### 1.1.2 Terminology

Objects - Sets, Regions, Graphs, annotative text, lines, and boxes.

Graphs are the collection of data sets, tick marks, titles, etc. drawn to display the data. By default, there are 10 graphs available in this version of ACE/gr. Operations on graphs are found in the View/Graphs pullright. There are numerous popups that allow the manipulation of graph parameters accessible through theView pulldown. Most operate on what is referred to as the ‘current graph’, marked by small filled rectangles at the

corners of one graph. The 10 graphs available in this version of ACE/gr are numbered from 0 to 9.

Sets are collections of points, with optional associated values at these points. The associated values can be used for error bars and high-low open-close plots. Sets are connected to a graph, up to a maximum of 30 per graph and are numbered from 0 to 29. The number of points in a set is limited by the size of virtual memory and are numbered from 1 to N where N is the total number of points in the set. Operations to manipulate sets are found in Edit/Set operations,and the particular representation of a set can be changed inView/Symbols.

Regions are sections of the graph defined by the interior or exterior of a polygon, or a half plane defined by a line. A region defined by a linecan be above, below, to the right, or left of the line. Defining regions and operations on points inside regions are found in Edit/Region operations.

Parameters are the settings of symbols, line styles, colors, fonts, etc. used to define a particular graph.

## 1.2 Execution

As ACE/gr comes in two flavors, XView and Motif, there are two different names for the respective executables. The XView version is called **xvgr**, the Motif version, **xmgr**. Which version to execute depends on which was installed on your system. As a rule of thumb, **xvgr** will be found on SUN systems using OpenWindows, and **xmgr** will be found everywhere else. Contact your system administrator for further details.

## 1.3 Command line parameters

### 1.3.1 SYNOPSIS

**xvgr** (XView) or **xmgr** (Motif) [ -autoscale *x*|*y*|*xy* ] [ -noauto *x*|*y*|*xy* ] [ -arrange *rows columns* ] [ -cols *columns* ][ -rows *rows* ] [ -device *device_number* ][-eps][-noask] [ -batch *batch_file*][ -graph *graph_number* ] [ -graphtype *graph_type* ] [-printfile *graphics_output_file* ] [ -type *XY_data_set_type* ][ -ihl *IHL_formatted_file* ] [ -log *x*|*y*|*xy* ] [ -nxy ][-xydx][-xydy][-xydxdx][-xydydy][-xydxdy][-xyr][-xyd][-xyz][-ihl][-ihilo] [-block *block_data* ] [ -parameter *parameter_file* ] [-pexec *parameter_string* ] [ -result *result_file* ] [ -rvideo ] [-mono ] [ -world x*min ymin xmax ymax* ] [ -view *xmin yminxmax ymax* ] [ -source *data_source* ] [ -legend load ] [-refresh *value*] [-GXxor] [-GXinvert] [-bs] [-nobs] [-dc] [-nodc] [maxcolors *number_of_colors*] [-redraw] [-noredraw] [-seed *seed_value*] [-maxplot *number_of_sets*][-maxgraph *number_of_graphs*][*data_files* ]

- -autoscale *x*|*y*|*xy*

Autoscale the axis denoted by *x* or *y* or both axes by *xy* overiding any parameter file specified by the -p option.

- -noauto x|y|xy

Suppress autoscaling on the given axis (axes).

- -arrange *rows columns*

Arrange graphs in column major order starting from the lower left corner in a grid rows by columns. .For example, '-arrange 2 2' will position the first four graphs (0, 1, 2, 3) in a 2x2 matrix of graphs starting with graph 0 in the lower left corner of the plotting surface. This function is not as complete as the similar function found in View/Graphs/Arrange and gives a different result.

- -cols *columns*

Arrange the first *columns* graphs in *columns* columns.

- -rows *rows*

Arrange the first *rows* graphs in *rows* rows.

- -results *results_file*

Write the results of regression and anything else that gets written to the results popup to file *results_file*.

- -device *device_number*

Set the hardcopy device to *device_number.*

The device numbering follows:

1 .. PostScript landscape

2 .. PostScript portrait

3 .. FrameMaker .mif landscape

4 .. FrameMaker .mif portrait

5 .. HPGL landscape

6 .. HPGL portrait

- -eps

Set the PostScript driver to produce EPS.

- -printfile *graphics_output_file*

Write the graphics output to file *graphics_output_file* rather than spool graphics data to a printer.

- *batch_file*

Execute the commands in batch_file on startup of the window system.

- -noask

Assume the answer is yes to all requests, This means that files will be overwritten without asking if they exist, and **ACE/gr** will exit abruptly.

- -graph *graph_number*

Make *graph_number* the current graph. *Graph_number* is an integer between 0 and 9.

- -graphtype *graph_type*

Set the current graph type to *graph_type*. Where *graph_type* is one of:

xy - linear scaling along both x and y.

logx - linear scaling in y, log scaling in x.,

logy - linear scaling in x, log scaling in y.

logxy - log scaling in both x and y.

bar - vertical bar chart.

hbar - horizontal bar chart.

stackedbar - vertical stacked bar chart.

stackedhbar - horizontal stacked bar chart.

- -type *data_set_type*

The format of the next data source is *data_set_type*. Where *data_set_type* is one of xydx (XY data with error bars along X), xydy (XY data with error bars along Y), xydxdx (XY data with error bars along X of differing values), xydydy (XY data with error bars along Y of differing values), xydxdy (error bars along X and Y), xyr (a circle drawn at XY of radius R) and xyhilo (XY data with high-low open-close data). This setting remains in effect until a new format type is specified.

- -ihl *IHL_formatted_file*

Assume *IHL_formatted_file* or the next source of data is in IHL format (a local format).

- -nxy

The format of the next data source is X1, Y1, Y2, ..., Yn.

- -xydx

The format of the next data source is X Y DX

- -xydy

The format of the next data source is X Y DY

- -xydxdx

The format of the next data source is X Y DX1 DX2

- -xydydy

The format of the next data source is X Y DY1 DY2

- -xydxdy

The format of the next data source is X Y DX DY

- -xyz

The format of the next data source is X Y Z where Z is a value drawn as text at X, Y.

- -xyr

The format of the next data source is X Y R, where a circle of radius R is drawn at X, Y.

- -ihl

The format of the next data souce is in IHL format, this format leads to a data set of type XY.

- -hilo

The format of the next data source is X HIGH LOW OPEN CLOSE, where a symbol denoting the open, close, the high and the low values for an observation is drawn at X.

- -block

Assume the format of the next data file is block data. No sets are formed by reading block data, and only one set of block data is allowed in an active session. To create sets from the block data, use Edit/Block data.

- -log x|y|xy

Set the current graph type to logarithmic depending the the string “x” or “y” or both axes by “xy”.

- -parameter *parameter_file*
- -p *parameter_file*

Read the parameter file *parameter_file*.

- -pexec *parameter_string*

Interpret parameter_string as a parameter setting.

- -result *result_file*

Write results from regression to *result_file*

- -rvideo

Exchange the color indices for black and white.

- -mono

Limit colors to black and white, affects the display only.

- -dc

Allow double click canvas operations, overriding any application default setting.

- -nodc

Disallow double click canvas operations.

- -redraw

Redraw the entire page if ACE/gr receives an Expose event.

- -noredraw

Do not redraw the entire page when Expose events are received.

- -maxcolors *number_of_colors*

Set the maximum number of colors allocated.

- -world *xmin ymin xmax ymax*

Set the scaling of the axes for the current graph.

- -view *xmin ymin xmax ymax*

Set the viewport of the current graph to a rectangle described by (*xmin, ymin*) and (*xmax, ymax*).

- -source *data_source*

Set source for the next data file, the argument takes on the following values:

disk - data file is on disk

pipe - read from a pipe Assume the next command line argument is a command to be executed.

stdin - ACE/gr is in a pipe, read from standard input

The source setting stays in effect until the next -source command is given.

- -pipe

Read from stdin and plot each set when the end of the set is seen. Use the -pipe option to monitor the results coming from a model or other data stream.

- -legend load

For each data set read, make the set legend label the same as the filename the data set originated.

- -seed *seed_value*

Initialize the random number generator using seed seed_value. This value is used in a call to the library function srandom().

- -GXinvert
- -GXxor

GXinvert and GXxor set the type of graphics operation used to draw rubberband lines and draw the graph focus markers. In a colormapped display, the xor of a source and destination pixel may not generate a color visible on the drawing area. The default is to use xor to draw rubberband lines, if the lines don't show, try -GXinvert. This draw mode can be set via X resource settings (see below).

- -maxplot *number_of_sets*

Set the maximum number of data sets per graph. *Number_of_sets* must be greater than 30 and there is no support in the graphical user interface for more than 30 sets. Note that this is a new item and there may be problems. If -maxplot is used in conjunction with -maxgraph, then the -maxplot setting must precede the -maxgraph setting.

- -maxgraph *number_of_graphs*

Set the maximum number of graphs per session. *Number_of_graphs* must be greater than 10 and there is no support in the graphical user interface for more than 10 graphs. Note that this is a new item and there may be problems. If -maxgraph is used in conjuction with -maxplot, then -maxplot must precede the use of -maxgraph.

- -usage

Display a brief explanation of command line settings

- *data_files*

The files or commands in the case of a pipe from which data are read.

## 1.4 Setting ACE/gr defaults

On startup ACE/gr will look for a .xvgrrc (XView version) or a .xmgrrc (Motif) file in the users' home directory allowing changes to the default behavior of ACE/gr. The construction of the ACE/gr default file is the same as for a parameter file.

The following X resource settings are supported:

- Xvgr.InvertDraw: False
- Xmgr.invertDraw: False

Use GXinvert to draw rubberband lines and the graph focus markers if this resource is set to True.

- Xvgr.InitialDraw: 0

Set the initial refresh of the graph on the startup of xvgr to be done in the resize procedure rather than the refresh procedure. If the value is 0 then the initial draw is done in the refresh procedure, otherwise the initial draw is done on the value of this resource.

- Xvgr.ReverseVideo: False
- Xmgr.reverseVideo: False

A boolean resource used to exchange the color indices for black and white.

- Xvgr.MaxSets: 30
- Xmgr.maxSets: 30

Set the number of sets per graph.

- Xvgr.MaxGraphs: 10
- Xmgr.maxGraphs: 10

Set the number of graphs.

- Xvgr.MaxColors: 17
- Xmgr.maxColors: 17

Set the number of colors allocated. There is no support in the graphical portion of the user interface for more than 16.

- Xvgr.VerifyAction: No
- Xmgr.verifyAction: No

Applies to the Pick set operations and allows an opportunity to accept or cancel any operation performed.

- Xvgr.AllowDoubleClick: Yes
- Xmgr.allowDoubleClick: Yes

When Yes, allow double clicks on the canvas to bring up various popups depending on the location of the pointer when the double click occurs. Double clicking to the right of a graph brings up the View/Ticks popup set to the Y-axis, below the graph brings up the View/Ticks popup set to the X-axis, inside the graph brings up the files popup if no sets are active or the View/Symbols popup set to the set nearest the pointer. Double clicking above the graph brings up the View/Title-subtitle popup, to the right of the graph, the View/Legend popup.

The command line option to set this resource is -dc or -nodc.

- Xvgr.AutoscaleOnRead: No
- Xmgr.autoscaleOnRead: No

When this resource is Yes or True, allow autoscaling to occur each time a set is read from the File/Read sets popup.

- Xvgr.Backingstore: No
- Xmgr.backingstore: No

Have ACE/gr handle backing store if the server doesn't.

The command line option for this is -bs or -nobs.

- Xvgr.AllowRefresh: Yes

When the server does not do backing store, redraw the entire page when expose events are generated. Set either AllowRefresh to True or Backingstore to True, but not both as they perform similar functions using different techniques.

The command line option for this is -redraw or -noredraw.

## 1.5 Environment variables

- GR_PS_PRSTR

Set the command string to use for spooling to a PostScript printer. The default (unless changed during the build process) is "lpr -h".

- GR_MIF_PRSTR

Set the command string to use for printing the FrameMaker .mif format. The default (unless changed during the build process) is "cat >acegr.mif < ", i.e., output is to a file.

- GR_HPGL_PRSTR

Set the command string to use for spooling to an HPGL capable printer or plotter.

- GR_HDEV

Select the hardcopy device.

The device numbering is as follows:

1. PostScript landscape
2. PostScript portrait
3. FrameMaker interchange format (.mif) landscape
4. FrameMaker interchange format (.mif) portrait.
5. HPGL landscape
6. HPGL portrait

## 1.6 Batch printing

Executing ACE/gr as *grbatch* suppresses the initialization of the toolkit (both versions) and allows hardcopy plots to be made without intervention.

## 1.7 Context sensitive help

The XView version has support for context sensitive help. To use this feature, the Help key must be defined and the environment variable HELPPATH must include the path to the file xvgr.info. I use

HELPPATH /usr/local/help

and in my .xinitrc file I have the line

xmodmap -e 'keysym F1 = Help'

Like the rest of xvgr, the xvgr.info file is a work-in-progress.

## 1.8 Contacting the author

Due to time constraints, please look for solutions to problems with ACE/gr locally before contacting the author. The author reserves the right not to respond to every request received.

For bug reports, comments, etc., send mail to pturner@amb4.ccalmr.ogi.edu. The home of ACE/gr sources and documentation is ftp.ccalmr.ogi.edu [129.95.72.34] in CCALMR/pub/acegr.

# CHAPTER 2 The Calculus of ACE/gr

## 2.1 Using ACE/gr

The first thing to do to get a feeling for **ACE/gr** is execute the script `dotest` in the examples subdirectory of the distribution, if this is available. The examples displayed by the script will serve as an introduction to some of the features and the philosophy of **ACE/gr**. If you are not sure where the examples are located or how to run the script, contact your system administrator.

The following discussion illustrates the use of **ACE/gr** using a small data set and traces the steps from the initial invocation of **ACE/gr** to the final graph.

Using the data set, in XY format:

4.5 89.3

7.9 100.1

11.5 25.6

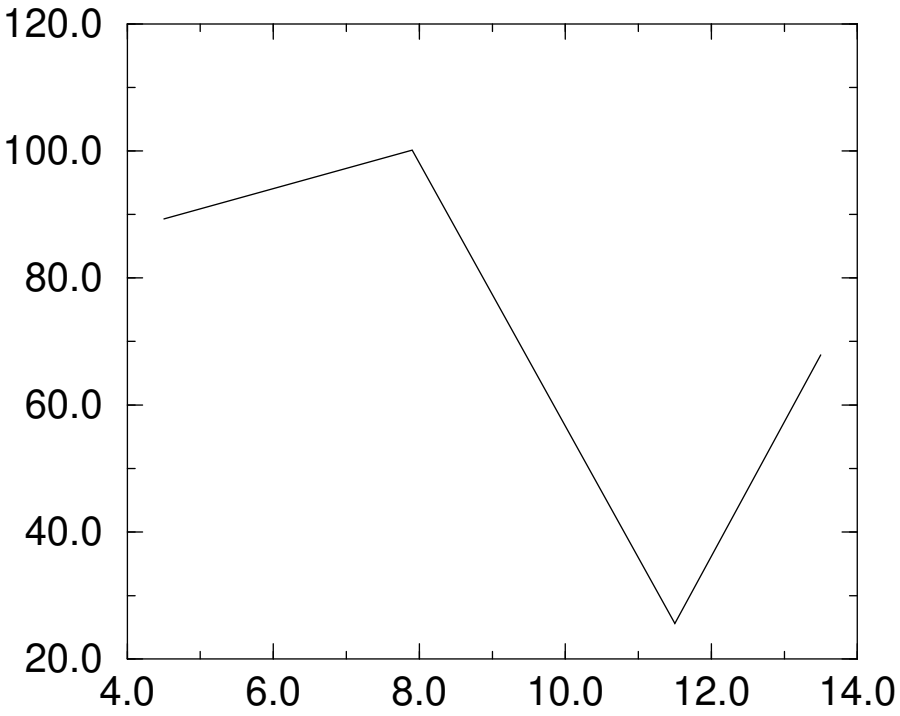
13.5 67.9

Saved to the file “intro.dat.”

Create the file using vi or whatever and execute **ACE/gr** as **xvgr** (XView/Open Look) or **xmgr** (Motif) depending on the desired interface:

```
unix% xvgr intro.dat
```

You should get a picture similar to the following:

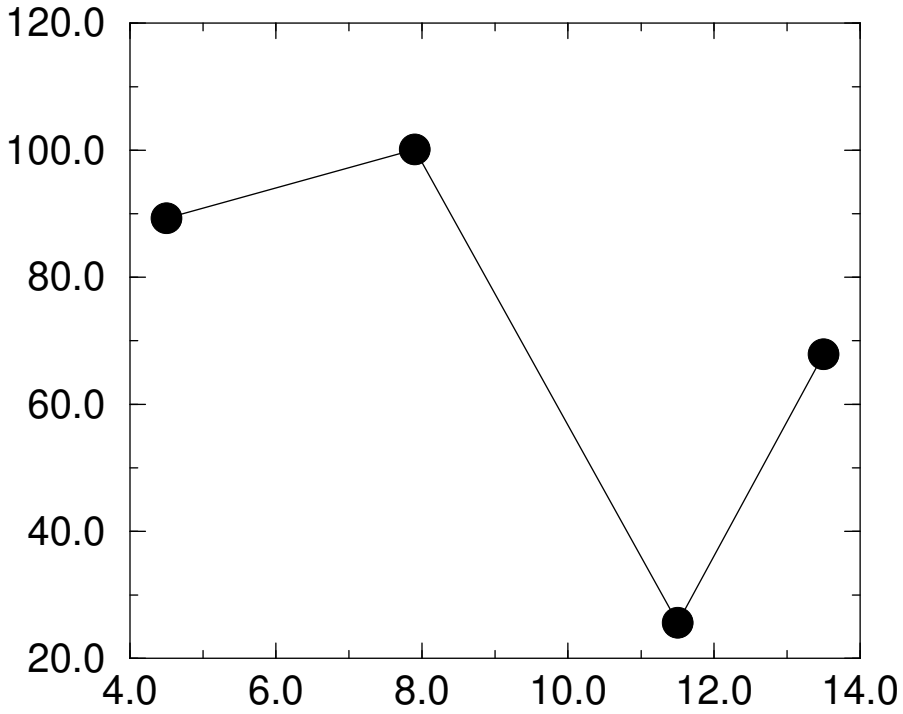



This graph was created using the FrameMaker .mif driver, and imported into this document. Because **ACE/gr** uses the Hershey fonts for screen text and FrameMaker uses other fonts there will be differences in the appearance of the graphs between the onscreen and FrameMaker graphs.

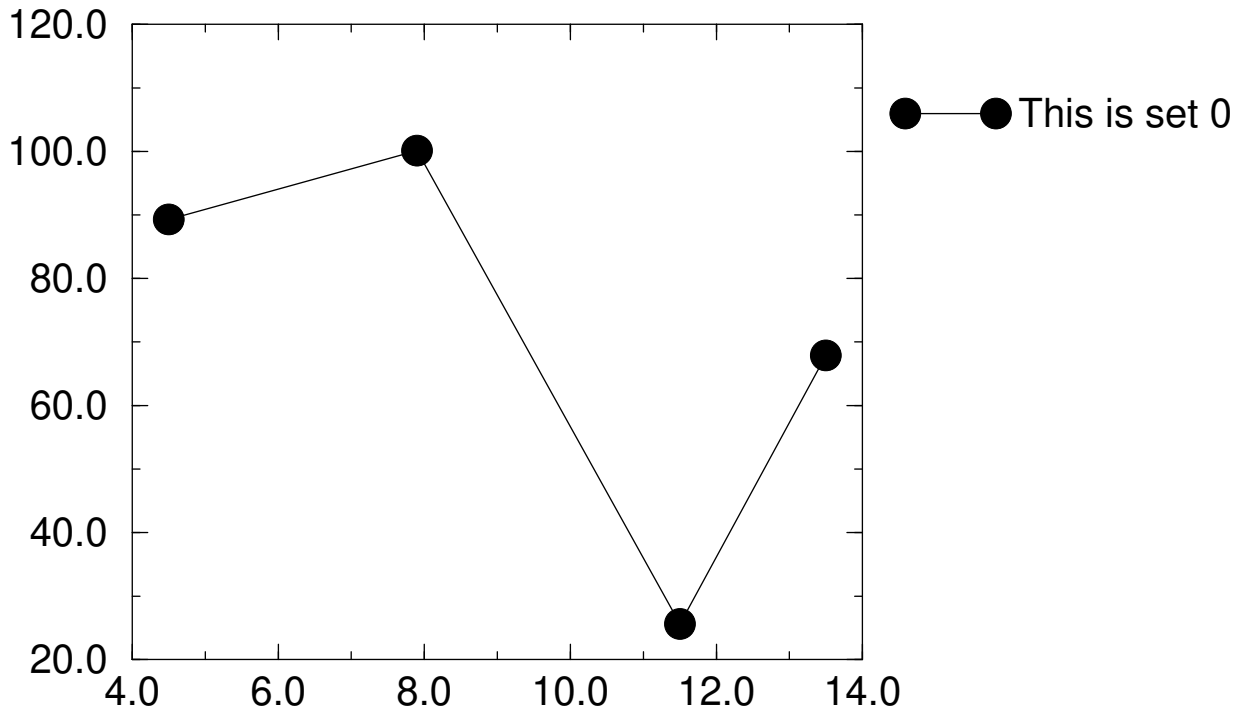
NOTE: **ACE/gr** has no understanding of inches or centimeters, a graph is positioned on the drawing area using viewport coordinates which are relative to the size of the drawing area or the hardcopy device. The graph printed above was printed using the .mif driver landscape mode using a viewport smaller than the default. Had the portrait version of the .mif driver been used with no change to the viewport, the graph would appear stretched along the Y-axis (the same thing would have happend with the PostScript or HPGL driver - there are plans to change this situation sometime in the future). To make the onscreen version look like the hardcopy version, it is necessary to resize

the **ACE/gr** drawing area to approximate the size of the hardcopy device (at least make it proportional to the hardcopy device).

The default for drawing a data set in **ACE/gr** is to connect each point with a line, given the small number of points, it might be better to add symbols to mark the data points. Open the popup View/Symbols to select a symbol for the data set. The graph below shows the data points drawn with a filled circle.

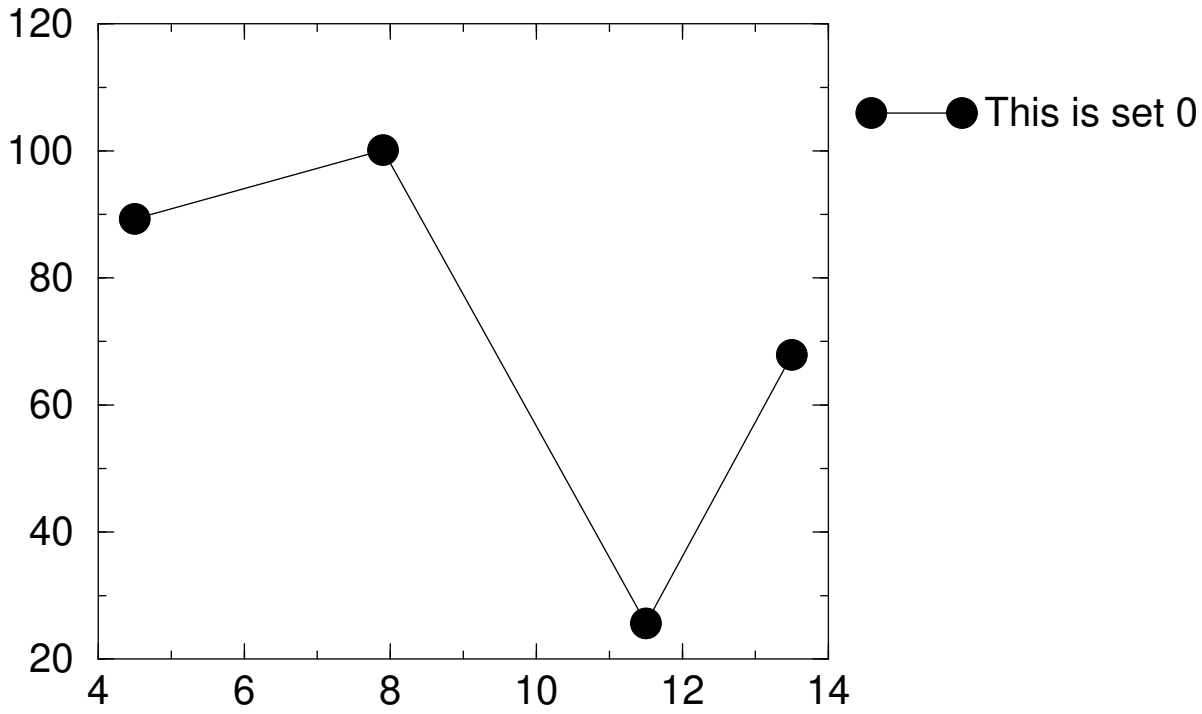



A legend is not really needed here, given only one data set, but let's add one anyway. Use View/Symbols to set the legend label for the data set (presently the only way to set the legend label is to define it here in View/Symbols or use the command interpreter). The relevent item is called "Legend:" and is a text item that will accept the legend label for the set. Next, open the View/Legends popup and with the first item in the popup, "Legend:," select ON to turn the legend on. The legend will appear in the default location, which is somewhere in the upper right corner of the drawing area (in general, this default location is not appropriate). The following graph is an approximation of what will be seen on the drawing area.

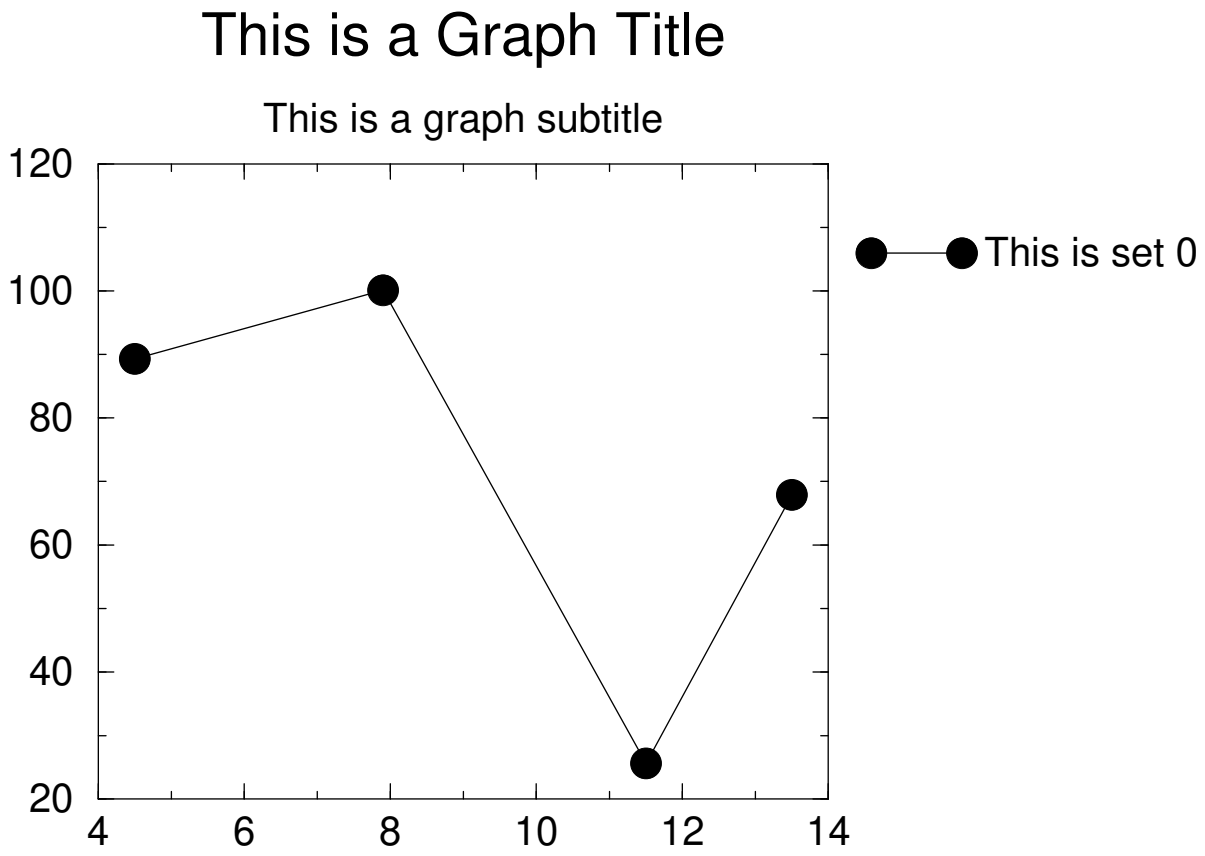



To make changes to the set legend, re-open the View/Symbols popup, and edit the “Legend:” text item.

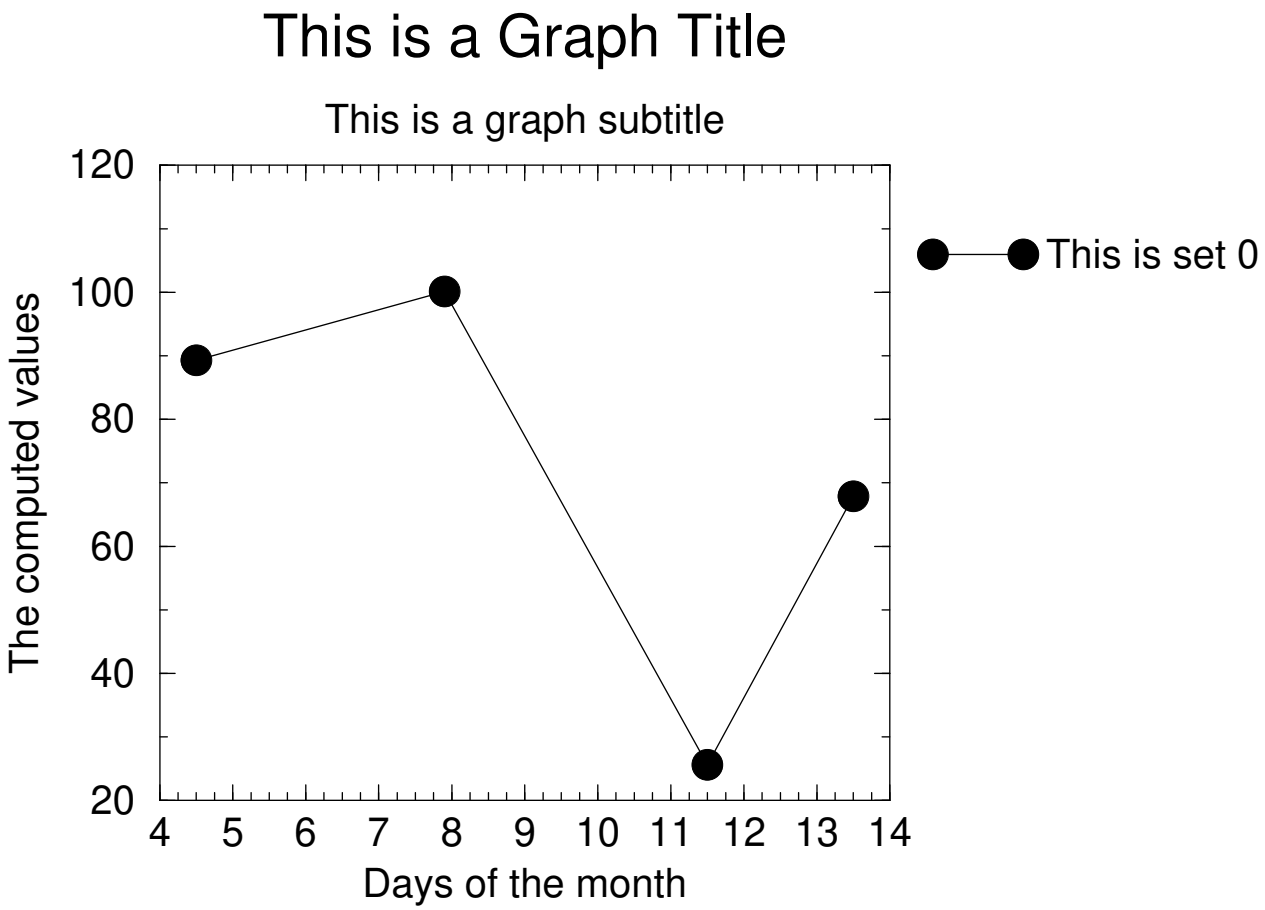
Things are looking better, but there are too many places to the right of the decimal point in the tick mark labels. To set the number of places to the right of the decimal point, spaces open the View/TIck/Ticklabel popup. Select the item to the right of the toggle “Tick labels” called “Props...” to open a popup that allows modifications to tick labels. The item “Precision:” is the one we want, a good value here is 0, for no places to the right of the decimal point. Since it would be nice to change the Y-axis tick label spacing also, use the “Apply to:” item to select “All axes, current graph.” Press the “Accept” button on this popup to register the change in precision for both the X and Y axes. This should give a graph as follows.




A title and subtitle for the graph would be a nice improvement. Select View/Title/Subtitle to open a popup that allows the definition of the graph title and subtitle. Fill in the text items and press “Accept” to register the new title and subtitle. Clicking on the button labeled “Props...” opens another popup allowing the selection of the fonts and colors for the title and subtitle. We now have something that looks like...

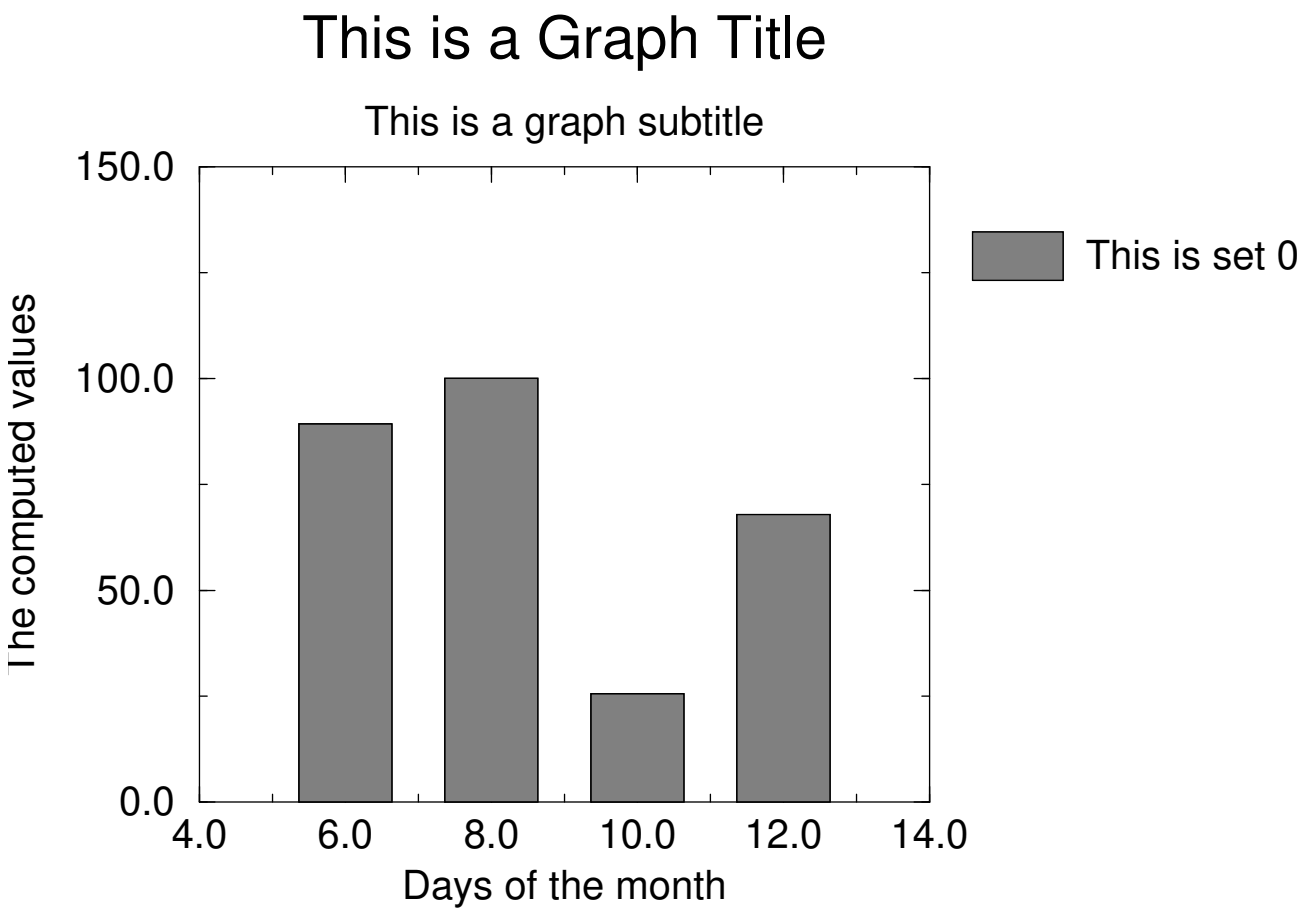



Would be nice to define the meaning of the axes, for this, open the View/Ticks/Tick labels popup again, filling in a text string at the item “Axis label.” Note that in the upper lefthand corner of this popup is a choice item labelled “Edit:.” The default is the X -axis and is the axis that will receive the label if the “Apply” button is pressed. To set the Y-axis label, change the “Edit” selection to Y-axis, fill in the text item as above and press “Apply.” While we have this popup open, let’s change the major and minor tick spacing on the X-axis. Select the X-axis to edit (using “Edit:”) and set the major tick spacing to 1 and the minor spacing to 0.25. With these changes to the axes labels and the tick spacing, the graph should appear as...




Just for fun, we'll change this graph from the default XY graph to a bar chart. To do this, requires some modification to the data, for illustrative purposes. Select the Edit/Transformations pullright (XView, the Motif version has a popup with similar selections) and select Load values. The Load values popup allows values to be assigned to a set in a sequential fashion. For the "Start:" item enter 6 (the selected set should be 0 and the "To:" item should be X). This indicates that for set 0, the X vector's values will begin at 6. Enter 2 for the "Step:" value, indicating that each point in X will be incremented by 2. Press the "Apply" button. The graph should redraw and the picture should reflect the modifications to the X vector, with the data now at X = 6, 8, 10, 12. Select the View/Graphs/Set type selection to bring up a popup used to change the type of an active graph. Select the graph type to be "Bar chart" and press "Apply." This should produce a bar chart. Since it is not obvious nor well documented, we'll set the fill pattern for the bar chart. Open the View/Symbols popup and in the upper right portion of the popup, peruse the "Fill" selections. In an XY graph context, these items select the type of fill for XY data sets. In the bar chart graph, the fill items select the type of fill for the bars. To produce a filled bar, set the "Fill:" choice to anything but "None" ("Fill as polygon" will do), and select the type of fill with the "Fill using:" choice item. In this case, select "Pattern." With the item "Pattern:" select the number value of the pattern. See the sec-

tion on Symbols for the mapping of the numbers to fill types. 10 is a good choice. Press the “Accept” button and you should be looking at something like...




Note the in the course of switching graph types that tick label’s precision changed, this is due to the autoscaling **ACE/gr** does whenever the graph type is changed. The fix here is to go back a few steps to the View/Tick/tick labels popup and with the “Props” popup for tick labels, reset the precision to 0 for both axes.

At this point we can call it a wrap, select File/Print to get a hardcopy. Note that **ACE/gr** is not WYSIWYG, although with some practice with resizing the drawing area and setting viewports, a close approximation can be achieved.

Before exiting **ACE/gr**, it might be wise to save the settings of the graph parameters, in order that this particular graph can be reconstructed given the data (or with a new set of data). Select File/Write parameters to open a popup to create a file that can be used for this purpose. Enter the file name in the item “Write parameters to:” and press the. “Accept” button to perform the write. If a file with the same name already exists, a

prompt will appear allowing the cancellation of the operation or the overwriting of the existing file. With this parameter file (called params.par in this case), **ACE/gr** can be fired up with the command (using xvgr as the program):

```
unix% xvgr -p params.par intro.dat
```

producing the graph constructed above.

CHAPTER 3

# Guide to Menus and Popups

## 3.1 Overview

Note: In the descriptions to follow, '->' indicates the item is a pulldown or pull-right menu, '...' indicates a popup, neither indicates an action item or button.

Main panel items

- File-> read, write data, printer setup, print, status, about
- View-> set graph parameters, define symbols and legends, axes scaling
- Edit-> perform operations on points, sets, region and block data.
- Z - Zoom, in **xmgr**, a magnifying glass.
- Z - Expand world
- z - Shrink world
- AS - Autoscale
- AutoT - Default ticks
- AutoOn - Click on a point in the set to use for autoscaling
- PU - Push current world

- PO - Make top of stack the current world
- SD:0 - Current graph stack depth
- Left arrow - Scroll left
- Right arrow - Scroll right
- Up arrow - Scroll up
- Down arrow - Scroll down
- PZ - Push current world and enable zoom
- CY - Cycle through the world stack
- CW:0 - Current world of current graph while cycling through the world stack
- G0:X, Y = [locx, locy] - Current graph and location of pointer with repect to the scaling of the current graph
- Calc: - Infix calculator (**xvgr** only)

## 3.2 FILE

Read data from disk or pipes, read and write plot parameters, read block data, write one or more **ACE/gr** datasets, set the printer options, and print (hardcopy), access the command line interpreter, describe the status of data sets, graphs and regions, and display hardwired default values.

### 3.2.1 Read sets...

Read one or more data sets. The list item at the top of the popup displays the contents of the current directory. Select a file or directory by clicking on the item with the right mouse button. The selection is placed in the text item labeled 'File:' and checked to see if it is a directory or a file. Selecting a directory causes **ACE/gr** to change to the new directory and reload the file list with the contents of the new directory. Selecting a file does nothing until the 'Accept' button is pressed or <return> is entered.

Before 'Accept'ing the file, the items describing the file type, file source, and the graph in which to read the data need to be properly set.

The type of data file can be one of ten formats:

- A 2 column multi-data set file. Sets are separated by a line containing non-numeric characters, that are neither comment lines (lines with a '#' in column 1) nor parameter lines (lines beginning with an '@'). The sets generated from this dat file type are all of type XY.
- A multi column data file. X is assumed to be in the first column, and Y1, Y2, ..., up to Y30 in the remaining columns. The sets generated with this data file format are all XY.
- IHL format. A 3 column data file with the first two lines giving one line of alpha and the second an integer value with the number of points to follow.
- Binary format. - not defined as of this writing.

- X Y DX format. A 3 column data file consisting of X, Y and a quantity used to form an error bar parallel to the X-axis, i.e., the point will be plotted with an error bar at (X+-DX, Y).
- X Y DY format. A 3 column data file consisting of X, Y and a quantity used to form an error bar parallel to the Y-axis, i.e., the point will be plotted with an error bar at (X, Y+-DY).
- X Y DX1 DX2 format. A 4 column data file consisting of X, Y, and the errors in X. The error bar riser is drawn from (X+DX1,Y) to (X-DX2,Y). If both DX1 and DX2 are >0 then the error bar will bracket the datum. If DX2 is <0 and DX1 > 0 the error bars are drawn to the right of the datum. Likewise, if DX1 < 0 and DX2 > 0 the error bar is drawn to the left of the datum.

There are 4 cases for the error bars' position with respect to the datum:

1. DX1 > 0 and DX2 > 0 ... Error bars bracket the datum.
2. DX1 > 0 and DX2 < 0 ... Error bars are drawn to the right of the datum.
3. DX1 < 0 and DX2 > 0 ... Error bars are drawn to the left of the datum.
4. DX1 < 0 and DX2 < 0 ... Error bars bracket the datum, but reversed from case 1.

- X Y DY1 DY2 format. A 4 column data file consisting of X, Y, and the errors in Y. The error bar riser is drawn from (X, Y+DY1) to (X, Y-DY2). If both DY1 and DY2 are >0 then the error bar will bracket the datum. If DY2 is <0 and DY1 > 0 the error bars are drawn to the right of the datum. Likewise, if DY1 < 0 and DY2 > 0 the error bar is drawn to the left of the datum.

There are 4 cases for the error bars' position with respect to the datum:

1. DY1 > 0 and DY2 > 0 ... Error bars bracket the datum.
2. DY1 > 0 and DY2 < 0 ... Error bars are drawn above the datum.
3. DY1 < 0 and DY2 > 0 ... Error bars are drawn below the datum.
4. DY1 < 0 and DY2 < 0 ... Error bars bracket the datum, but reversed from case 1.

- X Y DX DY format. A 4 column data file consisting of X, Y, and the errors in X and Y. The error bar risers are drawn from (X+DX, Y) to (X-DX, Y) and (X, Y+DY) to (X, Y-DY).
- X Y Z format. A 3 column data file consisting of X, Y, Z. Sets with this type are drawn with the Z value in text at (X, Y).
- X Y R format. A 3 column data file consisting of X, Y, R. Sets with this type are drawn with a circle of radius R at (X, Y).
- X HI LO OPEN CLOSE format. A 5 column data file consisting of the high, low, open and close values for an observation at X. A data set of this type uses the line style, line width and line color for a normal set when the symbol is drawn.

'Read from:' sets the source of the data, either a disk file or a pipe. In the case of a pipe, the information provided by the file filter item is not used, as the 'file' in this case will be a command as typed at the UNIX prompt.

'Read to graph:' selects a particular graph to serve as the repository for the incoming data. The 'Current' graph is the graph that has the focus, and can be any of the 10 available graphs. If you aren't sure which graph is current, the locator item on the main panel displays the current graph number.

Note: Data is read into the next available set. All data are assumed to be delimited by tabs or blanks.

For the xy, xydx, xydy, xydxdx, xydydy, xydxdy, xyz, xyr data file formats, sets are separated by a line containing non-numeric characters, that are neither comment lines (lines with a '#' in column 1) nor parameter lines (lines beginning with an '@'). Data files can have imbedded comments by placing a '#' in the first column of the comment line. Plot parameters can be set by placing a '@' in the first column of the line followed immediately by the parameter name and setting. See the command line reference for a description of plot parameters and other commands that may be placed in a data file using the '@' character. Comment lines and parameter lines can occur at any place in the data file, and are not used to indicate data set separators. I recommend a single '&' on a line to use as a data set separator.

Press 'Accept' to read the data. If the read was successful, the graph of the data is drawn using the current plot scaling parameters. If the data lies outside the current plotting limits, the new set(s) will not be visible until the plotting limits are changed by autoscaling or manually through the View/Define world popup.

Open the File/Status popup to get information on the state of sets. If reading the file generates more than 10 errors you'll be prompted by an alert requesting whether or not you'd like to continue. These errors generally arise when an attempt is made to read a text file - **ACE/gr** relies on the interpretive capabilities of sscanf() to parse the data. **ACE/gr** will read data sets until there are no more sets available, after the maximum number of sets have been used, you'll need to free some sets using the [Kill] or [Kill all] items in the Edit/Set operations pullright.

Note: **ACE/gr** scans and replaces commas with blanks and the 'D' in Fortran double precision exponential formats with 'e'.

### 3.2.2 Read parameters...

Enter the parameter file name on the line labeled "File:", press "Accept" to read a saved state of adjustable parameters, legends, strings, etc, to the graph selected by the 'Read to graph' cycle. As the graph number is saved in the parameter file this isn't used at the present time.

### 3.2.3 Read block data

Select the data source, either Disk or Pipe and enter the block data file name in the text item labeled "File:" to read a data set in block data file format. The block data file for-

mat consists of columns of data. For example, the following is a block data file consisting of 4 columns:

1 2 3 4 5

6 7 8 9 10

11 12 13 14 15

After reading the file of block data, the popup will close if no errors were found, and the Edit/Block data popup will take its place. Use the functions in the Edit/Block data popup to create sets from the block data.

NOTE: All items are assumed delimited by spaces or tabs. There is presently a 30 column limit. Only one set of block data per session, i.e., reading another set of block data will obliterate the previous set of block data.

### 3.2.4 Write sets...

Write one or all sets to disk. Select the set to write using the cycle displaying the set numbers (there is an item for selecting all active sets). The item denoted “Format” is the C language format string that will be used to format the data for output - the default should work well for most cases. Enter the file name on the line labeled “File:” and press the button marked [Accept], to write the data. The item, ‘Imbed parameters,’ causes **ACE/gr** to write not only the data, but the parameters describing the graph or graphs as well. The toggle button labeled ‘Write binary data’ overrides all the previous selections and only works for the current graph - this selection is there for local reasons and is not likely to be useful for general purposes.

A complete dump of the contents of **ACE/gr** may be accomplished by selecting all sets, imbed parameters, and all active graphs. The file generated this way can then be read as a normal data file, and should recreate the original environment.

### 3.2.5 Write parameters...

Select the graph from which to save the parameters. This can be the current graph, any particular graph, or all active graphs. Enter the parameter file name on the line labeled “Write parameters to”, press “Accept” to write the current state of adjustable parameters, legends, strings, etc. to disk. A prompt will appear if the action would overwrite an existing file. allowing an oportunity to cancel the operation.

### 3.2.6 Save all...

Perform a ‘Write sets’ using all active sets, all active graphs, and imbed parameters. Use this to save the current session. If there are gaps in the set structure, the pack set toggle will move the sets down to insure contiguous active sets starting from set 0. Use Edit/Set operations/Pack sets to remove the gaps during a session to achieve the same effect.

### 3.2.7 Clear all...

Kill all sets, graphs, and annotative text, lines, and boxes. After performing this action, it will be necessary to activate a graph using View/Graphs/Activate graphs.

### 3.2.8 Print

Generate a hardcopy of the current plot on the device specified in the File/Printer setup popup.

### 3.2.9 Printer setup...

Set the hardcopy device parameters. Select the device, where to print (either the printer or a file), and the string to use as a spooling command for the print job, or file name if output is directed to disk. Click on the ‘Accept’ button to make the selections above current, or ‘Close’ to close the popup and cancel the operation. If the selection is accepted, the next time File/Print is selected, hardcopy output will be to the selected device. Click on the ‘Print’ button to accept the settings and print a hardcopy. If the printer selected is PostScript, there is a toggle button that allows for a bounding box to be computed for the graph. Use this to generate Encapsulated PostScript.

### 3.2.10 Command interpreter...

Command driven version of the interface to **ACE/gr**. Here, commands are typed at the ‘Command:’ text item and executed when <return> is entered. The command will be parsed and executed, and if no errors are found, the command line is placed in the history list. Items in the history list can be recalled by simply clicking on them with the left mouse button. Save a history list by clicking on the ‘Save...’ button popping up the ‘Save commands’ popup. Clicking on ‘Read...’ will open the ‘Read commands’ popup and allows previously stored commands to be loaded into the history list for future access. ‘Clear’ empties the history list. ‘Replay’ cycles through each item in the history list and executes each instruction in the list. See the command line reference for a description of the command line syntax.

### 3.2.11 Status...

Display useful information about the state of sets, grids,and graphs. As of 2.10, additional functions allowing many set operations have been added. The bottom panel has buttons providing functions as follows:

- Close

Close the status popup

- Update

Refresh the status area

- Write

Write the status to the monitor window (a multi-line text item that is used for displaying this item and the results of the regression routine).

- Page

Page through the list of items displayed

- Home

Go to the first page.

- End

Go to the last page.

- Display status of...

Select the objects to display, sets, graphs, or regions

Immediately above are buttons for performing operations on the sets shown in the status window.

- Kill - Click on the set index to kill the set.
- Deact - Click on the set index to de-activate the set.
- React - Click on the set index to re-activate the set.
- Copy - Click on the set index to copy from, then on the set index to copy to.
- Move - Click on the set index to move, then on the set index number for the destination.
- Auto - Click on the set index number to use for autoscaling the current graph.
- Reverse - Click on the set index number to reverse.
- Join - Click on two sets, the first set is appended to the second set.
- Pack - Move all sets so that the sets occupy contiguous locations starting from set 0.
- Cancel - Cancel any of the above operations.

### 3.2.12 Results

Display the results of the regression routine and the output of the 'Write' button in File/ Status in the monitor window.

### 3.2.13 About...

Display the value of certain hardcoded limits in **ACE/gr**. Also displays the author's email address, the version and patch level, and where this version of **ACE/gr** originated.

### 3.2.14 Exit

Click on the Exit button to terminate the session with **ACE/gr** (as of this writing, abrubtly).

## 3.3 VIEW

The selections in the View pulldown menu allow the modification of graph parameters. There are popups for setting the scaling of the axes, the number of tick marks to display, legends, writing annotative text, and drawing objects such as lines and boxes.

**ACE/gr** uses 3 coordinate systems in drawing a graph. These are the world, viewport, and device coordinate systems (following Foley and Van Dam). The world coordinates system is the one in which the data are defined and constitutes the user's coordinate system. The viewport coordinate system (in **ACE/gr**), is a rectangle defined by the points (0.0, 0.0) or the lower left corner of the device and (1.0, 1.0) or the upper right corner of the device. The pipeline for drawing objects on the screen or hardcopy device is a pair of linear transformations that carry points in world coordinates to the viewport and then on to the device coordinate system. The upshot is that an object located in world coordinates can be drawn in various parts of the screen or hardcopy device depending on the scaling while an object located in viewport coordinates remains in the same spot regardless of the world scaling. Strings, lines, boxes, and the graph legend can be located in either world or viewport coordinates.

### 3.3.1 Graphs

Activate, kill, copy, display, arrange, select the graph with the current focus, and set the focus policy.

#### 3.3.1.1 Activate

Make a graph eligible for receiving the graph focus and other operations.

Select the graph to activate and press 'Apply'.

#### 3.3.1.2 Copy

Copy a graph to another graph.

Select the graph to copy from and the graph to copy to and press 'Apply'.

Note: Copying a graph will automatically conceal one of the graphs as they will both have the same viewport settings. Some adjustments in the viewport settings of either or both graphs will be required for each graph to be displayed in a non-overlapping manner.

#### 3.3.1.3 Swap

Exchange two graphs.

Select the first graph then the second and press 'Apply' to exchange the contents of the two graphs. This can be used to adjust the order in which graphs are drawn, as graphs are drawn in numerical order starting with graph 0 and proceeding to graph 9.

#### 3.3.1.4 Kill

Make a graph inactive and free all storage associated with sets.

Select the graph to kill and press ‘Apply’.

#### 3.3.1.5 Focus

Set the graph focus, the focus policy and toggle the display of focus markers.

Select the graph to have the focus, the focus policy, and the toggle for the focus markers and press ‘Apply’.

Note: Most operations in **ACE/gr** act on the current graph. Not noticing which graph has the current focus can be a source of frustration when working with **ACE/gr** - if you plan to work extensively with a particular graph, it might be useful to set the focus policy to ‘Set’ to fix **ACE/gr**’s notion of the current graph.

#### 3.3.1.6 Show

Toggle the display of a one or more graphs.

Select which graph or graphs to show, then press ‘Apply’. The default is to show all active graphs.

This item can relieve some of the drudgery when working with multiple graphs, especially when some of the graphs contain large data sets.

#### 3.3.1.7 Set graph type

Set the current graph type to XY, log-linear, linear-log, log-log, bar or stacked bar.

Select the graph type, and press ‘Apply’.

Note: As of this writing, very little is done to ensure that the data is acceptable for log plots.

#### 3.3.1.8 Arrange graphs

Place several graphs in a non-overlapping manner.

Select the number of rows and the number of columns, the packing method, the vertical and horizontal spacing between graphs in viewport coordinates, the start of the first graph in viewport coordinates, and how wide and how tall each graph should be. Graphs are laid out in column major order starting from the lower left. So, given 3 columns and 2 rows the graphs will be laid out thusly:

1 3 5

0 2 4

Given 3 colums and 3 rows:

2 5 8

1 4 7

0 3 6

Note that graphs are numbered from 0.

The packing selection packs the graphs vertically, horizontally or both.Use this item when there are several graphs with the same X or Y axis scaling so graphs on the outside of the packing arrangement provide the tick and axis labelling for all graphs in that row or column. In the 3x3 example above,selecting packing 'both', graphs 3, 6 will have X-axis tick labels, and graphs 1, 2 will have Y-axis tick labels. Graphs 4, 5, 7, 8 will have neither X or Y axis tick labels, while graph 0 will have both.

### 3.3.2 Define world

Define the world coordinate system by filling in the items Xmin, Xmax, Ymin, and Ymax. The plot is drawn in world coordinates in a rectangle described by the two points (Xmin, Ymin), (Xmax, Ymax). The input is scanned for arithmetic expressions so setting Xmin = -PI and Xmax = PI is legal input (see the section on transformations for a description of the syntax and available functions). The tick spacing can also be set in this popup by filling in the items for the major and minor spacing for each axis. 'Update world/ticks' is used, when, occasionally, the state of the items used to define the world scaling get out of sync with what is actually used to draw the graph, clicking on this item synchronizes the internal values with the displayed values. Press the button marked "Accept" to inform **ACE/gr** of the changes.

Note: make sure Xmin<Xmax and Ymin<Ymax.

### 3.3.3 Define view

Define the viewport by filling in the items Xmin, Xmax, Ymin,Ymax. Viewport coordinates run from (0.0,0.0), the lower left corner of the screen or hardcopy device, to (1.0,1.0), the upper right corner of the screen or hardcopy device.

Press the button marked "Accept" to make the change to the new viewport.

Press the "Pick view" button to use the mouse to define the viewport. Take the mouse to the lower left corner of the desired viewport and press the left button (there is no need to hold the button down). A rubberband box will show you the current size of the viewport you may select. When you are satisfied with the view, press the left mouse button again to activate the new viewport.

Press the button marked "Close" to close the 'Define view' popup.

### 3.3.4 Autoscale

Use the Autoscale popup to set parameters associated with autoscaling. Select the axis to scale, or none. Selecting none allows the autoscale type and the number of ticks to create to be set but no autoscaling is done. Select the particular set to use or All. The Autoscale type sets the method of autoscaling, either Heckbert or fixed. Heckbert uses a method based on routines posted to the net by Paul Heckbert and generates nice looking tick spacing, but alters the scaling of the axes. The fixed option uses the minimum and maximum values in the set (or over all sets) to set the scale of the axes. Select the number of tick marks to use for defining the tickmark spacing (in the case of Heckbert scaling, these settings are advisory only). The item labeled 'Apply to:' allows the autoscaling of the current graph, or all active graphs.

### 3.3.5 Draw options

Set the amount of scrolling, linked scrolling (scrolling the current graph scrolls all graphs), automatic redraw, and whether the screen is erased before refreshing the screen.

### 3.3.6 Title/subtitle

The title and subtitle are strings that appear centered at the top of the plot. Fill in the items indicated and press the button marked "Accept". To change the font, color, or character size used for titles use the items from the popup denoted 'Props...'.

### 3.3.7 Ticks/tick labels

Set the spacing and type of major and minor tick marks, definition of axes labels, and toggle features associated with the drawing of the graph axes. The top item, 'Edit', selects the axis for which the feature selections will apply. There are three axes defined for each co-ordinate direction, the primary axis, the zero axis, and an alternate axis. The primary axis will be used the most, the others are, by default, turned off. The button to the right of the 'Edit' item labelled 'Props...' allows the application of an alternate map to the tick labels and the capability to offset the axis by a specified amount in viewport co-ordinates. The item 'Axis label' accepts the definition of the label to be used when drawing the axis. Various properties of the label may be set in the popup just below labeled 'Axis label props...'.

The two items 'Major tick spacing' and 'Minor tick spacing' sets the spacing of major and minor ticks. Tick labels are drawn based on the settingfor the major tick spacing. These items are duplicated in the 'Define world'popup described previously. Fill in the appropriate blanks and press "Apply" to refresh the display. The units are positive deltas starting with the world minimums. Input is run through the scanner so expressions are allowed. The are special symbols "dx" and "dy" for setting a prescribed number of tick marks. Dx is the distance along the X-axis and dy is the distance along the Y-axis, as defined in the current window of the world coordinate system. Typing at "Major tick spacing" "dx/10" will give 10 major tickmarks.

Tick labels, tick marks, the axis bar, may be toggled by the check boxes. Properties for each of these can be set by the ‘Props...’ popups associated with each item.

Properties for tick labels are:

- Font - which font to use when drawing the labels
- Color - which color to use
- Line width - width of line to use when drawing the label
- Char size - size of characters to use
- Format - Decimal, exponential, power, general plus several time and date formats. To use the time and date formats, the data is required to be in Julian Date format, the long version. Two auxillary programs, jul2greg and greg2jul, are available with the distribution to assist in the conversion from Julian to Gregorian date formats.
- Stagger - each tick label may be offset with respect to its neighbor, the levels of staggering ranging from 0 to 3, This is especially useful with some of the time and date formats.
- Precision - sets the number of places to display to the right of the decimal point
- Skip every - tick labels are drawn at every major tick mark, use the skip factor to circumvent this.
- Start labels at - Graph minimum, or specified. To start the labels at some other point than the graph minimum, fill in the text item to the immediate right with the value where tick labels are to begin. There is a similar item in the property sheet for tick marks that can be used to set where the major tick marks begin. between these two, there should be sufficient flexibility.
- Stop labels at - same as above only works at the other end of the axis.
- Layout - allows the tick labels to be drawn horizontally, vertically, or a specified angle (in degrees).
- Draw tick labels - which side of the graph to draw tick labels, either the normal side (the bottom of the graph in the case of the X-axis, or to the left of the graph in the case of the Y-axis), The opposite side, or both. This can be handy when using multiple axes, or overlaying graphs with differingscales.
- Sign - allows the numerical value of the label to be displayed as is, as its absolute value, or negated. This can be used to effectively reverse the sense of the axis (negate), or reflect the axis about zero (absolute value).
- Apply to - Override the selected axis from the Ticks/tick label popup, causing the settings to be selected for all axes in the current graph, the current axis for all graphs, or all axes in all graphs.

Press the button ‘Accept’ when everything is OK, or ‘Cancel’ to close the popup.

Properties for tick marks are:

- Tick mark direction - in, out, or both. Which direction the ticks point.
- Tick marks on - Which side of the graph to draw ticks. The normal side is the bottom of the graph in the case of the X-axis, or to the left of the graph in the case of the Y-axis, the default is to draw ticks on both sides.
- Major tick length - sets the length of major tick marks

- Minor tick length - sets the length of minor tick marks
- Major grid lines - a check box that when toggled on, grid lines will be drawn at the settings for major tick marks.
- Minor grid lines - a check box that when toggled on, grid lines will be drawn at the settings for minor tick marks.
- Color, line width, and line style items set the characteristics of the lines use to draw ticks and grid lines.
- Apply to - allows the axis select in the Ticks/tick label popup to be overidden, causing the settings to be selected for all axes in the current graph, the current axis for all graphs, or all axes in all graphs.

Press the button ‘Accept’ when everything is OK, or ‘Cancel’ to close the popup.

Properties for the axis bar are:

- Color, line width, and line style items set the characteristics of the lines use to draw ticks and grid lines.

Press the button ‘Accept’ when everything is OK, or ‘Cancel’ to close the popup.

IMPORTANT NOTE FOR LOG PLOTS:

When log scaling is selected for a particular axis or axes, the meaning of major and minor ticks changes slightly. Major ticks should be set to integer values. Use minor ticks to set the number of interdecade tickmarks, these should be set to integer values ranging from 0 (no interdecade tickmarks) to5. A value of 1 will draw each interdecade position, 2 will skip every other position, etc.

The following applies to log scaled plots:

1. The major tick spacing should be in integer values.
2. The minor tick spacing works as follows:

- a value of 0 - draw a no minor ticks
- a value of 1 - draw a minor tick at each decade
- a value of 2 - draw a minor tick at decades 2, 4, 6, 8
- a value of 3 - draw a minor tick at decades 3, 6, 9
- a value of 4 - draw a minor tick at decades 4, 8
- a value of 5 - draw a minor tick at decade 5

3. Minor ticks are not drawn at integer powers of 10.
4. There are no provisions for drawing labels at other than integer powers of 10. I’m considering using fractional values of the major tick spacing to do this. Say a major tick spacing of 1.1 would mean label every magnitude and every decade. 1.5 would mean label every magnitude and decade 5 only.
5. Scrolling in any direction or in and out is not implemented.
6. Use the tick label skip parameter to draw tick labels when the scale covers many magnitudes but minor ticks are desired at every magnitude.

Please avoid throwing negative numbers or 0 at log graphs, there is little checking done for these values.

### 3.3.8 Frame

The frame is the box drawn around the clipping region of the graph, and the regioninside. Items in the frame popup are:

- Frame - turn the frame on or off
- Style - either ‘closed’ or ‘open.
- Color - set the color of the line drawn around the graph
- Line width - set the width of the line around the graph
- Line style - set the line style of the line
- Fill graph background - toggle the filling of the frame.
- Background color - the color with which to fill the frame

Press ‘Accept’ to register the settings, ‘Close’ to close the popup.

### 3.3.9 Symbols

The items in the symbols popup allow the selection of symbols, lines, and fill styleto be used when drawing a set.

Items in the symbols popup are:

- Select set - the set to which the following items will apply

Symbol:

- Symbol - select the symbol to use, or none for no symbol
- Fill - select the fill option, none, filled, or opaque. The opaque option allows the symbol to be filled with the background color (white as of this writing) overwriting the line used to connect each point.
- Size - set the size of the symbol
- Char - select a character to use instead of a symbol
- Skip - select the number of points to skip when drawing symbols
- Legend - A string defining the legend to use for this set
- Apply to - either this set or all sets

Line:

- Style - select the line style for the lines connecting the points in the set
- Width - select the line width for the lines
- Color - the color to use when drawing lines and symbols

Fill:

- Fill - select the type of fill, none, as polygon, to y=0.0, x=0.0, x=graph Xmin, x=graph Xmax, y=graph Ymin, y=graph Ymax.
- Fill using - either Color or Pattern.
- Color - select the color to use, or none.
- Pattern - select the pattern to use or none

There are three additional buttons in this popup that can be used to set the colors, symbols, and line widths in a regular fashion, avoiding the need to specify each.

Symbols:

1. No symbol
2. Dot
3. Circle
4. Square
5. Diamond
6. Triangle up
7. Triangle left
8. Triangle down
9. Triangle right
10. Plus
11. X
12. Star
13. Impulse at X
14. Impulse at Y
15. Vertical line at X
16. Horizontal line at Y
17. Histogram X
18. Histogram Y
19. Stair step X
20. Stair step Y
21. Bar X
22. Bar Y
23. Range
24. Location
25. Set #
26. Set #, location
27. *Bar and whisker (not active as of this writing)
28. Segments
29. Character
30. Tag first point

**31.** Tag last point
**32.** Tag center point
**33.** *String (not active)
**34.** Hi low X
**35.** Hi low Y
**36.** Open/close X
**37.** Open/close Y

NOTE: symbols 16-37 do not appear in the legends

- Error bars

Error bars are created by reading in the data as a set with error bars (see Files above). Properties of error bars are set by opening the 'Error bar props' popup.

### 3.3.10 Error bars

- Size

Set the length of the error bar.

- Line width

Set the line width of the error bar.

- Line style

Set the line style of the error bar.

- Riser

Toggle the display of the line connecting the error bar with the data point.

- Riser line width

Set the width of the riser.

- Riser line style

Set the riser line style.

- Display

Toggle the display of the error bar.

### 3.3.11 Legends

Legend items:

- Legend on/off - toggle the display of the graph legend

- Legend location type - Select the coordinate system to use when interpreting the legend x, and legend y items described below. Either the world or viewport coordinates. Legends placed in viewport coordinates make the legends stay put as the plot scale changes. Legends placed in world co-ordiantes will float as the graph scaling is changed.
- Font - specify the font for the legend labels
- Char size - select the size of text to use for the labels.
- Legend gap - specify the vertical gap between legend entries
- Legend length - specify how long the line representing the set should be.
- Legend X, Lengend Y - Legend location, either view or world coordinates, depending on the setting for location type above. The button described below, 'Place', allows the legend location to be set by clicking on the location for the legend.
- Frame - toggle a box drawn around the legend. The three items below set the line color, line width, and line style of the box.
- Fill frame - toggle the fill of the box around the legend. The frame fill will obliterate anything beneath, grid lines, sets, etc. Set the type of fill either color or pattern and the corresponding color or pattern.

The items Legend gap and Legend length refer to the gap between legend items and the length of the legend in units of characters (arbitrary).

Pressing the left mouse button on "Place" is a convenient method of placing the legend on the canvas. After pressing this button, move the mouse pointer to the desired location and press again with the left button. The location will be in world or viewport coordinates as selected above.

Press "Load comments" to make the set comment (generally the file name from which the set originated) the legend label. This affects all active sets in the current graph.

Press the button marked "Accept" to register the legend settings.

Press the button marked "Close" to close the popup.

Press "Edit" to edit all of the legend labels for the current graph.

### 3.3.12 Strings & Things

The numbers of strings, lines, and boxes are fixed, open File/About to see what the limits are for your version of **ACE/gr**.

To define a text string to be drawn on the canvas:

- Select font, pen, justification, rotation (in integer degrees [0,360]), size. Press the button marked "Text" to activate the text writing routines. Move the mouse to the canvas and press the left mouse button at the desired location and type away. Press <return> to advance to the next line - the right mouse button to stop.

The text cursor does not behave properly for rotated strings. To redefine string parameters you'll need to press the right button to leave text mode - adjust the parameters and press "Text" again to inform **ACE/gr** of the changes and resume writing. The cycle "Position in: World | viewport coordinates" allows the string position to be defined in the world coordinate system or viewport coordinates. The difference is that in World coordinates the string position on the display will change as the plot is re-scaled. Placing the string in viewport coordinates will force the string to remain in the same place as the plot scale is changed.

NOTE: It is necessary to press <return> to register the string.

To define a line or box to be drawn on the canvas:

The cycle "Position in: World | viewport coordinates" allows the line (box) position to be defined in the world coordinate system or viewport coordinates. The difference is that in World coordinates the line (box) position on the display will change as the plot is re-scaled. Placing the line (box) in viewport coordinates will force the line (box) to remain in the same place as the plot scale is changed.

Lines:

- Set the properties of the line using View/Strings & things/Line props if needed.
- Click on Line.
- Click at the start of the line, and again at the end of the line to define the line.

Boxes:

- Set the properties of the line using View/Strings & things/Line props if needed.
- Click on Box.
- Click on one corner of the box, then click again on the opposite corener to define the box

Move:

- To move a line, box or string:
- Press the button marked "Move", this should change the cursor to the "move" cursor.
- Press the left mouse button near the object to be moved.
- Move the cursor to the new location and press the left mouse button.

To delete a line, box, or string:

- Press the button marked "Delete", this should change the cursor to a bullseye cursor.
- Press the left mouse button near the object to be deleted.

NOTE:

The right mouse button cancels all operations (true throughout **ACE/gr** with respect to mouse operations).

Lines and boxes drawn in world coordinates scale with the plot, to draw fixed length boxes and lines, use viewport coordinates.

### 3.3.13 Flip X-Y

Release the right mouse button on “Flip X-Y” to swap the positions of X and Y (all sets, tickmarks, and labels will be swapped).

### 3.3.14 Invert X

Reverse the sense of the X-axis making increasing X toward the left. What this function does is to negate all X data values and exchange and negate the world settings for the X-axis. There could be problems with underflow if the original graph world has a minimum X of 0.0. This will show up as a -0.0 as the tick label for the inverted X-axis maximum. If this occurs, use View/Ticks/Tick labels to set the sign of the tick labels to Absolute.

### 3.3.15 Invert Y

Reverse the sense of the Y-axis making increasing Y toward the bottom of the graph. What this function does is to negate all Y data values and exchange and negate the world settings for the Y-axis. There could be problems with underflow if the original graph world has a minimumY of 0.0. This will show up as a -0.0 as the tick label for the inverted Y-axis maximum. If this occurs, use View/Ticks/Tick labels to set the sign of the tick labels to Absolute.

### 3.3.16 Locator

The locator is the message item on the main panel that displays the location of the pointer. There are several formats and options for this display. By selecting a fixed point (see below) the location is presented releative toa point on the canvas.

#### 3.3.16.1 Set fixed point

Select this item to mark a spot on the canvas to use as a reference point for the locator display types [DX, DY], [DIST], [R, THETA]. Click at a point on the canvas to set the fixed point, the point will be marked by a circle with a cross.

#### 3.3.16.2 Clear fixed point

Select this item to remove the fixed point previously selected. The fixed point becomes (0.0, 0.0).

#### 3.3.16.3 Props

Set the properties of the locator.

Toggle the continuous display of the pointer location by selecting Locator on or off. If the locator is off, the pointer location may be displayed by clicking on the canvas at the point of interest.

The Locator display type selects the transformation used to display the pointer location. [X, Y] uses the transformation provided by the scaling in the current graph. [DX, DY] gives the location of the point relative to the locator fixed points (set above or in this popup). [DISTANCE] gives the pointer location in terms of the Euclidian distance relative to the fixed point or 0.0 if no fixed point is defined. [R, Theta] gives the locator position in terms of polar coordinates. Use [VX, VY] to display the pointer location in viewport coordinates, and [SX, SY] to give screen coordinates.

The Format item selects the format of the display of the location of the pointer. Formats are decimal, exponential, power, general, several time and date formats. and several longitude and latitude formats. To use the time and date formats, the data is required to be in Julian Date format, the long version. Two auxillary programs, jul2greg and greg2jul, are available with the distribution to assist in the conversion from Julian to Gregorian date formats.

Set the number of decimal places to display using the Precision item.

Set the value of the fixed point using the two text items provided for X and Y, be sure to set the fixed point toggle to ON.

Press Accept to register the changes, and Close to close this popup.

## 3.4 EDIT

### 3.4.1 Transformations

NOTE:

Some of these operations generate new sets, if you run out of sets use Edit/Set operations (described below) to kill unneeded sets. Also, the scaling of the world coordinate system may be inappropriate for the results of many of these operations. Use the Status popup to determine the appropriate scaling factors to use in “Define world” (above), or use “Autoscale...” (in View/Autoscale above). Most functions operate on active sets only.

#### 3.4.1.1 Evaluate expressions

Evaluates a formula defined in infix fashion.

Select the set on which the formula will operate, (set must be active, use File/Status to find the current state of sets). If you desire the result be loaded to a new set rather than overwriting the set used for computations, use the panel cycle denoted ‘Result to’ to

inform **ACE/gr** you'd like the result placed in a new set (if there is one). A new set is created only if a single set is selected from step 1. If 'All sets' are selected then the results will overwrite all active sets.

Enter the formula, the syntax is:

(x,y,a,b,c,d)=<expression>

where (x,y,a,b,c,d) are defined as the x and y of the currently selected set and a,b,c,d are scratch arrays that can be used to perform operations between sets. Case is ignored, so X=COS(X) is correct.

Press the button "Apply" with the left mouse button when you are satisfied with everything or press <return> with the caret in the text item marked 'Formula'.

Functions:

- a == reference to scratch array
- abs(x) == absolute value
- acos(x) == arccosine
- asin(x) == arcsine
- atan(x) == arctangent
- atan2(y,x) == Fortran ATAN2
- b == reference to scratch array
- c == reference to scratch array
- ceil(x) == greatest integer function
- cos(x) == cosine
- d == reference to scratch array
- deg == 180.0/PI
- dx == span of world coordinate system in x
- dy == span of world coordinate system in y
- erf(x) == error function
- erfc(x) == complement of error function
- exp(x) == e^x
- floor(x) == least integer function
- index == the index of the current point in the selected set
- int(x) == truncation
- invn(p) == inverse of standard normal (p in [0,1])
- invt(p,id) == inverse of Student's t with id degrees of freedom
- irand(n) == random integer less than n
- lgamma(x) == log of gamma function
- ln(x) == natural log (should be log)
- log(x) == log base 10 (should be log10)
- max(x,y) == returns greater of x and y

- min(x,y) == returns lesser of x and y
- mod(x,y) == mod function (also x % y)
- norm(x) == gaussian density function
- normp(x) == cumulative gaussian density (-inf,x)
- pi == constant PI
- rad == PI/180.0
- rand == pseudo random number distributed U(0.0,1.0)
- rnorm(xbar,s) == psuedo random number distributed N(xbar,s)
- sin(x) == sine function
- sqr(x) == x^2
- sqrt(x) == x^0.5
- tan(x) == tangent function
- x == currently selected set X
- y == currently selected set Y

Note: See pars.yacc for the yacc grammer.

Examples:

- y=-y
- y=x*cos(2*x*PI/100)+sqr(x)
- x=(index>10)*(x-5)+(index<=10)*x

If the index of the current point is greater than 10 then x=x-5 else x=x.

- a=y

Store y of the current set into scratch array “a” you may now select another set and perform the operation y=somefunctionof(a)

#### 3.4.1.2 Load values

Load a sequence to (x,y) or (a,b,c,d).

#### 3.4.1.3 Histogram

Compute a frequency histogram.

- Select the set.
- Enter the width of a bin (all bin widths are the same).
- Enter the minimum and maximum values of the portion of the data you wish histoed. Xmin and Xmax refer to the RANGE of the set, not the domain.

Press ‘Apply’ to compute the histogram.

#### 3.4.1.4 Fourier

Compute the Discrete Fourier transform.

- Select the set
- Select the type of data window, the default is the rectangular window in which case the data is tranformed unmodified.

The data windows are defined as follows:

1. None

Use the default rectangular window

2. Triangular

1.0 - |(i-0.5*(N-1))/(0.5*(N-1))|

3. Hanning

0.5 * [1-cos(2*pi*i/(N-1))]

4. Welch

1-((i-0.5*(N-1))/(0.5*(N+1)))^2

5. Hamming

0.54-0.46*cos(2*pi*i/(N-1))

6. Blackman

0.42-0.5*cos(2*pi*i/(N-1))+0.08*cos(4*pi*i/(N-1))

7. Parzen

1.0 - |(i-0.5*(N-1))/(0.5*(N+1))|

- Select the form of the output, magnitude (spectrum), phase, or the coefficients.

The spectrum is computed by sqrt(x*x + y*y) where x, y arethe coefficients computed by the DFT or FFT. Only N/2 values (representing frequencies 0 to PI) are loaded to the resulting set.

- If the magnitude or phase is selected, then the next item, 'X = ', determines what values should be loaded to X.

The index runs from 0 to n/2, the frequency is the cyclical ith fourier frequency, the period is the reciprocal of the frequency with the period of the 0th fourier frequency plotted at T+delt, where T is the total length of the data and delt is the sampling interval.

- Select transform or inverse transform.
- Select real or complex data.

If real is selected, then the data to be transformed is assumed to be in Y, X is assumed equally spaced and is ignored. If complex is selected then the real part is assumed to be in X and the imaginary part in Y.

- Press DFT (for small data sets whose length is not a power of 2) - or
- FFT (for data sets whose length is a power of 2).
- Press 'Window only' to generated a windowed version of the data in a new set.

NOTE:

Small is < 1000 points. The DFT is O(N**2).

#### 3.4.1.5 Running

Compute a running average, median, minimum, maximum, or standard deviation.

- Select the method.
- Select the set.
- Set the length of the running method in the text item marked "Length", it must be less than the set length.

Press the button marked "Apply".

#### 3.4.1.6 Regression

Perform linear or polynomial regression.

- Select the set.
- Select the degree of fit.
- Select fitted curve or residuals to load.
- Press the button marked "Regress".

A set is loaded (if there is one) with the resulting curve and a summary of the statistical results are written to standard output.

For polynomial fits of degree > 1, the coefficient of the constant term is denoted by A[0], the first dgree term by A[1], etc.

#### 3.4.1.7 Non-linear curve fitting (prototype, use with extreme caution)

Fit a non-linear function with up to 10 parameters.

The underlying routine is LMDIF1 from Minpack with the original Fortran code converted to C using f2c. For details, please refer to the module lmdif1.c in the **ACE/gr** sources or LMDIF1.F from Minpack (available through Netlib). This routine handles unconstrained parameters only (there are techniques to convert a constrained problem to an unconstrained problem).

Note: This function is new and has an interface that will undoubtably change.

- Select the set
- Select fitted curve or residuals to load.
- Select the graph to use for the result.
- Enter the function to fit. The function must be defined as y= something. X (x) must be the independent variable and the parameters must be referred to as A0, A1, ..., A9, with no gaps in the parameters, i.e., using A0, A1, is OK, using A0, A2 (skipping A1) is not.

Example:

Correct,

y=1/(1+exp(-(a0-x)/a1))

Incorrect,

y=1/(1+exp(-(a0-x)/a2))

as a1 is skipped.

- Set the number of parameters to fit, this must correspond to the parameters used in the function. In the example above, the number of parameters is 2.
- Enter the tolerance to use, 1e-7 is the default.
- For each parameter to fit, enter an initial guess, the default is zero.
- Press the Apply button to launch the run.

If the routine is successful, the column of text items labeled “computed values” will contain the fitted parameters. A message item at the bottom of the column will give a diagnostic message regarding the run.

The messages are:

1. Improper input parameters.
2. Relative error in the sum of squares is at most tol.
3. Relative error between An() and the solution is at most tol.
4. Relative error in the sum of squares and A and the solution is at most tol.
5. Y is orthogonal to the columns of the Jacobian to machine precision.
6. Number of calls to fcn has reached or exceeded 200*(n+1).
7. Tol is too small. No further reduction in the sum of squares is possible.
8. Tol is too small. No further improvement in the approximate solution x is possible.

These messages are edited versions of the error messages given in LMDIF1.

#### 3.4.1.8 Differentiation

Numerical differentiation.

- Select the set.
- Select the method - one of forward, backward, or centered difference. Assumes unevenly spaced data, increasing in X.
- Press the button marked “Differentiate”.

A set is loaded (if there is one) with the resulting curve.

#### 3.4.1.9 Integration

Numerical integration.

- Select the set.
- Select the form of the results, the item marked “cumulative sum” will construct a set composed of the current value of the integral at a given X. Sum only reports just the value on the next line. Assumes unevenly spaced data, increasing in X.
- Press the button marked “Integrate”.

A set is loaded (if there is one) with the resulting curve if “cumulative sum” is chosen.

#### 3.4.1.10 X-corr

Cross/auto-correlation

- Select both sets (use the same set if autocorrelation is desired).
- Select the lag, N/3 is a reasonable value (your mileage may vary).
- Select bias - generally this will not make any difference for large data sets with lags << the length of the set, I was just curious. The difference is division by N (biased) or N-lag (unbiased).
- A set is loaded (if there is one) with the resulting curve.

#### 3.4.1.11 Spline

Compute a spline fit to a set

- Select the set.
- Select the starting value of X for the fitted curve.
- Select the ending value of X.
- Select the number of samples. The spline curve will be evaluated at X+i*(MaxX - MinX)/Nsteps for each i in (0, Nsteps-1).
- Press “Spline”
- A set is loaded (if there is one) with the resulting curve.

Notes: The code to compute the spline is a literal translation of the code in FMM.

#### 3.4.1.12 Sample

Sample a set pointwise or logically

- Select the set.
- Select the type of sample, either Start/step or Logical expression.

If Start/step is selected then enter the starting index to begin the sample.

- Select the number of points to skip between samples in Step.

If Logical expression is selected, enter the expression in the text item denoted “Expr:”. Values of the resulting evaluation of theexpression not equal to zero are interpreted as TRUE, and the point is accepted. Any expression evaluating to zero will result in the point being ignored.

#### 3.4.1.13 Digital filter

Apply a digital filter to a set.

- Select the set to be filtered
- select the set with the filter weights.

#### 3.4.1.14 Linear convolution

Perform convolution of 2 sets.

#### 3.4.1.15 Load and evaluate

Evaluate parametric functions.

- Enter the functions to be used to define X and Y.
- Select the independant variable (x,y,a,b,c,d).
- Enter the start, stop and the number of points items.
- Press the button “Apply’ to evaluate the functions and load the result to a new set.

### 3.4.2 Set operations

Set operations allow sets to be created, destroyed, written to disk, sorted.

#### 3.4.2.1 Pick operations

Pick operations allow the pointer to be used to perform a limited number of set operations. All operate by selecting the set nearest the pointer when the left mouse button is pressed. Copy and move are capable of working between graphs, with the restriction that the set to copy or move must be in the current graph.

Note: There is no undo function in **ACE/gr**. Make sure the set you wish to select is in the current graph.

- Kill nearest set

Remove the set nearest the pointer when the left mouse button is pressed. Presently, parameter settings are not preserved.

- Copy nearest set

Copy a set in the current graph to the next available set in the current graph or other active graph. Click near the set in the current graph to copy, then move the pointer to the destination graph and click again.

- Move nearest set

Copy a set in the current graph to the next available set in the current graph or other active graph. Click near the set in the current graph to move, then move the pointer to the destination graph and click again.

- Reverse nearest set

Reverse the order of points in a set.

- De-activate nearest set

Mark the nearest set as inactive. Data is preserved. Presently, there is no indication of exactly which set is deactivated, so how to get the set number to reactivate the selected set is a problem.

- Join nearest set

Join two sets together, the first set is appened to the second set.

- Delete range in nearest set

Click near a point in the set to begin the deletion, then again at the end of the range of points to delete.

#### 3.4.2.2 Activate

Make a set active and able to participate in operations.

- Select the set to activate.
- Set the length of the set.
- Select the type of set.
- Press the button marked “Apply”.

#### 3.4.2.3 De-activate

Make a set inactive and unable to participate in operations. The data associated with a deactivated set are still available and all plot parameters associated with the set are unchanged. Use the Re-activate set item below to make the set known to **ACE/gr** again. This item is include for those situations where a set is to be ignored temporarily, but needed later in the session.

- Select the set to Deactivate.
- Press the button marked “Apply”.

#### 3.4.2.4 Re-activate

Undo the effect of Deactivate.

- Select the set to Reactivate, it is a no-op to Reactivate an unused set.
- Press the button marked “Apply”.

#### 3.4.2.5 Set length

Set the length of a set.

- Select the set.
- Fill in the item marked Length.
- Press the button marked “Apply”.

#### 3.4.2.6 Change Set type...

Set the type of a set.

- Select the set
- Select the type to set
- Press the button marked “Accept”

#### 3.4.2.7 Copy

Copy one set to another in a possibly different graph.

- Select the set to copy from.
- Select the set to copy to.
- Select the graph to receive the copy.
- Press the button marked “Apply”.

#### 3.4.2.8 Move

Move one set to another in a possibly different graph.

- Select the set to move from.
- Select the set to move to.
- Select the graph to receive the set.
- Press the button marked “Apply”

#### 3.4.2.9 Swap

Exchange one set with another.

- Select the sets and the graphs these sets reside.

- Click on “Apply.”

The contents of the sets plus all parameters associated with the display of the sets are exchanged.

#### 3.4.2.10 Drop points

Drop points from a set.

- Select the set.
- Fill in the items “Start drop” and “End drop”.
- Press the button marked “Apply.”

NOTE: Elements in a set are numbered from 1 to N, where N is the number of elements in the set (as opposed to sets which are numbered from 0).

#### 3.4.2.11 Join

Merge 2 sets together.

- Select the set that will be appended.
- Select the set accepting the previous set.
- Press the button marked “Apply” to append the first set to the second.

#### 3.4.2.12 Split

Divide a set into other sets.

- Select the set to split.
- Enter the length of the resulting sets (say N).
- Press the button marked “Apply” to divide the selected set into (length of selected set) / N sets with the remainder to the last set.
- The first set will be the set selected to split and will contain the first N points.

#### 3.4.2.13 Kill

Eliminate a set.

- Select the set to kill or All to kill every active set.
- To save the parameter settings, toggle ‘Preserve parameters.’
- Press the button “Apply”.

NOTE:

There is no ‘undo’ for this operation.

#### 3.4.2.14 Kill all

Kill all active sets.

- This is an action item, you'll be asked if it is OK to kill all active sets.

#### 3.4.2.15 Sort

sort x or y of a set.

- Select set to sort.
- Select which component (x or y) as a key.
- Select the order of the sort.
- Press the button marked "Apply" to sort the set in place.

#### 3.4.2.16 Write set(s)

Write a set(s) to disk.

- Select the set to write or "All" for all sets.
- Fill in the format to use to write, syntax is C, default is "%lg %lg".
- Fill in the item marked "Write to file" with the filename to write.
- Select Imbed parameters to have parameter settings included in the file.
- Select Pack sets to have all sets fill contiguous locations starting from set position 0.
- Press the button marked "Apply".

Note: A complete dump of **ACE/gr** may be accomplished by selecting All sets, All active graphs and toggling Imbed parameters. The generated file may be read as a normal data file and will contain the necessary information to completely reconstruct all graphs in the current session.

#### 3.4.2.17 Reverse order

Exchange points in a set.

- Select set to reverse.
- Press the button marked "Apply" to reverse the order of a set.

#### 3.4.2.18 Coalesce sets

Merge active sets to an inactive set.

- Select set to receive points from all active sets. This selected setshould not be an active set.
- Press the button marked "Apply" to merge all sets to the selected set.

#### 3.4.2.19 Pack sets

Sometimes during a session in the course of creating and killing sets, gaps will form in the set structure of a graph. Pack sets moves all sets toward the lower numbers, giving a set structure with active sets filling contiguous positions starting from 0. Pack sets operates on the current graph only.

Note that deactivated sets are treated as inactive sets (although this may change in future versions).

### 3.4.3 Region operations

Note: Region operations operate only on sets of type XY. This will change in a future release.

#### 3.4.3.1 Define region

Define a region of interest on a graph or graphs.

- Select the region, there are 5.
- Select the type of region:

Inside polygon - points inside a closed, non-intersecting polygon are considered inside the region

Outside polygon - points outside a closed, non-intersecting polygon are considered inside the region

Above, below, to the left, and to the right - regions defining a half-plane, points inside the half-plane are considered in the region.

- Select the type of linkage, either a particular graph or all graphs, the default is the current graph.

Press ‘Apply’ to make the pointer ready for the region definition.

In the polygonal region type, define the region by successive clicks with the left mouse button, use the right mouse button to register the polygon. In the line case, define the line by clicking on the beginning point and end point of the line.

#### 3.4.3.2 Evaluate in region

Evaluate an expression applied to points within a region.

- Select the region to use.
- Enter the expression, regions are referred to by Rn.x or Rn.y where ‘n’ is the number of the region to use. Functions available are the same as those in ‘Evaulate expressions’ described above.

Examples:

R0.X = R0.X - 1

#### 3.4.3.3 Clear region

Remove a defined region

- Select the region to remove and press ‘Apply’.

#### 3.4.3.4 Extract points

Extract points from a region to a set

- Select the region to use.
- Select the set to receive the points.
- Select the graph to put the set.
- Click on ‘Apply’ to combine all points within the specified region to the specified set and graph.

#### 3.4.3.5 Delete points

Delete points in a region

- Select the region to use.
- Click on ‘Apply’ to delete all points within the specified region.

#### 3.4.3.6 Area/perimeter

Compute the area and perimeter of a region

This item doesn’t belong here as it does not use the region structures. Click on ‘Area’ or ‘Perimeter’ and use the mouse to define the region. The area or perimeter will be displayed in the text items ‘Area = ‘ or ‘Perimeter = ‘ as the case may be. Use the right mouse button to close the region.

- Compute the area of a polygon.

Press “Area” with the left mouse button to put **ACE/gr** into area computing mode. Mark the bounds of the polygon with the left mouse button (the area is zero until three points have been selected). A rubber band line will be drawn to mark the location of the polygon. There is no need to close the polygon as the algorithm assumes closure. Press the right mouse button to exit area mode. There is presentlya 100 vertex limit on the size of the polygon.

- Compute the length of connected line segments.

Press “Perimeter” with the left mouse button to put **ACE/gr** into distance computing mode. **ACE/gr** will total the successive lengths of line segmentscreated by pressing the left mouse button at each point.Press the right mouse button to exit “Perimeter” mode. There is presently a100 node limit.

### 3.4.4 Edit sets

The ‘Edit sets’ popup is an embryonic spreadsheet allowing the direct editing of points in a set. Use the ‘Edit set:’ item to select the set. Page up/down are buttons to page through the data. Home goes to the beginning and End goes to the end of the data set. To make a change to a point, place the cursor on the item to change, make the change and press <return> to register the change. This change is done on a copy of the set, so when the editing session is completed, press ‘Accept’ to bring the selected set up to date. The

<return> is essentially as is the ‘Accept’. Use the ‘Precision’ and ‘Format’ items to set the representation of the points in the text items.

NOTE: This is not available in **xmgr**.

### 3.4.5 Point operations

- Find - Find the set and point nearest the pointer.
- Tracker - Follow a set.
- Delete - Delete the point nearest the pointer.
- Add... - Add a point to the nearest set.
- Move - Move a point from one location to another.
- Move X - move the point in the X direction only.
- Move Y - move in the Y direction only.
- Goto... - Go to a given point on the drawing area.

#### 3.4.5.1 Find point

Report on a point in a set

- Press “Find points” with the left mouse button to activate.
- Position the pointer close to the data point to identify.
- Press the left mouse button.

The set, the location in the set, and (X, Y) for the datum nearest to the pointer will be displayed in a popup.

Press the right mouse button to shut off the “Find point” feature.

#### 3.4.5.2 Tracker

Track points in a set

Visit each point of a set in a sequential manner. Click nearest the set to track, and use the left mouse button to go forward through the set and the middle mouse button to go backwards through the set. The right mouse button exits tracking.

#### 3.4.5.3 Delete

- Press “Delete point” with the left mouse button to activate.
- Position the pointer close to the data point to delete.
- Press the left mouse button to delete the nearest point.

#### 3.4.5.4 Add

Add points to a set.

Select the set and press “Add points” to append points to the selected set by clicking on the location where the point is desired. This can be used to digitize (crudely) data from pages pasted on the screen.

#### 3.4.5.5 Move, Move X only, Move Y only

- Press “Move” with the left mouse button to activate.
- Position the pointer close to the data point to move.
- Press the left mouse button.
- Move the pointer to the new location and press the left mouse button agin to register the point’s new location.

#### 3.4.5.6 Goto

Enter the X, Y of the point to go to, and press ‘Goto point’ to have the pointer warp to that position on the drawing area.

### 3.4.6 Block data

Create sets from block data. Select the type of set and the columns to use for each vector of the new set. Press ‘Accept’ to create the set. Note that reading block data does not affect the scaling of the graph so when a set is created using this popup, the set may not appear within the graph’s scaling limits - use AS (Autoscale) to set the scale of the graph if needed (or use View/Define world).

## 3.5 Main panel items

### 3.5.1 Draw

Click on Draw to refresh the canvas with the current set of draw parameters and active sets. If View/Draw options/Auto redraw is False (toggle button is out), clicking on Draw will refresh the drawing area.

### 3.5.2 Locator

The item marked G0:[X,Y] on the top panel displays the number of the graph with the current focus and the location in world coordinates of the mouse pointer. Properties of the locator are set in View/Locator/Props. The location can be made relative to a fixed point by using View/Locator/Set fixed point.

### 3.5.3 Calc

**Note: (xvgr only)**

Evaluates infix expressions. There are 6 variables, x,y,a,b,c,d that can be used to store previous values. See the section on transformations for available functions. This item uses the same interpreter as the command line interpreter item in Files, but without a history list.

### 3.5.4 Scrolling and scaling buttons

Most of these are self-explanatory save the ones pertaing to the world stack. Each graph is allowed up to 20 different world and tick settings all saved on what is called the world stack. The world stack can be used to preserve the current axes scaling while panning and zooming about the graph.

Click on PU (push the current axes scaling on to the top of the world stack) to save the current scaling - if a zoom is used or one of the other commands that change the axes scaling, PO (pop the world stack) can be used to return to the saved scales. The command PZ is added as a convenience, clicking on PZ pushes the current scaling and prepares to accept a rectangular region for a zoom.

Assuming there is more than 2 items on the world stack, the CY button can be used to cycle through each of the settings stored on the world stack. The SD (stack depth) label item gives the depth of the current graph's world stack.

Summary:

- Z - Zoom (in **xmgr**, a magnifying glass).
- Z - Expand world
- z - Shrink world
- AS - Autoscale
- AutoT - Default ticks
- AutoOn - click on a point in the set to use to autoscale the current graph.
- PU - Push current world
- PO - Make top of stack the current world
- SD:0 - Current graph stack depth
- Left arrow - Scroll left
- Right arrow - Scroll right
- Up arrow - Scroll up
- Down arrow - Scroll down
- PZ - Push current world and enable zoom
- CY - Cycle through the world stack
- CW:0 - Current world of current graph while cycling through the world stack

## 3.6 Shortcuts

Double clicking on the canvas around various parts of the current graph will bring up what should be an appropriate popup:

- Double clicking to the left of the current graph brings up the View/Ticks-tick labels popup with the axis to edit set to the Y-axis.
- Double clicking below the graph brings up the popup for the X-axis.
- Double clicking above the current graph brings up the popup for the title and subtitle.

- Double clicking to the right of the current graph brings up the View/Legends popup.
- Double clicking inside the current graph brings up View/Symbols, set to edit the set nearest the pointer when the double click was made. If there are no sets, then the Files/Read sets popup is opened.

Note that if the graph focus policy is set to FOLLOWS, this stuff may not work very well in a multiple graph session.

CHAPTER 4

# Command interpreter reference and parameter file format

## 4.1 Introduction

ACE/gr provides a command line interface and the ability to save plot parameters in a file for later retrieval. The syntax for the command line interpreter and parameter files is the same. It is instructive to read a parameter file into the command line interpreter and press the 'Replay' button to verify this feature.

## 4.2 Reference

Case is ignored by the command line interpreter.

Description of parameters:

*color* - integer value from 0 to 29.

*string* - double quote enclosed character string.

*integer* - any integer >= 0

*number* - any floating point value >= 0.0

*expr* - any expression (includes negative floating point values)

*onoff* - ON or OFF

*torf* - TRUE or FALSE

*setnum* - set descriptor in the form Sn where n is the number of the set.

*graphno* - graph descriptor in the form Gn where n is the number of the graph.

*direction* - one of UP, DOWN, LEFT, RIGHT, IN, OUT.

*opchoice* - one of TOP, BOTTOM, LEFT, or RIGHT.

*formatchoice* - one of

- DECIMAL - 0.0
- EXPONENTIAL - 0e+-0
- POWER - $10^{10}$
- GENERAL - varies between 0.0 and 0e+0
- DDMMYY - requires data in Julian date format
- MMDDYY - requires data in Julian date format
- MMYY - requires data in Julian date format
- MMDD - requires data in Julian date format
- MONTHDAY - requires data in Julian date format
- DAYMONTH - requires data in Julian date format
- MONTHS - requires data in Julian date format
- MONTHL - requires data in Julian date format
- DAYOFWEEKS - requires data in Julian date format
- DAYOFWEEKL - requires data in Julian date format
- DAYOFYEAR - requires data in Julian date format
- HMS - requires data in Degrees format
- MMDDHMS - requires data in Degrees format
- MMDDYYHMS - requires data in Degrees format
- DEGREESLON - requires data in Degrees format
- DEGREESMMLON - requires data in Degrees format
- DEGREESMMSSLON - requires data in Degrees format
- MMSSLON - requires data in Degrees format
- DEGREESLAT - requires data in Degrees format
- DEGREESMMLAT - requires data in Degrees format

- DEGREESMMSSLAT - requires data in Degrees format
- MMSSLAT - requires data in Degrees format

The Julian date format is the long form. There are 2 programs in the subdirectory aux/ that can be used to convert data from Gregorian to Julian and vice versa.

The Degrees format is for latitude and longitude. Below the equator latitudes are negative. Longitude is assumed positive east of Greenwich and negative to the west.

### 4.2.1 Exit

- EXIT

Leave ACE/gr.

### 4.2.2 Devices

In the discussion to follow, *printer* refers to one of:

PSMONOP - PostScript portrait.

PSMONOL - PostScript landscape

MIFP - Maker Interchange Format portrait.

MIFL - Maker Interchange Format landscape.

HPGLP - HPGL portrait.

HPGLL - HPGL landscape.

FILE - print to a file rather than to the device.

HARDCOPY - the current hardcopy device.

- PRINT *printer string*

Set the printer command string to *string*.

Example:

print psmonop “lpr -Pps”

- PRINT TO HARDCOPY

Spool hardcopy output to the printer. This command sets the output destination, but does not generate a hardcopy, see the HARDCOPY command.

- PRINT TO FILE

Spool hardcopy output to a file, using the current value of the printer file string (see PRINT TO *string* above). This command sets the output destination, but does not generate a hardcopy, see the HARDCOPY command.

- PRINT TO FILE *string*

Spool hardcopy output to a file named *string*. This command sets the output destination, but does not generate a file, see the HARDCOPY command.

Example:

print to file “hardcopy.out”

- PRINT TO *printer*

Set the hardcopy device to *printer.*

- HARDCOPY

Print to the current hardcopy device or file.

- DEVICE *integer*

Set the screen device to *integer.* At the present time 0 (the X server) is the only value.

### 4.2.3 Display

- REDRAW

Refresh the display.

- AUTO REDRAW *onoff*

Toggle the automatic redrawing of the display.

- BACKGROUND COLOR *color*

Set the background color of the drawing area.

- CMAP *color, integer, integer, integer*

Set the RGB values of a colormap entry. The value of *color* ranges from 2 to 29, The three integer values following the color number are values from 0 to 255 and represent red, green, and blue resp. Colors 0 and 1 are black and white (reversed if the -rvideo command line option was selected), and cannot be changed. Xmgr presently dumps core when using this command after startup, colormap entries can be initialized only.

Example:

To set colormap entry 5, use

cmap 5, 0, 0, 255

This sets color 5 to solid blue.

- PAGE *direction*

Page left, right, up, down, in. out.

- PAGE *integer*

Set the amount of scrolling, and integer value giving the amount of scroll in percent of the graph scaling. For example:

PAGE 100

would set the amount to scroll left, right, up, down to 100 percent of the graph axis scaling.

- PAGE INOUT *integer*

Set the amount to increase or decrease the graph scaling when using the In/Out buttons on the main panel. An integer value expressing the percent to expand or shrink.

- LINK PAGE *onoff*

Toggle linked scrolling. Linked scrolling affects the scroll buttons on the main panel, when linked scrolling is on, all graphs are scrolled simultaneously. Linked scrolling OFF makes scrolling affect the current graph only.

### 4.2.4 Set operations

- ACTIVATE *setnum number*

Activate a set in the current graph and set the length

- COPY *setnum* TO *setnum*

Copy a set to another set

- COPY *graphno.setnum* TO *graphno.setnum*

Copy a set from a particular graph to a set in another graph

Example

copy g0.s0 to g1.s5

- MOVE *setnum* TO *setnum*

Move a set to another set.

Example

move s0 to s1

- MOVE *graphno.setnum* TO *graphno.setnum*

Move a set from a particular graph to a set in another graph

Example

move g0.s0 to g1.s5

- KILL *setnum*

Kill a set. Plot parameters are set to their default values.

- *setnum* POINT *expr, expr*
- *graphno.setnum* POINT *expr, expr*

Add a point to a set, create the set if the set is inactive. The first version adds a point to set *setnum* in the current graph, the second adds a point to *setnum* in the graph *graphno*.

### 4.2.5 Block data

The two block data commands allow block data to be read and sets formed from the active set of block data.

- READ BLOCK *string*

Read a file of block data from file string.

- BLOCK *xytype string*

Create a set of type *xytype* using columns coded in *string*. *String* describes columns as:

“c1:c2:...”

Where c1, c2, ... are the numbers of the columns to use in the construction of the set from the block data.

Examples:

Create a set with error bars using x from column 1, y from column 2 and the errors in column 5:.

read block “block2.dat”

block xydy “1:2:5”

Create an XY type data set from columns 7 and 2.

block xy “7:2”

### 4.2.6 Graph operations

- KILL *graphno*

Kill graph *graphno*.

- KILL GRAPHS

Kill all graphs and sets, but not annotative text, lines, and boxes.

- FLUSH

Kill all graphs, sets, and annotation.

### 4.2.7 Transformations

- LOAD VAR *setnum, expr, expr*

Load a set sequentially

- REGRESS (*setnum, number*)

Regress a set where *number* is the degree of the fit in the range 1-5.

- DIFFERENCE (*setnum, number*)
- DIFF (*setnum, number*)

Difference a set using the method specified by *number.*

0 - forward difference.

1 - backward difference

2 - centered difference

- INTEGRATE (*setnum*)
- INT (*setnum*)

Integrate a set using a trapezoid rule.

- SPLINE(s*etnum, start, stop, integer*)
- *fftype(setnum, integer*)

Compute a DFT either forward or inverse, using the DFT or FFT.

*fftype* is one of:

DFT - compute the DFT using the definition.

FFT - compute the DFT using an FFT.

INVDFT - compute the inverse DFT.

INVFFTa- compute the inverse using the FFT.

- *runtype*(*setnum, number*)

Compute a running average, standard deviation, median, maximum, or minimum.

*runtype* is one of the following:

RUNAVG - running average.

RUNSTD - running standard deviation.

RUNMED - running median.

RUNMAX - running maximum.

RUNMIN - running minimum.

- HISTO(*setnum, xmin, xmax, integer*)

Compute a histogram using set *setnum*. *Xmin* and *xmax* are the bounds of the histogram and the *integer* is how many bins to create in this range.

### 4.2.8 Autoscaling

- AUTOSCALE

Autoscale the current graph

- AUTOSCALE *setnum*

Autoscale the current graph on a particular set

- AUTOSCALE XAXES

Autoscale the X-axis and the axis at Y = 0 in the current graph.

- AUTOSCALE YAXES

Autoscale the Y-axis and the axis at X = 0 in the current graph.

- AUTOSCALE *torf*

Suppress autoscaling on startup. Use this command in a data file that uses imbedded parameter settings to set the axes scaling.

### 4.2.9 Graph focus

- FOCUS *graphno*

Set the current graph to *graphno.*

- FOCUS SET

Focus policy fixed on the current graph.

- FOCUS *onoff*

Turn the drawing of the focus indicators on or off.

- FOCUS FOLLOWS

Focus follows the pointer.

- FOCUS CLICK

Set the focus by clicking on a graph.

### 4.2.10 Locator

- LOCATOR *onoff*

Turn the locator on the front panel on or off.

### 4.2.11 IO

- SOURCE *sourcetype*

Set the source (disk or pipe) for reading XY data sets.

- READ *string*

Read an XY data set.

- READ BATCH *string*

Read a batch file.

- READ *xytype string*

Read a data set of a particular type.

- READ *xytype sourcetype string*

Read a data set of type *xytype* from source *sourcetype*.

- GETP *string*

Read a parameter file.

- PUTP *string*

Write a parameter file.

### 4.2.12 Boxes

NOTE: Boxes, used or not, are numbered from 0 to the maximum number of boxes. This is also true of lines and strings.

- WITH BOX

Get the next available box and make it current. A box needs to be current before any of the following commands will apply.

- WITH BOX *integer*

Get the box numbered *integer* and make it the current box

- BOX *onoff*

Toggle the display of the current box.

- BOX *expr, expr, expr, expr*

Set the location of the box in world or viewport coordinates depending on the value of BOX LOCTYPE. The 4 values represent xmin, ymin, xmax, ymax respectively.

- BOX LOCTYPE *worldview*

Set the location type of the next box created. World or viewport coordinates.

- BOX *graphno*

If the LOCTYPE is WORLD, set the graph to use.

- BOX LINESTYLE *integer*

Set the line style to use for the next box created.

- BOX LINEWIDTH *integer*

Set the line width of the next box created.

- BOX COLOR *integer*

Set the color to use for the box lines.

- BOX FILL *filltype*

Set the type of fill to either COLOR or PATTERN.

- BOX FILL COLOR *integer*

Set the color to use to fill the next box created.

- BOX FILL PATTERN *integer*

Set the pattern to use for the next box created.

- BOX DEF

Define the current box using the values set above.

- CLEAR BOX

Remove all boxes

### 4.2.13 Lines

**ACE/gr** uses an array of a data type called lines internally to hold the definition of lines. The actual construction of this data type is not important, but it is important to note that lines, used or not, are numbered from 0 to the maximum number of lines.

- WITH LINE

Make the next available line the current line.

- WITH LINE *integer*

Make line *integer* the current line.

- LINE *onoff*

Toggle the display of the current line.

- LINE *expr, expr, expr, expr*

Set the location of the current line created. The four expressions represent (X1, Y1), (X2, Y2) respectively.

- LINE LOCTYPE *worldview*

Set the location type of the current line created. World or viewport coordinates.

- LINE graphno

If the line LOCTYPE is WORLD, set the graph to use for scaling.

- LINE LINEWIDTH *integer*

Set the line width of the current line created.

- LINE LINESTYLE *integer*

Set the line style of the current line created.

- LINE COLOR *integer*

Set the color of the current line created.

- LINE ARROW *integer*

Define a line with or without arrow(s) and where they are located.

0 - no arrow

1 - arrow at start of line

2 - arrow at end of line

3 - arrow at both ends.

- LINE ARROW SIZE *number*

Set the size of the current line's arrowhead.

- LINE DEF

Define a line using the current set of line settings.

- CLEAR LINE

Remove all lines.

### 4.2.14 Strings

ACE/gr uses an array of a data type called plotstr internally to hold the definition of strings. The actual construction of this data type is not important, but it is important to note that strings, used or not, are numbered from 0 to the maximum number of strings.

- WITH STRING

Get the next available string

- WITH STRING *integer*

Get the particular string numbered *integer.*

- STRING *onoff*

Toggle the display opf the current string.

- STRING *expr, expr*

Location of the string, in world or viewport coordinates depending on the value of LOCTYPE..

- STRING LOCTYPE *worldview*

Set the location type of the string, either world or viewport coordiantes. If the setting is WORLD, then the strings position is affected by changes in the axes scaling, if VIEW, then the string is fixed to that spot in viewport coordinates.

- STRING *graphno*

Set the graph to uses for scaling when the LOCTYPE is WORLD.

- STRING LINEWIDTH *integer*

Set the line width of the current string.

- STRING COLOR *integer*

Set the color of the current string.

- STRING ROT *integer*

Set the rotation of the current string from-360 to 360 in degrees

- STRING FONT *integer*

Set the font of the current string.

- STRING JUST *integer*

Set the justification for the current string. 0 is left justified, 1 is right justified, and 2 is centered.

- STRING CHAR SIZE *number*

Set the character size of the current string.

- STRING DEF *string*

Define the current string using the values set above.

- CLEAR STRING

Remove all strings

### 4.2.15 World

- WORLD *xmin, ymin, xmax, ymax*

Set the scaling limits for the current graph.

- WORLD XMIN *xmin*
- WORLD XMAX *xmax*
- WORLD YMIN *ymin*
- WORLD YMAX *ymax*

### 4.2.16 World stack

- PUSH

Push the current graph scaling limits and tick spacing onto the graph's world stack.

- POP

Pop the current graph's world stack and set the new scaling limits and tick spacing from the new stack top.

- CYCLE

Cycle through the current graph's world stack.

- STACK *integer*

Set the current graph's scaling limits and tick spacing to the value at position *integer* of the current graph's world stack.

- STACK WORLD *expr, expr, expr, expr* TICK *expr, expr, expr, expr*

Push specific values onto the current graph's world stack - primarily for use in paramter files generated by ACE/gr.

- CLEAR STACK

Clear the current graph's world stack.

### 4.2.17 Viewport

- VIEW *xmin, ymin, xmax, ymax*

Set the currentt graph's viewport (where on the divice the graph is displayed).

- VIEW XMIN *expr*
- VIEW XMAX *expr*
- VIEW YMIN *expr*
- VIEW YMAX *expr*

### 4.2.18 Title and subtitle

- TITLE *string*

Set the graph title.

- TITLE FONT *integer*

Set the font for the graph title.

- TITLE SIZE *number*

Set the character size for the graph title.

- TITLE COLOR *integer*

Set the color for the graph title.

- SUBTITLE string

Set the graph subtitle.

- SUBTITLE FONT *integer*

Set the font for the graph subtitle.

- SUBTITLE SIZE *number*

Set the character size for the graph subtitle.

- SUBTITLE COLOR *integer*

Set the color for the graph subtitle.

### 4.2.19 Sets

*Setnum,* in the following descriptions, refer to the symbolic name of each set, i.e., the letter ‘s’ followed by the integer number of the set. S0 would refer to set 0, s1 to set 1, etc. The following commands for setting set parameters have two prefixes, GRAPHS and SETS, that allow the setting to be made for all graphs, all sets, or a given set in all graphs. So, to set the line width for set 0 to be 3 in all graphs that have a set 0 active, the command would be:

graphs s0 linewidth 3

Likewise, to set the line width for all sets in the current graph, give the command:

sets linewidth 3

To set the line width for all sets in all graphs to 3, execute:

graphs sets linewidth 3

- *setnum onoff*

Toggle the active/inactive status of sets. This can be used to force ACE/gr to ignore a set(s), even though data are still attached to the set. For example, assuming S0 is an active set:

S0 off

will allow ACE/gr to ignore S0 for all purposes, except any operation that kills a set. Autoscaling will ignore this set, etc - in effect, S0 is dead. To bring it back to life:

S0 on

will reintroduce the set with its data intact. At the present time, there is no corresponding popup to perform this action.

- *setnum* TYPE *xytype*

Set the type of set *setnum* to xytype, where *xytype* is one of xy, xydx, xydy, xydxdx, xydydy, xydxdy, xyz, xyr, or xyhilo

- *setnum* FONT *integer*

Set the font to use when the set type is xyz.

- *setnum* PREC *integer*

Set the precision when the set type is xyz.

- *setnum* FORMAT *formatchoice*

Set the format to use when the set type is xyz.

- *setnum* SYMBOL *integer*

Set the symbol for the set.

- *setnum* SYMBOL SIZE *number*

Set the size of the symbol.

- *setnum* SYMBOL CHAR *integer*

Set the character to use when using the symbol type character. The value is the decimal representation of the character using the ASCII collating sequence.

- *setnum* LINESTYLE *integer*

Set the lines style for the set.

- *setnum* LINEWIDTH *integer*

Set the line width for the set.

- *setnum* COLOR *integer*

Set the color to use to draw the lines and symbol.

- *setnum* FILL *integer*

Set the type of fill for the set.

- *setnum* FILL WITH COLOR

Set the type of fill for the set.

- *setnum* FILL WITH PATTERN

Set the type of fill for the set.

- *setnum* FILL COLOR *integer*

Set the color for the fill if the fill selected is color.

- *setnum* FILL PATTERN *integer*

Set the color for the fill if the fill selected is color.

- *setnum* SKIP *integer*

Set the number of points to skip before placing a symbol.

- *setnum* ERRORBAR TYPE *opchoice*

Set the the error bar display type. Opchoice is one of RIGHT, LEFT, or BOTH if the error bar type is xydx or xydxdx, or TOP, BOTTOM, or BOTH if the type is xydy or xydydy. The default in either case is BOTH, i.e. display both error bars.

- setnum ERRORBAR LENGTH *number*

Set the length of the error bar.

- *setnum* ERRORBAR LINEWIDTH *integer*

Set the line width for the error bar.

- *setnum* ERRORBAR LINESTYLE *integer*

Set the linestyle for the error bar.

- *setnum* ERRORBAR RISER *onoff*

Toggle the display of the error bar riser.

- *setnum* ERRORBAR RISER LINESTYLE *integer*

Set the line style for the error bar riser.

- *setnum* ERRORBAR RISER LINEWIDTH *integer*

Set the line width for the error bar riser.

- *setnum* COMMENT *string*

Set the comment string for the set.

### 4.2.20 Legend

- LEGEND *onoff*

Toggle display of the legend.

- LEGEND LOCTYPE *worldview*

Position the legend in either world or viewport coordinates.

- LEGEND VGAP *integer*

Set the vertical gap between legend entries in characters.

- LEGEND HGAP *integer*

Specify the gap between the display of the symbol and the legend label in units of characters.

- LEGEND LENGTH *integer*

Set the length of the legend in units of characters.

- LEGEND *expr, expr*

Set the location of the legend.

- LEGEND X1 *expr*

Set the X value of the location.

- LEGEND Y1 *expr*

Set the Y value of the legend.

- LEGEND FONT *integer*

Set the font to use for the legend labels.

- LEGEND CHAR SIZE *number*

Set the size of the characters in the legend label.

- LEGEND LINEWIDTH *integer*

Set the line width to use to draw the legend labels.

- LEGEND COLOR *integer*

Set the color of the legend labels.

- LEGEND BOX *onoff*

Toggle the display of the bounding box for the legend.

- LEGEND BOX FILL *onoff*

Toggle the filling of the bounding box for the legend.

- LEGEND BOX LINEWIDTH *integer*

Set the line width to use to draw the bound box for the legend.

- LEGEND BOX LINESTYLE *integer*

Set the line style to use to draw the bound box for the legend.

- LEGEND BOX COLOR *integer*

Set the color of the legend bounding box.

- LEGEND BOX FILL COLOR *integer*

Set the color to use for the filled legend bounding box.

- LEGEND BOX FILL PATTERN *integer*

Set the pattern to use for the filled legend bounding box.

- LEGEND BOX FILL WITH *colpat*

Set the type of fill for the bounding box, either COLOR or PATTERN.

- LEGEND STRING *integer string*

Set the legend label for *setnum integer.*

### 4.2.21 Graph frame

- FRAME *onoff*

Toggle the display of the current graph's frame.

- FRAME TYPE *number*

Set the type of frame for the current graph:

0 = rectangle

1 = Lines along the left and bottom of the graph.

- FRAME LINESTYLE *number*

Set the line style of the current graph's frame.

- FRAME LINEWIDTH *number*

Set the width of the line for the current graph's frame.

- FRAME COLOR *number*

Set the color of the current graph's frame.

- FRAME FILL *onoff*

Toggle the fill of the graph frame.

- FRAME BACKGROUND COLOR *number*

Set the color to use for filling the current graph's frame.

### 4.2.22 Graph axes

The are three axes in each coordinate direction. In the case of the X coordinate direction, there is one that follows the world scaling, one at Y = 0 and another that may be used to display an alternate scale. The names used by ACE/gr to refer to these axes are, XAXIS, ZEROXAXIS, and ALTXAXIS, respectively. Likewise along Y there is the YAXIS, ZEROYAXIS, and ALTYAXIS. There are names that can be used to refer to all the axes along a coordinate direction or to both directions or to all graphs, these being AXES (both coordinate directions, current graph), XAXES (along X in the current graph), YAXES (along Y in the current graph), and preceeding these with the key word GRAPHS, will cause the setting to be made throughout all the active graphs.

In the following descriptions, *axis* refers to the choices described above.

- *axis onoff*

Toggle the display of the axis or axes referred to by *axis*.

- *axis* COLOR *integer*

Set the color for the axis or axes specified by *axis*.

- *axis* LINEWIDTH *integer*

Set the line width for the axis or axes specified by *axis*.

- *axis* LINESTYLE *integer*

Set the line style for the axis or axes specified by *axis*.

- *axis* FONT *integer*

Set the font to use for text for the axis or axes specified by *axis*.

- *axis* CHAR SIZE *number*

Set the character size for text for the axis or axes specified by *axis*.

### 4.2.23 tick marks.

- *axis* TICK MAJOR *onoff*
- *axis* TICK MINOR *onoff*
- *axis* TICK MAJOR *expr*

Set the tick spacing for major tick marks.

- *axis* TICK MINOR *expr*

Set the spcing for minor tick marks.

- *axis* TICK OFFSETX *number*

Set the amount to offset the axis, in viewport coordinates in the X direction.

- *axis* TICK OFFSETY *number*

Set the amount to offset the axis in viewport coordinates in the Y direction.

- *axis* TICK ALT *onoff*

Toggle the use of the alternate map for the axis scaling.

- *axis* TICK MIN *number*

Specifiy the minimum value to use for the alternate map.

- *axis* TICK MAX *number*

Specify the maximum value to use for the altermate map.

- *axis* TICK DEFAULT *number*

Set the default number of ticks to use when autoscaling.

- *axis* TICK *inout*

Set the display of tick marks to IN, OUT, or BOTH.

- *axis* TICK SIZE *number*

Set the size of tick marks.

- *axis* TICK MAJOR SIZE *number*

Set the size of major tick marks.

- *axis* TICK MINOR SIZE *number*

Set the size of minor tick marks.

- *axis* TICK COLOR *number*

Set the color to use for tick marks.

- *axis* TICK MAJOR COLOR *number*

Set the color to use for major tick marks.

- *axis* TICK MINOR COLOR *number*

Set the color to use for minor tick marks.

- *axis* TICK MAJOR LINEWIDTH *number*

Set the line width to use for grid lines at major tick marks.

- *axis* TICK MAJOR LINESTYLE *number*

Set the line style to use for grid lines at major tick marks.

- *axis* TICK MINOR LINEWIDTH *number*

Set the line width to use for grid lines at minor tick marks.

- *axis* TICK MINOR LINESTYLE *number*

Set the line width to use for grid lines at minor tick marks.

- *axis* TICK MAJOR GRID *onoff*

Toggle the display of grid lines at major tick marks.

- *axis* TICK MINOR GRID *onoff*

Toggle the display of grid lines at minor tick marks.

- *axis* TICK OP *opchoice*

Set the display of tick marks on the axis to LEFT, RIGHT, or BOTH if the axis is in the X direction, or TOP, BOTTOM, or BOTH if the axis is in the Y direction. The default is BOTH in either case.

- *axis* TICK TYPE AUTO

Use the values for major and minor spacing for draw the tick marks. See the next it for tick marks at specified locations.

- *axis* TICK TYPE SPEC

Use specified values for drawing tick marks. These values are for major tick marks only.

- *axis* TICK SPEC *integer*

Give the number of specified tick marks.

- *axis* TICK n*umber, expr*

Set the value of specified tick mark number *integer* to value, *expr.*

### 4.2.24 Axes tick mark labels

- *axis* TICKLABEL PREC *integer*

Set the number of places to the right of the decimal point when drawing tick lables.

- *axis* TICKLABEL FORMAT *format*

Set the format to use for drawing tick labels.

- *axis* TICKLABEL LAYOUT HORIZONTAL

Set the angle of the axis tick labels to be horizontal.

- *axis* TICKLABEL LAYOUT VERTICAL

Set the angle of the axis tick labels to be vertical.

- *axis* TICKLABEL SPEC

Specifiy the use of the specified tick label angle.

- *axis* TICKLABEL ANGLE *number*

Specifiy the angle to use for drawing the tick labels in degrees from 0 to 360.

- *axis* TICKLABEL JUST *justify*

Sepcifiy the type of justification to use when drawing the tick label.

- *axis* TICKLABEL SKIP *integer*

Set the number of major tick marks to skip before drawing a tick lablel.

- *axis* TICKLABEL STAGGER *integer*

Set the number of characters to use to offset the tick labels.

- *axis* TICKLABEL OP *opchoice*

Set the side(s) to draw tick labels, LEFT, RIGHT, or BOTH for tick labels in the Y direction and TOP, BOTTOM, or BOTH for tick labels in the X direction. The default is LEFT for the Y axis tick labels and BOTTOM for the X axis tick labels.

- *axis* TICKLABEL SIGN *signchoice*

Set the function to uise on the numeric value used to create the tick mark label. One of NORMAL, ABSOLUTE, or NEGATE. The default is NORMAL, i.e., no transformation is applied. ABSOLUTE indicates that the absolute value of the tick label location is used, NEGATE is the negative of the tick mark location. The latter two can be used to achieve a reversal of axis or a reflection effect of the axes. Of course, the data will need to be transformed also.

- *axis* TICKLABEL START *expr*

Set the value to use to begin drawing tick lables.

- *axis* TICKLABEL STOP *expr*

Set the value to stop drawing tick labels.

- *axis* TICKLABEL START TYPE SPEC

Use the specified starting value for drawing tick mark labels.

- *axis* TICKLABEL STOP TYPE SPEC

Use the specified stopping value for drawing tick mark labels.

- *axis* TICKLABEL START TYPE AUTO

Use the graph minimum to use as the starting point for drawing tick labels, this is the default.

- *axis* TICKLABEL STOP TYPE AUTO

Use the graph maximum as the stopping point for drawing tick mark labels. This is the default.

- *axis* TICKLABEL VGAP *number*
- *axis* TICKLABEL HGAP *number*
- *axis* TICKLABEL CHAR SIZE *number*

Set the size of characters to use when drawing the tick mark labels.

- *axis* TICKLABEL FONT *integer*

Set the font to use when drawing tick mark labels.

- *axis* TICKLABEL COLOR *integer*

Set the color to use for drawing tick mark labels.

- *axis* TICKLABEL LINEWIDTH *integer*

Set the line width to use for drawing tick mark labels.

- *axis* TICKLABEL TYPE AUTO

Use the values of major tick marks for position the tick mark labels.

- *axis* TICKLABEL TYPE SPEC

Specify the tick mark label to use at each major tick mark.

- *axis* TICKLABEL *integer, string*

Set the value for the *integer*'th tick mark label.

### 4.2.25 Axes label strings.

- *axis* LABEL *string*

Set the text string to use for the axis label.

- *axis* LABEL LAYOUT PERP
- Set the layout of the axis label to be perpendicular to the axis.
- *axis* LABEL LAYOUT PARA

Set the layout of the axis label to be parallel to the axis.

- *axis* LABEL CHAR SIZE *number*

Set the character size of the text used for the axis label.

- *axis* LABEL FONT *integer*

Set the font to use for the axis label.

- *axis* LABEL COLOR *integer*

Set the color to use for the axis label.

- *axis* LABEL LINEWIDTH *integer*

Set the line width to use for drawing the axis label.

Axes bar

- *axis* BAR *onoff*

Toggle the display of the axis bar. The dfault is OFF.

- *axis* BAR COLOR *integer*

Set the color to use for the axis bar.

- *axis* BAR LINEWIDTH *integer*

Set the line widht to use when drawing the axis bar.

- *axis* BAR LINESTYLE *integer*

Set the line style to use for drawing the axis bar.

### 4.2.26 Graphs

- WITH *graphno*

Set the current graph to graphno.

- *graphno onoff*

Set graphno on or off.

- *graphno* AUTOSCALE TYPE AUTO

Set the method of autoscaling to a type developed by Paul Heckbert. This method makes nice tick spacing, but fiddles with scales of the axes.

- *graphno* AUTOSCALE TYPE SPEC

Set the method of autoscaling to use the minimum and maximum values of the data.

- *graphno* HIDDEN *torf*

Toggle the display of graphno.

- *graphno* TYPE *graphtype*

Set the type of *graphno* to *graphtype*, where *graphtype* is one of:

XY - Linear scaling on both X and Y axes (the default).

BAR - bar chart.

STACKEDBAR - stacked bar chart.

LOGX - logarithmic X axis, linear Y axis.

LOGY - linear X axis, logarithmic Y axis.

LOGXY - Log-log graph.

- *graphno* FIXEDPOINT *onoff*

Toggle the use of the graph fixed point, i.e., the point used as a reference for the locator display.

- *graphno* FIXEDPOINT XY *number, number*

Set tghe value of the graph fixed point.

- *graphno* FIXEDPOINT TYPE *number*

Set the type of display for the locator on the main panel.

- *graphno* FIXEDPOINT FORMAT *format format*

Select the format to use for both X and Y in the locator display.

- *graphno* FIXEDPOINT PREC *integer, integer*

Set the number of places to display to the right of the decimal point in the locator display for both X and Y.

# CHAPTER 5 Reference

## 5.1 Hot keys

When the mouse is on the canvas (where the graph is drawn), there are some shortcuts that can be taken to bring up several popups. They are:

- ^A - autoscale plot
- ^B - draw a box (using the current settings for boxes)
- ^D - delete an object (string, line, or box)
- ^N - move an object (string, line, or box)
- ^P - draw a line (using the current settings for lines)
- ^V - set the viewport with the mouse
- ^W - write a string (using the current settings for strings)
- ^X - exit ACE/gr
- ^Z - enable zoom

## 5.2 Fonts

### 5.2.1 Changing fonts within a string

To change fonts within a string preceed the font number listed below by a backslash. To turn subscripting or superscripting on or off use \s for subscripts and \S for superscripting. Font selection is current for the remainder of the string or until the next font change. Subscripts and superscripts remain for the remainder of the string or until \N is seen. To print a backslash use \\. To backspace use \b. To begin underlining use \u, to stop underlining use \U. \+ increases the size of the characters and \- decreases the size of the characters.

ACE/gr uses the Hershey fonts to draw text on the screen, but PostScript fonts for hardcopy. There are discrepancies between the two sets of fonts and of this writing, there are problems with the mapping of Greek and special characters.

Font # Font

- 0 .... Complex Roman on screen, maps to Times-Roman.
- 1 .... Triplex Roman on screen, maps to Times-Bold.
- 2 .... Complex Italic on screen, maps to Times-Italic.
- 3 .... Triplex Italic on screen, maps to Times-BoldItalic.
- 4 .... Simplex on screen, maps to Helvetica.
- 5 .... Duplex on screen, maps to Helvetica-Bold
- 6 .... Complex Italic on screen, maps to Helvetica-Oblique.
- 7 .... Triplex Italic on screen, maps to Helvetica-BoldOblique.
- 8 .... Simplex Greek on screen, maps to the lower 128 characters of the Symbol font.
- 9 .... Symbols on screen, maps to the upper 128 characters of the Symbol font.
- x .... Special symbols

Summary of other special commands

- + .... increase size
- - .... decrease size
- b .... backspace length of previous character
- s .... begin subscripting
- S .... begin superscripting
- u .... begin underline
- U .... stop underline
- N .... return to normal

Example:

\0F\sX\N(\8e\0) = sin(\8e\0)*e\S(-X)\N*cos(\8e\0)

prints roughly $F_X(\varepsilon) = \sin(\varepsilon)*e^{-X}*\cos(\varepsilon)$

using font 0 and e prints as epsilon from the Simplex Greek font.

NOTE:

Special characters are mapped to the keyboard, the present mapping is not very well organized and may change.