


Easy Plot

*Scientific Graphing
and Data Analysis*

A product of

 Spiral Software

Easy Plot™

For Microsoft Windows

Scientific Graphing and Data Analysis

by Stuart Karon

easyplot@spiralsoftware.c o m

 Spiral Software

This document was prepared using Framemaker for Windows.

The manual describes EasyPlot for Windows Version 4.0.

Copyright © 1988 - 2002 by Spiral Software and Massachusetts Institute of Technology

All rights reserved. No part of this manual may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of MIT and Spiral Software.

Neither MIT nor Spiral Software make any guarantees as to the suitability or accuracy of this program for individual applications.

Table of Contents

List of Figures	vii
List of Tables	ix
Preface	xi
1 Getting Started	1
1.1 System Requirements	1
1.2 Installing and Running EasyPlot for Windows	1
1.3 Selecting Curves.	2
1.4 A Guided Tour	2
1.4.1 Loading Data	3
1.4.2 Data Viewing Tools	3
1.4.3 Mathematical Analysis	4
1.4.4 Customizing Axes	5
1.4.5 Annotating and Titling Graphs	6
1.4.6 Previewing and Printing	8
1.4.7 Multiple Graphs on a Page	8
2 Making Graphs	11
2.1 Plotting Data.	11
2.1.1 Listing Specific Files Types	12
2.1.2 Browsing Files	12
2.1.3 Entering Data	12
2.1.4 Plotting Data in the Clipboard	13
2.1.5 Reading Binary Data Files	13
2.1.6 ASCII File-Reading Options	14
2.1.7 File-Reading Preferences	15

2.2	Styling the Data	17
2.2.1	Selecting the Data Mark, Color, Dash, & Fill	18
2.2.2	Data Marks	18
2.2.3	Marker Preferences	19
2.2.4	Connecting Lines	20
2.2.5	Line chart Preferences	21
2.2.6	Error Bars	22
2.2.7	Three D Style	24
2.2.8	The Bar Style	24
2.3	Font and Font Size.	25
2.4	Data Style Dialog	26
2.5	Polar Plots	27
2.6	Graph and Axis Titles	28
2.7	Annotations	28
2.8	Text Toolbar.	30
2.9	Lines and Arrows	32
2.10	Grid Lines	33
2.11	Axis Setup Dialog	34
2.11.1	Setting the Range	35
2.11.2	Positioning Tic-Marks on Axes	36
2.11.3	Turning Tic Marks Off	37
2.11.4	Floating Tic Marks	37
2.11.5	Tic Size and Direction	39
2.11.6	Axis Placement	40
2.11.7	Log Scales	40
2.11.8	Reverse-Direction Axes	41
2.12	Creating a Legend	41
2.13	Multiple-Axis Graphs	42
2.14	Axis Linking and Mirroring.	43
2.15	Zooming	44
2.16	Scrolling.	44
2.17	Autorange	45
2.18	Displaying Multiple Graphs.	45
2.19	Maximizing and Minimizing Graphs	47
2.20	Saving Graphs	48
2.21	Deleting Curves	50

3	Input	51
3.1	Creating Data Files	51
3.2	Naming Columns	53
3.3	Reading Data Files	55
3.4	Defining the Columns	55
3.5	Date and Time Data	58
3.6	Comments	59
3.7	Equations	60
3.7.1	Recomputing an Equation	62
3.7.2	Setting Curve Resolution	63
4	Data Analysis	65
4.1	The Curve Toolbar	65
4.2	Transforming Data	65
4.2.1	Computing Error Bars	67
4.2.2	Math on Multiple Curves	67
4.3	Difference Equations	69
4.4	Dragging Curves	69
4.5	Curve Fitting	70
4.5.1	Splines	73
4.5.2	Surface Splines	73
4.6	Statistics	74
4.7	Histogram	74
4.8	Integration	75
4.9	Differentiation	76
4.10	Data Smoothing, Range Averaging & Decimation	77
4.11	Interpolating Evenly Spaced Data.	78
4.12	Fourier Transform	78
4.13	Analyzing Frequency Change Over Time.	79
4.14	Reading Coordinates	80
4.15	Digitizing from Screen to File.	80
5	3-D	81
5.1	3-D Data	82
5.2	3-D Graphs	83
5.2.1	Rotating 3-D graphs	83
5.2.2	Perspective	83
5.2.3	Color	84
5.2.4	Hidden-Line Removal	84

5.2.5	Contour Lines on 3-D Graph	84
5.2.6	Shadowing XYZ Data onto Axis Planes	85
5.2.7	Axes & Surrounding Boxes	85
5.3	Contour Plots	86
5.4	Image Plots	87
5.5	Gridding	87
5.6	Color as a 3rd or 4th Dimension	88
6	Batch & Programming Interfaces	89
6.1	Batch Command Overview	89
6.2	Menu Commands	90
6.3	Non-Menu Commands	91
6.3.1	Data File Commands	91
6.3.2	Batch Macros	92
6.4	The “profile.ep” File	94
6.5	Automating EasyPlot to Speed Your Work	95
6.6	Setting F-Key Macros	96
6.7	Concatenating Strings in Batch Mode.	97
6.8	Programming Interface	97
6.9	Command-Line Parameters	103
7	EasyPlot in Detail	105
7.1	Date and Time Stamps on Printouts.	105
7.2	Using Expressions to Enter Numbers	105
7.3	The Undo Feature	106
7.4	Quitting EasyPlot	106
7.5	Remembering the Current Directory	106
7.6	The EasyPlot Initialization File	107
7.7	Refreshing the Screen	107
7.8	The Menu System	107
7.9	Editing Data	108
7.9.1	The Graphical Data Editor	108
7.9.2	Editing Data in the Data Table	108
7.10	The Data Table	109
7.11	Setting Screen Colors	112
7.12	Running EasyPlot on a Network	113
7.13	Scratch Files.	113
7.14	The Preferences Dialog	114

7.15	Cyclical X -Values	114
7.16	On-line Help.	115
8	Printing	117
8.1	Printing a Graph	117
8.2	Print Preferences	118
8.3	Previewing.	119
8.4	Exporting Graphs to Other Applications	120
8.5	Printing Multiple Data Pages	120
8.6	Differences Between What-You-See and What-You-Get	121
Appendix A	Menu Batch Commands	123
Appendix B	Non-Menu Batch Commands	129
Index		139

List of Figures

2.1	File reading Preferences.	15
2.2	The Data Style dialog	17
2.3	The 14 Data Marks	19
2.4	Marker Preferences	19
2.5	The 7 Dash Patterns	20
2.6	Line Chart Preferences	21
2.7	A data file with error values.	23
2.8	Bar chart Preferences	24
2.9	The Font dialog	26
2.10	The Annotation Properties dialog	29
2.11	The Line/Arrow Properties dialog	33
2.12	Grid Setup and Grid Preferences dialogs	34
2.13	The Axis Setup dialog.	35
2.14	The Floating Tics dialog	38
2.15	Example batch commands for adding tic marks	39
2.16	Tic Preferences	40
2.17	The Tile dialog	46
2.18	The Save dialog	48
2.19	Saving Preferences	49
3.1	Example batch commands for defining data columns	58
3.2	How not to comment a file	59
5.1	The 3-D User Interface	81
5.2	Floating 3-D contours	82
5.3	Shadowing <i>xyz</i> data onto back planes.	85
5.4	Contour map.	86
5.5	Image plot	88
6.1	A few example batch commands	91
6.2	Sample profile.ep file.	95

6.3	Sample template file – template.ep	95
6.4	Sample batch file – stdgrph.ep	95
6.5	Batch file that creates multiple graphs	96
7.1	The Color Setup dialog	112
8.1	The Print dialog	117

List of Tables

2.1	Text formatting codes	32
3.1	Characters used to define columns	56
3.2	Default column definitions	57
3.3	Math functions	61
3.4	Mathematical operators	62
6.1	Equation Constants and Macros.	93
7.1	Spreadsheet Keyboard Shortcuts	110

Preface

EasyPlot is a tool for viewing, analyzing, and plotting scientific data. It is designed for anyone who works with scientific or engineering data. EasyPlot provides a powerful graphical environment in which you interact directly with graphs and data.

The philosophy behind EasyPlot is to make what most people do most of the time as simple as possible. You should find EasyPlot's menu system compact and straightforward. You will not find layers of complicated dialog boxes; EasyPlot has a few neatly laid-out dialogs that control key aspects of graphs. Details of fine-tuning EasyPlot and your graphs are organized in a **Preferences** dialog that gives you access to loads of special features.

If you are not familiar with EasyPlot, don't be fooled by its small size and uncluttered appearance. The entire program is one executable file that occupies around 700K of disk space. A great deal of engineering has gone into keeping EasyPlot small and the result is a program that does a lot, runs quickly, and takes up relatively little disk space.

The original EasyPlot was developed over a two and a half year span (1986–1989) at MIT Lincoln Laboratory. Lincoln Laboratory provided an ideal environment for developing a general-purpose plotting program; I was surrounded by scientists who used the software, provided endless constructive feedback, and could get updates whenever I implemented a new feature. Today, the Internet is extending Spiral Software's corridors around the world and, once again, scientists and engineers (that's you) can participate in EasyPlot's development by submitting ideas and receiving software updates minutes after a new feature is implemented. I urge more of you to communicate even the smallest suggestions; your ideas fuel EasyPlot's growth and development.

Stuart Karon
stuart@spiralsoftware.com

Chapter 1: Getting Started

1.1 System Requirements

EasyPlot for Windows will run on any PC system running Windows 95, Windows 98, 2000, ME, XP and Windows NT.

1.2 Installing and Running EasyPlot for Windows

Place the EasyPlot diskette into your computer and run **setup.exe**.

EasyPlot for Windows consists of one file, **epw.exe** for 16-bit EasyPlot, or **epw32.exe** for the 32-bit version. EasyPlot also comes with a supporting help file (**help.ep**) and a few small data files. EasyPlot does not add or modify any files in your Windows directory. If you need to remove EasyPlot from your hard drive, simply delete the files in the EasyPlot directory and remove the EasyPlot icon from your Windows desktop. Or use the **Uninstall** utility loaded with EasyPlot.

When you run EasyPlot, an empty graph appears and EasyPlot prompts you for something to plot (an ASCII or binary data file, a spreadsheet file, an equation,...). Answer the prompt and a graph appears. To overlay more curves on the same graph, click **File / Open** and load another data file. Choose **File / New** to create a new, empty graph.

EasyPlot has two menus (§7.8). One is more compatible with earlier versions of EasyPlot; the other is more Windows standard. The Windows-standard menu is selected when you first install EasyPlot. If you are familiar with old versions of EasyPlot, or plan to use EasyPlot's batch language to automate graphing, you may prefer the EasyPlot-standard menu. To switch, pull down **File**, run **Preferences**,

and go to the **User interface** topic. The two menus are very similar and you can switch from one to the other at any time.

1.3 Selecting Curves

Much of what you do with EasyPlot centers around curves on graphs. There are several ways to select curves on graphs with many data sets displayed. If you run a function that operates on a curve (such as **Tools / Curve fit**), you need to select a curve, assuming there is more than one curve on the graph. You can select a curve before you choose the operation, or after. If no curve is selected when you run a curve function, EasyPlot displays a list of curves and prompts you to select one.

If you select a curve before choosing the curve function, you can do it right on the graph. Click on a curve to select it. A piece of the curve appears on the help line at the bottom of the EasyPlot window (the status bar) to show you which curve is selected. If a curve is already selected and you click near the selected curve but want to pick another nearby curve, you may notice a small delay before the next curve shows up as selected; the delay gives you time to double-click before EasyPlot changes the selected curve.


You can also select curves with the **<space>** bar or the right mouse button. If you use the right mouse button, be sure to click on an empty part of the graph; otherwise, you'll activate a context-sensitive right-mouse menu. Each time you press the right mouse button or **<space>** bar, EasyPlot selects the next curve on the graph. When you reach the last curve, EasyPlot cycles back to the beginning. If you hold the **<space>** bar or right mouse button, the selected curve glimmers, making it easy to spot on a crowded graph.

1.4 A Guided Tour

This section guides you through creating a graph and using a few of EasyPlot's special features. We'll begin by reading a couple data files and looking at the data with EasyPlot's visual-analysis tools. We'll do some mathematical analysis on the data, title and annotate the graph, and then take a look at the final results.

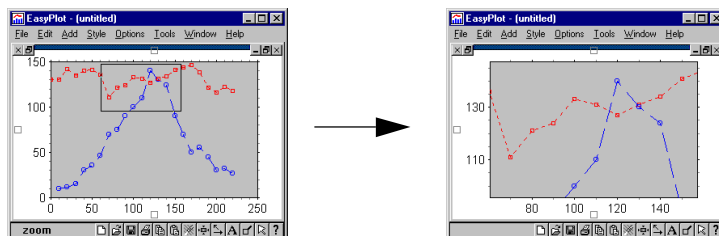
The tour covers commonly used features and should help you get started using the program. It is not a comprehensive introduction to EasyPlot; the topics discussed represent only a fraction of EasyPlot's capabilities. Throughout the introduction, notice that you always work directly with graphs, not columns of numbers.

1.4.1 Loading Data

To begin, we'll load a few data sets. Run EasyPlot and, at the **Open** dialog, select **Data2** in the EasyPlot directory. Click the **Open** icon () on the help line (the status bar), or click **File / Open**, and load **Data3**. **Data2** and **Data3** are short, ASCII data files provided with the program.


1.4.2 Data Viewing Tools

Zooming in Click and drag the cursor to define a zoom rectangle on the graph. When you release, EasyPlot expands the data inside the zoom rectangle (§2.15).



Zoom in a few more times. There's no limit to the number of times you can zoom.

Scrolling Once you've zoomed in, you can use scroll bars (§2.16) to shift your view of the data. Pull down **Tools** and select **Scroll**. Try scrolling. If you do a lot of scrolling, turn on **<Scroll Lock>** to use cursor arrow keys to scroll the graph.

Zooming Out There are several ways to zoom back out. To return to the previous zoom level, hold **<Ctrl>** and click on the graph. EasyPlot remembers up to five zoom ranges. To zoom all the way out, **Autorange** the graph. Click the **Autorange** icon () on the help line, or **Options / Autorange**. The **Autorange** feature (§2.17) sets axis ranges to include all data and major ticks at the axis ends.

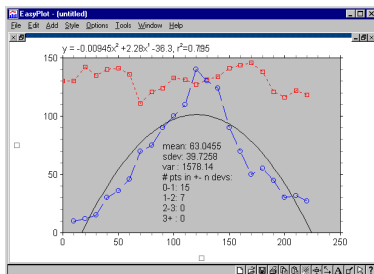
Reading Coordinates from the Graph Turn scroll bars off and select **Tools / X-hair**. Move the cursor inside the graph. EasyPlot displays the *x*- and *y*-coordinates of the cursor in two small windows above the graph (§4.14). You can lock the cursor to a curve by clicking on the curve you want to track. Hit **<esc>** or move the cursor off the graph to unlock from the curve. You can disable the lock-to-curve feature by right clicking on one of the cross-hair windows. Turn off the **X-hair** tool; click **Tools / X-hair**, or right click on a cross-hair window.

1.4.3 Mathematical Analysis

EasyPlot's mathematical analysis tools compute curves and statistics based on existing curves. EasyPlot gives you the option to save the computed data to a separate file. If you don't care about saving computed data to separate files (it is saved with the graph otherwise), you can turn off the save prompt. For the analysis steps below, we assume the **save computed data** option is off. Click **File / Preferences**, scroll to the **Safeties** topic, and make sure that the option **Prompt to save computed data** is off.

Curve Fitting Pull down the **Tools** menu and select **Curve fit**. Choose to fit the **Data2** curve, the one with open-circle markers. If you select a curve before running **Curve fit** (by clicking it on the graph), EasyPlot would not prompt you to select a curve. At the **Curve fit** dialog (§4.5), enter “2” and hit **OK**. The fitted curve appears on the graph and EasyPlot puts the fit equation above the graph. The equation is an annotation that you can move, edit, or delete. Because curve-fits can shoot way off the graph, EasyPlot locks the axis ranges (turns autoranging off) before plotting fits. Later on, we'll need to autorange an axis as a result of having fit a curve.

Statistics Click on the **Data2** curve; a piece of the curve appears on the help line (or “status bar”) to show it's selected. Pull down **Tools**, go to **Stats**, and run **Mean, Std Dev,...** (§4.6). Notice you're not prompted to select a curve. EasyPlot places the statistics on the graph as an annotation which, like the curve-fit equation, you can move, edit, or delete. Click on the note and drag it to the middle of the graph:



Smoothing Click on the **Data3** curve and run **Tools / Smooth**. You can smooth data (§4.10) with a sliding average or by averaging all the points within range segments on the x -axis. Choose the **sliding data window** and select a 5-point window. The smoothed curve appears on the graph.

Transforming Data In spreadsheets, you can compute new columns of data from mathematical operations on existing columns. In EasyPlot, you can create new curves by applying equations to existing curves (§4.2). Click on the **Data3** curve again and run **Tools / Transform**. There are three options in the bottom-left corner of the **Transform** dialog. Make sure **Keep original curve** is checked and enter **$y = 7\log(y)$** . Multiplication is implicit; you do not need to type **$7*\log(y)$** . The transformed curve appears near the bottom of the graph.

1.4.4 Customizing Axes

The new curve (the log of **Data3**) does not show up well on the y-axis that runs from 0 to 150. Let's plot the transformed curve on its own y-axis and then adjust the range and tic positions on an axis.

Private Axes Any curve on an *xy* graph can have its own “private” *x*- and/or *y*-axis (§2.14). Private axes are color-coded to the curve and while they are called “private”, you can plot any curve against another curve's private axes. Right click on the transformed curve, select **Private axes...**, and click **Y-axis**. A new axis appears to the left of the primary y-axis.

Range & Tic Positions So far, EasyPlot has set the range and tic positions for all the axes. Double click on the private axis. At the top of the **Axis Setup** dialog (Figure 2.13), enter **14** in **Coor of a major tic**, **0.2** for the **Maj tic increment**, and **1** for **# of minor tics**. Click **OK**. Note that the range changes to 14.2 to 15.2; autoranged axes include all data and major tics at the axis ends and when you change tic positions on an autoranged axis, the range can change, too.

Axis Markers Click from one y-axis to the other. EasyPlot draws a small color-coded marker for each curve plotted on the selected axis. Now click from one curve to another. A single color-coded marker appears above the axis on which the curve is plotted. The markers show you how curves and axes are organized and can be very helpful on complicated graphs.

Floating Tic Marks You can add individual “floating” tic marks anywhere on an axis to highlight special values (§2.11.4). Double click on the private y-axis (the transformed curve's y-axis) and select **Floating tics...** in the top right of the **Axis Setup** dialog. Add a tic at 14.86, and enter “median” in **label for tic**. (14.86 is not the true median but it's close.) Click **OK** to close both dialogs. You can label entire axes with floating tics to read day-of-week or month, for example. Or use them to produce custom, nonlinear scales, such as for probability plots.

Secondary Axes In addition to primary and private axes, you can put a second x - and/or y -axis on graphs (§2.14). Secondary axes appear on the right and top sides of the graph. Right click on the primary y -axis and select **Add 2nd Y-axis**. The new axis inherits the range and tic positions of the primary axis. The open marker above the new axis tells you that no curves are plotted against it. Let's move the two top curves (\cdot^{\square} and \cdot^{\bullet}) to the right-hand axis. Right click on one of them and choose **Select axis...** EasyPlot shows you a piece of the selected curve on the help line along with the prompt "Click on axis to use...". Click on the right axis. The only change you should see is in the marker above the axis. If the secondary axis were set to autorange, it would have rescaled automatically upon having a curve assigned to it. But it inherited the range of the primary axis and because of the curve fit we ran earlier, the primary axis was not autoranged. Double click on the right-hand y -axis and set the **Autorange** checkbox. Hit **OK**. Right click on the other top curve and use **Select axis...** to move it to the secondary y -axis. Notice that two of the curves (**Data3** and its log) almost overlap but click on them and watch the axis markers; you'll see that they are plotted on different scales and based on the axis ranges, they are quite different.

1.4.5 Annotating and Titling Graphs

Assuming we have all our data on the graph and are finished analyzing it, it's time to make the graph presentable to others. We start by creating a legend.

Legend Click **Options / Legend**. A legend box appears on the graph (§2.12). Depending on your **Legend Preferences**, the box should have one, three, or no entries. Right click on the legend and select **Legend Preferences**. The bottom two checkboxes tell EasyPlot whether or not to assign default legend titles to curves generated from equations or loaded from files. When you first install EasyPlot, **Put eqn in legend** is checked but **Use file & column #** is not. EasyPlot assigns default legends only when a curve is created; changing the legend preferences will not add or remove legend entries for existing curves. Turn on the two bottom **Legend Preference** checkboxes and click **OK**.


Detour — Saving a graph If **Data2** and **Data3** do not appear in the legend, here's a trick to get EasyPlot to assign default legend titles (now that we've turned the options on). Click **File / Save** and, in the **Save** dialog, make sure **Put data in file** is not checked. Save the graph. With **Put data in file** off, the save file contains links to the data files you loaded, in this case **Data2** and **Data3**. Other curves that


do not correspond to data files will be stored right in the save file. Save the graph and then open the save file (**File / Open** or **File / New**). Now **Data2** and **Data3** should appear in the legend.

To add or edit legend titles, right- or double-click on a curve and choose **Legend**. If the curve already appears in the legend, you can double click on its legend title. Right click on the transform curve, select **Legend**, and enter "Log of data3".

Rearranging Legend Entries Let's move **Data2** and **Data3** to the top of the legend. Click **Window / Curve toolbar**. On the right of the Curve Toolbar is a 3-part legend button. The arrows move a curve up or down in the legend. Click on **Data2**, on the graph or in the legend, and then click the up arrow until **Data2** is at the top of the legend. Select **Data3** and move it up and below **Data2**.

Annotations The statistics information we computed earlier on **Data2** was placed on the screen as an annotation. **<Alt>** click on the note and remove all but the first two lines. Click off the note or hit **<Ctrl><Enter>** when finished. If the text overlaps other graph objects, drag the note to an empty spot on the graph. Right click on the note, select **Properties...**, and set the color of the note to blue to match the **Data2** curve. The equation above the graph is also an annotation. Click and drag it into the top of the graph.

You can add any number of annotations to a graph (§2.7). Create a few new ones by clicking the annotation icon () on the help line, **Add / Annotation**, or by **<Alt>**clicking on an empty spot on the graph. When finished typing, hit **<Enter>** twice at the end of a note; or hit **<Ctrl><Enter>** if the text cursor is not at the end of the note; or just click off the note.

Lines and Arrows You can sketch lines and arrows on the graph, to point from a note to the curve it describes, for example. Click the line-draw icon () on the help line, or **Add / Line** (§2.9). The cursor turns into a pencil. Hold the left mouse button and move the cursor to draw a line. Hold **<Alt>** when pressing or releasing the mouse button to put an arrow head on the line. Draw a line on the screen from the statistics note to the **Data2** curve. Draw another line from the equation to the fit curve. Hit **<Esc>** to leave line-draw mode. Right click on the first line, select **Properties...**, and make the line blue. Repeat for the second line to match its color to the fit curve and the equation.

Titling the Graph and Axes The open boxes around the graph let you enter graph and axis titles (§2.6). Click on the boxes and enter a title or two.

1.4.6 Previewing and Printing

When finished laying out the graph, you can see what it will look like when printed. Pull down **File** and run **Print Preview** (§8.2). You can zoom in on the preview the same way you zoom in on data in “working” mode. In the previewer, however, you zoom the entire view rather than just the graph range. Zooming out also works the same; use the **Autorange** feature or right click on the preview and select **Fit to window**.

You can move objects on the graph and even edit titles and notes. Editing in the previewer is a little slower than in “working” mode; if performance is important to you, use the previewer for final touch-up work only. When ready to print, click **Print...** in the main menu; or right click and select **Print...** or **Print now**. The **Print...** buttons open the **Print** dialog and let you modify settings before printing. The **Print now** button and the print icon on the help line send the graph directly to the currently selected printer.

1.4.7 Multiple Graphs on a Page

Pull down **File** and choose **Close all**. EasyPlot creates a new, blank graph and asks for something to plot. Click **Graph a function...** Make sure the **Trig mode** is set to **Degrees** and enter “ $\sin(x)$ ”. Because there is no data on the graph and thus no defined x -range, EasyPlot prompts you for a range over which to compute the equation. Enter “0 1440”. Click **File / New** and, again, choose **Graph a function...** Before entering the function, click on the button labeled **150 Points...** (150 is EasyPlot’s default; your **Points** button may show a different number.) For this curve, we should compute more than 150 points. Enter 600. For the equation, type “ $\sin(x) + .2\sin(10x)$ ” and use the same x -range, 0 to 1440.

If you plot a lot of functions and use EasyPlot’s more Windows-standard menu (§7.8), use **Add / Function** instead of **File / Open**. Or, use the EasyPlot-standard **Open** dialog which switches automatically between file and equation modes based on what you type. You choose which menu and **Open** dialog to use in the **User Interface** topic of the **Preferences** dialog (**File / Preferences**).

We now have two graphs, one on top of the other. Hit **<Ctrl><Tab>** to cycle from one to the other. Pull down the **Window** menu and you’ll see the two graphs at the bottom of the menu with a check mark next to the top graph. Select **Tile** and choose a 1x2 layout. The bottom of the **Tile** dialog shows you the graph layout. You can use the **Gap** setting to add extra space between graphs or to bring graphs

closer together but for our purposes, leave the **Gap** at 0. Click **OK** and you should see both graphs.

You can link axes together so that a change in one is reflected in all linked axes. EasyPlot's "Axis Linking & Mirroring" feature (§2.14) is a handy tool for keeping both x - and y -axes identical. Right click on an x -axis and select **Mirror axis...** The cursor turns into a small cross-hair and, on the help line, EasyPlot prompts you to click on another axis. Click on the other x -axis. A small mirror marker appears below each axis near x_{\max} . The two axes are now linked but since they were already identical, neither graph changes.

Right click on the y -axis of the first graph (the one with the smooth sine curve) and mirror it to the y -axis on the other graph. This time, the first graph changes to take on the y -range of the other graph.

Double click on an x -axis and enter "1500" for the axis maximum. Click **OK** and both graphs display the new range. Double click on an x -axis and enter "180" for the distance between major tics (**Maj tic increment**). Again, the change is reflected on both graphs.

Double click on an x -axis. Turn on **Let EasyPlot Place tic marks** and click **OK**. Zoom in on one of the graphs. With the x - and y -axes linked, zooming on one graph also zooms the other.

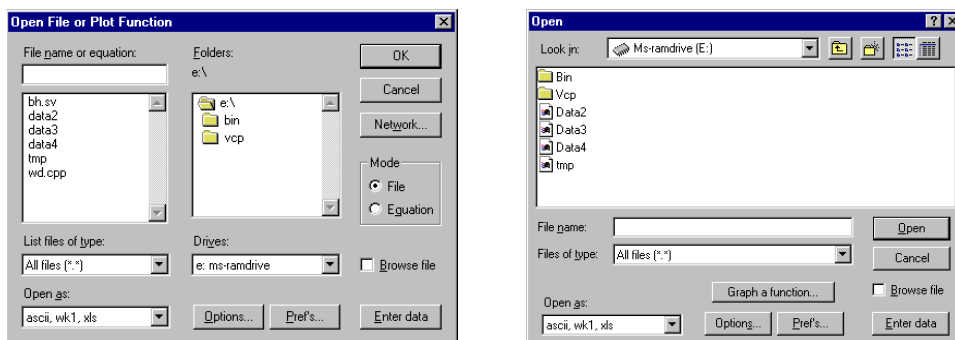
To break the link between axes, right click on an axis and choose **Break mirror**.

That concludes our tutorial. You should now be ready to use EasyPlot to create your own graphs. If you have questions that aren't answered by the manual or on-line help, feel free to contact Spiral Software or Cherwell Scientific by phone or e-mail. Enjoy the software!

Chapter 2: Making Graphs

2.1 Plotting Data

To plot a data file (ASCII, binary, or spreadsheet), an equation, or to enter data, pull down **File** and select **New** or **Open**; or click on the **New** (□) or **Open** (□) help-line icons. **New** creates a new graph on top of existing graphs; **Open** adds data to a graph. EasyPlot displays one of two **Open** dialogs:



Under Windows NT or 95, EasyPlot uses the Explorer-style **Open** dialog by default. Under Windows 3.1, EasyPlot uses the dialog on the left. The W3.1 **Open** dialog automatically switches to equation mode if you start typing a function. To plot a function with the Explorer-style dialog, click on the **Graph a function...** button. If you run NT or 95 and prefer the **Open** dialog that switches automatically between file- and equation-mode, go to the **User Interface** topic of the **Preferences** dialog, and select to use the EasyPlot-standard **Open** dialog.

From the **Open** dialog, you can read ASCII and binary data files, spreadsheet files (**.wk1**, **.xls**, or **.wq1**), an EasyPlot save file, or you can generate curves with equations. If you enter the name of a file that does not have one of the spreadsheet

extensions listed above, EasyPlot assumes it contains ASCII data and/or EasyPlot batch commands. (EasyPlot save files are ASCII files containing EasyPlot batch commands and possibly data.)

To read a binary data file, use the **Open as** dropdown to select the file format: 1-, 2-, or 4-byte integer, 1-, 2-, or 4-byte unsigned integer, and 4- or 8-byte float. After choosing a format, EasyPlot displays a list of file-reading options that lets you specify the number of data columns, the number of header bytes to skip before reading data, etc. See Section 2.1.5 for details on reading binary files.

By using the **Open** button repeatedly, you can overlay any number of data sets on a graph. To keep data sets distinct, EasyPlot supports seven dash patterns and fourteen data marks. You can also differentiate data by displaying them with different styles – one as a scatter plot and another as a line plot, for example. Section 2.2 discusses how to customize the appearance of your data.

2.1.1 Listing Specific File Types

By default, the **Open** dialog lists all files types. You can modify the default file types listed by entering custom file specifications in the **User Interface** topic of the **Preferences** dialog. You can enter several file specifications separated by a space or comma, for instance “*.dat, *.ezp”.

2.1.2 Browsing Files

To view and/or modify data before plotting, turn on the **Browse file** checkbox in the **Open** dialog. EasyPlot loads the file you select into its built-in Data Table. Once in the Data Table, click on **Plot** or hit **<alt>P** to plot the file. See Section 7.9.2 for more details on editing files. The Data Table lets you browse and edit ASCII and spreadsheet files; it does not load binary files. To load binary data, plot it and then **Edit** the **Data**.

2.1.3 Entering Data

To type data directly into EasyPlot, click on **Enter data** in the **Open** dialog or choose **File/Enter data**. You enter data into EasyPlot’s Data Table (§7.9.2). Move from cell to cell with the cursor-arrow keys or the mouse. After entering a number, hit **<space>** or **<tab>** to move to the next column. Hit **<enter>** to move down a row and back to the first column.

EasyPlot assumes the first data column is x and subsequent columns y (§3.4). Click on the definition at the top of a column to change its definition. Click on

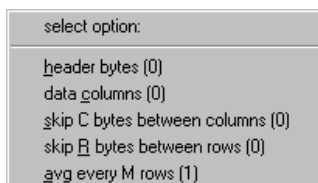
Plot or hit **<alt>P** to plot what you've entered. If **Close after plot** (under the Data Table **File** menu) is not checked, the table stays open after plotting data and you can continue entering or editing data. If the table closes, you can load data back into the table by pulling down EasyPlot's **Edit** menu and selecting **Data**. See Section 3.1 for details on how to format data files.

2.1.4 Plotting Data in the Clipboard

If you have copied data to the Clipboard in another application, you do not need to import the data into EasyPlot's Data Table before plotting. Simply **Paste** the data onto the graph (**^V** or **Edit/Paste**) and EasyPlot automatically converts the text data on the Clipboard to curves.

2.1.5 Reading Binary Data Files

To read binary data, click on **Open as** in the **Open** dialog and select the binary format that matches your data: 1, 2, 4, or 8-byte integer (signed or unsigned), or 4 or 8-byte floating point. EasyPlot then displays a file configuration menu:



```

select option:
header bytes (0)
data columns (0)
skip C bytes between columns (0)
skip R bytes between rows (0)
avg every M rows (1)
  
```

If a file has header information, use the **header bytes** button to tell EasyPlot how many bytes to skip before reading data. You can also use the header to skip over a section of data at the top of the file. If a file has 1000 pairs of 2-byte integers, for example, and you want to plot only the second half of the file, you can set the header to 2000 bytes (500 pairs * 4 bytes/pair).

The **data columns** specifies how many columns of data the file contains. A file with alternating *x* and *y* values, for example, has 2 columns. If the number of data columns is 0 (as it is by default), EasyPlot will not read any data from the file.

The next two options, the skip values, let you read every *n*th data row or only one or two columns out of a multiple-column file. The first skip value tells EasyPlot how many bytes to skip after reading a data point. The second specifies how many bytes to skip after reading a data row. To read every 10th row of a 6-column, 2-byte-integer file, for example, select **skip M bytes between rows** and enter 108 (12bytes/row * 9rows). If you want only the 2nd and 4th columns of the

file, add 2 bytes to the header (to skip the first data point), skip 2 bytes between columns (to step over columns 3 and 5), and add 4 to the between-row skip (to step over column 6 and the first column of the next row).

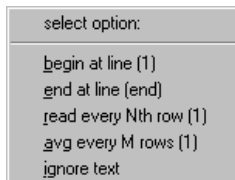
The last option lets you average every M rows to produce an average row, thereby reducing the number of data points without losing magnitude information. Averaging every 10 points, for example, turns a 100,000-point file into a 10,000-point file. If you set the between-row skip, EasyPlot averages only the rows that are read, not those that are skipped.

When finished setting the file-reading options, click outside the popup menu. If you need to go back to the options list, click on the **Options...** button.

2.1.6 ASCII File-Reading Options

When reading ASCII files, you can have EasyPlot start and stop reading the file at specific lines, read every N^{th} data row, and average every N data rows to produce M/N average rows, where M is the total number of rows read.

With the **Open** dialog set for reading ASCII files (in **Open as**), click **Options...** and EasyPlot displays the following options:



The **begin** and **end** options let you tell EasyPlot where in the file to begin looking for data and at which line to stop reading the file altogether. When skipping lines at the beginning of a file, EasyPlot ignores all data but will process batch commands. Once it reaches the “end” line (if one is set), EasyPlot stops reading the file and will not process any subsequent data or batch commands.

The **read every Nth row** option lets you take every N^{th} data row of a file. The **average every M rows** lets you reduce the number of data points by averaging every M rows together to produce one average row. The average feature takes into account only those rows which are read; if you read every N^{th} row, EasyPlot averages every N^{th} row until it gets M rows averaged.

The **ignore text** option tells EasyPlot to ignore all text that is not part of a batch command. By default, EasyPlot jumps ahead to the next line after finding any non-numeric character. With **ignore text** on, EasyPlot skips over text and continues scanning the line for data.

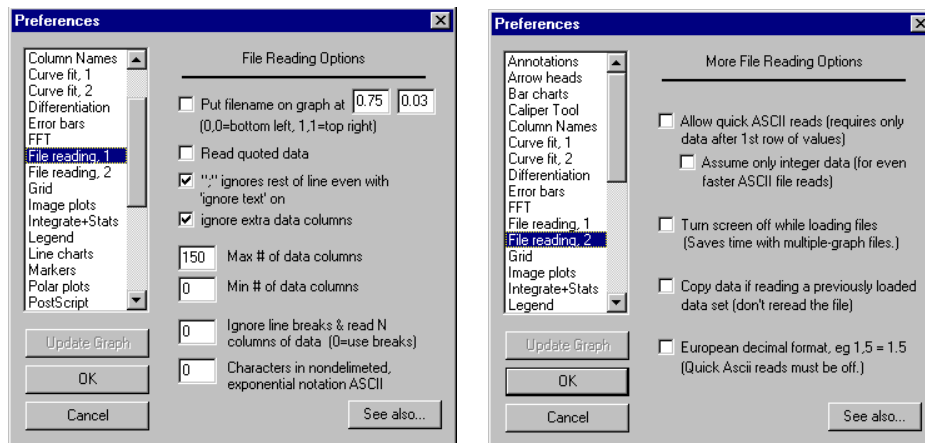


Figure 2.1 File reading Preferences

2.1.7 File-Reading Preferences

There are two **File reading** topics in the **Preferences** dialog (Figure 2.1) that provide several options for customizing the way EasyPlot reads files.

Put filename on graph: You can put the name of each file on the graph by turning on **Show filename**. The name appears as an annotation at the coordinates specified below the **Show** checkbox. If you read several files, the names appear on top of one another unless you move them. You can move, edit and delete the filenames as you would any annotation.

Read quoted data: By default, EasyPlot does not read quoted numbers as data. If files have columns of quoted data, turn on **Read quoted data**.

“;” ignores rest of line: With ASCII files, EasyPlot ignores the remainder of a line upon finding a non-numeric character. If you turn “ignore text” on (§2.1.6), the only way to comment out portions of a file is to turn on **“;” ignores rest of line** and put semicolons in front of lines you want to ignore.

Ignore extra data columns: EasyPlot determines the number of columns in ASCII files by counting how many numbers are on the first data row. If it finds N numbers, it ignores rows that do not have exactly N numbers. If some of the rows have extra values after the N columns, you can ignore the extra columns rather than the entire row by turning on **ignore extra data columns**.

Max # of data columns: By default, EasyPlot reads up to 150 data columns. If you read files with more than 150 columns, increase the number in the **Max # of**

data columns edit box. The absolute maximum is 1024. Larger values require more memory for reading files.

Min # of data columns: When reading ASCII data files, EasyPlot determines the number of columns by the first row of data it finds. If the first data row is missing one or more values and you don't want to modify the files to insert missing value place-holders (`//m -- §6.3`), you can tell EasyPlot the minimum number of data columns to accept. EasyPlot will ignore data until it finds the first row with at least the specified minimum number of columns.

Ignore line breaks: Rows of ASCII data are usually delimited by line breaks. Otherwise, you can tell EasyPlot how many columns of data to read and it will ignore line breaks altogether. With XY pairs strung over a single line, for example, enter 2 in the **ignore line breaks** edit box. EasyPlot scans for two data points, installs them as a data row, looks for the next two points, and so on. You need to enter 0 to go back to using line breaks. The batch command for turning this feature on is `"//columns N"`, where *N* is the number of columns. After reading the data, you should reset **ignore line breaks** back to zero.

Fixed exponential notation: If a file contains data in fixed-length exponential notation, some values may not have a space or other character separating them from their neighbors. To read such a file, enter the length of the data values in the **fixed exponential notation** edit box.

Allow Quick ASCII reads: If you read large ASCII data files, EasyPlot will load them much quicker with this option on. Files can contain header comments and EasyPlot batch files but only before the data. The quick ASCII read does not support missing values or some other file-reading options, such as European decimal format. If you are having trouble loading data and **Quick ASCII read** is on, try turning it off. The quick ASCII read can save a significant amount of time with very large data files (tens of thousands of points). With smaller files, the difference can be insignificant and we recommend you leave this option off.

Assume only integer data: The **Quick ASCII read** can load data about twice as fast if the values are all integers. If values do have fractional components and this option is set, the data will not load correctly.

Turn screen off while loading files: EasyPlot draws graphs as it reads files so that you see the graphs you're loading. If your save files contain many graphs, you can save time by not updating the screen until EasyPlot finishes reading the file.

Copy data if reading a previously loaded data set: When plotting different columns of the same data set on separate graphs, EasyPlot does not let you share

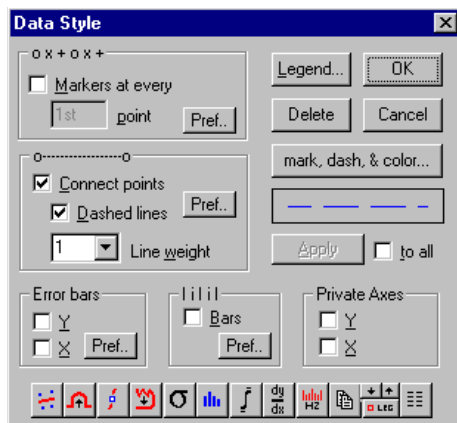


Figure 2.2 The Data Style dialog

data from one graph to the next. Rereading can be time consuming but you can save time by allowing EasyPlot to make a copy of the data it has already loaded instead of rereading the file. If the file contains EasyPlot batch commands, they will not be executed for the copied data.

European decimal format: Normally, EasyPlot treats commas as column delimiters. The **European decimal format** option tells EasyPlot to assume commas separate the integer from the decimal part of a number. With this option off, EasyPlot reads ‘2,4’ as two values, ‘2’ and ‘4’. With it on, it reads ‘2,4’ as ‘2.4’.

2.2 Styling the Data

You can display data in a variety of formats, such as scatter, line, or bar. EasyPlot provides a list of style attributes which you can turn on or off for the entire graph or for individual curves. Pull down the **Style** menu and you see the style options:



New curves inherit the styles that are on at the time. Switching styles on and off does not affect data on the screen. To change the appearance of data, use the **Restyle** button (under **Style**) or double click on the data and turn style attributes on or off in the **Data Style** dialog (Figure 2.2 – §2.4).

EasyPlot starts with **Connect Pts**, **Mark Pts**, and **Dash** as defaults styles. You can use EasyPlot batch commands to change the defaults (§6.4) or to toggle styles on and off from within a data file (§6.1).

2.2.1 Selecting the Data Mark, Color, Dash, & Fill

EasyPlot assigns an attribute set to each curve which determines the curve's color, data mark, dash pattern, and bar fill pattern. To change the attribute set of a curve, double-click on the curve and select **mark, dash & color...** in the **Data Style** dialog. EasyPlot lets you pick one of the 14 predefined attribute sets or create a custom set. Curves can share the same attribute set.

EasyPlot supports 14 geometric data markers, 14 bar fill patterns, 7 curve colors, and 7 dash patterns. Attribute sets 1-7 share the same colors and dash patterns as sets 8-14 but have unique data marks and bar fill patterns.

In addition to the geometric markers, you can mark data with any ASCII character or its data values. Click on **mark, dash & color...** and choose the **custom** option. EasyPlot prompts you to select a marker (assuming markers are on). To use a character, pick **letter** and enter the letter. To use data values, pick **X-coor** or **Y-coor**. If the data has a column of z -values, you can choose **Z-coor**.

Another method for changing the attribute set of a curve is to hold the right mouse button (be sure the cursor is not on any graph object) or space bar so that the curve glimmers (§1.3) and hit **c** for "color". EasyPlot assigns the next attribute set to the curve; if it had set 5, for example, it gets 6. Repeat until you find a set you like. At set 14, EasyPlot cycles back to 1.

You can specify an attribute set for a curve in a data file with the command:

/sa m n [c]

where n is a number from 1 to 14 and c is an optional column number. Place the command before the data it is to describe. If you don't specify the column number, EasyPlot assigns the attribute set to column 2.

2.2.2 Data Marks

EasyPlot can mark data with any of the 14 marks in Figure 2.4, any ASCII character, or the x -, y -, or z -value of each point. Section 2.2.1 discusses how to select a particular data mark.

If a data set has many points, you can mark at every N^{th} point. Double click on a curve with markers and enter a number in the **at every** **point** edit

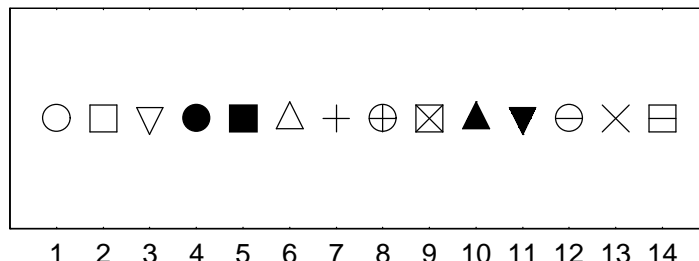


Figure 2.3 The 14 Data Marks

box. If you enter a 2, EasyPlot places a mark at every other point. If you enter 10, it marks every 10th point.

On contour maps, the **mark pts** style creates a color map or image plot. Each data point is marked with a color or gray scale representing its height. You can select color or gray-scale images in **Image plots Preferences**.

2.2.3 Marker Preferences

Mark size: Marker size affects all curves on all graphs. Enter 2 to double marker size, 0.5 to half their size, or 1 to use the default size. EasyPlot bases the default mark size on the size of the graph. If you use 0 as the mark size, EasyPlot places a dot at each data point. On screen, EasyPlot does not draw markers smaller than 3 pixels so that you can distinguish the mark shapes. When you print, they will be scaled exactly as you specify, down to a dot if the size you enter is small enough.

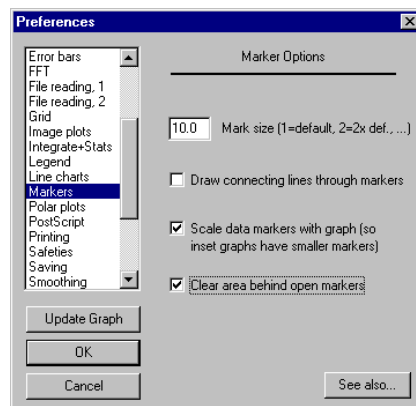


Figure 2.4 Marker Preferences

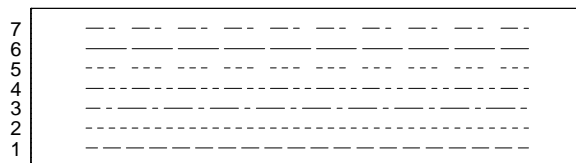
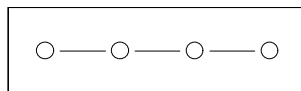
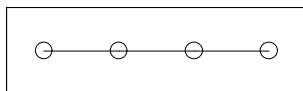


Figure 2.5 The 7 Dash Patterns

Draw connecting lines through markers: If a curve has connecting lines and markers, you can draw lines through markers or break them to not hit the marks:

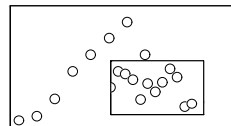
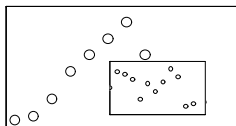
**Draw
through**



**Miss
marks**

Scale data marks with graph: With different sized graphs on the same page, you can have EasyPlot scale markers so that small, inset graphs have smaller markers than larger graphs. When not scaled, all graphs have the same size markers:

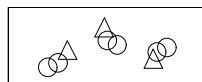
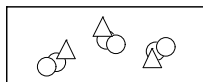
**Scale
marks**



**Don't
scale**

Clear area behind open markers: EasyPlot can draw open markers as truly open or fill them with the background color. When open markers overlap, clearing can make the graph look neater.

Clear



**Leave
open**

2.2.4 Connecting Lines

If a curve is drawn with connecting lines, EasyPlot draws a straight line from each point to the next. Connecting lines are drawn dashed if the **dash** style is on and solid if **dash** is off. EasyPlot supports seven dash patterns (Figure 2.4). EasyPlot assigns a color and dash pattern to each curve. Section 2.2.1 discusses how to select the color and dash pattern used for a curve.

On curves with connecting lines and data markers, you can pass the lines through the data marks or break them so they do not hit the marks (§2.2.3).

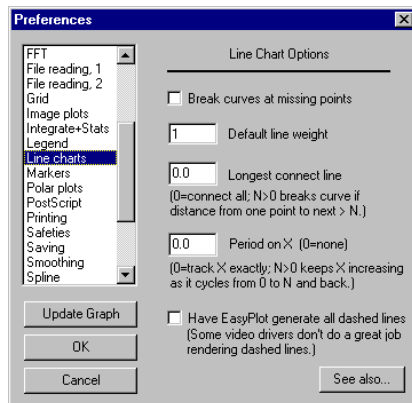


Figure 2.6 Line Chart Preferences

2.2.5 Line chart Preferences

The **Preferences** dialog (under **File**) has a **Line charts** topic that lets you set the default line weight for connecting lines. You can also choose to break curves at missing data points (otherwise EasyPlot connects the last non-missing point to the next non-missing point).

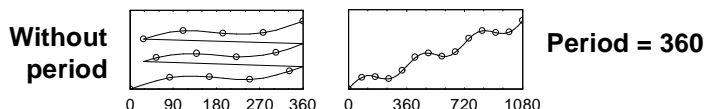
Break curves at missing points: If a curve has missing values (§6.3), EasyPlot can break the curve at missing values or keep the connecting lines contiguous by ignoring missing points.

Default line weight: Normally, EasyPlot assigns a line weight of 1 to curves, meaning that the curves are drawn with the same line weight as the axes and tic marks. You can increase the line weight for individual curves by double clicking on the curve. If you want all curves to get a heavier line weight by default, enter the desired number in the **Default line weight** edit box.

Longest connecting line: This option lets you highlight discontinuities in data. Enter the longest line you want EasyPlot to draw and if two points are further apart than the specified length, a break appears in the curve. Specify the distance in graph coordinates. Enter '0' to connect all points.

Period on X: If data is periodic on X, for instance 0,1, 2,... 360,0, 1, 2,..., the curve will trace back and forth across the graph. If you want the X-values to appear as though they are sequential in time, enter a period value for X. EasyPlot

adds the value to a base X-coordinate each time the data cycles backwards.



Have EasyPlot generate all dashed lines: Maintaining a consistent dash pattern along curves consisting of many short line segments is a tricky procedure. EasyPlot generates all dashed curves when printing, but on screen, it lets the Windows video driver do the drawing (assuming it's a single-pixel width line; Windows doesn't support multiple-pixel-width dashed lines). Many video drivers do not do a very good job rendering complicated dashed lines. If you see strange-looking dash patterns, turn on this option and EasyPlot will draw all dashed lines.

2.2.6 Error Bars

For any curve on an xy plot, EasyPlot can draw x -error bars, y -error bars, or both. To create error bars, you must place error values in the data file along with the x - and y -values and you must tell EasyPlot which column or columns are error values. You can then turn x - and/or y -error bars on and read in or restyle a data set.

Error Values in Data Files

In the data file, error values occupy columns. With one error column, you can draw either x - or y -error bars. With two error columns, you can draw x - and y -error bars or asymmetric error bars. For x - and y -error bars, the first, or left-most error column maps to the independent axis (usually x) and the second maps to the dependent axis. For asymmetric error bars, the first error column represents the 'up' error and the second column the 'down'.

The **Error bars** topic in the **Preferences** dialog controls how EasyPlot interprets error values. The upper two radio buttons determine whether error values specify the actual size of error bars or whether they are fractional percentages, and thus are multiplied by the x - or y -values to obtain the size of the error bars. The bottom two radio buttons tell EasyPlot whether with one error column, error values specify the entire up-and-down size of error bars, or whether they specify only half the total bar size (just the up or down height).

Normally, each y -column has its own column of error values and you would arrange the columns as **XYEYEE...** You can apply one error column to several y -columns by placing the error between the x - and first y -column, as in **XEYYY**. An example data file which includes error information is listed in Figure 2.7.

```

This file results in two curves,
each with a column for making error bars.
/td      xyeye      ;define the columns

  X      Y1      error      Y2      error
10      125      0.020      158      0.011
20      141      0.013      176      0.009
30      137      0.044      172      0.005
.
.
.

```

Figure 2.7 A data file with error values

Defining the Error Columns

You need to tell EasyPlot which columns of a data file are error values. Otherwise, EasyPlot assumes they are y-values and plots curves with them. Read in a data file that includes error information. Pull down **Tools** and select **Define Data**. EasyPlot lets you edit the “column definition string” associated with the data file. This string has one character for each column of data; the first character in the string defines the first column of data, the second character defines the second column, and so on. To define error columns, place an **e** in the string positions that correspond to error values. The **e** stands for “error”. For the example in Figure 2.7, the definition string is set to **xyeye** to define the first column as x, the second and fourth as y, and the third and fifth as error values. See Section 3.4 for more details on defining columns.

You can also define columns in the Data Table. In the **Open** dialog, turn on **browse file** and load the data file. Or, plot the file and select **Edit/Data** to load it into the Data Table. At the top of each data column, you see its definition in a small box. Click on the box to change the definition or double click to ignore the column. Select **Plot** to apply changes to the graph.

If you generate your data files, you can define the columns by placing a batch command (§3.4) in the file, as illustrated in line 3 of Figure 2.7. That way, the columns will be defined properly every time you read the file into EasyPlot.

When EasyPlot first reads a file, it computes the data range for each column. If error values are predefined, EasyPlot takes the error into account in setting data ranges. Then, if you plot error bars, the autorange feature assures that data and the

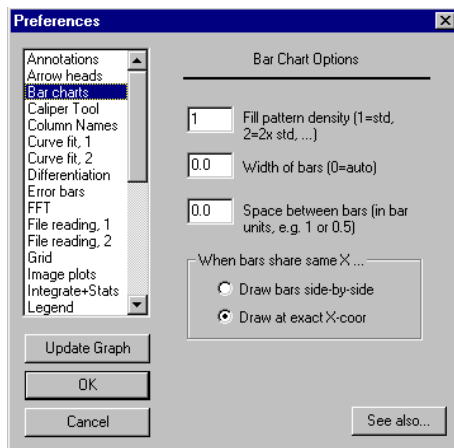


Figure 2.8 Bar chart Preferences

error bars fit in the graph. If you define error columns after plotting, you may have to stretch the range yourself to keep error bars from extending outside the graph.

Turning On Error Bars

If you've read in data and defined error columns before turning the **Error bar** style on, double-click on the curve and turn on error bars in the **Data Style** dialog. Otherwise, pull down **Style**, select **Error Bars**, and choose an axis. Read in the data. If the file contains a batch command to define the error columns (see Figure 2.7), you should see error bars on the graph. Otherwise, define error columns as described above and you should then see error bars.

2.2.7 Three D Style

The **Three D** style feature allows you to view data in three dimensions. You can plot data as 3D fishnet surfaces or as *xyz* triplets as a 3D scatter plot. You rotate the data by clicking on axis rotation buttons that appear in 3D mode. EasyPlot's 3D feature is very fast and interactive, making it a powerful tool for visualizing 3D data. See Chapter 5 for details on working with 3D graphs.

2.2.8 The Bar Style

If a curve is drawn with the **Bar** style, EasyPlot draws a bar from each point to the *x*-axis. EasyPlot assigns a default fill pattern to each curve but you can select any of EasyPlot's 14 fill patterns (§2.2.1).

The width of the bars depends on the number of data points, the number of curves using the bar style, and a “bar spacing” value which you can adjust. If the bar spacing is 0, EasyPlot draws bars as wide as it can without overlapping them. If more than one data set is drawn with bars, EasyPlot makes the bars narrower so that bars can be displayed side-by-side without overlapping. Bar width is calculated as $n / (s + n)$, where n is the number of data sets displayed with bars and s is the bar spacing. To set the bar spacing or customize the width of the bars, go to the **Bar charts** topic of the **Preferences** dialog (Figure 2.8).

If you use a large value for the **Space between bars**, bars become very thin. You can force EasyPlot to draw a line, or a needle, instead of a bar by using a value greater than or equal to 100.

If two data sets share the same x -values, you can display bars side-by-side or at their exact x -coordinates (and thus on top of each other). Use the two radio buttons labelled **side-by-side** or **at exact X coord** to choose.

If a data set has two x -values that are very close together, for example 1, 2, 2.01, 3,..., the bars will be very thin so that the 2 and 2.01 don't overlap. You can force EasyPlot to draw wider bars by entering a **Width of bars** value. Specify the width in x -axis units. Enter 0 to have EasyPlot compute the bar width.

You can adjust the darkness of the fill-patterns EasyPlot uses to distinguish bars of different data sets. For **Bar fill density**, enter a line-density scale value, where 1 is the default density, 2 doubles the line density, etc.

If you turn bars on for a polar plot, EasyPlot draws a line from each point to the center of the circle, producing what some people call a “wind rose”.

2.3 Font and Font Size

The **Font** dialog (Figure 2.9) lets you select the font and text sizes used for graphs. The typeface you choose is used for all text on all graphs but you can make any text bold, italic, underlined, and any size independent of the rest of the text. To open the **Font** dialog, go to **Style / Font**. You can also set point sizes in the **File / Page Setup** dialog.

There are five types of text you can put on graphs: tic-mark labels, axis titles, graph titles, annotations, and legend text. EasyPlot draws tic-mark labels at the base point size. You can set the size for each of the other types. You can override the defaults to make any text (or piece thereof) larger or smaller. Any annotation or part of a title, for instance, can be made larger or smaller than the default by using the point-size button in the Text Toolbar (§2.8).

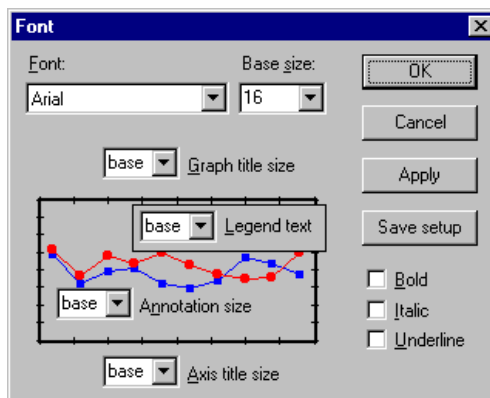


Figure 2.9 The Font dialog

In “working mode” on screen, EasyPlot does not render fonts in their true, physical size relative to the graph so that it can use as much of the screen as possible for displaying data. At 640x480 resolution, EasyPlot displays 12-point text with characters that are 12-pixels high, 20-point text with 20-pixel-high characters, and so on. With higher resolution screens, it scales the height so that characters appear about the same size as on a standard VGA screen. When you print or preview, text is scaled to its true point size.

EasyPlot can scale printed text based on the graph size so that small, inset graphs use smaller text than larger graphs. On a graph that takes up half the window, for example, 20-point text would print as 10-point text. You can turn text scaling on or off in **Printing Preferences**. You can also choose whether text is scaled relative to the largest graph on the page or relative to the full EasyPlot window. With two equal-sized graphs side-by-side, 12-point text will print at 12-points if text scaling is off or scaled relative to the largest graph; otherwise text will appear smaller than the set size when printed.

2.4 Data Style Dialog

EasyPlot’s **Data Style** dialog (Figure 2.2) lets you set up a data set’s appearance. Double click on a curve, or right click and select **Properties...** If a graph has a legend, you can also double click on a data segment in the legend.

If markers are on, you can set marker frequency in the **at every** **point** box (§2.2.2). If points are connected, you can set the weight, or thickness, of connecting lines in the **line weight** dropdown. You can use line-weight to differ-

entiate data sets. The **mark, dash, & color...** button (§2.2.1) lets you choose the data marker, dash pattern, bar fill pattern, and the color used to display a curve.

The **Prefs...** buttons jump you to **Preferences** topics for data markers, line charts, error bars, etc.

As you change styles, EasyPlot updates the curve-viewer button on the right side of the dialog. Click on **Apply** to see the new styles reflected on the graph. If you update the graph and don't like the new appearance, hit **Cancel**.

You can restyle all curves at once by turning on the **to all** checkbox next to the **Apply** button. The curves will not look alike because they retain their individual mark, dash, and color attributes but they will all be drawn with the same styles.

You can switch from one curve to another without closing the **Data Style** dialog. Click on the curve-viewer button (the one with a piece of the current curve displayed inside it) and select a curve from the popup menu. If you've made changes, EasyPlot applies them before switching curves.

You can enter or edit a curve's legend title by clicking on the **legend...** button (§2.12). Click on **Delete** to delete the selected curve.

The **Data Style** dialog has a toolbar along the bottom that lets you perform a variety of math and other operations on a curve. The buttons are identical to the Curve Toolbar (§4.5).

2.5 Polar Plots

For polar plots, enter data as radius and angle values and define the columns as r and t (t stands for theta, or angle). EasyPlot determines the graph type (xy or polar) by how the column are defined; if columns are defined with x s and y s, EasyPlot draws an xy plot. If r s and t s, it creates a polar plot. Define columns in the Data Table or with **Define Data** under **Tools** (§3.4).

You can label angles around the plot by turning on **Label angles around plot** in the **Polar plots** topic of the **Preferences** dialog. The angular labels appear at intervals specified by the **Angle between radial grid lines** (also in **Polar plots Preferences**). Specify the angle in degrees. EasyPlot labels angles in degrees or radians depending on which trig mode is selected in the **Open** dialog.

You can plot the entire circle, a semicircle, or a single quadrant. To set the angular range, you must be using the EasyPlot-standard menu (in **User Interface Preferences**). Pull down **Edit**, choose **Move**, and select x - or y -axis. See Section 2.11.6 for more details on setting angular range.

The angle origin ($\theta = 0$) faces east by default. If you want the origin at the top of the circle or at any other location, go to **Polar plots Preferences** and enter the **Angle origin** in degrees; 90 puts the origin due north, 180 west, or 270 south.

The angle can increase clockwise or counter-clockwise (the default). Set the angle direction in **Polar plots Preferences**, with the **Increase angle clockwise** checkbox. You can also plot only positive angles by setting the **Plot absolute value of angles** checkbox.


You can select data styles for polar plot data just as you would for xy curves. The **Bar** style on polar plots draws a line from each data point to the center of the circle, creating what in atmospheric studies is called a “wind rose”.

The x - and y -axes on polar plots are labeled identically. Turn major tics off (§2.11.3) for one axis to make the plot appear less cluttered. To change the radial range or location of tic marks, double click on the x -axis; the y -axis mirrors the x .

2.6 Graph and Axis Titles

To place titles on graphs and axes, click on the small boxes above, to the left, and below the graph. After you click, the box disappears, the Text Toolbar (§2.8) pops up, and you can enter text. You enter graph and x -axis titles right on the graph. Because the y -axis title is rotated, EasyPlot opens a window for editing the y -axis title. When done, hit **<enter>** twice at the end of the text, **^<enter>** with the cursor anywhere in the text, or click off the text. Once you’ve entered a title, click on it to edit. You can set the point size of titles independently from the rest of the text on a graph (§2.3). To title a second x -axis, add extra lines to the graph title.

2.7 Annotations

You can type annotations anywhere on a graph. Pull down **Add** and select **Annotation**; or click the text icon () on the help line. Or **<alt>**click where you want to put a note. When done entering text, hit **<enter>** twice with the cursor at the end of the text, **^<enter>** with the cursor anywhere in the text, or click the mouse off the text. To leave a blank line, type a space before hitting **<enter>**.

If a note is not exactly where you want it, click and drag it to a new position. You can edit an existing note by double- or **<alt>**-clicking on it. To delete a note, right click on it and select **Delete** from the popup menu. You can also click on the note and, while holding the left mouse button, hit **** or **<backspace>**. If you click and release the mouse button without moving, the note remains selected and **** or **<backspace>** deletes it.

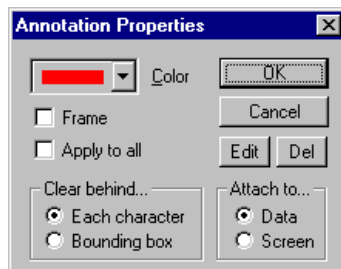
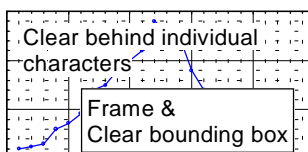


Figure 2.10 The Annotation Properties dialog

You can rotate annotations to any angle. Click on a note and, if you don't move the note before releasing the mouse button, a selection box appears with arrows on each corner. Click and drag any corner to rotate the note. Hold **<shift>** while rotating to move in 10 degree increments. To return a note to its default orientation, right click on it and select **Clear rotation**. Depending on the font you use and the font support in your version of Windows, rotated text may or may not appear in exactly the same typeface as unrotated text.

You can set the color and other details of how a note is displayed in the **Annotation Properties** dialog (Figure 2.10). Right click on a note and select **Properties...** The dialog appears next to the annotation. You can set defaults for the properties in **Annotation Preferences**.

You can draw a frame around a note. Turn the frame on or off in the **Annotation Properties** dialog or by **<ctrl>** clicking on the note. You can also choose to clear behind the entire bounding box or behind individual characters.



You can place any number of annotations on a graph and each note can have up to 800 characters. Annotations and other text can include Greek characters, math symbols, superscripts, subscripts, and bold, italic, and underlined text (§2.8).

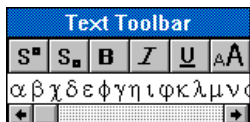
Annotations placed inside a graph can be attached to a data coordinate or to a fixed screen location. (A note is considered inside or outside the graph based on the location of its first character.) Notes attached to data move with the data when axis ranges change. New annotations inherit the default specified in **Annotation Preferences**. You can change the “attach” setting for existing notes in the **Annotation Properties** dialog (Figure 2.10).

Annotations are attached to a specific graph, whether inside a graph box or not. With multiple graphs on a page, EasyPlot assigns notes to the graph closest to where you begin the note. If you delete or minimize a graph, notes belonging to the graph go with it. You can drag a note from one graph onto another but it remains attached to its original graph and may be hidden if you make the graph onto which it was moved the current, or top graph. While you cannot assign a note to a different graph, you can copy it to another graph. Right click on the note, select **Copy**, click on another graph, and Paste.

When saving graphs, EasyPlot can store annotation positions in graph or screen coordinates. When creating templates from save files, save annotations in screen coordinates. That way, if axis ranges change to accommodate new data, annotations appear where they were saved. You can set this option in **Saving Preferences** with the **Save annotations in screen coordinates** checkbox.

2.8 Text Toolbar

When entering or editing graph text, the **Text Toolbar** makes formatting mathematical text easy. Use it to create superscripts and subscripts, bold, italic, and underlined text, set point sizes, and insert Greek, math, and European characters.



The first two buttons on the left create superscripts and subscripts. The button labeled **B** turns bold text on or off. The **I** italicizes and the **U** underlines text. The right-most button, labeled **A A**, changes font sizes.

Click on **S²** to enter a superscript or, **S₃** to enter a subscript. The cursor moves up or down and the characters you type become part of the super- or subscript. If you click on the superscript button again, you create a superscript within a superscript. To leave the superscript and go back to typing normal text, hit **<return>** and the cursor moves just to the right of the superscript. You can also move the cursor out of the superscript using cursor-arrow keys or with a mouse click. (If you click, be sure to click inside the text block, where the cursor is an I-beam, otherwise you end the edit session).

If you highlight a block of text and click on the super- or subscript button, EasyPlot moves the block of text up or down. There is no button to unsuperscript text. You can remove the super- or subscript codes, however. Move the cursor to

the beginning or end of the superscript and use **<backspace>** or **** to delete one of the control-code characters ('^', '_', '{', or '}') surrounding the superscript. Use the help line display (described below) to aid in positioning the cursor before or after one of the control characters. When you delete one of the control characters, EasyPlot removes the others automatically.

The bold, italic, and underline buttons toggle styles on and off. If the cursor is before the 'a' in the line "How are you?" and you hit the bold button, the text to the right of the cursor becomes bold. If you hit the button again, EasyPlot unbolds the text. If you highlight a block of text and select bold, italic, or underline, EasyPlot turns the style on for the entire block if any character within the block does not have the style; it turns the style off if the block already has the style.

As with super- and subscripts, there is no "plain" button that removes all bold, italic, and underline styles. To remove text styles, highlight the block of text and click on style buttons in the Toolbar once or twice until the text appears without the styles. Or you can delete the control codes that turn the styles on by positioning the cursor before or after the code and hitting **** or **<backspace>**.

The Text Toolbar buttons insert and remove formatting codes in the text. While you edit, EasyPlot maintains a WordPerfect-like "reveal-codes" display on the help line. Actual text is shown in black, formatting codes in gray, and the cursor is displayed as a solid pink rectangle. (The numeric character position of the cursor is displayed in pink surrounded by pink parentheses.) Don't try to click on the help line to move the cursor; clicking there will take you out of edit mode.

The help line is useful for positioning the cursor relative to formatting codes. If the pink rectangle cursor is inside the curly braces of a superscript, for example, any characters you type will be part of the superscript. Or, if you want to delete the code for turning bold text on, the help-line display shows you whether the cursor is before or after the code.


To insert Greek, math, European, or other special characters in text, use the Text Toolbar's scrollable Symbol Bar. The Symbol Bar contains the entire Symbol font in a slightly rearranged order (lowercase Greek letters followed by uppercase Greek and the extended characters) and ends with the extended ASCII characters of the current graph font. Click on a character to insert it in your text.

EasyPlot uses standard ASCII characters in the text to control formatting. For example, the caret ('^'), underscore ('_'), and curly braces ('{' and '}') denote superscripts and subscripts, and the backslash denotes a Greek character. The codes used to turn bold, italic, and underlining on and off, and to change point sizes are listed in Table 2.1. ASCII character codes let you enter formatted text in batch mode.


\Bo/	turn bold style on
\bo/	turn bold style off
\It/	turn italics on
\it/	turn italics off
\UI/	turn underlining on
\ul/	turn underlining off
\ptN/	change point size; <i>N</i> is new size, as in \pt20/ or \pt8/

Table 2.1 Text formatting codes

2.9 Lines and Arrows

When annotating a graph, you can draw lines and arrows to point from a note to a specific curve or data point. You can draw lines and arrows freehand on the graph. Pull down **Add** and select **Line**. Or click the line-draw icon () on the help line. The cursor turns into a small pencil. Click and drag to draw a line. Hit ****, **<esc>**, or **<space>** while drawing to cancel the line and start over.

To place an arrowhead on a line, hold **<alt>** while pressing or releasing the mouse button. If **<alt>** is down at the start, EasyPlot places an arrowhead at the first point selected. If **<alt>** is down when the line is finished, an arrowhead appears where you release the button. You can customize the size, shape, and fill of arrowheads in **Arrow head Preferences**.

If you draw a less-than-perfect line, you can delete or move it. Hit **<esc>** or click the select icon () on the help line to leave line-draw mode. Click and drag an endpoint to move just one end of the line. Click and drag on the body of the line to move both endpoints. You can hit **** or **<backspace>** while moving a line to delete it. Hit **<space>** or **<esc>** to cancel the operation.

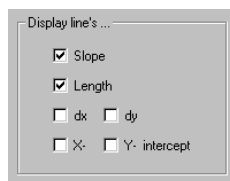
Lines and arrows are attached to data coordinates. EasyPlot moves and rescales lines if the scale or range of an axis changes.

The line-drawing function has a built-in caliper feature. If you want to know the distance or slope between two points, draw a line and, before releasing the mouse button, hit **m**, **s** (for “slope”) or **l** (for “length”). EasyPlot places an



Figure 2.11 The Line/Arrow Properties dialog

annotation at the midpoint of the line displaying information about the line. In **Caliper Tool Preferences**, you can choose what information is displayed:



If you reshape the line, EasyPlot updates the information automatically. You can edit, move or delete caliper information as you would any annotation.

You can set the line weight, color, dash pattern, and other properties of a line in the **Line/Arrow Properties** dialog (Figure 2.11). Right click on a line and select **Properties...** The **Line/Arrow Properties** dialog also lets you turn caliper information on or off and set which end or ends of a line have arrow heads.

2.10 Grid Lines

To draw a grid on the graph, pull down **Options** and select **Grid**. Or you can right click on an axis and select **Grid**. Dotted lines create a grid on the graph. You can draw grid lines at major tics, major and minor tics, or at manually added “floating” tics. If you want grid lines only in one direction, you can turn them off for one axis while leaving them on for the other. The **Grid Style** dialog (Figure 2.12) lets you select which tic marks get grid lines.

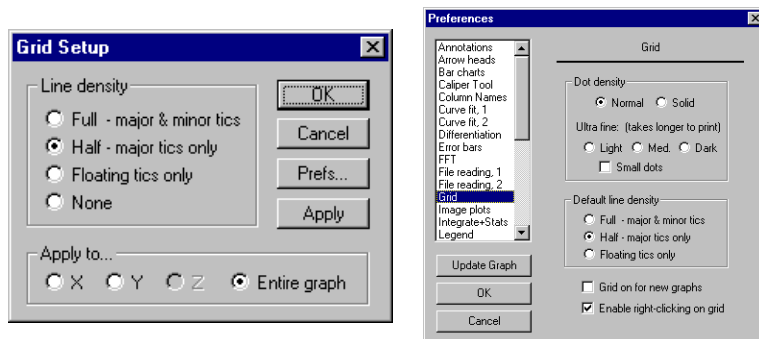


Figure 2.12 Grid Setup and Grid Preferences dialogs

In **Grid Preferences** (Figure 2.12), you can customize the default grid setup (full, half, or floating-tics only), and specify whether the grid is on or off for new graphs. It also gives you several dot densities options, from thin, solid lines to ultra-fine dotted lines. With the dot density set at normal or solid, EasyPlot lets Windows draw the grid lines. With any of the ultra-fine settings, EasyPlot generates the dot pattern itself; each dot is a separate line and, as a result, screen redraws and printing can be slower with many ultra-fine grid lines.

On polar plots, the grid consists of concentric circles around the origin and radial lines extending outward from the origin. If the x -axis is configured with a full grid, a dotted circle is drawn at every tic mark. If the grid style is half, a circle is drawn only at major tic marks. EasyPlot draws radial lines at fixed increments which you can set in the **Polar plots** topic of the **Preferences** dialog. The default increment is 30° . The value must be in degrees.

2.11 Axis Setup Dialog

When you double click on an axis, EasyPlot displays the **Axis Setup** dialog (Figure 2.13). The **Axis Setup** dialog lets you set ranges, tic-mark placement, linear or log scale, and more.

You can enter simple expressions in any of the edit boxes that take a graph coordinate, such as $\text{PI} * 4$ (which evaluates to $3.14159 * 4 = 12.56637$).

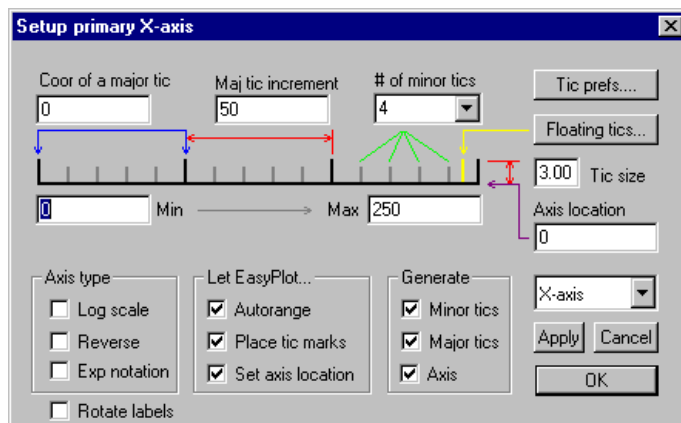


Figure 2.13 The Axis Setup dialog

2.11.1 Setting the Range

Axes are autoranged by default; an autoranged axis includes all data and the first major tic at or beyond the data. Because EasyPlot tries to put a major tic at each end of an axis, the range on an autoranged axis can change if you change the position of tic marks.

To customize an axis range, double click on the axis and enter values in the **Min** and **Max** edit boxes. You can enter simple expressions for **Min** or **Max**, such as $\pi*4$, or $360*3$. Once you enter a custom range, EasyPlot turns off the **Autorange** flag (under **Let EasyPlot...**). The range you set remains fixed until you change it or turn **Autorange** back on.

With the more EasyPlot-standard menu (§7.8), there is a **Range** button under **Options**. Choose an axis and enter two numbers separated by a space or a comma. If you enter one number, EasyPlot sets whichever end is closer to the number. Entering 240 with the range $0 \leq x \leq 250$ sets $x_{\max} = 240$. If you want to set an end to be very close to the other end, you can use a comma to signify the missing number. With $0 \leq x \leq 250$, “240,” sets $x_{\min} = 240$. Likewise, “,10” sets $x_{\max} = 10$.

If you do not care about the exact endpoints of an axis but want to expand your view of a portion of the graph, use EasyPlot’s zoom feature (§2.15). To undo any range setting, use **Undo** (under **Edit**) or the **Autorange** button (§2.17).

To set ranges on polar plots, you must use the EasyPlot-standard menu. The **Range** button (under **Options**) lets you specify the radial range. To adjust the angular range, you **Move** the axes (**Edit / Move**, and select the axis you want to move – §2.11.6). Moving polar axes allows you to plot a hemisphere or quadrant instead of the entire circle.

2.11.2 Positioning Tic-Marks on Axes

There are two types of tic marks: ones generated automatically based on a starting position and distance between tics, and “floating” tics that you add one-by-one, providing a coordinate and label for each. You can tell the autolabeling feature where to put tics and whether to draw only major tics, only minor tics, or to draw no tic marks at all. You can use “floating” tics to highlight special values or to create custom labelings, such as day-of-week or probability scales (§2.11.4).

To customize tic-mark placement, double click on an axis. The three edit boxes along the top of the **Axis Setup** dialog (Figure 2.13) let you tell EasyPlot how many minor tics to put between major tics, the distance between major tics, and the position of one major tic, which tells whether to put major tics at 0, 50,..., or 10, 60,..., for example.

When you customize tic positions, EasyPlot will not change the position or spacing of the tics even if the range changes. If you extend the range of the axis, EasyPlot adds tics until it reaches the maximum and minimum or until it reaches its limit of thirty labels. Likewise, if you zoom, the scheme you provide is stretched out and you see fewer tic marks. To go back to having EasyPlot place the tics, turn on the **Let EasyPlot... Place tic marks** checkbox. EasyPlot will then choose tic positions for you and update its choice any time the range changes.

On autoranged axes, EasyPlot sets the range to include all the data and major tic marks at the ends. If you change the tic positions, the axis range may change, too. See Section 2.17 for more details on the **Autorange** feature.

To customize log axes, you must turn off the autolabeling feature (turn major and minor tics off – §2.11.3) and mark the axis with floating tic marks.

The autolabeling feature allows up to thirty labels on an axis. If the scheme you provide results in more than thirty labels, a EasyPlot tells you there are:

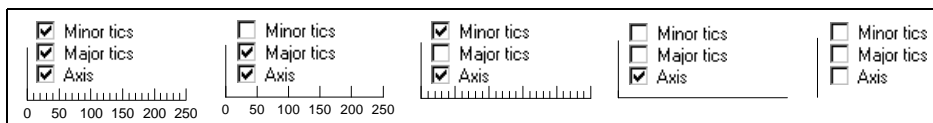
too many labels.

EasyPlot proceeds to mark the axis but only until the thirtieth label. If you do need more than thirty labels, you can add tic marks and labels individually; the thirty label limit applies only to automatically generated major tic marks. You can have any number of minor tics between labels.

With the EasyPlot-standard menu, the **Auto-tics** buttons (under **Options**) let you position and turn tics on or off. The **Set Locs** button prompts you for tic position, increment, and the number of minor tics. On contour plots, the **Set Locs** button places contours the same way it places tic marks on the x - and y -axes. Contours are treated as tic marks on the z -axis.

2.11.3 Turning Tic Marks Off

Double-click on an axis. In the **Generate** group, the **Minor tics** and **Major tics** checkboxes toggle minor and major tics on or off for an axis. The graphic below illustrates the different options. With minor tics on and major tics off, for example, EasyPlot draws tics at the major and minor positions but does not label the major tics. This setup is useful for polar plots; you can turn major tics off for one axis to make the graph appear less cluttered. If major and minor tics are off, EasyPlot draws no tics on the axis and you can leave the axis blank or label it with “floating” tics (§2.11.4). The **Axis** checkbox turns the entire axis off, including the axis line itself.



The **Minor tics** and **Major tics** checkboxes control only tic marks generated by EasyPlot’s automatic labeling feature. To turn off floating tic marks, you must delete them in the **Floating Tics** dialog (§2.11.4).

With the EasyPlot-standard menu, you can use the **Min tics** and **Maj tics** buttons (under **Options / Auto-tics**) to turn tics on or off.

2.11.4 Floating Tic Marks

You can add tic marks one at a time anywhere on an axis. Use “floating” tic marks to emphasize axis values of particular importance. Or you can turn off the automatic labeling feature and mark an entire axis with floating tics. The labels you provide for these tic marks can be any text. With floating tics, you can label an axis with the days of the week, the months of the year, or with numbers other than the coordinate of the tic.

Double-click on an axis and select **Floating tics....** EasyPlot displays the **Floating Tics** dialog (Figure 2.14). With the EasyPlot-standard menu, **Add / Tic** also opens the **Floating Tics** dialog. Enter a coordinate for the tic. To use a word or a special number for the tic label, edit the text in **label for tic**. Labeling a tic with a **<space>** creates a major tic mark (and accompanying grid line) without a visible label. To create a minor tic, click on **minor tic** or remove all text from the **label for tic** box. Click on **Add—>** to add the tic to the axis.

To change a label or delete a tic, select it from the list and make changes in the edit boxes. If you change the coordinate of a tic, EasyPlot leaves the original.

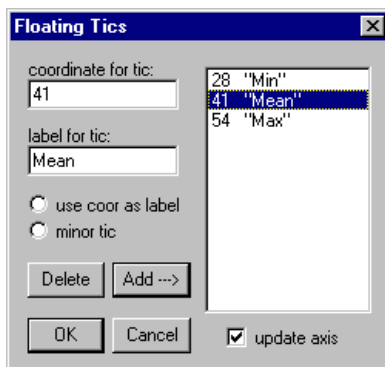


Figure 2.14 The Floating Tics dialog

Click on **Delete** to remove the selected tic.

If **update axis** is checked, the axis updates when a tic is added, changed, or removed. Turn **update axis** off if updates result in inconvenient delays.

If you want a denser linear labeling than EasyPlot generates, you do not have to add floating tics to fill in the axis. Use the **Axis Setup** dialog to create a denser labeling automatically (§2.11.2).

On contour maps, floating tic marks on the z -axis add major and minor contour lines. Use the EasyPlot-standard menu to **ADD / Tics** to the z -axis. If you provide a label for the tic (or contour), it appears in the same color as the contour. If you do not provide a label, you get a minor contour.

Nonlinear Labelings

The ability to label floating tic marks with numbers other than their coordinates lets you create any nonlinear scale. You transform the data to a linear scale but mark the axis to display a nonlinear scale. To relabel an axis, you should first turn off the automatic labeling feature (§2.11.3). Next, decide where you want tic marks and transform those coordinates to the linear scale. Use the linear coordinates to place the marks and the nonlinear coordinates as labels.

If you label an entire axis with individually added tics, you can use EasyPlot batch commands to create a tic mark template file. §Figure 2.15 lists example batch commands which add tic marks. Once you create a file with these commands, you can read it into EasyPlot as though it were a data file. Read it into EasyPlot with any data you want plotted on that scale.

You might want to add the following lines to your tic mark file to turn off the automatic labeling feature for the specified axis:

/at	x	15	"15"	;tic at X=15, label: 15
/at	x	15	=	;tic at X=15, label: 15
/at	x	50	".95"	;tic at X=50, label: 0.95
/at	y	100	"cutoff"	;tic at Y=100, label: cutoff
/at	x	60		;minor tic at X=60
/at	z	30	=	;draw contour line at Z=30

Figure 2.15 Example batch commands for adding tic marks

```
/oaam x OFF      ;turn minor tics off on X-axis
/oaamm x OFF     ;turn major tics off on X-axis
```

You can use the **Save** feature (under **File**) to create a template tic-mark file. Add all the tic marks you need. Turn automatic labeling off wherever necessary. Delete any data, annotations, lines, or titles until you have just the tic marks on an empty graph. Save the graph. The save file is your template file. The **Save** button creates a file containing EasyPlot batch commands and comments which explain the commands. You can read and edit the file with any text editor or word processor.

There is no limit to the number of floating tic marks you can add.

2.11.5 Tic Size and Direction

EasyPlot sizes tic marks automatically based on the size of the graph and a tic scaling factor. To adjust the length of tics, double click on an axis and enter a number in the **Tic size** control. Values greater than 1 make tics longer than the default; values less than 1 make them shorter. Enter '2', for example, to make tics twice as long as the default. Tic size affects all axes on all graphs.

To change the direction in which tics are drawn, go to **Tic Preferences** (Figure 2.16). You can draw tics inside or outside the graph, or centered on axes. The **Flip tics** checkbox draws tics onscreen in the opposite direction from how they will print; with tics set up to print inside the graph, the **Flip** option lets you keep the graph area clear for viewing and analyzing data.

The **Major to minor size** at the bottom of **Tic Preferences** lets you adjust the length of minor tics relative to major tics. Enter 1 to make all tics the same length. Larger numbers make minor tics shorter relative to the fixed length of major tics.

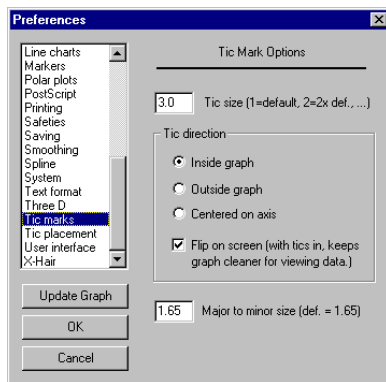


Figure 2.16 Tic Preferences

2.11.6 Axis Placement

To move an axis away from its default location, double click on it and, in **Axis location**, enter a y -coordinate for an x -axis or an x -coordinate for a y -axis. The coordinate can be inside or outside of the graph.

If you customize an axis position and then change the graph range, the axis moves with its coordinate, even if it ends up way off the graph. To go back to having EasyPlot place the axis, turn on **Set axis location** under **Let EasyPlot...**

On polar plots, moving axes modifies the angular range of the plot. You can plot a semicircle or quadrant instead of the entire circle. With a polar plot on the screen, pull down **Edit**, select **Move**, and choose an axis. (You must be using the EasyPlot-standard menu; go to **User Interface / Preferences** if you need to switch.) For the x -axis, you can choose **bottom**, **center**, or **top**. For the y -axis, choose **left**, **center**, or **right**. Moving the x -axis to the bottom of the graph, for example, plots the top hemisphere, $0 \leq \theta \leq 180$, or $0 \leq \theta \leq \pi$. Moving the x -axis to the bottom and the y -axis to the left plots the first quadrant, $0 \leq \theta \leq 90$, or $0 \leq \theta \leq \pi/2$.

2.11.7 Log Scales

To switch an axis to logarithmic scale, double click on the axis and turn on **Log scale** under **Axis type**. Or right click on the axis and select **Log Scale**. With the EasyPlot-standard menu, the **Log** button (under **Options**) toggles between linear and log scales.

When graphing data on a log scale, EasyPlot ignores data that is less than or equal to zero. EasyPlot does not support logarithmic axes on 3D plots.

2.11.8 Reverse-Direction Axes

By default, EasyPlot places y_{\min} at the bottom of the y-axis and y_{\max} at the top. To reverse the direction of an axis, double click on the axis and turn on **Reverse**, under **Axis type**. With the EasyPlot-standard menu, you can also pull down **Options** and choose **Reverse**.


2.12 Creating a Legend

To create a legend, pull down **Options** and select **Legend**. Only curves which have been titled appear in the legend. If you choose the **Legend** button and get a small, empty white box, none of the curves have been titled.

To enter or edit a curve's legend, double-click on the curve and select **Legend...** Or right click on a curve and choose **Legend**. Or select a curve and click on the **Leg** button in the **Curve Toolbar**. With the EasyPlot-standard menu, you can also pull down **Edit**, select **Title**, and choose **Curve**. EasyPlot displays the Text Toolbar (§2.8) and a window for entering or modifying the curve's legend title. Click **OK** or inside the box but off the text to end editing.

When you turn the legend on, EasyPlot places it in the top-right portion of the graph. (You can set the default legend position in **Legend Preferences**.) Click and drag to move the legend. The legend can extend off the EasyPlot window. When you print, the whole legend appears, provided it doesn't overshoot the edge of the printed page.

To remove the legend, select the **Options / Legend** again. Or right click on the legend and select **Delete**. You can also turn off the legend by hitting **** while holding it selected with the left mouse button. If you bring the legend back after removing it, it reappears where you last left it.

You can change the order in which curves appear in the legend. Turn on the **Curve Toolbar** (under the **Window** menu), select a curve (click on it in the legend or on the graph), and click on the up or down arrows in the **Curve Toolbar** (). You can also double click on the curve and use the up- and down-legend arrows in the **Data Style** dialog (§Figure 2.2). A third way to rearrange legend entries is to hold a curve selected (§1.3) with the space bar or right mouse button and, while the curve glimmers, hit **t** or **b** to move the curve to the top or bottom of the legend.

You can place legend titles in data files with the command:

```
/sa 1 "title" [n]
```

The n is an optional parameter specifying the data column the title describes. If n is not given, EasyPlot assigns the title to column 2.

When you print, the legend may appear larger or smaller than its screen counterpart depending upon the point size of the legend text. The top-left corner positions the legend on the page and the bottom-right corner moves to accommodate the legend contents. Preview to see the exactly where the legend will print. If the legend is too large, go to **Style / Font** and choose a smaller point size for the legend text.

You can choose whether or not to draw a box around the legend in **Legend Preferences**. Without the box, you can also choose whether or not to clear the entire area behind the legend. If the **Box** and **Clear** are off, EasyPlot clears behind each character in the legend but not behind the curve portion of the legend.

If you use a lot of superscripts and subscripts in legend titles, you may need to increase the space between the entries. Go to **Legend Preferences** (right click on the legend) and increase the **Entry spacing**. The value is a scale factor; if $N > 1$, you increase the space between entries. $N = 2$, for example, doubles the spacing.

If all curves have solid connecting lines and data markers, the connecting lines do not provide any information for differentiating curves on black-and-white output. The **Dashed lines only** option (in **Legend Preferences**) displays only data markers in the legend if all curves have solid connecting lines and markers.

EasyPlot can assign a default legend title to curves. In **Legend Preferences**, you can select to use the filename and column number or column name as a curve's default title. For curves generated from equations, you can have EasyPlot use the equation as the default title.

2.13 Multiple-Axis Graphs

There are three types of axes: primary, secondary, and private. Every graph has primary x - and y -axes, even if you can't see them (2.11.3). Primary axes appear on the bottom and left sides of the graph. Secondary axes appear on the top and right sides of the graph. "Private" axes are linked and color-coded to specific curves. Private axes appear next to the primary axes. Any curve can have a private x - and/or y -axis and while they are called "private", you can plot any curve against another curve's private axes.

To add a second x - or y -axis, pull down **Add** and select **X-axis** or **Y-axis**. Or right click on a primary axis and select **Add 2nd Axis**. EasyPlot draws a second x -axis on top of the graph and a second y -axis on the right side of the graph. The new axis inherits the same range as the primary axis. If the range on the primary axis is manually set, the new axis holds the same scale until you either manually

set a new range or autorange. If the primary axis was autoranged, the range on the new axis changes to accommodate data which is plotted against it.

To create a private axis, double click on a curve and turn on the **X** and/or **Y Private axes** checkboxes. Or right click on a curve and select **Private axes...**

Small markers next to x_{\max} or y_{\max} point to the selected axis. New curves are plotted against the selected axis. Click on an axis to select it. If no curves are plotted against an axis, the marker is open (∇). Otherwise, you get a colored marker for each curve plotted on the axis.

You can move curves from one scale to another by right clicking on a curve. With only one secondary axis, choose **Plot on other axis**. With more axes, choose **Select axis...** and then click on the axis you want the curve to use.

You can title secondary y-axes and private axes by clicking on the title box next to the axes. Private y-axis titles appear above the axes. Private x-axis titles appear to the left of the axes. Once you've entered a title, click on the text to edit it. EasyPlot does not support an explicit title for secondary x-axes; add additional lines to the graph title to label secondary x-axes, using a blank line or two between the titles if desired. (Type a space before hitting <Enter> to leave a blank line.)

To delete a secondary axis, right or double click on it and select **Delete**. (With the EasyPlot-standard menu, you can also use the **Delete** button under **Edit**). Data plotted against the axis is deleted with the axis. To remove private axes, right click on the axis and select **Private X** or **Private Y**. Turning off private axes does not delete any data. You cannot delete primary axes but you can turn major and minor tic marks off (2.11.3).

You can customize the positions of primary and secondary axes (§2.11.6) but not private axes. Grid lines are drawn on primary axes only. Annotations (§2.7) and lines you sketch on the graph (§2.9) are plotted against the primary axes.

2.14 Axis Linking and Mirroring

You can link axes together so that the linked axes 'mirror' each other; set the range or tic positions on one axis and all linked axes mirror the change. The mirroring feature is handy when working with several stacked graphs all of which need to have the same x-axis range, or when you want the x- and y-axes on a graph to be identical. If you zoom or scroll one graph, other graphs with axes linked to the current graph also zoom or scroll.

To link axes, right click on an axis and select **Mirror axis...** The cursor changes to a cross-hair ("+") and, on the help line, EasyPlot prompts you to click


on the axis to mirror. The first axis takes on the range and tic positions of the second axis. From then on, the mirrored axes behave like identical twins. You can chain any number of axes together and you can have any number of mirror chains. EasyPlot draws a color-coded ‘mirror marker’ (☞) next to each mirrored axis; each mirror chain has its own color marker.

To remove an axis from a linked chain or to break an entire mirror chain, right click on one the axis and select **Break mirror** or **Break all N mirror links**. N is the number of axes in the chain.

2.15 Zooming

To zoom in on part of a graph, click and drag on the area of interest. When you release the mouse button, EasyPlot expands the area inside the rectangle to fill the entire graph.

Hit any key before releasing the mouse button to cancel a zoom. Or, if you release without having defined a rectangle, the graph remains unchanged. If you zoom in by mistake or want to return to previous ranges, **<alt>click** or use **Undo**. You can unzoom up to five times.

You can zoom as many times as you want. Zooming turns autoranging off for the entire graph. To restore a graph to its unzoomed state, use **Autorange** (under **Options**) or hit the autorange icon () on the help line (§2.17).

You cannot zoom while in draw-line mode (§2.9). In draw-line mode, clicking and dragging draws a line instead of a zoom rectangle. Hit **<esc>** or the space bar to leave draw-line mode. You also cannot start zooming with the cursor on an annotation, the legend, or a line or arrow; you will move the object.

2.16 Scrolling

Once you have zoomed, you might want to view a different window of data, or slide along the data at the current zoom magnification. Pull down **Tools** and select **Scroll**. The area between the scroll arrows represents the entire range of data. The highlighted section inside the scroll bar represents the data visible on the graph. If all the data is visible, the full area between the arrows is highlighted.


Click on a scroll arrow to shift the range by one major-tic increment. You can also click on the area between the two arrows to jump from one portion of the graph to another. Clicking just to the left of the right arrow on the x -axis, for

example, brings the maximum end of the x -range onto the screen.

If you turn **<scroll lock>** on, the cursor-arrow keys scroll the graph instead of moving the cursor. Scrolling works with linear-scale axes only; it does not work with log scales or polar plots.

If an axis is mirrored to another axis (§2.14), all linked axes will scroll.

2.17 Autorange


When you create a graph, EasyPlot's autorange feature sets axis ranges so that all the data appears on the graph. If you change the range by zooming, scrolling, or setting a specific range, you disable the autorange feature. To undo any of these range changes, pull down **Options**, select **Autorange**, and choose an axis or the entire graph. The autorange icon () on the help line autoranges the entire graph. You can also double click on an axis and turning on **Autorange** (§2.11.1).

Autoranging the x - or y -axis only autoranges the axis within the range defined by the other axis. If, for example, data extends from 0 to 1000 on the x -axis and you limit the x -range to $0 \leq x \leq 2$, autoranging the y -axis sets the y -range based only on the data points within the range $0 \leq x \leq 2$.

The autorange feature uses the data bounds and the labeling scheme to set the range for an axis. It places the axis minimum at the first major tic mark that is less than or equal to the data minimum and the axis maximum at the first major tic mark that is greater than or equal to the data maximum. If, for example, the data goes from 12 to 475 and the labeling scheme specifies major tic marks at multiples of 100, the autorange feature sets the range to go from 0 to 500. If you specify a labeling scheme that places major tic marks at multiples of 150, the autorange feature sets the range to go from 0 to 600, 600 being the first multiple of 150 which is greater than 475.

Setting the maximum and/or minimum to a specific value (§2.11.1) turns autoranging off for that axis. Zooming and scrolling also turn autoranging off. EasyPlot also turns autoranging off when you compute a curve fit.

2.18 Displaying Multiple Graphs

You can place any number of graphs on the screen at one time. Each time you run **File / New** or click the **New** icon () on the help line, EasyPlot lays an empty graph on top of old graphs and prompts you for data to plot. (With the EasyPlot-standard menu, you can also **Add a Graph**.) The new graph hides existing graphs

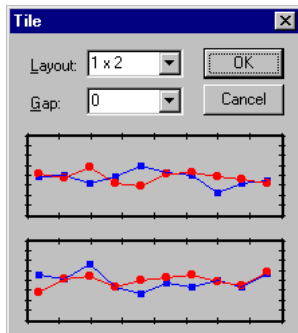


Figure 2.17 The Tile dialog

but if you reduce its size, you'll see the hidden graphs.

The easiest way to lay out multiple graphs on a page is with the **Tile** tool (in the **Window** menu). The **Tile** dialog (§Figure 2.17) lets you select the graph layout and the spacing between graphs. Use positive **Gap** values to increase the space between graphs; use negative values to bring graphs closer together. The **Tile** tool tries to put each graph in the position closest to its current location. To rearrange graphs, tile once, move each graph close to its final position, and then **Tile** again to line them up. If there are more tile positions than graphs, EasyPlot leaves some of the positions empty. If there are more graphs than positions, EasyPlot minimizes the extra graphs. The **Tile** tool ignores minimized graphs.

You can click and drag any edge or corner of a graph to change its size and shape. If a graph is not maximized, you can move the whole box by clicking and dragging the blue bar across the top of the graph, or by **<alt>**clicking on its border. You cannot move a graph off the edge of the EasyPlot window. If you start moving and want to cancel or start over, hit any key (except for **b**, **h**, or ****) before releasing the left mouse button.

All graph operations such as adding curves, titles, or setting ranges, affect the current, or top graph. Click on any part of a graph to make it the current graph. Hit **<Ctrl><Tab>** to cycle from one graph to the next. You can send a graph to the back of the stack by picking up an edge or corner and, before releasing, hit **b** for "back", or **h** for "hide". With the EasyPlot-standard menu, you can use **Next** (under **Edit**) to change the current graph.

The **Window** menu also lets you select the current graph. Each button at the bottom of the menu represents a graph. The label in the button is either the name of the graph save file, the title of the graph, or the equation or data file that is plotted on the graph. If a graph has been minimized, an "i" (for "icon") in parentheses follows the name.

To delete a graph, pull down **File** and choose **Close**; or click a close icon in the top left or right corners of the graph. Or start moving the graph and hit **** or **<backspace>** before releasing the mouse button. With the EasyPlot-standard menu, **Edit / Delete / Graph** deletes the current graph. If you delete or close the only graph, EasyPlot creates a new graph and prompts you to for data to plot.

If you close a graph that has been modified, EasyPlot asks whether you want to save the graph. If you open and close many graphs and don't want to be prompted to save when closing, go to **Safeties Preferences** dialog and turn off **when closing graphs** under **Prompt to save...**

With multiple graphs on a page, EasyPlot can scale text by the size of the graph so that small, inset graphs are drawn with smaller text than full-size graphs. Turn off **Scale text size down on inset graphs** in **Printing Preferences** to use the same size text on all graphs.

If you work with multiple graphs in batch mode, the batch commands for deleting a graph or moving to the next graph look for an optional graph number:

```
/ed    g    3           ;delete the third graph in the list
/en    g    2           ;bring the second graph to the front
```

The current, or top, graph is number 1; the graph on the bottom of the stack has the highest number. If you provide a number which is less than 1 or greater than the number of graphs, EasyPlot deletes the current graph or pops the bottom graph to the front, as if no graph number were given. The command for positioning a graph is **//pos**, followed by four numbers:

```
//pos  x1  y1  x2  y2
```

where (x_1, y_1) is the lower left corner and (x_2, y_2) is the top right of the graph. Use numbers between 0 and 1, where (0,0) is the lower-left corner of the EasyPlot window and (1,1) is the top-right.

2.19 Maximizing and Minimizing Graphs

The top corners of each graph contain buttons for maximizing, minimizing, and closing the graph. Click on a maximize button to have the graph fill the entire EasyPlot window. If you decrease the size of the EasyPlot window, maximizing refits the graph to the new window space.

The minimize buttons reduces the graph to an icon on the help line. To restore a minimized graph, click on the blue rectangle on the help line or by its button in

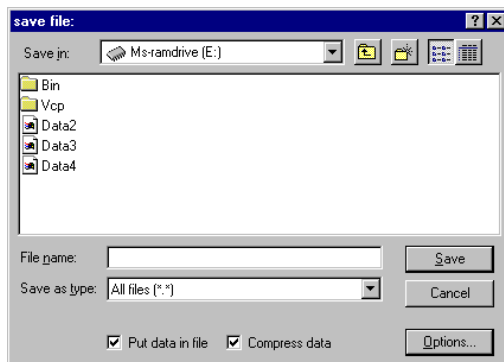


Figure 2.18 The Save dialog

the **Window** menu. Minimized graphs have an ‘i’ (for ‘icon’) following the name in the **Window** list. If you select the button of a minimized graph, EasyPlot restores the graph and displays it as the current graph.

2.20 Saving Graphs

To save a graph or graphs, pull down **File** and select **Save**. Or click the **Save** icon (📁) on the help line. With several graphs open, you can save all graphs, only visible graphs, or just the top graph. To restore the graph or graphs, read the save file as you would any data file, with **File / New** or **Open**.

The Save dialog (Figure 2.18) has options for storing data in the save file. With **Put data in file** checked, EasyPlot puts all data into the save file. If unchecked, EasyPlot links the save file to data files, provided the data file did not use the **//nc** command (§6.3) and the data was not altered in any way within EasyPlot. When using data-file links, you must leave the original data files intact to properly restore the graph.

Compress data selects whether data EasyPlot puts in save files is stored in ASCII or compressed format. With large data files, compressing data results in much smaller save files that EasyPlot can write and read much more quickly.

With **Compress data** off, EasyPlot save files are fully commented ASCII files containing batch commands and possibly data. You can learn about EasyPlot’s batch language by looking at a save file, editing, and reloading the file to see the result of your change. See Section 6.1 for more details on batch commands.

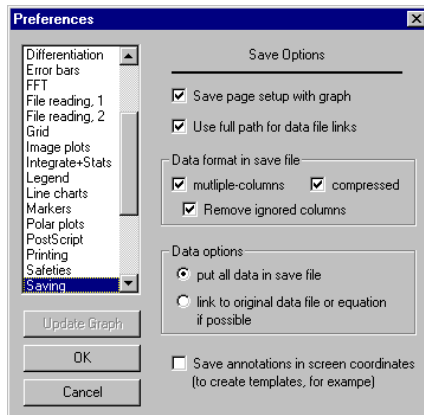


Figure 2.19 Saving Preferences

The **Options...** button opens **Saving Preferences** (§Figure 2.19). The top checkbox, **Save page setup with graph**, selects whether save files include page dimensions, page orientation, font and font sizes, and other page-setup information. The page setup is a global and if you don't want save files to change the settings, save your graphs with the **Save page setup** option off.

By default, EasyPlot refers to data files with a full path. If data and save files are in the same directory and you want the files to be portable (so that you can move them to a different drive or directory), turn off **Use full path to data files**.

The **Data format** group lets you control the format in which data is stored in save files. EasyPlot can write multiple-column data in a single block or in separate sections of just the columns needed for each curve. With **multiple columns** on, you can choose to store all columns or just those that are needed to recreate the graph. Use all three **Data format** options to produce the smallest save files.

The two **Data Options** radio buttons set the same option as the **Put data in file** checkbox described above.

When using a save file as a graph template and changing only the data, you might want to **Save annotations in screen coordinates** when creating the template. That way if axis ranges change to accommodate different data, annotations appear in the same place on the graph.

2.21 Deleting Curves

To delete a curve, double- or right-click on the curve and select **Delete**. You can also delete a curve by holding the **<space>** bar or left mouse button (§1.3) and hitting **** or **<backspace>** while the curve glimmers. With the EasyPlot-standard menu, you can pull down **Edit**, select **Delete**, and choose **Curve**.

If a graph contains curves from a multiple-column data file and you want to view only a few of the columns, you do not have to delete the curves (or columns) you don't want; use **Define Data** (under **Tools** – §3.4). Or load the data into the Data Table (**Edit / Data** – §7.9.2) and double click on definition letters at the top of columns to turn columns on or off.

Chapter 3: Input

You can read data into EasyPlot from ASCII, spreadsheet, or binary files. You can also enter an equation and have EasyPlot generate data for you. Or, you can type numbers into EasyPlot's Data Table. When you invoke EasyPlot and when you choose **File / Open** or **New**, EasyPlot asks for something to plot (§2.1). Provide a file or an equation and EasyPlot graphs the data. Pull down **File** and choose **Open** to plot more data on the current graph.

3.1 Creating Data Files

EasyPlot reads ASCII, spreadsheet, and binary files, as well as data on the Clipboard. You can prepare EasyPlot data files with any word processor, text editor, spreadsheet, data acquisition program, your own computer programs, or with EasyPlot's Data Table. If you use a wordprocessor, be sure the file you create is an ASCII file; most PC applications have some facility for generating ASCII files. If you use a spreadsheet, you can read **xls** (Excel), **wk1** or **wk3** (Lotus 123), or **wq1** (Quattro Pro) format files directly into EasyPlot.

When preparing data for EasyPlot, arrange numbers in rows and columns. Each row represents one data point or set of data points. Each column represents a set of x , y , z , error, radius, or theta values. Separate numbers with a space or comma. In spreadsheets, each cell is automatically distinct from its neighbors.

By default, EasyPlot can read files of up to 150 columns. If your files have more than 150 columns, go to **File reading Preferences** and increase the **max # of data columns**. Don't use an exceedingly large number because when reading a file, EasyPlot requires a memory block of size proportional to the maximum number of columns. When generating files with many columns, you can use EasyPlot's extend-line character, "V", to limit line lengths. When EasyPlot finds a backslash, it advances to the next line without starting a new data row.

EasyPlot lets you arrange and define data columns any way you like. If you don't predefine the columns, EasyPlot starts with its own defaults. With one column of data, EasyPlot plots the data as y -values; it assigns an x -value to each point, starting at zero and counting up to $N - 1$, where N is the number of data rows. If a file has 19 or fewer, EasyPlot assumes the first column is x and plots a curve with each subsequent y -column. If a file contains than 18 y -columns, error columns, or an arrangement other than **xyyy...**, you need to define the columns either in the Data Table (§7.9.2) or by plotting the data and then using the **Define Data** button under **Tools** (§3.4).

For polar plots, place radius and angle values in a data file. As with xy data, you can have multiple r - and/or t -columns (r stands for “radius” and t for “theta”). If you define columns (§3.4) with rs and ts , EasyPlot generates a polar plot instead of an xy plot.

For contour maps, image plots, or 3D surfaces, EasyPlot needs a grid of z -values. The matrix must be rectangular (20x20, 30x50, etc.) and all data in the grid should be evenly spaced z -values. As an example of how to arrange surface data, assume you want to generate a contour map of one square mile of your neighborhood and that you have a map showing elevation lines. Start at the top-left corner of the map and open a new file. Read the height at the top-left corner and put the value into the file. Move one inch to the right on the map and read another value. Place that number to the right of the first number using a space, tab, or comma to separate the values. Continue moving to the right on the map and adding numbers to the first data row. When you get to the right side of the map, move down an inch, go back to the left side of the map, and start a new row in the file. Repeat until you reach the bottom-right corner of the map. The top-left portion of the matrix maps to the top-left corner of the contour map.

The x - and y -coordinates of each z -value in a surface grid is implied by its position in the matrix. By default, EasyPlot assigns the ranges $0 < x < N$ and $0 < y < M$, where N and M are the dimensions of the surface matrix. To specify ranges, use the “**//cx**” and “**//cy**” commands (§6.3):

```
//cx  50  100    ;set x-range of surface
//cy  0   15    ;set y-range of surface
```

Place the surface range-setting commands directly in the data file.

If a matrix has 20 or more columns, EasyPlot automatically assumes it is a surface grid. If it has fewer than 20 columns, you have to define the columns as z -values: Put “**/td z**” directly in the file; or plot the data, run **Define Data** (under **Tools**), and enter a single “**z**” as the definition string (§3.4).

If you want to plot contour maps or 3D surfaces from unevenly spaced data or from random *xyz* triplets, read the data into EasyPlot in 3-column, *xyz* format and use the **XYZ->Grid** feature (under **Tools**) to convert the *xyz* triplets to an evenly spaced surface grid.

EasyPlot places no limit on the number of data rows in a file. It reads data until it reaches the end of the file or until it runs out of memory.

Rows cannot vary in length. EasyPlot determines the number of columns in a file by the first line of data it finds. It ignores subsequent lines with more or fewer columns. If a file contains the **//nc** (“new curve”) command (§6.3), EasyPlot resets the number of columns based on the first line of data after the **//nc**.

The data can be integer, decimal, or exponential. EasyPlot accepts numbers in the range $-10^{35} < n < 10^{35}$. It ignores numbers outside this range. It treats numbers in the range $-10^{-35} < n < 10^{-35}$ as zero. If you want to represent a singularity in your data, use the **//inf+** and **//inf-** commands (§6.3) instead of very large positive or negative numbers; these commands represent positive and negative infinity and can be used in place of numbers.

ASCII and spreadsheet files can contain data, batch commands, or any mix of data and commands. All text other than data and commands is ignored.

3.2 Naming Columns

You can name data columns to make selecting curves easier. If columns are named, EasyPlot displays the name along with a piece of the curve any time you select a curve. When you run **Define Data** (under **Tools**), you selectively view and ignore columns by name rather than column number. You can also have EasyPlot use column name as default legend titles.

Many data acquisition programs place a row of column headers at the top of data files. You can use these or a row of headers you create to name columns. Before reading a file, go to **Column names Preferences**. Enter the row number of the column names. The row should have a name for each data column.

You can choose whether to define columns automatically or manually when turning columns on and off. With automatic definitions, EasyPlot defines columns using the current column-definition string, or, if no definition was provided, using the defaults in Table 3.2. To specify a definition, put a “**/td xy...**” command in the data file or in a batch file that reads the data file (§3.4). With manual definitions, EasyPlot prompts you to define a column whenever you choose to view it. Manually defining columns lets you change a column’s definition; automatic definitions is quicker but only lets you turn a column on or off.

You can build column definitions into the names, as in “**time=x**”, or “**velocity=y**”. The equal sign and definition letter do not count toward the ten-character maximum for the name. EasyPlot uses the latest definition for subsequent names that do not have a definition character:

```
time=x    vel=y    acc                ;define columns as xyy
time=y    vel      acc                ;define columns as yyy
```

You can assign column names with a batch command, as in the following:

```
//col_name    time    vel    acc        ;set the names
/ac           tva.dat        ;read the file
```

With the **//col_name** command, names can have a column number in addition to a definition character, as in:

```
//col_name    time    vel=5    acc        ;name columns 1, 5, & 6
```

Use column numbers if want to name only selected columns of a file.

If a file has column names and you want to load the file in batch mode, use the command **//cn_row** to specify which row has the names:

```
//cn_row      4                ;use 4th row to name columns
/ac           data.dat        ;read file w/ names in 4th column
```

Put names in quotes if they include spaces or numerical characters (including +, -, ...). Names can have up to 10 characters and do not have to line up with the data columns. If the number of strings on the name row does not match the number of data columns, EasyPlot displays a warning and lets you edit the file. If you choose not to edit the file, EasyPlot assigns the first name to column 1, the second to column 2, and so on until it runs out of names or columns.

If you want a user to select columns to view from a batch file, use the **//pick_cols** command. It temporarily turns off batch mode and runs the **Define Data** button, letting the user provide the input.

As described below in Section 3.4, the **/td** command defines columns from within a file. The **/td** command expects a column-definition string, as in “**/td xy.y**”. If you use **//col_name** to name columns, you can define columns by name in a batch file with the **//tdn** command:

```
//col_name    time    vel    acc        ;name columns
//tdn         time    vel                ;define as xy.
```

You can use column names as legend titles for curves that have not been titled. Go to **Legend Preferences** and turn on **Use file & column #/name in legend**. If a column is not named, EasyPlot uses the file name and column number.

3.3 Reading Data Files

To read a data file, pull down **File** and select **Open**. With the EasyPlot-standard menu (§7.8), you can also **Add** a **Curve**. EasyPlot reads and plots the file you select. If it finds batch commands in the file, it executes them. To read a spreadsheet, enter a file with extension “**.wk1**” or “**.wk3**” (123), “**.xls**” (Excel), or “**.wq1**” (Quattro Pro). EasyPlot scans the spreadsheet for named ranges and prompts you to select a range to read. Select a named range or specify an explicit range (such as **a5..c20**) or hit **<enter>** without typing anything to read the entire spreadsheet. When EasyPlot reads a spreadsheet, it creates a temporary file with the same name as the spreadsheet but with a “**.tmp**” extension. If a file with that name already exists, it is destroyed.

See Section 2.1 for details on reading ASCII, binary, and spreadsheet files.

3.4 Defining the Columns

EasyPlot maintains a “column definition string” for each data set. The string has one character for each column of data in the file. The first character defines column 1, the second column 2, and so on. You can change definitions to hide columns, transpose graphs, and to select the graph format (*xy*, polar, or surface).

You edit column definitions with **Define Data** (under **Tools**) or in the Data Table. With **Define Data**, you define strings character by character. If columns have been named (§3.2), you define columns by clicking on names in a list. With the Data Table, you define columns by clicking on the definition letter at the top of each data column. Double click on the letter to ignore a column, or if it is already ignored, to plot it. See Section 7.9.2 for more details on defining columns in the Data Table.

Use an **x** to define a column of *x*-values and **y** for *y*-values. For polar data, use **r** for radius values and **t** for theta values. For surface data, use a single **z** to define all the columns. If EasyPlot finds *xs* and *ys* in a definition string, it generates an *xy* plot. If it finds *rs* and *ts*, it makes a polar plot. If it finds a single *z*, it generates a contour map or 3D-surface.

You can hide, or ignore, columns by defining them with a period, comma, or any other character not listed in Table 3.1. In the Data Table, you can double click on the definition letter to ignore a column. Ignored columns remain in memory provided at least one column is displayed.

character	meaning
x	x value
y	y value
z	z value
e	error value
c	color value
r	radius for polar plot
t	angle (θ) for polar plot
any other character	ignore column

Table 3.1 Characters used to define columns

Replacing all x s with y s and all y s with x s transposes the graph. Replacing all x s and y s with r s and t s or vice versa switches between xy and polar format. Changing any string to a single **z** switches from xy or polar to surface format.

You can change definitions any number of times. Changing definitions affects only the way the data is interpreted; it does not transform the data in any way.

If a definition string contains x - and y -columns or r - and t -columns, the first defined column becomes the independent variable. The independent variable can be plotted against any number of dependent-variable columns. If the first column is x , all subsequent y s are plotted against the same x until another x -column is encountered. A six-column file defined as **xyxyxy** produces four curves, the second and third columns vs. the first, and the fifth and sixth columns vs. the fourth. Each time a column of the independent variable is found, it becomes the current independent-variable column. Only columns of the dependent variable produce curves. If a six-column file is defined with the string **xyxxxy**, only two curves result, the second column vs. the first, and the sixth column vs. the fifth.

If a definition string contains only y s, only x s, or only r s or t s, EasyPlot treats all columns as dependent variables and uses a counter for the independent variable. A three-column file with a definition string **yyy**, for example, produces

three curves; the first point of each curve is at $x = 0$, the second at $x = 1$, and so on up to $x = N - 1$, where N is the number of points.

When you do not specify a definition, EasyPlot defines the columns for you. A one-column file is given the string **y**; the one column is plotted as y-values and EasyPlot assigns a counter for x . Two to nineteen columns are defined as “**xyyy...**”. Files with twenty or more columns are defined with a single “**z**”. Default definitions are summarized in Table 3.2.

If all your data files are arranged with alternating x - and y -columns, for example, you can set the default definition string to **xyxyxy** and not have to define each file individually. Onscreen, you customize the default by running

columns	definition string
1	y
2	xy
3	xyy
...	
19	xyyyyyyyyyyyyyyyyyyy
20+	z

Table 3.2 Default column definitions

Define Data on an empty graph. From a batch file, use the command **//def_str** followed by a definition string to set the default.

You can define columns with a batch command in the file. Then every time you read the file, the columns will be properly defined. Place the command:

/td ‘definition string’

in a data file or in a batch file *before* the line that reads the data file. The **/** is EasyPlot’s command character. The **td** following the slash pulls down **Tools** and run the first button beginning with **d**, **Define Data**. Some examples are listed in Figure 3.1. See Section 6.1 for a detailed discussion of batch commands.

/td	yyy	;col1-3 vs. X=0,1,2,...,N
/td	xye	;col2 vs. col1, col3=error value
/td	xyxyy	;col2-3 vs. col1 & col5-6 vs col4
/td	rt	;col2 vs. col1 in a polar plot
/td	trrr	;col2-4 vs. col1 in a polar plot
/td	z	;contour map with matrix of Zs

Figure 3.1 Example batch commands for defining data columns

3.5 Date and Time Data

EasyPlot accepts time and date data. By default, EasyPlot assumes dates are in the form MM/DD/YY. If your dates are formatted differently, go to **System Preferences** and edit the date format. Enter 8 characters, such as ‘DD/MM/YY’, or ‘YY-MM-DD’. The delimiter character (the ‘/’ or ‘-’) and the order of the day, month, and year are used for dates that EasyPlot prints on date axes and where you use the **%date** macro.

Time must be in the form HH:MM:SS. The seconds can include a decimal point, as in **05:22:54.31**. The hour runs between 0 and 24. EasyPlot automatically detects the time and date columns by their format; you do not need to specify that columns are date or time. To use the date and time on the graph, define a date or time column with an *x* or *y* (with **Tools / Define Data** or in the Data Table).

Files can contain date, time, or both date and time columns. With date and time columns, you should select one of the columns as your *x* (usually the date) and ignore the other. If you define the date as *x*, EasyPlot combines the date and time for the *x*-axis. If you define time as *x* (or if you do not have a date column), you can represent a day of data and sequential days will wrap around on the graph. You can enter a **Period on X** of 1 in **Line charts Preferences** to keep a time axis that spans several days increasing on *x* instead of wrapping back on itself.

EasyPlot translates date and time into a numeric representation where days are integers and time runs from 0 to 1 (0 is the beginning of the day and 1 the end). For each graph, EasyPlot sets a base date and computes all numeric dates relative to the base. The base is set to the first date read into the graph. If the first date read is ‘01/01/90’, for example, then Jan. 1 ‘90 is represented internally as 0, Jan. 2 ‘90 as 1, and so on. Jan. 1 ‘91 would be 365 (or 366 if 1990 were a leap year).

When EasyPlot chooses tic marks for a date/time axis, it uses the numeric equivalent of the axis and, as a result, you may not always get the best tic marks; EasyPlot chooses nice tic marks for a base-10 axis but time is an odd mixture of base 60, base 24,... To set tic-mark locations, use the **Axis Setup** dialog (§2.11.2).

When you set ranges or place tic marks on a time axis, you can enter values as date and time or as a number. Use a plus (“+”) sign to combine a date and a time into one value. If you are setting the range, do not put any spaces between the “+” and its neighbors. Using the numeric equivalent can be handy as a quick way to enter a date or time. If you are setting the labeling scheme and want 1 day between major tic marks, enter 1 as the distance between labels. When setting the range, for example, you can use the XMIN and XMAX constants (Table 6.1) to adjust the range relative to the current range; entering “xmax+.5” would extend the *x*-axis by 12 hours.

3.6 Comments

You can place comments anywhere in a data file. When EasyPlot finds a non-numeric character (other than a batch command), it assumes it is a comment and ignores the remainder of the line. Be sure not to begin a comment line with a number; EasyPlot may read the number as data and ignore the real data. In the file listed in Figure 3.2, EasyPlot reads the “40” in the first line as data and assumes the file contains one column of data. It ignores subsequent lines with two columns. Placing a semicolon or another non-numeric character in front of the “40” would solve the problem. You can also use the **Options** button in the **Open** dialog to begin reading the file at line 4.

```

40 measurements of bicycle tire wear taken at 50
kilometer intervals.
km  tread thickness (in mm.)
0   2.36
50  2.31
.
.
.
```

Figure 3.2 How **not** to comment a file (line 1)

3.7 Equations

You can plot equations in EasyPlot. Pull down **File** and select **Open** or **New**. (With the more Windows-standard menu, you can **Add a Function**; with the EasyPlot-standard menu, **Add / Curve** or **Add / Graph** also works.) Type an equation such as $y = 3x - 2$. EasyPlot plots the function along the full range of the independent axis, in this case the x -axis. If there are no curves on the graph, EasyPlot prompts you to enter a range for the independent axis (§2.11.1). Equations can be any of the following forms:

$$\begin{array}{lll} y = f(x) & r = f(t) & z = f(x, y) \\ x = f(y) & t = f(r) & \end{array}$$

The first two forms use x and y as variables and produce xy plots. The second two use r and t (for ‘radius’ and ‘theta’) and produce polar plots. The last defines z in terms of x and y and produces surfaces which you can view as contour maps, image plots, or 3D fishnets.

If you enter equations with no variables, such as “ $\sin(45)$ ”, EasyPlot acts as a calculator and displays the result on the screen.

Unless you are entering the equation of a horizontal or vertical line, such as $y = 2$, or $x = 3$, you do not need to type the left side of the equation, the $y =$, $x =$, etc. EasyPlot deduces where to put the result when you do not specify it. If you enter a function in terms of x , EasyPlot assumes the form $y = f(x)$. Likewise, typing just $f(x, y)$ or $f(t)$ is equivalent to typing $z = f(x, y)$ or $r = f(t)$.

You can use any of the functions listed in Table 3.3. Functions must have their arguments enclosed in parentheses, as in the following:

$$\begin{array}{l} y = \sin(x) \\ r = 5\cos(3t) \\ z = \ln(xy) + \exp(2x(y + 0.1)) \end{array}$$

The trigonometric functions can operate in degrees or in radians. Switch by clicking on **Radians** or **Degrees** when entering an equation.

You can use the constants **pi** and **ne**, (**pi** = π , or 3.1415927, and **ne** = e , or 2.7182818) as numbers, as in $y = \sin(2\text{pi}/x)$, or $y = \text{ne}^x$ (type as **y = ne^x**).

Table 3.4 lists mathematical operators. Note that an asterisk, “*”, acts as the multiplication symbol, not an “x”; “x” represents the variable x . The order of operations follows standard rules. Expressions inside parentheses are evaluated according to nesting levels. Within expressions, exponentiations are performed first, multiplications and divisions second, and additions and subtractions are last. Multiplication is implicit; if no operator is given, the two values are multiplied.

$\text{abs}(n)$	absolute value
$\text{acos}(n)$	arccosine
$\text{asin}(n)$	arcsine
$\text{atan}(n)$	arctangent
$\text{cos}(n)$	cosine
$\text{sin}(n)$	sine
$\text{tan}(n)$	tangent
$\text{cosh}(n)$	hyperbolic cosine
$\text{sinh}(n)$	hyperbolic sine
$\text{tanh}(n)$	hyperbolic tangent
$\text{int}(n)$	integer part of n
$\text{log}(n)$	\log_{10}
$\text{ln}(n)$	\log_e
ptn	point # counter: 0, 1, 2,...
$\text{pow}(n)$	10^n
$\text{px}(n), \text{py}(n)$	n^{th}
$\text{exp}(n)$	e^n
$\text{rnd}(n)$	random number from 0 to n (uniform distribution)
$\text{rndn}(n)$	random number from $-n/2$ to $n/2$ (pseudo-normal distribution)
$\text{sqrt}(n)$	\sqrt{n}
$\text{step}(n)$	step function: 1 when $n \geq 0$ 0 when $n < 0$

Table 3.3 Math functions

The expression $5x$, for example, is equivalent to $5 * x$. Implicit multiplication works with expressions, constants, variables, function calls, and numbers but not with two numbers next to each other: “4 5x” produces a “missing operator” error.

EasyPlot skips over points that cause numerical errors such as dividing by zero or taking the log of a negative number. If you compute the function $\log(x)$ over the range $-5 \leq x \leq 0$, the resulting curve will not have any data points because the $\log(x)$ is undefined over the entire range.

You can enter equations of up to 1000 characters. Variables (x , y , z , r , and t) and function names can be upper or lower case.

+	addition
-	subtraction
*	multiplication
/	division
^	exponentiation
%	mod

Table 3.4 Mathematical operators

3.7.1 Recomputing an Equation

Once an equation is plotted, it is a set of numbers like any other data set you load. If you change the graph range or curve resolution (§3.7.2), EasyPlot does not recompute equation curves automatically. To calculate over a new range or at a new resolution, pull down **Add**, select **Function** or **Curve**, and use the same equation. EasyPlot remembers the last equation; hit **<enter>** or click **OK** to recompute. If you hit **<enter>** without changing the equation, EasyPlot recomputes the curve only if the graph range or resolution has changed. You can force EasyPlot to recompute by making a meaningless edit, such as changing “3” to “3.0”, or “ $x+5$ ” to “ $x + 5$ ”. If you recompute a curve, you must delete the older curve if you do not want it (§2.21).

3.7.2 Setting Curve Resolution

When entering a function to plot or curve-fit, EasyPlot displays the number of points it will compute in the *N points...* button. Click on the **points** button to change the number of computed points. For functions with rapid fluctuations, use more points to better represent the function. For smooth, slow-moving functions, you can use fewer points to speed calculations. For surfaces, enter two values, the number of points along x and y .

For xy curves, the resolution must be an integer from 2 to 100,000. For surfaces, each dimension must be an integer between 0 and 100. By default, EasyPlot generates 150 points for xy or polar equations and 400 points (20x20) for surface matrices.

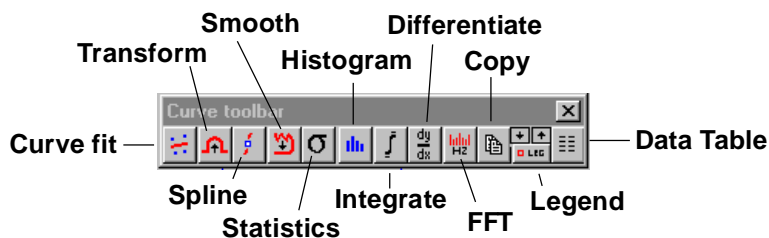
With the EasyPlot-standard menu, the **Resolution** button under **Tools** also lets you set the number of points in computed curves.

Chapter 4: Data Analysis

EasyPlot has tools for analyzing data. You can transform data, fit curves, read x - and y -coordinates from a graph, compute statistics, smooth data, and more.

4.1 The Curve Toolbar

For quick access to many of the math tools described below, turn on EasyPlot's **Curve Toolbar** (under the **Window** menu):



The **Curve Toolbar** floats on top of the EasyPlot window. Click on a curve and then select an operation from the **Curve Toolbar**. If you prefer double clicking to select a curve, the **Data Style** dialog has the same tool buttons along its bottom.

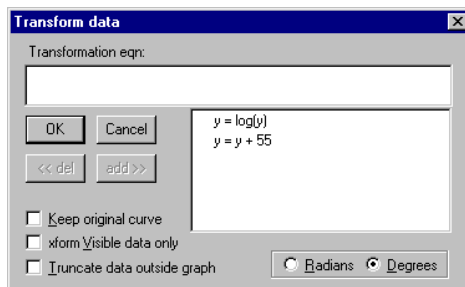
4.2 Transforming Data

EasyPlot lets you perform mathematical transformations on data sets. You can enter equations in any of the following forms:

$$x, y, \text{ or } e = f(x), f(y), \text{ or } f(x,y)$$

$$r, t, \text{ or } e = f(r), f(t), \text{ or } f(r,t)$$

Run **Tools / Transform** or, with the EasyPlot-standard menu, **Edit / Xform**. Pick **Curve** or **Graph**. EasyPlot prompts you for a transformation equation:



Enter an equation such as $y = \sin(y)$, $y = y + 10$, $y = xy$, or, if working with polar plots, $r = 3r$ or $t = t + \pi/2$ (type the last equation as **t = t + pi/2**).

When transforming curves, EasyPlot adds the result to the current graph. If you choose to transform the whole graph, EasyPlot transforms every curve and places the results in a new graph; the original graph remains unchanged.

When done, EasyPlot lets you save transformed data to a separate file. If you do not save the data, EasyPlot saves it with the graph. You can turn off the prompt for saving computed data in **Safeties Preferences**; the **save computed data** option affects all tools that operate on curves, such as fits, smoothing, and FFTs.

The checkboxes in the dialog customize how EasyPlot performs transformations. With **xform Visible data only** checked, EasyPlot transforms only data that is visible on the graph; data outside the visible graph remains unchanged. Transforming visible data only lets you zoom on a portion of a data set and transform that portion alone.

Truncate data outside graph removes data outside the graph from the transformed curve. If you want a subset of a large data set, zoom on a portion and apply the transformation “ $y = y$ ” with **Truncate** checked.

Keep original curve leaves the original, untransformed curve on the graph; if **Keep** is off, the new curve replaces the original. This affects only curve transforms; if you transform the entire graph, EasyPlot leaves the original graph.

You can add up to 10 often-used transform equations to the listbox. Type the equation, click on **add >>**, and it appears in the listbox. Equations can have up to 250 characters. To remove an equation, click on it in the listbox and select **<< del**. EasyPlot remembers equations for future sessions.

You enter transform equations the same way you enter curve-generating equations. With transformations, however, you can define a variable as a function of

itself, such as $y = 2^y$ (type as **y = 2^y**), or $t = \cos(t)$. Section 3.7 discusses equation syntax and lists the functions and constants you can use in equations.

You do not need to specify the left side of the equation, the $y=$, or $x=$. EasyPlot assumes the destination if you do not provide one. If you enter a function in the form of $f(y)$ or $f(x)$, EasyPlot assumes $y = f(y)$ or $x = f(x)$. Likewise, with polar equations, typing $f(r)$ or $f(t)$ is equivalent to typing $r = f(r)$ or $t = f(t)$. For transformations involving both variables, $f(x, y)$ or $f(r, t)$, you do need to specify the left side of the equation.

You can reference the minimum and maximum values of a curve in transform equations with the CYMIN and CYMAX macros. Use for normalizing curves, for example; the equation $y = (y - \text{CYMIN}) / (\text{CYMAX} - \text{CYMIN})$ transforms y -values of any data set to the range 0 to 1. If you have run the **Mean, Std Dev** button (§4.6), you can also use the MEAN and SDEV macros in transform equations.

For translating curves, you can enter functions such as $y = y + 5$, or $x = x - 50$. To move a curve visually and without worrying about the exact number of units shifted, hold **<shift><ctrl>** and you can click and drag the entire curve (§4.5).

4.2.1 Computing Error Bars

You can use transformations to compute and/or adjust the size of error bars for a data set. The equation $e = 0.05$, for example, assigns each point an error value of 0.05. In its default mode of interpreting error data, EasyPlot multiplies the error value by the y -value to compute the full size of an error bar. If your equations compute the full size of error bars or only the up/down values, set the appropriate buttons in **Error bars Preferences**. The MEAN and SDEV macros (§4.6) can be useful for computing error bars.

4.2.2 Math on Multiple Curves

The transformation feature can perform math on multiple curves. You can add curves together, divide one by another, or perform any other mathematical function on up to ten curves at once.

You must first tag each curve with a number. Hold the space bar or right mouse button so that the curve you want to tag glimmers and, while it is glimmering, hit a number from 1 to 9. EasyPlot tags the curve with the number. Tag each curve you want to reference with a different number. If you use the right mouse button, be sure the cursor isn't on an object before pressing.

Next, pull down **Tools** and select **Transform**; with the EasyPlot-standard menu, use **Edit / Xform**. Choose to transform a **Curve** and EasyPlot prompts you for an equation. In addition to using x and y as variables, you can now use $x1$, $y1$, $x2$, $y2$,..., as in the equation:

$$y = y1 + y2$$

The number following an x or y corresponds to a tag number. You can tag up to ten curves at once. When you enter the equation, you must also have a curve selected, not just tagged. A piece of the selected curve appears in on the help line in the lower-left corner of the EasyPlot window. The currently selected curve is curve 0 and can be referenced as $x0$ and $y0$ or simply x and y . You can reassign tags any number of times.

EasyPlot uses the first point in each curve to compute the first point of the new curve. It uses the second point in each to compute the second point, and so on. It does not correlate x - or y -values. It computes one point for each point in the selected curve. It assigns zero values to any unknown variables. If $y5$ appears in an equation but no curve is tagged with a 5, the variable $y5$ is zero for every computation. Likewise, if curve 2 has fewer data points than the currently selected curve, curve 2 is padded with zeros once all its points have been referenced.

There are three special functions for use in math on multiple curves. They return the point-by-point maximum, minimum, and average of all tagged curves:

max()
min()
avg()

You can tag curves in polar plots, too. Use rs and ts instead of xs and ys . Reference polar curves with variables $r1$, $t1$, $r2$, $t2$,...:

$$r = r1 + r2 / 5$$

$$t = \cos(t1 + t2 + t3) / r4$$

Tagging Curves Automatically

EasyPlot can automatically tag up to 9 curves with their curve numbers. Put a **//tag** command in a batch file or in **profile.ep**. You can then do math on multiple curves in batch mode.

The **//tag** command takes an optional “A” parameter that tells EasyPlot to tag curves automatically every time you do a transformation, in batch mode or interactively. Automatic tagging overrides tags you assign manually. You can turn automatic tagging off with another **//tag** command without the “A”.

If you put a **//tag** command in a file with the data you want to tag and you do not use the “A” option, put the **//tag** after all the data and put a **//nc** (new curve) command between the last row of data and the **//tag**. Without the “A”, the **//tag** command tags only curves that are on the graph and EasyPlot does not put data on a graph until it finishes reading the file or finds a **//nc** command.

After seeing a **//tag** command, all batch commands that operate on a curve (transforms, stats, curve-fits,...) prompt for a curve number, even if there is only one curve on the graph. This makes batch commands work the same way for graphs with one or more than one curve.

4.3 Difference Equations

When calculating or transforming curves, you can reference 10 previous points to create difference equations, useful in digital filtering, for example. The `px()` and `py()` functions return previous x - and y -values respectively. The parameter of the `px()` and `py()` specifies which previous point to take. $N = 0$ (as in `px(0)`) returns the current x - or y -value. $N = 1$ returns the previous point, and so on. N can be any integer from -10 to 10. If $N < 0$, EasyPlot reverses the sign, making $-N$ equivalent to N . If $N > 10$ or N is greater than the number of points that have been processed, `px()` and `py()` produce a math error and the point is discarded. The two equations:

$$y = px(0) + 0.5px(1)$$

$$y = x + 0.5px(1)$$

are equivalent, and, when computed at $x = 0, 1, 2, 3$, and 4 , produce the points:

x	y	
0		error: <code>px(1)</code> undefined
1	1.0	$(1 + 0.5 * 0)$
2	1.5	$(2 + 0.5 * 1)$
3	4.0	$(3 + 0.5 * 2)$
4	5.5	$(4 + 0.5 * 3)$

4.4 Dragging Curves

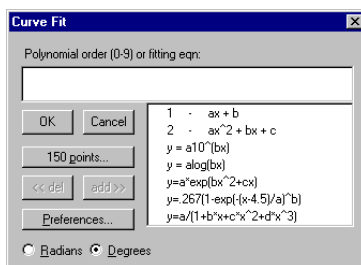
You can pick up and drag curves to translate them. Curve dragging gives you a quick, graphical way to shift a curve along the x - and/or y -axis. It is equivalent to using the **Transform** tool (§4.2) with the functions $x = x + N$, or $y = y + N$. Use

the curve-dragging tool to place a curve visually when you're not concerned about the exact graph coordinates of the data.

Hold **<ctrl><shift>** and click-and-drag on a curve. Once you've grabbed the curve, you can release the **<ctrl>** and **<shift>** keys. To restrict motion to only one direction, hold **<shift>** while moving; you can switch from one direction to the other by dragging the curve back through its original position.

4.5 Curve Fitting

Pull down **Tools** and select **Curve Fit**. Choose a curve if asked and enter or select a fitting equation:



Enter just a number N , from 0 to 9, to compute an N^{th} -order polynomial fit. Enter "1", for example, for a first-order line, $y=ax+b$; enter 2 for you a second-order parabolic fit, $y=ax^2+bx+c$; and so on up to 9.

For nonlinear fits, enter any equation using a , b , c ,... as unknowns. Fit equations can have up to 20 unknowns. Use any letter except those used for defining data: x , y , z , e , r , and t . The equation parser is not case sensitive; A and a are equivalent. You can fit 2-D and 3-D equations of the form $y=f(x)$ or $z=f(x,y)$. EasyPlot prompts you for an initial guess for each unknown. Enter a value or just hit **<enter>** or **OK** to start with 0.

Two nonlinear functions are built into the listbox, an exponential ($y = a10^{bx}$) and a logarithmic fit ($y = a\log(bx)$). You can add up to 10 custom fitting equations to the listbox. Type the equation and click on **add >>**. Equations can have up to 1000 characters. To remove an equation, click on it in the listbox and select **<< del**. EasyPlot remembers equations for future sessions.

EasyPlot has three least-square fitting routines. When you enter a number from 0 to 9, EasyPlot uses a routine optimized for polynomial fits. For nonlinear equations, EasyPlot uses either a downhill-simplex routine or the Marquardt-Levenberg method to determine the best fit. It uses the Marquardt-Levenberg

routine if you have **Curve fit Preferences** set to display parameter uncertainties. Like the polynomial routine, the Marquardt-Levenberg fitter is very fast, but its ability to find a best fit can be sensitive to the model and initial guesses. The downhill-simplex does not generate uncertainty statistics but it can fit just about any equation to any data, even with no or poor initial guesses.

The downhill-simplex routine graphs intermediate stages of the fit so that you can watch as it homes in on a solution. If the fit appears to be homing in on a bad solution, click a mouse button or hit **<esc>** to stop the fit and start over with a different equation or better initial guesses. The simplex searches until the change in parameters from one iteration to the next reduces the fit error by an amount that is below a predefined threshold. You can adjust the threshold to increase or in **Curve fit,2 Preferences**. To increase fit accuracy, decrease the magnitude of the threshold. If you don't need very high precision, you can save time by increasing the threshold value and thus stopping the fit sooner. The downhill-simplex can be fooled by local minima in the error; EasyPlot takes the results of the first pass through the least-squares fit and uses them as initial guesses for a second pass. If your fit equations are fairly simple and a single pass produces good results, you can stop EasyPlot from running the second pass by turning on Skip 2nd pass in **Curve fit,2 Preferences**. If a fit does not look right with one pass or two, try using different initial guesses or a different model, or decrease the fitting tolerance. You can interrupt the simplex fitter by hitting a mouse button or **<esc>**. If EasyPlot is in the first pass of the fit, it proceeds to the second. If in the second, it asks whether you want to compute the curve or discard the fit.

If the curve being fit is displayed with error bars, the downhill simplex weights the fit by the error so that points with large errors affect the fit less than points with small errors. The other fitting routines do not support error weighting. In **Curve fit Preferences**, you can choose whether EasyPlot computes the resulting fit over the entire independent axis or over the range of the fitted data. To project a fit beyond the data, expand the *x*-axis range before doing the fit and compute the fit over the whole graph.

You can save curve-fit data to a separate file after each fit. In **Safeties Preferences**, turn on **Prompt to save... ☐ computed data**. The fit file contains some statistics on the fit and the actual data points computed.

The button labeled **150 points...** sets the number of data points computed for the fit curve. You can compute 2 to 100,000 points. By default, EasyPlot uses 150 samples. Use larger values to better represent curves with many fluctuations. The number of points you select also affects curves generated by equations. See Section 3.7.2 for more details on curve resolution.

EasyPlot places the residual error between the fit and the data in the file **ep.res**. To see the residual, plot **ep.res** after computing the fit. You'll find the file in the directory specified in your system's **EASYPLOT_TEMP**, **TEMP**, or **TMP** environment variable (§7.12).

EasyPlot places the fit equation on the screen as an annotation color-coded to the fit. If you do not want equations on the graph, turn off **Display fit equation** in **Curve fit Preferences**. When computing fits from batch files, you can turn fit equations on or off with the command **//fit_eqn** followed by "on" or "off" .

In **Curve fit Preferences**, you can set the number of significant digits displayed for fit parameters and statistics. EasyPlot uses 3 significant digits by default. For high-order fits, use more significant digits to see results accurately.

EasyPlot can include standard error as part of the fit information displayed on the graph. The standard error is the square root of the variance between the fit and the original data. By default, variance is computed as $\Sigma \text{error}^2 / (N - O - 1)$, where N is the number of data points and O is the order of the fit (# of unknowns). In **Curve fit Preferences**, you can choose to compute variance without subtracting 1 from the denominator. For large data sets, the difference is negligible.

The curve fitting routines can be sensitive to the range of the data, particularly along the independent axis. Fitting 4th-order polynomials over the range $1000 < x < 1001$, for example, can result in curves which look nothing like 4th-order polynomials. This is caused by loss of precision. A quick glance at the polynomial coefficients should let you know whether the fit is numerically sound. If the largest coefficient is more than five or six orders of magnitude greater than the smallest, the fit may suffer from loss of precision, or round-off error. To avoid the problem, use the transformation feature (§4.2) to translate the data so that it is centered around the dependent axis (usually the y-axis) and to scale it so that the range along the independent axis is limited to reasonable values. What "reasonable" is depends on the polynomial order. A 4th-order fit on data running from $-100 < x < 100$ results in computations which span eight orders of magnitude, from 0 to 100^4 , or 10^8 , and can produce significant round-off error. A 2nd-order fit on the same data results in computations which span only four orders of magnitude and should not produce significant round-off error. The equations to translate and scale data are very simple. For the range $100 < x < 110$, use the equation $x = x - 105$ to translate the data. The translation reduces the computational range by more than five orders of magnitude ($\log(110^4) - \log(5^4)$). For the range $-100 < x < 100$, use the equation $x = 0.1x$, $x = 0.01x$, or any similar function which reduces the distance between x_{\min} and x_{\max} .

4.5.1 Splines

Pull down **Tools**, select **Spline**, and choose a curve if asked. EasyPlot computes the spline and, if **Prompt to save...** ☐ **computed data** (in **Safeties Preferences**) is checked, it prompts you for a file in which to save the spline data. The spline file contains the equation for every cubic segment in the spline.

The spline passes through every data point and is constructed from 3rd-order polynomials that match slope and curvature at each data point. A spline fit to N data points is constructed from $N-1$ cubic segments. The curvature at the ends of the spline are set to 0, making it a “natural” cubic spline. For each cubic segment, EasyPlot generates twenty evenly spaced samples.

EasyPlot has two spline routines. One assumes that x -values are uniformly increasing. If the data violates this assumption, EasyPlot automatically switches to the second method. The two methods can produce different splines for the same data. If you get too much overshoot on a spline, you can try the other method by changing the **Basis for Spline computation** in **Spline Preferences**.

If a data set has missing values or unevenly spaced data, the spline routine can compute evenly spaced points. Set the **Point distribution** in **Spline Preferences** to **evenly spaced on X**. Compute a spline and EasyPlot scans the data set to find the smallest x -distance between any two points and computes spline points at that interval. If all the points on the original curve are 10 units apart on x but there are missing values, the spline will be identical to the original curve except it will have interpolated the missing values.

4.5.2 Surface Splines

You can use splines to interpolate additional points for a surface grid. With a surface onscreen, pull down **Tools** and run **Spline**. EasyPlot asks for the:

of X/Y filler points:

Enter the number of points to insert between real data points on the x - and y -axes (two numbers). EasyPlot computes a spline along each row of the surface and plots the result. It then computes a spline along each column of the new surface and plots the results again. It runs a third computation to transpose the surface to the original orientation. If you start with a 10x10 grid and enter “1, 1” for the

number of filler points, the result is a 19x19 grid. The new surface replaces the original; to go back to the original surface, reread the data file.

You can also use the gridding routine (**Tools / XYZ->grid**) to interpolate a finer grid. The gridding routine can also reduce the size of a grid. The spline produces more curvaceous interpolations.

4.6 Statistics

EasyPlot can compute the mean, standard deviation, variance, and the number of data points which fall within $\pm 0-1$, $\pm 1-2$, and $\pm 2-3$ standard deviations from the mean. Pull down **Tools**, highlight **Stats**, and select **Mean, Std Dev...**

Statistics are placed on the screen as an annotation. You can move, edit, or delete the statistics as you would any annotation (§2.7). If you prefer to see standard error instead of the standard deviation, go to the “Integrate+stats” topic in the **Preferences** dialog and select **Show standard error**.

If you want to label an axis with tic marks at the mean and at $\pm n$ standard deviations, double-click on the axis and enter the mean as the coordinate of a major tic mark, and the standard deviation as the distance between labels.

The **Mean, Std Dev...** button sets two macros, MEAN and SDEV, for use in transformation, curve-fitting, or other equations. (With the **Show standard error** selected, the SDEV macro returns the standard error.) If you use MEAN or SDEV without having run the **Mean, Std Dev...** function, the macros return 0. The MEAN and SDEV macros have two counterparts, %mean and %sdev, which access statistics in batch files. The following sequence of batch commands reads a data set and labels the y-axis with major tic marks at the mean and $\pm n$ standard deviations from the mean, with 1 minor-tic between the major tics:

/ac	data2	;read a data set
/tsss		;run statistics on data
/oaas	y %mean %sdev 1	;set up axis labeling

4.7 Histogram

To compute a histogram, pull down **Tools**, highlight **Stats**, and select **Histogram**. You can compute 1- or 2-dimensional histograms. 1-D histograms slice graphs into horizontal bands and count the number of points that fall within each band. Bands are defined by tic marks on the y-axis. With $0 \leq y \leq 100$ and tic marks (major

or minor) at 0, 10, 20, etc., EasyPlot counts how many points have y-coordinates between -5 and 5, between 5 and 15, and so on.

EasyPlot creates a new graph with the histogram data. The x-coordinates on the histogram are the tic-mark locations from the original graph. If 12 points fall between $y=5$ and $y=15$, the histogram will have a point at $x=10$ and $y=12$.

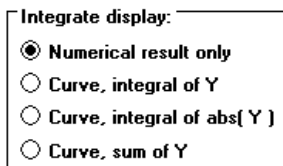
A 2-D histogram slices the graph into a grid and counts the number of points that fall within each box. The result is a surface in which the z-values represents point count. The number of rows in the surface is based on the number of y-axis tic marks on the original graph and the number of columns on the x-axis tic marks.

To specify the number and location of buckets, double-click on an axis and change tic positions (§2.11.2) before running the histogram.

If **Prompt to save... ☐ computed data** (in **Safeties Preferences**) is checked, EasyPlot lets you save the computed histogram data to a separate file.

4.8 Integration

To find the area under a curve or the volume under a surface, pull down **Tools**, highlight **Stats**, and select **Integrate**. EasyPlot uses the trapezoidal rule to sum the area or volume under the curve. The “Integrate+stats” topic of the **Preferences** dialog lets you select what EasyPlot displays:



With **Numerical result only**, EasyPlot displays the final integral value:

integral area: 13685

If any data is negative, producing ‘negative’ area, EasyPlot displays the integral and the absolute area under the curve. To compute the volume under a surface, display the surface as a contour map (§5.3) and run the **Integrate** button. EasyPlot displays the volume on the screen:


integral volume: 0

With the three ‘curve’ options, EasyPlot graphs a curve representing the selected statistic at each point of original data. If **Prompt to save... ☐ computed data** (in **Safeties Preferences**) is checked, EasyPlot lets you save the computed

integration data to a separate file. If you look at the data for an integral curve (**Edit / Data**, or save the data to a file and look at the file), you see four columns:

column:	1	2	3	4
value:	x	$\int f(x) $	$\int f(x)$	Σy

You can select what the integration tool displays from a batch file with the command “**//plot_int N**”, where N specifies which y-column to plot; $N = 1$ plots column 2, $N = 2$ plots column 3, and $N = 3$ plots column 4. If $N > 3$ or $N < 1$, EasyPlot does not plot the integral data.

You can subtract a baseline from integrations. Go to the “Integrate+stats” topic of the **Preferences** dialog and turn on **baseline subtraction**. Draw a line on the graph (**Add / Line** or click ) that represents the baseline. If there is one user-drawn line, EasyPlot uses it as the baseline. If there are multiple lines, select one (click on it so that a small box appears at each endpoint) before integrating. If there are no user-drawn lines or if there are multiple lines and one is not selected, EasyPlot computes the integral with the default baseline of $y = 0$. Use the caliper tool (§2.9) to see the slope and intercepts of the baseline. If you save the integration data to a separate file, the file also contains the equation of the baseline. You can move the baseline and recompute the integral. The baseline can be any length; it does not have to extend across the entire graph.

4.9 Differentiation


To compute the derivative of a curve, pull down **Tools**, highlight **Stats**, and select **Differentiate**. EasyPlot has two methods for computing derivatives. One fits a spline to the data and plots the derivative of the spline. Since the spline defines a continuous curve, the spline method can interpolate a smooth curve between points. The second method computes the straight-line slope from one point to the next. For the first point, EasyPlot uses the slope to the second point. For each subsequent point, it uses the slope from the previous point. (The first two points always show the same slope.)

Select the computation method in **Differentiation Preferences** dialog. EasyPlot uses the spline by default. With the spline derivative, EasyPlot asks for the number of filler points. If you enter 0, the derivative curve will have 1 point for each point on the original curve. If you enter $N > 0$, EasyPlot interpolates N points between each pair of original points.

If the spline is not well behaved for a data set (if it has a lot of overshoot), its slope will not accurately represent the slope of the real data. To check, compute a

spline on the original data and see whether it follows the data smoothly. Or, compute the straight-line derivative and confirm that the values are similar.

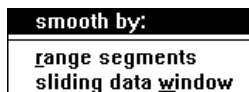
When finished computing, EasyPlot lets you save the derivative data to a separate file. In **Safeties Preferences**, you can turn off the save prompt.

If you need the slope only at a particular point, you can use the line-drawing feature (**Add / Line** or click ) to sketch a line on the graph and hit 's', 'm', or 'l' to get the slope and length of the line (§2.9). Draw the line tangent to the curve to get the slope.

4.10 Data Smoothing, Range Averaging & Decimation

EasyPlot has two smoothing algorithms. One uses a sliding average window and produces one point for every original point. The other groups points by ranges of x -values and produces one point that is the minimum, maximum, or average of all the points in each range segment, reducing the size of the data set. Use range-averaging to summarize data or to reduce the size of large data sets, such as turning a year of daily samples (365 points) into weekly or monthly averages.

Pull down **Tools**, select **Smooth**, pick a curve if asked, and select the smoothing method:



If you select **sliding data window**, EasyPlot asks you to specify the size of the smoothing window. Enter any positive, nonzero integer. EasyPlot computes a curve by replacing each original data point by the average of it and $N-1$ of its neighbors, where N is the size of the smoothing window. EasyPlot centers the smoothing window on the data points. When the window extends off the end of the data set, EasyPlot averages fewer than N points; it does not pad the data. At the first and last data point, it averages only $N/2$ data points. The smoothed curve has the same number of points as the original curve.

If you select range segment smoothing, EasyPlot prompts you for the:

start pt & bucket width:

Enter two numbers, the minimum x -coordinate to start bucketing points, and the width of the buckets. Separate the numbers with a space or a comma. If your x -axis is day-of-year (from 1 to 365), for example, and you want weekly averages,

use 1 for the start point and 7 for bucket width. The resulting data set has one point for each bucket. The x -values are centered on the buckets and the y -values are the average of all the data points whose x -values fall within the bucket.

By default, the range-smooth feature returns the average of all points within each range segment. If you want the maximum or minimum value instead of the average, go to the **Smoothing** topic of the **Preferences** dialog.

EasyPlot adds the smoothed curve to the graph and lets you save the computed data in a separate file. You can turn off the save prompt in the “Safeties” topic of the **Preferences** dialog.

4.11 Interpolating Evenly Spaced Data

If you have a data set with unevenly spaced x -values, the **Smooth** routine can create a new data set with equally spaced points. Pull down **Tools**, run **Smooth**, and use the sliding window. Enter a window size of 0. EasyPlot scans the data set to find the smallest x -distance (dx_{\min}) between any two points and determines the number of evenly spaced points, N , required to fully sample the original data, where $N = (x_{\max} - x_{\min})/dx_{\min}$. EasyPlot prompts you to confirm the computation of the N data points. EasyPlot uses linear interpolation to compute the points.

You can also use the Spline routine to interpolate evenly spaced points (§4.5.1). The spline produces smoother, more curvaceous interpolations.

4.12 Fourier Transform

Pull down **Tools**, highlight **FFT**, and select **Forward**. EasyPlot asks whether to apply a window to the data:

data window:
<u>H</u> amming
Kaiser
<u>n</u> one

For a Kaiser window, specify the fall-off in dB from the peaks to the sidelobes:

sidelobe reduction (in dB):

EasyPlot plots the FFT in a new graph, leaving the original graph intact, and lets

you save the FFT data. You can turn the save prompt off in the **Safeties** topic of the **Preferences** dialog.

If the number of data points is not an even power of two, EasyPlot pads with $M - N$ zeros, where N is the number of data points and M is the first power of two greater than or equal to N .

You can run FFTs on real or real and imaginary data. From xy plots, EasyPlot provides only real data (the y -values) to the FFT routine. To compute a complex FFT, you must put your data in polar form, specifying a magnitude (radius) and phase (theta). Define the columns with rs and ts (§3.4) and you should see a polar plot. From polar plots, EasyPlot converts data to real and imaginary parts, $a + bi$, where $a = r \cos(t)$, and $b = r \sin(t)$, and it uses the real and imaginary parts as input to FFT.

The second half of an FFT computed on real data is a mirror image of the first half. To ignore the redundant data, go to **FFT Preferences** and turn on **Plot only 1st half of FFT**. EasyPlot then graphs the first half of FFT data. This option does not affect FFTs you already computed.

The FFT data has four columns, magnitude, real and imaginary parts, and the angle of the FFT. EasyPlot graphs the magnitude of the FFT. To see the magnitude in dB, transform (§4.2) the FFT with the equation $y = 20 \log(y)$. To plot other columns, run **Define Data** (under **Tools** – §3.4) or **Edit the Data** and turn columns on or off in the Data Table (§7.9.2).

The **Inverse** button (under **FFT**) is the same as the **Forward** FFT button except that it computes the inverse FFT.

EasyPlot can label the x -axis of an FFT with point number or frequency. To display frequency, go to **FFT Preferences** and select the Frequency radio button. You also need to set the **Time scaler** to describe the units of time the x -axis on the original graph. Enter 1, for example, if the original x -axis reads seconds, or 1000 if it reads milliseconds.

4.13 Analyzing Frequency Change Over Time

If you want to analyze frequency change over time in a signal, you can compute FFTs on consecutive segments of a data set. Go to **FFT Preferences** and enter a segment size, N , where N is a power of 2. Choose the resulting display: multiple xy curves, or a surface in which each row is the FFT of a segment. Pull down **Tools** and run **FFT**. EasyPlot computes an N -point FFT with the first N points, another with the next N points, and so on until it reaches the end of the data set.

EasyPlot computes FFT segments only if there are enough data points for two segments; otherwise it computes one FFT with the entire data set. To go back to computing FFTs on entire data sets, reset N in **FFT Preferences** to 0.

4.14 Reading Coordinates

To read x - and y -coordinates from a graph, pull down **Tools** and select **X-hair**. Two coordinate windows appear above the graph. When the cursor is inside the graph, the coordinate windows display the x - and y -position of the cursor. You can improve the precision of the reading by zooming on the point of interest. To turn the cross-hair off, select the **X-hair** button again.

The **X-hair** tool has a “Lock to curve” feature. Turn it on or off in **X-hair Preferences**, or by right clicking on one of the **X-hair** windows. With the locking feature on, click on a curve and the X-hair stays on the selected curve as you move left and right within the range of the curve data.

4.15 Digitizing from Screen to File

You can use the graphical data editor to pick points from a curve and write their x - and y -coordinates to a file. To initialize the digitizing feature, you need to open a digitizing file. Pull down **File**, choose **Open**, and type:

//dig filename [A] [X/Y]

where “filename” is the name of the file in which to place the data. You must be using the EasyPlot-standard **Open** dialog to open a digitizing file; go to **User Interface Preferences** if you are using the Explorer-style **Open** dialog. The “A” is an optional parameter that tells EasyPlot to append data to the given file. The “X” or “Y” is another optional parameter that tells EasyPlot to write only the x - or y -coordinate into the file.

Use the graphical data editor (§7.9.1) to select a point and, while holding it selected, hit the letter **d**, for “digitize”. If you move the point, EasyPlot returns it to its original position when you hit ‘**d**’ provided you have not released the mouse button. EasyPlot places the coordinates of the point in the file and displays the coordinates on the help line.

EasyPlot closes the file when you leave the graphical editor (by hitting the **<esc>** key) or when you enter another **//dig** command.

Chapter 5: 3-D

EasyPlot can display 3-D data as 2-D contour maps, 2-D image plots, or as 3-D wireframes or contours. The **Three D** button (under **Style**) toggles between 2-D and 3-D representations. In 3-D mode, 3-D user-interface buttons appear along the left side of the screen (Figure 5.1).

If you have a 2-D contour map or image plot on the screen, EasyPlot displays a 3-D fishnet surface when you turn on the 3-D style. With an xy plot, you get the same curve or curves but with a third dimension. If an xy curve does not have a column of z -values associated with it (§3.4), EasyPlot sets $z=0$ for every point.

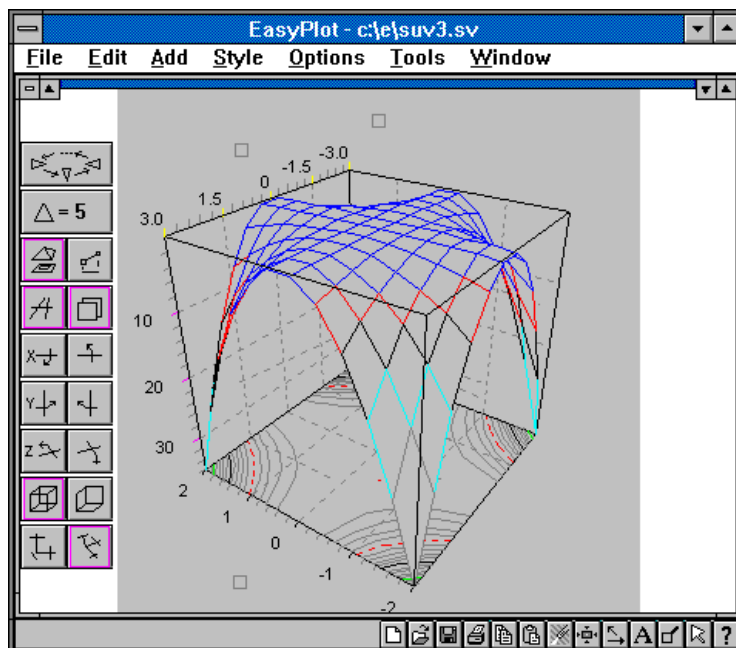


Figure 5.1 The 3-D User Interface

5.1 3-D Data

EasyPlot handles two types of 3-D data, surfaces and xyz triplets. Figure 5.1 contains a surface. Surfaces come from matrices in which every value is a z -value; x - and y -coordinates are implied by position in the matrix. Surface matrices can be any rectangular size of up to 1024 columns. See Section 3.1 for more details on setting up surface data files.

If a file has 20 or more columns, EasyPlot assumes it is a surface and defines the entire data set with a single “**z**”. To graph a surface matrix of fewer than 20 columns, put the command “**/td z**” in the file; or plot the file, run **Define Data** (under **Tools** – §3.4), and replace the “**xyyy...**” with “**z**”.

In 3-D mode, EasyPlot graphs fishnet grids (Figure 5.1) or floating contours (Figure 5.3) from surface data. With fishnets, it can shadow contours onto the top or bottom xy plane (Figure 5.1). In 2D, EasyPlot generates contour maps and image plots from surface data.

Surfaces grids can have missing values. Put a “**//m**” (§6.3) wherever a value is missing. With EasyPlot’s Data Table, blank cells are considered missing values if **Autofill Missing Values** is on (under the Data Table **File** menu). EasyPlot’s gridding routine interpolates the missing values.

You can graph three-column xyz data to create 3-D scatter or line graphs. Enter or read in data with xyz columns and define the columns in the Data Table (§7.9.2) or with **Define Data** (under **Tools** – §3.4). You choose data display styles (connect points, markers, etc. – §2.2) just as on 2D xy graphs. EasyPlot has a gridding routine (**Tools / XYZ->Grid**) that converts xyz data to surface grids, letting you plot surfaces and contour maps from random xyz data.

You can plot any number of data sets, surface or xyz -triplets, on 3-D graphs.

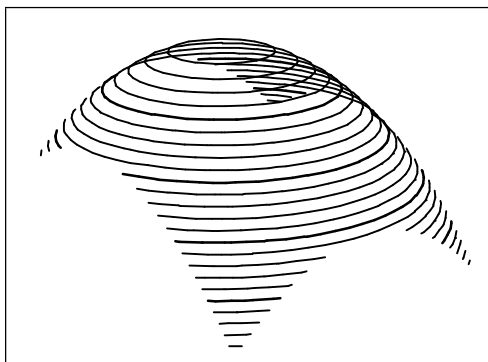
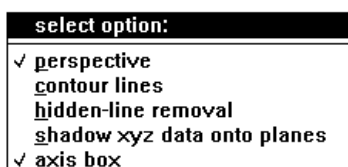


Figure 5.2 Floating 3-D contours

5.2 3-D Graphs

In 3-D mode, you can rotate data to any orientation, turn perspective on or off, draw the data with or without axes, and even animate a rotation. Most of the 3-D buttons replot the data. If you have a lot of data and want to rotate several times or run other buttons before looking at the data, hit **<esc>** or a mouse button after each operation to interrupt the plotting.

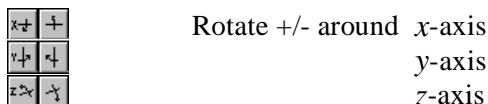
Several features controlled by 3-D buttons can also be turned on or off by running **Three D Options** under **Style**. EasyPlot displays the options:




When EasyPlot saves 3-D graphs, it uses the **Three D Options** dialog (with **/stt** commands) to set up the graph. You can use the same commands if you create 3-D graphs from batch files (§6.5).



5.2.1 Rotating 3-D graphs

To rotate data, click on one of the six axis-rotation buttons:




EasyPlot rotates the data in the selected direction by the number of degrees displayed in the **Δ=5** button. Click on 'Δ = n' to change the rotation increment; select a value from the listbox or enter any integer. The angle is in degrees.

The "animate" button () repeats the last rotation until you hit a key or mouse button. The rotation increment ('Δ = n') controls the speed of the rotation. Use larger angles to turn faster or smaller angles to move slower.

You can rotate relative to the screen () or data () axes. The screen x-axis is horizontal, its y-axis vertical, and z runs in and out of the screen. EasyPlot uses data-relative rotations by default.

5.2.2 Perspective



The  button toggles perspective on and off. You can adjust the amount of perspective with which EasyPlot draws the data by holding **<alt>** and clicking on

the perspective button. EasyPlot prompts you to enter a scale factor:


scale viewing distance by: [1.00]

The number in parentheses is the current scale factor. Enter $n < 1$ to increase perspective effect or $n > 1$ decrease it. The default scale is 1.


5.2.3 Color

EasyPlot uses color to signify depth in a surface by assigning a color to each line of a surface based on its location on the z -axis. With screen-relative rotation () , EasyPlot uses the screen z -axis, or the distance from your eye, to assign colors. With object-relative rotation () , it uses the actual z -values. Blue and red color the surface where z -values are largest, magenta and gray where smallest.

5.2.4 Hidden-Line Removal

Click on  to toggle hidden-line removal on or off. With hidden-line removal on, EasyPlot draws surfaces from back to front, filling the grid. Filling grid cells slows redraws. With larger surfaces, you can speed your work by turning hidden-line removal off until the surface is oriented correctly or ready to print.


5.2.5 Contour Lines on 3-D Graph

You can draw contour lines on 3-D surface plots shadowed at the top or bottom of the graph or at the actual z -coordinates in place of the fishnet grid. With a surface onscreen, click on  and EasyPlot asks where you want the contour lines:

3D contours:
<u>off</u>
at z-min
at z-max
at z-coor

Choose **at z-min** or **at z-max** to draw contour lines at the top or bottom of the box along with the fishnet grid. If you choose **at z-coor**, contour lines replace the fishnet grid. You can adjust the number and location of contour lines by double clicking on the z -axis; major and minor contour lines appear at the same coordinates as major and minor z -axis tic marks. EasyPlot draws contour lines from surface data only; to draw contours from three-column xyz data, use the gridding routine (**Tools, XYZ->Grid**) to convert the data to a surface.

5.2.6 Shadowing XYZ Data onto Axis Planes

You can project xyz data onto each of the 3 coordinate planes (Figure 5.3). The  button toggles this feature on or off. The shadow lines are drawn in gray regardless of the color of the original curve. This feature works for xyz data only; it does not work with surface data.

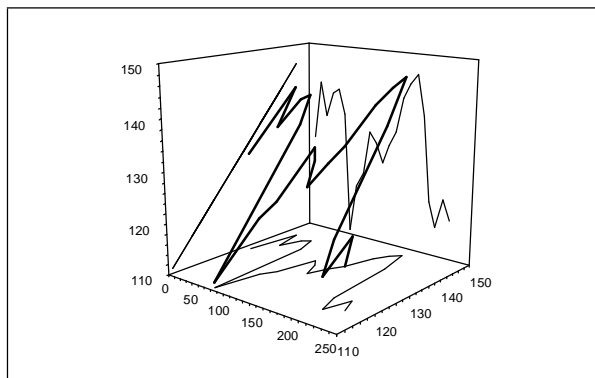




Figure 5.3 Shadowing xyz data onto back planes

5.2.7 Axes & Surrounding Boxes

You can draw a full box around 3-D data (Figure 5.1), the back three planes of a box (Figure 5.3), or no box at all (Figure 5.3). The  and  buttons toggle full and rear boxes on and off. With back planes only, you may get the entire box if EasyPlot cannot determine which planes are in the back; rotate the graph a few degrees if you get the entire box. EasyPlot draws axis tic marks along edges of the box. If you turn the box completely off (click on whichever box button is highlighted), you also remove the axes.

Double click on 3-D axes to set ranges and tic mark placement just as you would on 2-D xy graphs. EasyPlot does not support log or reverse-direction axes on 3-D graphs. EasyPlot chooses the edges on which axes appear based on the orientation of the box. If you do not like where EasyPlot puts the axes, rotate the graph a few degrees.

To title 3-D axes, click on the small title boxes next the axes or on the existing title, as you would on 2-D graphs. With small graph windows, a title box may end up under the 3-D interface buttons; rotate the box so that the title button is accessible, enter the title, and rotate back.

5.3 Contour Plots

To create a contour map as in Figure 5.3, plot a surface grid (§5.1) or plot xyz data and convert it to a grid (§5.5). If the graph is in 3-D mode, pull down **Style** and choose **Three D**. For surfaces displayed in 2-D, the **Connect Pts** style turns contour lines on or off; the **Mark Pts** style turns color maps, or image plots, on or off. If you do not see contour lines or if the graph is filled with colors, pull down **Style**, turn **Connect Pts** on, **Mark Pts** off, and **Restyle** data.

Contour lines are equivalent to tic-marks on the z -axis. Major contours appear at major tic locations and each has a unique color. Minor contours appear in gray. EasyPlot chooses the contour levels automatically but you can override any of its choices. The **Auto-tics** buttons (under **Style** in the EasyPlot-standard menu – §2.11.2) give you the same control over the placement of contours as they do over x - and y -tic marks. Or, you can switch to 3-D mode, double click on the z -axis, and position z tic marks in the **Axis Setup** dialog. You can even add floating contour lines just as you would add floating tic marks (§2.11.4).

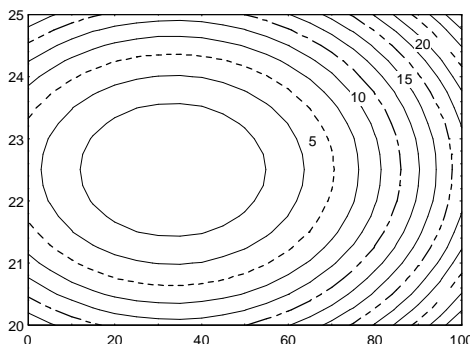


Figure 5.4 Contour map

Major contours can be drawn solid or dashed. Pull down **Style**, set **Dash**, and **Restyle** the data. When you print, EasyPlot draws major contour with heavier lines than minor contours.

EasyPlot places labels for the major contours outside the top-right corner of the graph. It draws labels in the same colors as the contours they describe. The labels are placed on the screen as annotations (§2.7). You can drag the labels around the graph to place them next to their respective contours. If you need another set of labels, pull down **Options** and select **Legend**. With a contour map on the screen, the **Legend** button generates a new set of contour labels.

EasyPlot can generate contour maps from fairly large data arrays. Arrays on the order of 512x512 approach the practical limit of EasyPlot's (and the PC's) data handling capabilities. You can read up to 1024 columns and any number of rows. If your surface files have more than 150 columns, go to **File reading Preferences** and make sure the **Max # of data columns** is greater than the number of columns in your file.

By default, EasyPlot allows up to 35 major and minor contour lines on a single plot. If you want more than 35 gradations, place the command

```
//max_conts N
```

in the **profile.ep** file, where N is the maximum number of contour lines. EasyPlot reserves a block of memory with size proportional to N for contour-map computations. Use modest values of N (less than 100 or 200).

5.4 Image Plots

To create a color map or image plot (Figure 5.3), plot a surface grid or xyz data and convert it to a grid (§5.5). If the graph is in 3-D mode, pull down **Style** and choose **Three D**. For surfaces displayed in 2-D, the **Mark Pts** style turns color maps, or image plots, on or off. If you see contour lines or if the graph is blank, pull down **Style**, turn **Mark Pts** on, **Connect Pts** off, and **Restyle** the data. EasyPlot fills the graph with a color or gray scale based on surface height.

You can draw image plots using color or gray scales to represent height. To choose, go to **Image Plot Preferences**. For color images, EasyPlot can produce approximately 1000 gradations from red, to green, to blue. Red represents the highest values, green the median values, and blue the smallest. For gray-scale images, you get 256 gradations. The number of colors or gray scales you see onscreen and printouts depends on the device and driver you use and, of course, the data you graph.

5.5 Gridding

EasyPlot's gridding routine converts random xyz data to evenly spaced, rectangular surface grids. EasyPlot draws contour maps, image plots, and 3-D fishnet surfaces from surface grids only. If you have xyz triplets that you want to plot as a 3-D surface, contour map, or image plot, you must first use the gridding routine to convert the data to a surface grid. If all the points are linear in the xy plane, however, EasyPlot will not be able to grid the data.

The gridding routine can also reduce or increase the size of existing surfaces.

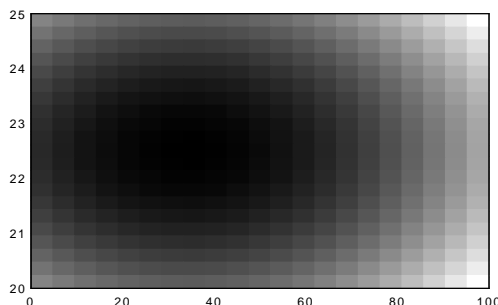


Figure 5.5 Image plot

If you have very large arrays, the gridding routine can average rows and columns to produce a smaller grid that takes less time to replot. If you have small surface data sets, the gridding routine can interpolate a larger grid. The **Spline** can also interpolate additional points for a surface; when expanding grids by a factor of 3 or 4, the spline produces a smoother, more curvaceous surface.

To run the gridding routine, pull down **Tools** and choose **XYZ->Grid**. Enter the number of rows and columns to create (10x10, 50x50,...). EasyPlot interpolates an evenly-spaced grid of data and plots the result on the current graph. If you read in a surface that has missing values ("**//m**" – §6.3), EasyPlot automatically uses the gridding routine to interpolate missing values.

5.6 Color as a 3rd or 4th Dimension

You can use a column of data to determine the data marker color. If you have *xyz* triplets, for example, and at each point you have a temperature measurement, you can give EasyPlot four columns of data and define the temperature as '**c**':

/td	"xyzc"		
X-coor	Y-coor	Z-coor	Temp
10	12	15	18
20	118	34	96
30	85	72	244

Display the data with **Mark Pts** on (in 2-D or 3-D view) and EasyPlot draws the markers different colors based on the '**c**' value. The minimum '**c**' value is drawn in blue, the median in green, and the highest value in red, with cyan and magenta in between. EasyPlot supports 1000 different colors along the blue to red range. The number of gradations you see depends on your video driver and/or printer.

Chapter 6: Batch & Programming Interfaces

EasyPlot has a complete batch language that lets you speed repetitive tasks and automate graph production. It also has a programming interface that lets you use EasyPlot as though it were a subroutine of your own Windows applications

6.1 Batch Command Overview

There are two types of batch commands, menu and non-menu. Menu commands begin with a single forward slash. Non-menu commands begin with two slashes. The slash, “/”, is EasyPlot’s command character.

Most menu buttons in EasyPlot have a batch-command equivalent. Menu commands are made up of the same letters you would type onscreen to pull down a menu, run a button, and answer prompts. Non-menu commands provide functions specific to batch processing. Appendix A lists the more commonly used menu commands. Appendix B lists all the non-menu commands.

You can place batch commands in any ASCII or spreadsheet file you read into EasyPlot. You can enter double-slash commands in the EasyPlot-standard **Open** dialog. Files can contain batch commands, data, or any combination of the two. Place each command on a separate line; EasyPlot ignores the remainder of a line after finding a command.

The best way to become familiar with batch commands is to create a graph, save it, and look at the save file. Save files are fully commented batch files containing all the commands needed to recreate the graph.

6.2 Menu Commands

Most menu buttons have batch-command equivalents. Menu commands consist of a forward slash followed by two or more letters, the same letters you would type onscreen to pull down a menu and run a button.

For compatibility with earlier versions of EasyPlot, batch commands are based on the EasyPlot-standard menu (§7.8). You do not need to have the EasyPlot-standard menu selected when you read batch files. EasyPlot ignores batch commands based on buttons that are unique to the Windows menu (such as **Add, Function**), or in a different place from their counterparts in the EasyPlot-standard menu (such as **Tools / Transform** vs. **Edit / Xform**).

The first letter after the “/” tells EasyPlot which menu to pull down. (In batch mode, the menus are never really displayed.) A **/f** opens the **File** menu, **/e** the **Edit** menu, and so on. The second and subsequent letters specify which button in the menu to run. The commands **/ag** and **/ac** pull down the **Add** menu; **/ag** adds a **Graph** and **/ac** adds a **Curve**.

When buttons in a menu share the same first letter, you may need a third or fourth letter after the slash: **/OL**, for example, runs the first button under **Options** beginning with **L**, **Log**, and **/OLL** runs the second **L** button, **Legend**. A third **L** would highlight the third button beginning with **L**, if there were one.

Some buttons have sub-buttons which appear when the parent is highlighted. To access these, start with the letter sequence which highlights the parent button and add the letter or letters you need to run the sub-button. **Auto Tics** (under **Options** in the EasyPlot-standard menu), for example, has three sub-buttons; **/oaam** runs **Min Tics** and **/oaamm** runs **Maj Tics**. If a sequence of letters does not lead to a button, EasyPlot ignores the command.

In batch mode, EasyPlot reads information needed to run a function from the file. The **Log** button, for example, prompts you to select an axis. In batch mode, EasyPlot looks for an **x** or **y** in the file directly after the command:

```
/ol x ;set x-axis to log scale
```

The **Range** button requires you to select an axis and provide two numbers, a minimum and maximum. The two numbers should follow the axis selection:

```
/or x -50 50 ;set range on x-axis  
/or y 100 200 ;set range on y-axis
```

All input for a function must be on the same line as the command. If a command reaches the end of a line before finding the input it needs, it quits without doing

anything. Figure 6.1 lists a few sample batch commands. Look at and edit

/ed	g		;delete current graph
/ag	y=x^2	-2 2	;plot x^2 from -2 to 2 on a new graph
/ac	data3		;add curve(s) from file data3
/td	rt		;define data as Radius, Theta
/ol	x	ON	;set X-axis to log scale
/og	ON		;turn grid on
/oaam	x	OFF	;turn minor tics on X-axis off

Figure 6.1 A few example batch commands

EasyPlot save files with any text editor for hands-on experience with batch files.

The last three examples in Figure 6.1 have an optional “ON” or “OFF” parameter. All the functions that act as toggles look for this when run in batch mode. If no “ON” or “OFF” is found, the functions act as they would interactively: they toggle their features from on to off or vice versa. The optional parameter lets you specify the state for a feature. It is good practice to provide the state to ensure the command does what you want and to make your batch commands more readable. Commands and parameters can be upper or lower case.

6.3 Non-Menu Commands

There are many batch commands which are not part of the menu system. Some are particular to loading data, some to click-and-drag features (such as resizing or moving a graph), and some are specific to writing graph templates. Appendix B lists all the non-menu, double-slash commands.

Non-menu commands begin with two slashes instead of one. Normally, you would put double-slash commands in batch files. You can also type double-slash commands directly into the EasyPlot-standard **Open** dialog. (The Explorer-style **Open** dialog supports opening files only.)

6.3.1 Data File Commands

Missing Data Points: If a file has missing values, put a **//m** wherever there is a missing value. For multiple-column files, the **//m** keeps EasyPlot from ignoring an entire line of data when only one or two of the values are missing.

The Data Table can automatically interpret empty cells as missing values. Pull down **File** (in the Data Table) and turn on **Autofill Missing Values**. When you plot data in the table, EasyPlot fills empty cells inside the data matrix with the **//m**.

By default, EasyPlot connects the point before a missing value to the point after. You can break curves at missing values by turning on **Break curves at missing pts** in the **Line charts** topic of the **Preferences** dialog.

Splitting a Data File: A data file can contain any number of data sets. You can use multiple columns to represent multiple data sets. Or you can use the **//nc**, or “he w curve” command to split a file into sections, where each section represents a distinct data set. When EasyPlot finds the **//nc** command, it stops reading the file, installs whatever data it has read, and resumes reading the file as though starting a new file. You can change the number of columns after a **//nc** command.

Infinity: If a data set has a discontinuity which goes to positive or negative infinity, you can represent it with the **//inf+** or **//inf-** commands. Use the commands instead of very large numbers. You can use infinity commands in the x - or y -direction. Adjacent infinity values are not connected. No matter how far you stretch the range, lines going to infinity remain vertical or horizontal. The autorange feature ignores infinity points when setting axis ranges. Math tools (curve-fits and statistics, for example) also ignore infinity values.

X and Y Contour Range: EasyPlot generates contour maps from matrices of z -values (§5.1) and the x - and y -coordinates are not part of matrices. If you want the x - and y -axes to display particular ranges, you must specify them in the data file with the commands:

```
//cx      xmin      xmin
//cy      ymin      ymax
```

The **//cx** and **//cy** label the x - and y -axes of contour maps and 3D surface plots with the given range. Without **//cx** or **//cy**, EasyPlot labels axes with row and column numbers. A 21x21 matrix, for example, would have axes labeled 0 to 20.

6.3.2 Batch Macros

There are two batch commands for defining macros in batch files:

```
//macro  n    “string”
//input  n    “prompt”  [“initial value”]  [-file]
```

The n selects a macro number from 1 to 20. The **//macro** command assigns the given string to macro n . You can also use the **//macro** command to set macros

from the N^{th} string in a specified file as described below. The **//input** command displays the given prompt and assigns what you type to macro n . You can specify an initial value for the string. You can also use the **Open** dialog by adding the “-file” parameter.

After defining a macro, you use it by placing %1 or %2, for example, anywhere in a batch file. EasyPlot replaces the % n with the text assigned to macro n . The commands:

```
//macro 1 "data2" ;define macro 1 as "data2"
/ac %1 ;read file defined by macro 1
/ag %1 ;read file defined by macro 1
```

define macro 1 as the name of a file and then reads the file into two graphs.

You can redefine macros any number of times and they can have up to 100 characters. Macros remain defined until you quit from EasyPlot.

EasyPlot has a number of built-in macros, or constants, which provide access to curve and graph statistics (Table 6.1). The **%mean** and **%sdev** can be used

%mean, MEAN	mean of a data set
%sdev, SDEV	standard deviation of data set
CXMIN, CYMIN, CZMIN, CXMAX, CYMAX, CZMAX	min & max x, y, z values of selected data set
DXMIN, DYMIN, DZMIN, DXMAX, DYMAX, DZMAX	min & max x, y, z values of all data on graph
XMIN, XMAX	x -axis min & max
YMIN, YMAX	y -axis min & max
PTN	point-number counter
PTS	# of points in selected curve

Table 6.1 Equation Constants and Macros

anywhere in a batch file (except as part of a quoted string). All the others are used

in equations. The **%mean**, **%sdev**, and **PTS** return values computed by the last **Mean, Std Dev...** (§4.6) operation; if you have not computed statistics for a curve, they return 0. Here is an example of how you can use the built-in macros:

```

/ac      data2           ;read a data set
/tsssm                   ;run statistics on the data
/ac      "y=CYMAX"       ;draw line at  $y_{\max}$ 
/ac      "y=CYMIN"       ;draw line at  $y_{\min}$ 
/at y    %mean =         ;add major tic at mean

```

Extracting Text from Files: You can set a macro to be the N^{th} string of a file. You can then place the string on a graph as a title, annotation, etc. The syntax for the command is:

```
//macro  m  -file  filename  N
```

where m is the macro number. The “-file” tells EasyPlot to scan the specified file for a string. N is an integer that tells EasyPlot which string to read. If $N = 1$, EasyPlot takes the first string. If $N = 2$, it takes the second, and so on. EasyPlot counts numbers as strings.

You can concatenate strings (§6.7) to insert macro strings (be they extracted from files or not) into titles, annotations, equations, etc. If your data files all begin with the comment “creation date:” followed by a date, “01/01/92”, for example, you can extract the third string from the file and put the date on the graph:

```

//macro 1 -file run5.dat 3      ;macro 1=3rd string in run5.dat
/aa      100 55 "data date:" + % 1 ;put date on graph as note

```

6.4 The “profile.ep” File

On start-up, EasyPlot looks in the directory where the program resides (or in the alternate setup directory you specify – §7.12) for a file called **profile.ep** and, if found, it reads the file as though it were a data file. You can put batch commands in the file to customize EasyPlot. Figure 6.2 lists a sample **profile.ep** file which contains a few of commands one might want executed automatically. See Section 6.1 for a general description of batch commands. See Appendices A and B for lists and brief descriptions of all the batch commands. EasyPlot uses the **profile.ep** file to store curve-fit and transform equations that you save for future sessions.

/sm	OFF	;turn data marks off
/sc	ON	;connect points with a line
/sd	OFF	;turn off dash style
/st	ON	;turn 3D mode on

Figure 6.2 Sample **profile.ep** file

6.5 Automating EasyPlot to Speed Your Work

Most buttons in the EasyPlot-standard menu (§7.8) have a batch-command equivalents. You can use batch commands to speed repetitive tasks such as titling and setting ranges. Figure 6.3 lists a short batch file that sets up some graph

/et g	"Power Curve	;title the graph with
%date, %time"		; date & time on 2nd line
/et x	"motor RPM"	;title x-axis
/et y	"torque, ft-lbs"	;title y-axis
/or x	0 6000	;set x-range
/or y	0 100	;set y-range
/oaas y	0 20 1	;set y-tic marks

Figure 6.3 Sample template file – **template.ep**

basics. Read the file into EasyPlot along with a data file to create a graph. If you create many similar graphs, you can set an F-key macro (§6.6) to read a template file and have the template file prompt you for a data file. For example, name the file in Figure 6.3 "**template.ep**" and name the file in Figure 6.4 "**stdgrph.ep**". In

/ag		;open new graph
/ac	template.ep	;load template (Figure 6.3)
//input 1	"data file:" -file	;prompt for data file
/ac	%1	;read data file

Figure 6.4 Sample batch file – **stdgrph.ep**

your **profile.ep** file, add the line:

```
//fkey 1 "/ac c:\lep\stdgrph.ep"
```

and restart EasyPlot. (See Section 6.6 for details on setting FKey macros.) Now, when you hit **F1**, EasyPlot reads the file **stdgrph.ep**. (“/ac” stands for **Add / Curve**.) The “stdgrph” opens a new graph, reads in the template (the template file could be copied directly into **stdgrph.ep**), prompts you for a data file, and reads the file. Thus, by hitting **F1** and entering a data file, you set up an entire graph. You can add a “/fp p” command to print the graph.

To further automate graph production, you can create batch files that generate multiple graphs, as in Figure 6.5. To run a batch file, read it as though it were a data file; pull down **File**, choose **Open** or **New**, and enter the file.

To conserve memory resources, it’s a good idea to delete graphs in your batch files once you’re finished with them.

/ag		run1.dat	;read file named “run1.dat”
/et	g	“run1.dat”	;title graph with file name
/fp	p		;print
/ed	g		;delete the graph
/ag		run2.dat	;read next file
/et	g	“run2.dat”	;title
/fp	p		;print
/ed	g		;delete the graph
...			

Figure 6.5 Batch file that creates multiple graphs

6.6 Setting F-Key Macros

You can assign batch commands to F-Keys for quick access to often-used functions with the command

//fkey *N* “command”

Put the command in your **profile.ep** file (§6.4) and restart EasyPlot or read **profile.ep** like a data file. The *N* is an F-Key number from 1 to 12 and “command” is any complete or partially complete EasyPlot batch command. If the function initiated by the macro requires input when the command ends, EasyPlot prompts you for the input. Here are a few examples:

//fkey	1	“/ac”	;prompt for file/eqn when F1 is hit
//fkey	2	“/fp s”	;preview graph when F2 is hit

```
//fkey 3  "aa"      ;ADD an Annotation by hitting F3
//fkey 4  "sa l"    ;add/edit the legend title of a curve
```

You can assign only one line to an F-Key but that line can be the command for reading a multiple-line batch file. If, for example, you have many binary data files of the same format, you can program an F-Key to set up the file format and read the file in one step. In your **profile.ep** file, add the line:

```
//fkey 1  "/ac c:\ep\bin.ep"      ;read "bin.ep" when F1 is hit
```

The file **bin.ep** could be as follows:

```
//input 1  "binary file:" -file      ;prompt for file
/ac f 2i h 10 c 2 s 0 r 0 a 1        ;2-byte int, hdr=10 bytes, cols=2
/ac      %1                          ;read file
```

6.7 Concatenating Strings in Batch Mode

In batch mode, you can join strings and macros to make new strings and/or macros. If EasyPlot finds a plus sign (“+”) after a quoted string or macro, it joins the string following the “+” with the preceding string. You can join any number of strings but the total length is limited to the length of the string for which EasyPlot is scanning. All literal strings must be quoted (single or double) and all macros must have spaces separating them from the “+” signs. For example, the line s

```
//macro 1  "by"
/et g      "Graph" + %1 + "EasyPlot"
```

title the graph “Graph by EasyPlot”. Concatenating strings can be useful with the **//macro** function’s ability to extract text from other files (§6.3.2).

6.8 Programming Interface

If you develop Windows applications and need to plot and analyze data, you can use EasyPlot as though it were a subroutine of your program. You control EasyPlot by having it read files containing data and EasyPlot batch commands. There are also special command-line parameters and programming hooks that give you additional control over the appearance and operation of EasyPlot.

You use standard Windows API function calls to launch EasyPlot and communicate with it. The process of launching EasyPlot and linking your program to EasyPlot is different under 32- vs. 16-bit Windows. We’ll be describing one

method for each but there are surely other ways. In either case, the goal is to get EasyPlot running and to get the window handle to EasyPlot's main window.

The following lines illustrate (in C) how to invoke a 16-bit version of EasyPlot under Windows 3.1. The syntax using other programming languages will be different but the function names and parameters will be the same.

```
if (inst = WinExec( "c:\\ep\\epw.exe data3", SW_SHOW)) {
    eph = GetNextWindow( hWnd, GW_HWNDPREV);
    MoveWindow( eph, 10, 10, 200, 150, DO_REPAINT);
}
```

The first line runs EasyPlot and returns the instance handle of the newly created task. The second line retrieves the handle of the EasyPlot window. Once you have the window handle, you can move the EasyPlot window and send it messages. The third line positions the EasyPlot window.

Launching a 32-bit version of EasyPlot under Windows 95 or NT is a bit more complicated. You start EasyPlot with the Windows **CreateProcess** function which returns (in a passed structure) the thread ID of the EasyPlot process. With the thread ID, you can use the **EnumThreadWindows** function to list IDs of all windows belonging to the given thread. EasyPlot opens only one window and so the first window enumerated is the one you want. You also need to attach your application's input thread to EasyPlot's; attaching input threads lets you send messages to the EasyPlot window. The following lines illustrate:

```
eph = 0;
if (CreateProcess( exec, cmdln, 0, 0, 0, DETACHED_PROCESS, 0,0, &si, &pi)) {
    for (ct=0; !eph && ct < 100; ct++) {
        Sleep( 20);                               /* give ep time to create its window */
        EnumThreadWindows( pi.dwThreadId, enum_epwh, 0);
    }
    if (eph) {
        AttachThreadInput( pi.dwThreadId, GetCurrentThreadId(), 1);
        EnableWindow( eph, 0);
    }
}
```

The example is taken from the Programming Interface Demo which you should have received along with the software. The **enum_epwh** routine, which **EnumThreadWindows** calls, does nothing but set the **eph** variable to the EasyPlot window handle. For more details, look at the on-line source code and documentation that comes with the demo.

Once you have EasyPlot running and linked up to your program, you can send messages to EasyPlot. There are twelve messages:

<u>Message name</u>	<u>Msg #</u>	<u>Description</u>
API_READY	70	Check if EP is ready to go
API_REMOVE_FILE	71	Have EP remove the given file
API_GET_XTICD	74	Get distance between x -axis major tics
API_GET_XMAJS	75	Get # of major tics on x -axis
API_SHFT_X	76	Shift x -axis range
API_SHFT_X_NOUPDATE	77	Shift x -range w/out updating graph
API_LOCK_YAXIS	78	Lock y -axis for cleaner redraw
API_RUN_FILE	80	Tell EP to read a file
API_CLOSE	81	Tell EP to exit
API_ADD_DATA	82	Add data only to graph
API_SET_CURVE	83	Select curve to add data to
API_NEW_DATA	84	Replace all data for existing curve

For simple applications, you can make due using only a few of the messages. Message #80 tells EasyPlot to read a file. The file can contain data and/or EasyPlot batch commands. Message #82 also tells EasyPlot to read a file but, for speed, EasyPlot assumes the file contains only data and only redraws data when finished reading the file. By default, the **ADD_DATA** command adds data sets to a graph. If you first select a curve with message #83, **ADD_DATA** appends data to the selected curve. Use this feature if you are gathering or computing data and periodically want to show the latest data. When appending data, EasyPlot draws only the new data, providing exceptionally fast updates. (For fastest results, display the curve with solid connecting lines only; with dashed lines, EasyPlot has to redraw the entire curve to maintain a consistent dash pattern.)

Apart from adding data and closing the EasyPlot window, all other aspects of what EasyPlot can be controlled by standard EasyPlot batch commands. Place the commands in the files you have EasyPlot read.

To tell EasyPlot to read a file, use the Windows **SendMessage** or **PostMessage** functions. Set the "message" parameter to WM_COMMAND and wParam to 80. The simplest way to read a file is to use a fixed "run file" which you set on the command line ("-run=filename") that starts EasyPlot; or you can put the **//run_file** command in **profile.ep**. These commands are described below. With a fixed run-file, place batch commands and data into the file and use the following line to have EasyPlot read it:

```
SendMessage( eph, WM_COMMAND, 80, 0L); /* read the run-file */
```

where ‘eph’ is the handle to the EasyPlot window. If you have not set a run-file or want EasyPlot to read a different file, place the name of the file into global, shareable memory and pass the memory handle as the IParam:

```
#define EPW_RUN_FILE 80

char huge *gmem;
HANDLE mh;

if (mh = GlobalAlloc( GMEM_SHARE, 60))
    if (gmem = (char huge *)GlobalLock( mh)) {
        copy_string( "c:\\e\\data4", gmem);
        GlobalUnlock( mh);
        SendMessage( eph, WM_COMMAND, EPW_RUN_FILE, (LONG)mh);
    }
```

Once you pass the memory handle to EasyPlot, you do not have to free it; EasyPlot frees the memory for you.

When you launch EasyPlot, you can pass it command-line parameters. If you want EasyPlot to read one or more files, place the filenames on the command line. You can also place any of the following on the command line, or use the double-slash batch-command equivalents in **profile.ep** or another file EasyPlot reads:

command-line double-slash

-cap=caption	//cap	set caption on window
-nomenu		don't display menu on window
-ret	//ret	return focus to parent (W3.1 only)
-ret=N	//ret N	return focus to window handle N (W3.1)
-run=filename	//run_file fname	set name of run-file

The **-cap** command replaces the word ‘EasyPlot’ in the window caption with custom text. The **-nomenu** command creates an EasyPlot window with no menu.

Under Windows 3.1, you can use the **-ret** command to return the input focus to your application as soon as EasyPlot finishes processing the **profile.ep** file and any files on the command line. (In W95/NT, your application continues running even while EasyPlot is initializing.) The **-ret** command can include an optional window handle (“-ret=3510”, for example) to specify the window to receive input focus. Use integer notation (not hexadecimal) for the window handle. If you do not specify a window, EasyPlot returns focus to the window which created it.

The **-run** or **//run_file** commands must specify the full path of a fixed ‘run file’. EasyPlot does not read the file until told to do so and so the file need not exist when you invoke EasyPlot. Once you have set a run-file, you can place

EasyPlot batch commands and data in the file and then use a **SendMessage** or **PostMessage** command with lParam=0 to have EasyPlot read the file:

```
inst = WinExec( "c:\\ep\\epw.exe -run=c:\\ep\\epbat.run", SW_SHOW);
if (inst) {
    eph = GetNextWindow( hWnd, GW_HWNDPREV);
    :
    : /* create file "epbat.run" */
    SendMessage( eph, WM_COMMAND, 80, 0L); /* read file */
    :
    : /* put new data in epbat.run */
    SendMessage( eph, WM_COMMAND, 80, 0L); /* read file */
    :
    :
    SendMessage( eph, WM_COMMAND, 81, 0L); /* close EP */
}
```

Using a fixed run-file saves you from having to allocate a block of global memory, lock the memory and copy the filename into the block, unlock the memory and pass the memory handle to EasyPlot.

If you are computing a family of curves and want to display each as it is computed, set the wParam to 82 (for "Add Data") instead of 80. EasyPlot will then redraw only the data on the graph and not rescale the axes, resulting in faster redraws. You can also use "Add Data" to append data to existing curves :

```
SendMessage( eph, WM_COMMAND, 83, 1L); /* select curve #1 */
SendMessage( eph, WM_COMMAND, 82, 0L); /* add data in read-file to curve */
```

The "Add Data" command adds data to one curve at a time. The added data has to have the same number of columns as the original curve. When selecting a curve, the curves are numbered from 1 to the number of curves on the graph. The order is the same as if EasyPlot were prompting you to select a curve on the screen.

Appending data to a curve is the fastest way to update a graph. If the curve does not have dashed connecting lines, EasyPlot draws only the new data. If the curve extends outside the graph range, you may not see all the data. To update or scroll the graph, put an autorange or range command in a file EasyPlot reads:

```
/oa x          ;autorange the X axis
/oa g          ;autorange the X & Y axes
/or x 100 200   ;set X range from 100 to 200
```

You can also use messages 76 ("**API_SHFT_X**") and 77 ("**API_SHFT_X_NOUPDATE**") to shift the *x*-axis range. To close the EasyPlot window, send it message #81:

```
SendMessage( eph, WM_COMMAND, 81, 0L); /* close EP window */
```

Windows will usually continue executing commands in your program before EasyPlot has finished initializing itself. If you want EasyPlot to read files when you invoke it, put the files on the command line and EasyPlot will read them when it is ready. You should not open an EasyPlot window and immediately send it messages to read files. If you use a fixed run-file, EasyPlot stores one “read” message and will read the run-file when ready. If you send more read messages, EasyPlot ignores them. The return value from the **SendMessage** command is 0 if EasyPlot was not ready to read the file; it is 1 if EasyPlot was ready. The **Postmessage** command does not return a value from EasyPlot.

When starting EasyPlot, you can use send message #70 (“**API_READY**”) to EasyPlot to see if it’s ready. Use the **SendMessage** command and if the return value is nonzero, EasyPlot is ready to process commands.

If you send EasyPlot messages with the **PostMessage** command, your application continues executing while EasyPlot does its job in the background. When sending data to EasyPlot at fairly short intervals, you might want to be sure that EasyPlot has finished its previous task before you ask it to graph other data. If you use a single “run file”, you must be sure EasyPlot has finished reading the file before you put new data into it. Message #71 tells EasyPlot to delete the specified file. If no file is given, EasyPlot deletes the run file if it exists. By posting **Read** (#80) and **Remove-File** (#71) messages in tandem (set to operate on the same file), EasyPlot will read the file and when done, it will delete the file. To keep your application synchronized with EasyPlot, you can check to see if the file you’ve told EasyPlot to delete still exists. If it does, EasyPlot has not finished processing the last messages you sent.

If you are using EasyPlot to graph a steady flow of data, you may need to scroll the x -axis periodically to move old data off the graph and make room for new data. Your display will look much nicer if you scroll by major-tic increments. That way, the x -axis labels don’t dance around the window. You can use the **//oaas** batch command to set the tic increments and you’ll then know the distance between major tics. If you let EasyPlot choose the tic positions, you can send message #74 to have EasyPlot return the distance between major tics. EasyPlot returns a 4-byte floating point value. Message #75 returns the number of major tic marks along the x -axis. Use it to help decide how much to scroll the graph.

You can use the **/or** batch command to shift the x -axis range; EasyPlot replots the entire graph after reading a file and you have to have EasyPlot read a file to process a batch command. If you are trying to create a smooth display of scrolling data, use message #76 to shift the x -axis range. The **IParam** value for message 76 is a 4-byte floating point number that specifies how much to shift the range.

Specify the shift in x -axis units. Message #77 also shifts the x -axis range but EasyPlot does not update the graph. Use message #77 if you plan to replace all the data on the graph after shifting the range.

If the y -axis range on the graph remains fixed while scrolling data, you can use message #78 to lock the y -axis and reduce redraw flicker. If the **IParam** value is non-zero, EasyPlot locks the y -axis.

EasyPlot V4 ships with a small Programming Interface demo. The demo uses EasyPlot to graph “real-time” data and comes with source code (in C) and documentation. If you plan to use EasyPlot’s programming interface, run the demo and look through the documentation and code to see to get a feel for how the programming interface works.

6.9 Command-Line Parameters

You can specify up to ten data files as command-line parameters. When EasyPlot finishes reading **profile.ep**, it looks at the command-line parameters and, if they specify files, EasyPlot reads the files.

Chapter 7: EasyPlot in Detail

7.1 Date and Time Stamps on Printouts

To put a date and/or time stamp on a printout, place the text **%date** or **%time** anywhere on a graph. When you print or preview the graph, EasyPlot replaces the **%date** or **%time** with the system date or time:

%date	→	mm/dd/yy
%time	→	hh:mm:ss


The order of the day, month, and year in the date is controlled by the date format as specified in the **System** topic of the **Preferences** dialog.

7.2 Using Expressions to Enter Numbers

When entering graph coordinates (for placing tic marks, setting axis ranges,...), you can use expressions such as $1024*4$, or $PI/2$. The equation cannot contain x s, y s, or unknowns but it can reference any of the equation constants, such as XMAX, YMIN, SDEV, listed in Table 6.1.

If you use expressions with the **Options / Range** (in the EasyPlot-standard menu), do not include spaces in the expression; EasyPlot looks for a space or comma to separate the minimum range value from the maximum. You cannot use expressions in the Data Table.

7.3 The Undo Feature

EasyPlot's **Undo** feature lets you undo most operations, such as range changes, deletions and moves, and changes to a data set's column-definition string. To undo an operation, select **Edit / Undo**; or click the undo button () on the help line.

The **Undo** feature stores five operations. To save memory, **Undo** does not save deleted graphs or curves that have more than 1000 data points. EasyPlot prompts you to confirm deletes that cannot be undone but you can turn this check off in the **Safeties** topic of the **Preferences** dialog. The **Undo** feature does not record additions to graphs (such as a line you sketch or annotation you type) because you can just as easily delete the object.

7.4 Quitting EasyPlot

When you quit without having saved the current graph, EasyPlot asks whether you want to save before quitting. If you jump in and out of EasyPlot often and don't want to be prompted for saving graphs, go to the **Safeties** topic of the **Preferences** dialog and, under **Prompt to save**, turn off **when closing graphs**; EasyPlot then quits as soon as you select **Exit** or hit **<alt>F4**.

You can create a "hot-key" for quitting. Place the **//quit_key** command in your **profile.ep** file followed by the ASCII code for the key. The line

```
//quit_key 27
```

makes the **<esc>** key quit EasyPlot.

7.5 Remembering the Current Directory

When you start EasyPlot, the current directory is set to the working directory specified by the icon properties in the Program Manager. If you change directories within EasyPlot, you can have EasyPlot save the directory upon quitting so that when you next start EasyPlot, you will be in the same directory. Open the **Preferences** dialog and go to the **System** topic. The bottom checkbox labeled **Remember current directory for next session** turns this feature on or off.

7.6 The EasyPlot Initialization File

Every time you quit, EasyPlot stores its current state in a file called **epw.inn** (for the 16-bit executable) or **epw32.inn** (for 32-bit versions of EasyPlot). When you next start the program, EasyPlot reads the file and comes up at the same screen position and with the same options selected. The file is in binary format and should not be modified. If you delete it, EasyPlot starts with its default settings as though you just installed the software.

EasyPlot puts the initialization file in the directory where the program resides or in the alternate setup directory you specify (§7.12).

EasyPlot does not save the state of the **Style** buttons. To customize default settings for styles, use batch commands in **profile.ep** (§6.4).

7.7 Refreshing the Screen

If the screen needs to be refreshed, you can do so with the **Refresh** button under **File** in the EasyPlot-standard menu. With either menu, you can force a redraw by minimizing and restoring the window.

7.8 The Menu System

EasyPlots has two menu systems, one more similar to earlier versions of EasyPlot, and the other more “Windows standard”. To switch menus, go the **User Interface** topic of the **Preferences** dialog. The main menu bar of the two systems is identical; when you switch from one to the other, you do not see any change. To see the differences, pull down each of the menus (**File**, **Edit**, **Add**,...). The Windows menu removes several buttons which duplicate functions found elsewhere, such as **Add / Curve** and **Add / Graph** which are equivalent to **File / Open** and **File / New**. It also moves a few buttons to more logical places.

As with any Windows application, you can pull down a menu by hitting **<alt>** and a menu letter. With EasyPlot, you do not need to hit the **<alt>** key; hitting ‘**f**’ pulls down the **File** menu, for example.


For compatibility with earlier versions of EasyPlot, the batch language (§6.1) is based on the EasyPlot-standard menu. Batch commands are assembled from the letters you would type at the keyboard to run menu buttons (“**sm**” for **Style / Mark Pts**, for example). If you plan to use the batch language, it’s a good idea to use the EasyPlot-standard menu to familiarize yourself with the buttons on which the commands are based.

7.9 Editing Data

You can modify an entire graph or curve by applying transformation equations to the data (§4.2). Or you can modify data on a point-by-point basis with EasyPlot's Graphical Data Editor or Data Table.

7.9.1 The Graphical Data Editor

EasyPlot's Graphical Data Editor lets you delete or move data points directly on the graph. It is ideal for removing outliers from experimental data or for sculpting a curve into a particular shape.

Select **Edit / Data / on graph**. With the EasyPlot-standard menu (§7.8), click **Edit / Move / Curve**. You can also click the data editor icon () on the help line. Move the cursor onto a data point, hold the left mouse button, and move the cursor. The data point moves with the cursor until you release the button.

To delete a point, hit **** before releasing the mouse button. If you start moving a point by mistake, hit any key to cancel the move. You can undo the five most recent moves or deletions with **Edit / Undo**.

When you click on a data point, EasyPlot searches all the curves on the graph for a point inside the cursor box. If a graph has many curves each with many points, you can speed the search by preselecting a curve (with the space bar or right mouse button); EasyPlot then searches only the selected curve.

As you make changes, EasyPlot updates data internally but does not update any files. To save the modified data to a file, load it into the Data Table (§7.9.2) and save the table; or save the graph and EasyPlot stores the data in the save file.

You can zoom in and out without leaving the graphical editor. If you click when the cursor is not on any data, EasyPlot goes into zoom mode. You can force EasyPlot to zoom or unzoom (and thus avoid any delay caused by searching data) by holding **<alt>** (to zoom) or **<ctrl>** (to unzoom) when clicking.

The Graphical Data Editor works with *xy* plots only. You cannot add data points to a curve graphically. Use the Data Table (§7.9.2) if you need to add points. Hit **<esc>** to leave the Graphical Data Editor.

7.9.2 Editing Data in the Data Table

Once you have plotted a data set, you can load its data into EasyPlot's Data Table. Pull down **Edit**, select **Data**, and choose **in text editor** if asked. If there are multiple curves on the graph, select the curve to edit.

Make changes and, when ready, click on **Plot**. EasyPlot updates the curve with the new data. If **Close After Plot** (in the Data Table **File** menu) is not checked, the table remains open and you can continue editing.



With multiple column data, you can load all columns or data for one curve only. The **Whole Data Set** button (in the table's **File** menu) toggles between displaying one curve or all data columns.

See the next section for details on using the Data Table.

7.10 The Data Table

EasyPlot's spreadsheet data editor lets you enter data directly into EasyPlot and, once plotted, lets you look at and modify data. You can also use it to browse or edit files before plotting.

To enter data, pull down **File** and select **Enter data**. Or, click on **Enter data** in EasyPlot's **Open** dialog. To browse or edit a file before plotting, open it with **Browse file** checked.

Once data is plotted, select **Edit / Data** to load the data into the Data Table. You can also click on the edit-data icon () in the Curve Toolbar (). With multiple-column data, you can load all columns or just the data for one curve. The **Whole Data Set** button (in the Data Table **File** menu) toggles between displaying all columns or just the data of one curve.

EasyPlot's Data Table is a cross between a spreadsheet and a text editor. If you enter numbers, it behaves like a spreadsheet. If you enter text (the first character is non-numeric), EasyPlot treats the row as a line of text; it extends the cell across the window and does not let you enter data to the right of the text.

The Data Table does not have the mathematical capabilities of a spreadsheet. To perform math on data, plot it and use EasyPlot's Transform feature (§4.2).

To enter a number or text in a cell, simply begin typing. If a cell already has information in it, what you type replaces the old text. If you overwrite a cell by mistake, hit **<esc>** or, if you already hit **<enter>** or moved to another cell, use the **Undo** button. The **Undo** feature remembers the last five operations that removed or modified text in the data table.

To edit the contents of a cell, make it the current cell and click on it. When entering data, hit **<space>** or **<tab>** to move to the next column. Hit **<return>** to move down a row and back to the left-most column.

EasyPlot displays the definition of data columns at the top of the table. If you enter data in the table or load a file that does not have its columns defined (with a

/td command), EasyPlot defines the columns automatically by assuming the first column is x and subsequent columns y . Click on the definition box at the top of a column to change a definition. Click on **Plot** to apply changes to the graph.

If you have many columns and want to plot a few at a time, you can turn columns on or off by double clicking on the definition boxes. The definition appears gray if the column is ignored. Click on **Plot** to apply the changes.

If a column corresponds to a curve on the current graph, EasyPlot displays a piece of the curve (a data marker or its line style) at the top of the column. When you click on a curve on the graph, EasyPlot moves the data table cursor to the column of the selected curve, assuming its data is in the table.

The **Edit** menu lets you insert, delete, and duplicate rows and columns. You can move rows and columns by deleting, moving the cursor, and recovering. The **Recover** button restores the most recently deleted column, row, or block of rows to the left of or above the current cell. To move column 3 to column 1, for example, put the cursor on column 3 and delete (**Edit, Delete, Col;** or hit **<alt>O**), move to column 1, and hit **<alt>r** or select **Edit / Recover**.

The row and column operations (insert, delete,...) have keyboard shortcuts that are listed in Table 7.1.

	row	column
insert	<alt>a	<alt>i
delete	<alt>d	<alt>o
duplicate	<alt>=	<alt>l
recover	<alt>r	<alt>r

Table 7.1 Spreadsheet Keyboard Shortcuts

You can select a block of rows by clicking and dragging the mouse or by holding **<shift>** and moving the cursor up or down. Once selected, you can delete the block by hitting ****, **<Backspace>**, or **<alt>D**. If you delete a block by mistake, **Undo** recovers the rows.

EasyPlot expands and contracts the width of columns automatically to accommodate data. If you enter a number that is wider than the column, the text scrolls while you type. When you hit **<return>** or move the cursor to another cell, EasyPlot widens the column to fit the data.

Once in the spreadsheet, you can load ASCII or spreadsheet files with the **Open** button under **File**. If you open a file and the table is not empty, EasyPlot asks whether to replace the contents. Answer “yes” and EasyPlot clears the table and loads the file. “No” tells EasyPlot to load the file starting at the current cell and overwrite cells to the right and below the current cell. To append a file to the table, for example, move the cursor cell below the last data row and into the left-most column. Open a file and tell EasyPlot not to replace the table contents.

You can move data in and out of the spreadsheet via the Clipboard. The **Copy** button (under **Edit**) copies the entire table to the Clipboard. The **Paste** button pastes whatever text is in the Clipboard onto the Data Table. Pasting is identical to opening a file; if the table is not empty, EasyPlot asks whether to clear the table before pasting. As with files, answer “no” to start filling cells at the current cell and overwrite cells to the right and below the current cell.

The **Plot** button graphs the data in the table. EasyPlot closes the table if **Close After Plot** (under **File**) is checked. If the data in the table is linked to a file (look on the caption bar for a filename), EasyPlot may ask you to confirm overwriting the file. If you answer ‘no’, EasyPlot plots the data but maintains no link between the data and the original file. If you save the graph, EasyPlot puts the data directly into the save file rather than referring to the file. You can turn this overwrite check off in **Safeties Preferences**.

The **Save** button (under **File**) lets you save the contents of the Data Table to a comma-delimited ASCII file. Files created with the spreadsheet can be loaded back in or read directly onto an EasyPlot graph. You can also view and/or modify them with any text editor.

EasyPlot’s Data Table can handle missing data values automatically. If **Autofill Missing Values** is checked (under **File**), EasyPlot marks empty cells inside the data area as missing values when you plot or save to file. The data area is the smallest rectangle that includes all the data. If you look at what EasyPlot exports (via Save or Copy, or by choosing to **Edit** the **Data** again), you will see missing value markers (`‘//m’` – §6.3).

The Data Table can handle up to 30,000 rows or columns; it is not designed to handle an unlimited number of points the way EasyPlot can on a graph. A practical upper limit is 15 or 20 thousand *xy* pairs.

You can open only one table at a time. See Section 3.1 for details on how to arrange data for EasyPlot.

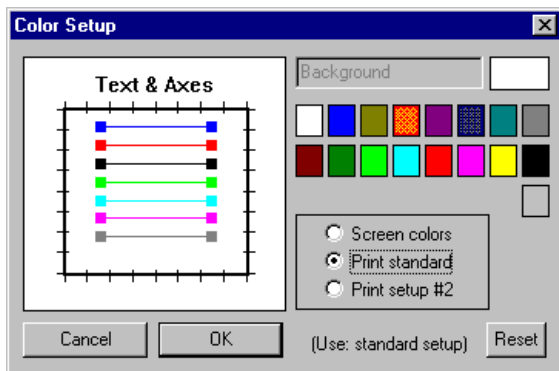


Figure 7.1 The Color Setup dialog

7.11 Setting Screen Colors

You can customize the color palettes EasyPlot uses onscreen and on printouts. Pull down **Style** and select **Colors**. If an annotation (§2.7) or a curve (§2.2.1) is selected, EasyPlot prompts you to choose its color. Otherwise, EasyPlot displays the **Color Setup** dialog. (Hit **<esc>** or click away from any object to make sure no object is selected.)

There is one screen palette and two print palettes. The print setups let you switch easily between palettes for paper (dark colors on a white background) and for slides or transparencies (light colors on a dark background).

The radio buttons in the **Color Setup** dialog determine which palette is used for printing. If **Screen colors** or **Print standard** is selected, EasyPlot uses the standard palette. If **Print setup #2** is selected, it uses setup #2. To print color, make sure that **color** is on in the **Print** dialog. Use the print preview to check the color setup before printing.

The left side of the Color Setup dialog shows you what graphs look like with the selected colors. To change a color, click on an object in the sample graph. The top-right of the dialog tells you what object you selected and shows its current color. Click on any of the color boxes to change the color. The sample graph updates automatically. You can set colors for the page background, the graph-box background, text and axes, and the seven curve colors.

The background and text colors you choose for the screen do not affect printing; they are for screen aesthetics only. Curve colors print as they appear on the screen unless you map them to different colors in the print palettes.

EasyPlot remembers color selections for future sessions.

7.12 Running EasyPlot on a Network

By default, EasyPlot manages its setup files (**profile.ep** and **epw.inn/epw32.inn**) in the directory from which the program is run. If several people are running EasyPlot from a single network installation, each user should have his/her own setup files in a private directory. There are two ways to tell EasyPlot which directory to use for setup files. You can put the command:

```
//setup_dir      path
```

in the **profile.ep** file in the directory where EasyPlot resides. The “path” is the full path to a directory, for example “c:\eplot”, and must be a valid path for everyone running EasyPlot. Alternatively, you can use an environment variable to tell EasyPlot where to manage setup files. EasyPlot looks for one of the following environment variables:

```
EASYPLOT_SETUP  
EASYPLOT  
EP
```

It looks for **EASYPLOT_SETUP** first, followed by **EASYPLOT** and then **EP**. To set one of the variables, use the DOS **SET** command, for example:

```
set EASYPLOT=C:\EPlot
```

Put the command in **autoexec.bat** or in a DOS batch file that launches EasyPlot.

As with setup files, EasyPlot also creates scratch files. For network installations, you should be sure EasyPlot is using a local or at least a private directory for each user's EasyPlot scratch files. See the next section.

7.13 Scratch Files

When you transform data or fit curves, for example, EasyPlot creates scratch files for computed data. Upon start-up, EasyPlot looks for the environment variables:

```
EASYPLOT_TEMP  
TEMP  
TMP
```

It looks for **EASYPLOT_TEMP** first, followed by **TEMP** and **TMP**. If found, EasyPlot uses the directory specified by the variable for all its scratch files. Alternatively, you can use the **//tmp_dir** batch command:

```
//tmp_dir      path
```

Put the **//tmp_dir** command in the **profile.ep** file in the same directory as EasyPlot.

When running EasyPlot on a network (§7.12), you should make sure EasyPlot puts scratch files in different directory for each user. With standard Windows installations, the **TEMP** or **TMP** environment variable should already be set to a local or private directory; you should not have to do anything specifically for EasyPlot scratch files, unless you want them in a directory other than what Windows uses by default.

7.14 The Preferences Dialog

The **Preferences** dialog gives you quick and easy access to a wide variety of features. The Preferences dialog has a list of topics on the left. As you move from topic to topic, the controls on the right change to display options for the selected topic. When you close EasyPlot, preference settings are saved automatically in the **epw.inn** or **epw32.inn** file.

There are many ways to access EasyPlot **Preferences**. You can pull down **File**, choose **Preferences**, and scroll to the desired topic. Or you can right-click on an object and if there are preference topics related to the object, you can select a topic from the context-sensitive right-mouse menu. Many dialog boxes (such as the Data Style dialog – Figure 2.2) have **Preferences** buttons that send you directly to a specific **Preferences** topic.


When you see “**Safeties Preferences**” or “**Arrow head Preferences**” in the manual, we are referring the a particular topic of the **Preferences** dialog.

7.15 Cyclical X-Values

If the x -values of a data set are cyclical (0,... 360, 0,... 360,...), and you want the points to be plotted without wrapping around, go to the **Line charts** topic of the **Preferences** dialog and enter the maximum x -value as the **Period on X**. If x represents an angle measurement in degrees, for example, use 360 as the period value. With the **Period on X** set, EasyPlot treats x -values as offsets from a base. The base is initially 0 and increases by the period value every time an x -value is smaller than the previous x -value.

The first time EasyPlot graphs the data, the x -axis may not be autoranged properly. If the data extends off the graph, pull down **Options** and choose **Autorange**. Reset the period to 0 to go back to plotting x -values at their actual coordinates.

7.16 On-line Help

EasyPlot's Help system provides context-sensitive help. To use Help, pull down **File** and choose **Help** or click the help icon () on the help line. If you don't know how to do something, click on **Index** and find the desired topic.

If **Auto Update** (under **Help**) is checked, the help window updates as you work with EasyPlot. Turn **Auto Update** off to hold the current help topic; you can still change the topic with the index or a cross reference. Cross-reference topics are displayed in blue. Click on the blue text to go to that topic.

You can scroll back through 10 previously viewed topics with the << button. The >> button moves forward through topics.

Chapter 8: Printing

EasyPlot prints to any Windows device except for pen plotters. If you must print to a pen plotter, you'll need to use EasyPlot for DOS. To print a graph, you must have installed at least one printer in Windows.

8.1 Printing a Graph

Pull down **File** and select **Print**. EasyPlot displays the **Print** dialog (Figure 8.1). The **Print** dialog lets you select a printer, configure the page size, etc. You can print directly to a printer, to a disk file, or to the screen to preview the graph.

You can select landscape or portrait orientation in the **Print** or **Page Setup** dialogs. Switching orientation does not affect the size or aspect ratio of graphs; it merely turns the graphs on the page.

The **Color** and **B&W** radio buttons choose whether EasyPlot generates color or black-and-white output. The **Color / B&W** option also affects graphs copied to the Clipboard.

The **Line weight** dropdown lets you set line thickness. The dropdown lists values from 1 to 9 but you can enter a number up to 20. Axes, data, tic marks, and

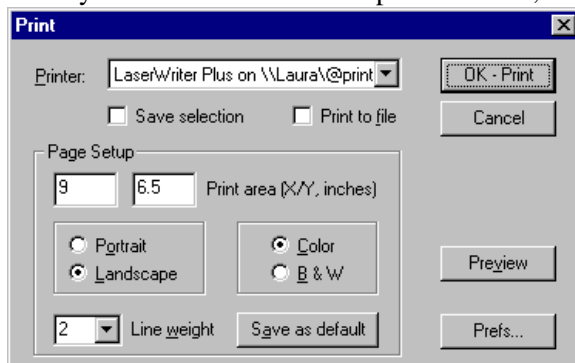


Figure 8.1 The Print dialog

lines and arrows are drawn at the specified line weight. Grid lines are always drawn with single-pixel-wide lines.

You adjust the size of printed graphs by setting the **Print area**. You can also set the print area graphically in the **Preview** (§8.2). The aspect ratio of printed graphs is directly related to the aspect ratio of the print area. If you make the area square, you get square graphs (assuming the graphs are maximized). The entire EasyPlot window maps into the print area. If you print one maximized graph, the graph box will be 75% of the print area; a 4x4 print area, for example, will produce a 3" square graph. The other 25% leaves space for axis labels and titles.

You cannot change the aspect ratio of polar plots; EasyPlot forces the printed page to be square. If you set the page dimensions to 5"x4", EasyPlot places graphs in a 4"x4" square, resulting in circular plots with 3" diameters. For two quadrant polar plots, EasyPlot forces the page to be a $2n \times n$, or $n \times 2n$ rectangle.

When copying graphs to the Clipboard for pasting into other applications, it's a good idea to set the print area as close as possible to the size it will be when pasted. That way, graph text and lines appear fairly close to the selected sizes and weights. Otherwise, increase the line weight and point sizes so that when the graph is reduced, text and graphics end up at a reasonable size and weight.

The **Page Setup** and the **Font** dialogs let you set point sizes. See Section 2.3 for details.

Click on **Save as default** or **Save Setup** in the **Print** or **Page Setup** dialogs to save the page setup for future sessions. The page setup is a global; there is only one page setup per EasyPlot session. If you save graphs with the page setup (selected in **Saving Preferences**), loading a saved graph will reset the page setup to what it was when you saved the graph. EasyPlot saves the setup in the file **attrib.ep** in the EasyPlot setup directory (§7.12).

8.2 Print Preferences

Distance from graph to graph & axis titles: EasyPlot positions the graph and axis titles automatically. If you want the titles closer or farther from the graph, enter a distance-scaling value in the "Distance from graph" edit box. If you enter a number greater than 1, the titles will be farther from the graph. Enter 1.5, for example, to move titles 50% farther from the graph.

Draw box around graph: With **Draw box** on, EasyPlot draws a box around the data area. With the axes in their standard positions, the box overlaps the axes on the bottom and left sides of the graph. If you want just data and axes, turn **Draw box** off.


Frame around graph & titles: You can have EasyPlot draw a frame around the entire graph and titles. By default, EasyPlot leaves a 1/10 inch margin between the frame and the graph. You can customize the margin by changing the value in the **Inches between graph & frame** edit box.

Scale text size down on inset graphs: When you print multiple graphs on a page, EasyPlot can scale text so that small inset graphs use smaller text than larger graphs. On a graph that takes up half the window, for example, 20-point text would print as 10-point text. You can also choose whether text is scaled relative to the largest graph on the page or relative to the full EasyPlot window. With two equal-sized graphs side-by-side, 12-point text will print at 12-points if text scaling is off or scaled relative to the largest graph; otherwise text will appear smaller than the set size when printed.

Dash & dot length: EasyPlot scales the length of dots and dashes based on line weight and printer resolution. If you want the patterns to be longer than what EasyPlot produces, increase the **Dash & dot length**.

8.3 Previewing

In “working” mode, EasyPlot configures the screen so that graphs are as large as possible for viewing and analyzing data. The layout changes in subtle ways when you print. To see what graphs will look like when printed, pull down **File** and choose **Print Preview**. You can also click the **Preview** button in the **Print** or **Page Setup** dialogs.

You can zoom on the preview the same way you zoom in working mode, with a simple click-and-drag. Zooming on the preview enlarges the entire page; it does not change axis ranges. You can zoom in until you reach the maximum magnification EasyPlot allows. As in working mode, **<ctrl>**click brings you to the previous zoom level. To zoom all the way out, click on **Fit to window** in the main menu; or right click and select **Fit to window** in the context-sensitive popup menu. You can also click the **Autorange** icon () on the help line.

You can edit graphs in the previewer. Screen updates in the previewer are slower than in working mode; use the previewer for final touch-up work only, especially with very complicated graphs. Click on graph and axis titles to edit them. Double click on annotations and legend entries to edit them. Click and drag annotations, legends, and lines/arrows to fine-tune placement. You can add annotations and lines and arrows just as you would in working mode.

You can move and resize graphs in the previewer. If you click on a graph, but not on a note, line, legend, or title, you select the graph and a gray selection box

surrounds the graph. You can click and drag a corner or edge selection handle to resize the graph within the print area. If the graph does not occupy the entire print area, you can click inside the graph (after selecting it) and drag it around the print area without resizing. The print area is not the page; it is the area defined in the **Print** or **Page Setup** dialog and it shows up as a blue border in which you can move the graph.

If you click on the page but not on a graph, a blue border frames the print area. The blue border has selection handles that you can use to adjust the size of the print area. EasyPlot keeps the print area centered on the page. You can click inside the print-area selection box, however, and drag the entire image to a different spot on the page, thereby shifting it from center. If you shift the print area, you can clear the offset by right clicking on the preview and selecting **Clear page offset**.

When you're ready to print, click on **Print...** in the main menu or the print icon on the help line. Or right click on the preview screen and select **Print...** or **Print Now**.

8.4 Exporting Graphs to Other Applications

The easiest way to put an EasyPlot graph into a wordprocessor or other application is to copy the graph to the Clipboard (**Edit / Copy**) and paste it into the other program. You can also print to a Windows Metafile and import the file. To create a metafile, choose **Windows Metafile** as your printer in the **Print** dialog. EasyPlot asks you to enter or choose the name of the file.

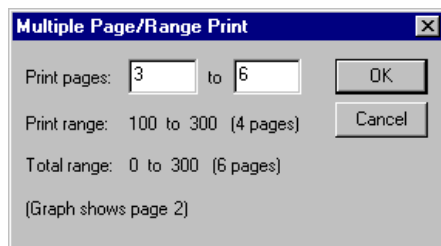
To copy graphs to the Clipboard, make sure that no graph object (a curve, annotation, or line) is selected when you click **Edit / Copy**. If you choose **Copy** when a curve is selected, for example, the data for the curve is copied to the Clipboard instead of the entire graph image.

8.5 Printing Multiple Data Pages

If you zoom in on a graph, you can print several pages of data at the given magnification in a single operation. Print as many pages as it takes to span the entire x -range of data; or print a range of pages to span a subset of the x -range. When printing multiple pages, EasyPlot scrolls the x -axis automatically from page-to-page, shifting the range by the entire width of the graph.

You must have only one graph visible on the screen. Zoom in or set the x -axis range to show part of the data. Go to the **Print** dialog (**File / Print**) and select the **Multi-page...** button. The **Multi-page...** button appears only if one graph is being

printed. EasyPlot displays the **Multiple Page/Range Print** dialog:



Select the pages you want to print and click **OK**. Under **Print range**, the dialog shows you the data range covered by the pages you select.

8.6 Differences Between What-You-See and What-You-Get

When you print, you get almost exactly what you see on the screen. There are a few differences, however. Use the previewer (§8.2) to see what graphs will look like when printed.

If **Flip on screen** is on (**Tic mark Preferences**), EasyPlot prints tic marks in the opposite direction as it displays them onscreen. The **Flip** option keeps the graph clear for viewing data when tics are configured to print inside the graph. If you want graphs to be a bit more WYSIWYG, turn off **Flip on screen**.

You can adjust the *x*- and *y*-dimensions for the print area but these options are not reflected on the screen; they affect only the printed or previewed graph.

Other differences between the screen and the printout arise from text sizes relative to the size of the printed graph. Onscreen, EasyPlot displays point sizes in a relative fashion; 20-point text appears larger than 14-point text. The size of text relative to the size of the graph onscreen may not accurately reflect the relative sizes on the printout, however. Onscreen, EasyPlot displays the graph as large as possible to make viewing and analyzing data easier. If you set the page area to create a very small printed graph, for example, the text will appear large relative to the printed graph. Annotations shrink and grow with point size but the first character appears exactly where it appears on the screen.

The distance between the graph and axis titles and the graph box may be different on printed graphs. Onscreen, the position of the graph and axis titles is fixed so that the text is visible on the screen and doesn't interfere with the graph; EasyPlot ignores point size and the number of lines in the titles. When you print, EasyPlot places titles at an aesthetically pleasing distance from the graph, taking point sizes and extra title lines into account.

The legend often appears slightly different when printed. The size of the legend depends on the size of the characters inside it. If printed text occupies twice the area of screen text (relative to the area of the graph), then the legend appears much larger on the printout than it does on the screen. The top and left sides of the legend remain fixed and appear in the same place they appear on the screen. The bottom and right sides move to accommodate the text and graphics inside the legend.

The Print Preview shows you exactly how graphs will print. If your graphs require very accurate placement of annotations and other graph objects, use the previewer to fine-tune the graph before printing.

Appendix A

Menu Batch Commands

This appendix lists the more commonly used menu batch commands (§6.1). Remember that every menu function has a batch command equivalent. If a command is not listed here, start up EasyPlot and jot down the keystrokes needed to do what you want and create the batch command yourself.

A great way to familiarize yourself with EasyPlot's batch language is to create a graph onscreen, save it, and read the save file. EasyPlot save files are short, fully commented batch files.

Commands which have an ON/OFF parameter will also function without this parameter. Buttons which toggle features on and off look for this parameter when run in batch mode. If found, they set their features to the given state; otherwise, they toggle the features from on to off or vice-versa.

Each entry in this appendix lists, as part of its description, the menu button which corresponds to the batch command. The **/ac** command, for example, has its menu button listed as “**(Add, Curve)**”. The first item is the menu under which the button is found.

/aa Add an annotation to the graph. **(Add, Annotation)**

usage: **/aa** *x* *y* ‘text’ [A]

The *x* and *y* specify the location of the note in graph coordinates and ‘text’ is the text of the note. The text can be enclosed in single or double quotes and can extend over multiple lines. The “A” following the annotation is an optional parameter that lets you place annotations in screen rather than graph coordinates. With the A parameter, *x* and *y* should be between 0 and 1, where (0,0) is the lower-left corner of the screen.

/ac	<p>Add a curve to the current graph. (Add, Curve)</p> <p>usage: /ac file, eqn, or directory</p> <p>If you provide an equation, be sure to enclose it in quotes if it contains embedded spaces. If you provide a directory, you set the default directory.</p>
/ag	<p>Opens a new, empty graph. (Add, Graph)</p> <p>usage: /ag [file or eqn]</p> <p>The data file or equation is optional.</p>
/al	<p>Draw a line or arrow on the graph. (Add, Line)</p> <p>usage: /al x_1 y_1 x_2 y_2 [a]</p> <p>The x_1, y_1, x_2, and y_2 specify endpoints in graph coordinates. The a is an optional parameter which places an arrowhead at x_1, y_1 if $a = 1$ or at x_2, y_2 if $a = 2$.</p>
/at	<p>Add a tic mark to the x-, y-, or z-axis. (Add, Tic)</p> <p>usage: /at a l "label"</p> <p>The a is the axis (x, y, or z) and l is the location of the tic mark in graph coordinates. The "label" is the label for the tic. The label can be any text and can include spaces. Enclose the label in quotes if it has embedded spaces. If the label is "=", EasyPlot uses the coordinate as the label.</p>
/ax	<p>Add a second x-axis to the graph. (Add, X-axis)</p> <p>usage: /ax</p>
/ay	<p>Add a second y-axis to the graph. (Add, Y-axis)</p> <p>usage: /ay</p>
/ed c	<p>Delete a curve. (Edit, Delete, Curve)</p> <p>usage: /ed c n</p> <p>The n is a curve number or a to delete all curves.</p>
/ed g	<p>Delete the top, or current graph. (Edit, Delete, Graph)</p> <p>usage: /ed g</p>

/em x	<p>Move the x-axis. (Edit, Move, X-axis)</p> <p>usage: /em x l</p> <p>With xy plots, l specifies the y-coordinate for the x-axis. With polar plots, l is a letter: t for placing the axis at the top of the graph, c for placing it in the center of the graph, or b for placing it at the bottom of the graph.</p>
/em y	<p>Move the y-axis. (Edit, Move, Y-axis)</p> <p>usage: /em y l</p> <p>With xy plots, l specifies the x-coordinate for the y-axis. With polar plots, l is a letter: l for placing the axis at the left side of the graph, c for placing it in the center of the graph, or r for placing it at right side of the graph.</p>
/et g	<p>Title the graph. (Edit, Title, Graph)</p> <p>usage: /et g "title"</p> <p>Enclose the title in single or double quotes.</p>
/et x	<p>Title the x-axis. (Edit, Title, X-axis)</p> <p>usage: /et x "title"</p> <p>Enclose the title in single or double quotes.</p>
/et y	<p>Title the y-axis. (Edit, title, Y-axis)</p> <p>usage: /et y "title"</p> <p>Enclose the title in single or double quotes.</p>
/ex g	<p>Transform the graph. (Edit, Xform, Graph) Apply a mathematical transformation to all the curves on a graph and place the result into a new graph.</p> <p>usage: /ex g "fcn"</p> <p>The "fcn" is a mathematical function. Enclose the function in quotes if it contains embedded spaces.</p>
/fp	<p>Print the graph or graphs. (File, Print)</p> <p>usage: /fp p or file</p> <p>The "p" tells EasyPlot to send the image to the printer; "file" tells EasyPlot to print to the given file.</p>

/fpo	<p>Set a print option. (File, Print, Options)</p> <p>usage: /fpo code input</p> <p>The “code” is a two-letter code. The “input” can be a point size, page size, line weight, etc. To see a list of all the print option batch commands, go into EasyPlot and run the print options button. Select the sv-save configuration button and look at the attrib.ep file.</p>
/fx or /fq	<p>Quit out of the program. (File, Exit or Quit)</p> <p>usage: /fx y</p>
/oaam	<p>Turn minor tic marks on or off for an axis. (Options, Auto-tics, Min tics)</p> <p>usage: /oaam a ON/OFF</p> <p>The <i>a</i> specifies the axis as <i>x</i>, <i>y</i>, or <i>z</i> and the “ON” or “OFF” specifies whether to turn the marks on or off for the axis.</p>
/oaamm	<p>Turn major tic marks on or off for an axis. (Options, Auto-tics, Maj tics)</p> <p>usage: /oaamm a ON/OFF</p> <p>The <i>a</i> specifies the axis as <i>x</i>, <i>y</i>, or <i>z</i> and the “ON” or “OFF” specifies whether to turn the marks on or off for the axis.</p>
/oaas	<p>Set the locations of tic marks for an axis. (Options, Auto-tics, Set locs)</p> <p>usage: /oaas a <i>n1</i> <i>n2</i> <i>n3</i></p> <p>The <i>a</i> specifies the axis as <i>x</i>, <i>y</i>, or <i>z</i>, <i>n1</i> is the location of a major tic mark, <i>n2</i> is the increment between major tic marks, and <i>n3</i> is the number of minor tics between major tic marks.</p>
/og	<p>Turn the grid on or off. (Options, Grid)</p> <p>usage: /og ON/OFF</p>
/ogs	<p>Select the grid style, or density. (Options, Grid, Style)</p> <p>usage: /ogs a <i>s</i></p> <p>The <i>a</i> specifies the axis as <i>x</i> or <i>y</i>, and <i>s</i> is f for a full grid (a grid line at every tic mark), h for a half grid (a grid line major tic marks only), m for a manually-added tics grid (a grid line at manually-added tics only), or n for no grid.</p>

- /ol** Set axis to log scale. (**Options, Log**)
 usage: **/ol** *a* ON/OFF
 The *a* specifies the axis as *x* or *y*. An “ON” following the axis sets it to log scale and an “OFF” sets it to linear scale .
- /oll** Turn the legend on. (**Options, Legend**)
 usage: **/oll** *x y*
 The *x* and *y* are optional parameters that specify the location of the upper-left corner of the legend in graph coordinates.
- /or** Set the range of an axis. (**Options, Range**)
 usage: **/or** *a min max*
 The *a* specifies the axis as *x*, *y*, or *z*, and the “min” and “max” specify the minimum and maximum for the axis.
- /sa c** Center tic marks (**Style, Attributes, center tics**)
 usage: **/sa c**
 Draw tic marks half inside and half outside graph.
- /sa d** Scale size of data markers. (**Style, Attributes, data mark size**)
 usage: **/sa t** *s*
 The *s* is an integer or fractional scale factor. If $s > 1$, tics are drawn longer than the default size.
- /sa l** Title a curve. (**Style, Attributes, legend title**)
 usage: **/sa l** “title” *c*
 Assign a legend title to a curve. Place in data file before the data. The *c* specifies the data column. If *c* is not specified, the title is assigned to column 2.
- /sa m** Select data mark, dash pattern, bar fill patten, & color. (**Style, Attributes, mark, dash & color**)
 usage: **/sa m** *a c*
 The *a* is an integer from 1 to 14 which specifies the attribute set and *c* is an integer which specifies the data column. If *c* is not specified, the color is assigned to column 2.

/sa t	Scale size of tic marks. (Style, Attributes, tic size) usage: / sa t <i>s</i> The <i>s</i> is an integer or fractional scale factor. If $s > 1$, tics are drawn longer than the default size.
/sb	Turn bar style on or off. (Style, Bar) usage: / sb ON/OFF
/sc	Turn connecting-lines style on or off. (Style, Connect) usage: / sc ON/OFF
/sd	Turn dash style on or off. (Style, Dash) usage: / sd ON/OFF
/se	Turn error-bar style on or off. (Style, Error Bars) usage: / se <i>a</i> ON/OFF The <i>a</i> specifies the axis as <i>x</i> or <i>y</i> .
/sm	Turn data mark style on or off. (Style, Mark Pts) usage: / sm ON/OFF
/st	Turn 3D style on or off. (Style, Three D) usage: / st ON/OFF
/td	Define columns in a data file. (Tools, Define Data) usage: / td "string" where "string" is the column definition string .
/tr	Set resolution, or number of computed points, in calculated curves. (Tools, Resolution) usage: / tr <i>n</i> where $1 < n < 100,000$. Curve resolution affects curves generated from equations and polynomial curve fits.
/ts	Turn scroll bars on or off. (Tools, Scroll) usage: / ts ON/OFF
/tx	Turn cross-hair feature on or off. (Tools, X-hair) usage: / tx ON/OFF

Appendix B

Non-Menu Batch Commands

The non-menu, or double-slash commands begin with two slashes. The commands can be in upper- or lowercase. You can put the commands in any ASCII file you read into EasyPlot. In early versions of EasyPlot, starting with EasyPlot for DOS, non-menu commands were used to customize the software while keeping the user-interface simple. Today, most configuration commands are built into the **Preferences** dialog, or have otherwise been integrated into interface. Most of the commands listed here are specific to batch processing or are kept for compatibility with older versions of EasyPlot.

//bat_esc By default, EasyPlot stops reading data and batch files when you hit the **<esc>** key. You can either disable the **<esc>** key or make it close EasyPlot altogether.

usage: **//bat_esc** OFF/ABORT

The **//bat_esc** command affects the operation of the **<esc>** key only when EasyPlot is reading data and/or batch files.

//binary Read a block of binary data.

usage: **//binary** *c* *r*

where *c* is the number of column, *r* the number of rows. Follow *r* with a carriage return (no line feed) and the data in 4-byte floats.

//bfd Use denser fill patterns on bar charts.

usage: **//bfd** *n*

where *n* is an integer greater than or equal to 1. If *n* = 2, for example, fill patterns will be twice as dense. Interactive equivalent in **Bar charts Preferences**.

<code>//caption</code>	Set caption on EasyPlot window. usage: //caption "your window caption"
<code>//col_name</code>	Name columns of data. See Section 3.2 for details. usage: //col_name <i>p1 p2 p3 ...</i>
<code>//columns</code>	Have EasyPlot configure ASCII data files with <i>n</i> data columns, regardless of how the file is formatted. usage: //columns <i>n</i> With the //columns command set, EasyPlot ignores line breaks in files. The //columns command is not reset after a file is read; turn it off by reissuing //columns with <i>N</i> = 0. Interactive equivalent in File reading Preferences .
<code>//ctp</code>	Map comma to period when entering data into spreadsheet, for entering data in European decimal format. usage: //ctp
<code>//cwt</code>	Set line-weight of a curve. This is the batch equivalent of setting line-weight in the Data Style dialog box. usage: //cwt <i>w [c]</i> where <i>w</i> is the weight of the line (1, 2, 3,...) and <i>c</i> is the column number to which the weight applies. If you do not specify <i>c</i> , EasyPlot assigns the weight to column 2.
<code>//cx</code>	Tell EasyPlot to label the <i>x</i> -axis of a contour map from x_{\min} to x_{\max} . This command works only with contour maps. The "cx" stands for "contour <i>x</i> -range". usage: //cx x_{\min} x_{\max} The minimum and maximum values can be integer or real numbers. By default, EasyPlot labels the <i>x</i> -axis from 0 to the number of columns in the <i>z</i> -matrix. The //cx and //cy commands must be placed in the data file.

<code>//cy</code>	Same as //cx for the y-axis.
<code>//def_leg_pos</code>	Set the initial, default legend position. Interactive equivalent in Legend Preferences . usage: //def_leg_pos <i>x</i> <i>y</i> The <i>x</i> and <i>y</i> are absolute screen coordinates for the top-left corner of the legend. Use numbers between 0 and 1. (0,0) is the lower-left corner of the screen and (1,1) is the top right.
<code>//def_str</code>	Set the default column-definition string. See Section 3.4. usage: //def_str "string"
<code>//degrees</code>	Set trig functions to operate in degrees. usage: //degrees Use //radians to switch to radians. Interactive equivalent in the Curve fit, Transform , and Open (in eqn mode – §2.1) dialogs.
<code>//dig</code>	Digitize data from a graph on the screen to a file (§4.15). usage: //dig filename [A] [X/Y]
<code>//disp_fn</code>	Display the names of files read on a graph as annotations. usage: //disp_fn If a file contains links to other files, only the original filename is displayed. EasyPlot places all filenames at the same location. If you read multiple files, move the notes to make them readable. Interactive equivalent in File reading Preferences .
<code>//enot</code>	Use exponential notation on a primary or secondary axis. usage: //enot <i>x/y</i> ON/OFF
<code>//env</code>	Set a batch macro (§6.3.2) from a system environment variable. usage: //env var-name <i>n</i> where "var-name" is the name of the environment variable and <i>n</i> is the EasyPlot macro number to set.
<code>//fbw</code>	Force width of bars on bar chart. usage: //fbw <i>W</i> where <i>W</i> is the width of the bars in graph coordinates. Use if the bars EasyPlot draws are too thin due to two data points that are

much closer together than other points. Available interactively in **Bar chart Preferences**.

- `//fit_eqn` Suppress curve-fit equations. Interactively equivalent in **Curve fit Preferences** as **Display fit equation on graph** checkbox.
usage: `//fit_eqn` ON/OFF
- `//fit_fcn` Add function to pick-box list for curve fitting.
usage: `//fit_fcn` “eqn”
Interactive equivalent: **Add >>** button in **Curve fit** dialog.
- `//fix_ms` Use the same size data markers on all printed graphs. Otherwise, EasyPlot scales data marks based on graph size. Available interactively in **Markers Preferences**.
usage: `//fix_ms`
- `//fkey` Assign a batch command to an F-Key.
usage: `//fkey` *n* “command”
where *n* is an F-Key number (1 to 10) and “command” is a complete or partially complete EasyPlot batch command (§6.6).
- `//frame` Print a frame around all the text and graphs on a page.
usage: `//frame` *D*
D is the distance in inches between the outer edge of the graph and the frame. Available interactively in **Printing Preferences**.
- `//hide_ax` Batch-command equivalent of unchecking the **Axis** checkbox (under **Generate**) in the **Axis Setup** dialog.
usage: `//hide_ax` *a*
where *a* is the axis, either **x** or **y**.
- `//hor_y` Print y-axis title above y-axis without rotating it. Available interactively in **Text format Preferences**.
usage: `//hor_y`
- `//inf+ & //inf-` Use as a data value to represent positive and negative infinity.
- `//input` Prompt user for input from a batch file (§6.3.2).
usage: `//input` *n* “prompt” “string” [-file]

<code>//lad</code>	<p>Label all decades on log-scale axes.</p> <p>usage: //lad</p>
<code>//leg_ls_sc</code>	<p>Scale the spacing between entries in legends</p> <p>usage: //leg_ls_sc <i>s</i></p> <p>where <i>s</i> is a scale factor. If $s > 1$, you increase the space between legend lines. Available interactively in Legend Preferences.</p>
<code>//leg_msc</code>	<p>Scale size of data marks in legends.</p> <p>usage: //leg_msc <i>s</i></p> <p>where <i>s</i> is any number greater than 0 that scales legend marks relative to the default size. Available in Legend Preferences.</p>
<code>//ls</code>	<p>Set number of characters allowed for tic-mark labels. EasyPlot switches to scientific notation if the digits required for labels exceeds this number. Decrease to force use of exponential notation. Increase to suppress exponential notation.</p> <p>usage: //ls <i>n</i></p> <p>where $n > 0$. Default label space is 5. Interactively, you can switch an axis to exponential notation in the Axis Setup dialog.</p>
<code>//m</code>	<p>Missing data point.</p> <p>usage: //m</p> <p>Place in a data file wherever there is a missing point.</p>
<code>//macro</code>	<p>Set a batch-command macro. There are two forms:</p> <p>usage: //macro <i>n</i> "string"</p> <p>usage: //macro <i>n</i> -file filename <i>m</i></p> <p>The first assigns "string" to macro <i>n</i>. If the string contains spaces, place quotes around it. The second assigns the m^{th} string in the given file to macro <i>n</i>. See Section 6.3.2 for details.</p>
<code>//max_cols</code>	<p>Allow for more than 150 columns of data.</p> <p>usage: //max_cols <i>n</i></p> <p>Use values on <i>N</i> between 150 and 1024. Available interactively in File reading Preferences.</p>

<code>//max_conts</code>	<p>Raise the maximum number of contour levels allowed on a contour map above the default of 35.</p> <p>usage: //max_conts <i>n</i></p>
<code>//mirror</code>	<p>Set up axis linking / mirroring.</p> <p>usage: //mirror <i>A1 n A2</i></p> <p><i>A1</i> and <i>A2</i> specify the axes to link. Use x / y (primary axis), x2 / y2 (secondary axis), or xpN / ypN (private axis #<i>N</i>, where 1 is the private axis closest to the graph). The <i>n</i> specifies the graph; 0 is the current graph being set up in the batch file, 1 is the next graph, -1 the previous graph, etc. To see examples, set up axis mirroring interactively, save the graph or graphs, and look at the save file.</p>
<code>//nc</code>	<p>Split a data file into separate parts. The //nc tells EasyPlot to stop reading the file, install data it has read, and continue reading as though from a new file. The //nc stands for “new curve”.</p> <p>usage: //nc</p>
<code>//no_border</code>	<p>Do not print a border around the current graph.</p> <p>usage: //no_border</p> <p>The //no_border command stops EasyPlot from drawing a rectangle around <i>xy</i> plots and a circle around polar plots. It does not stop EasyPlot from drawing the axes which, on <i>xy</i> plots, make up the bottom and left sides of the border. Available interactively in Printing Preferences as Draw box around graph.</p>
<code>//pause</code>	<p>Pauses batch processing by <i>n</i> seconds.</p> <p>usage: //pause <i>n</i></p>
<code>//pick_cols</code>	<p>Let user select columns interactively with Define Data button while still in batch mode (§3.2).</p> <p>usage: //pick_cols</p>
<code>//pos</code>	<p>Position a graph on the screen.</p> <p>usage: //pos <i>x₁ y₁ x₂ y₂</i></p> <p><i>x₁</i>, <i>y₁</i>, <i>x₂</i>, and <i>y₂</i> are numbers from 0 to 1. (0,0) is the lower-left corner of the screen and (1,1) the top right. (<i>x₁</i>, <i>y₁</i>) sets the lower-left corner of the graph.</p>

- //pr_shift** Adjust position of printed graph.
 usage: **//pr_shift** *x* *y*
 The *x* and *y* specify how much to shift the graph in the respective directions and are given in dimensions of 300dots/inch. You can shift the page in the Print Previewer (§8.3).
- //priv** Set up a private axis.
 usage: **//priv** *x/y* [**Column**=*N*] [**Range**=*A,B*]
 [**Log**=on/off] [**Exp**] [**MAjor**=on/off] [**MInor**=on/off]
 [**REv**=on/off] [**ROt**=on/off] [**Tics**=*c,d,m*] [**Float**=*t*,“Title”]
 The parameters can be in any order. Parameters in brackets are optional. For optional parameters, only the capitalized letters are required (**R** is equivalent to **Range**, for example). The “=on” or “=off” is optional; if not given, EasyPlot uses **ON**. The **Column** parameter specifies the data column which gets the private axis; the default is 2. For the **Range**, *A* and *B* are the minimum and maximum axis ends. **MAjor** and **MInor** turn major and minor tic marks on or off (they are on by default). **REv** reverses the axis direction. **ROt** rotates *x*-axis tic labels; it has no affect on *y*-axes. **Tics** sets up the starting point (*c*), distance between tics (*d*), and the number of minor tics (*m*). **Float** adds a floating tic at coordinate *t* with the given label. You can have any number of **Float** parameters. To see examples, set up graphs with private axes, save them, and look for **//priv** commands in the save file.
- //qdata** Read quoted data as data. Available in **File reading Preferences**.
 usage: **//qdata**
- //quit_key** Create a hot-key for quitting EasyPlot.
 usage: **//quit_key** *a*
 where *a* is the ASCII code of the key. A few examples are 27 for the <esc> key, 120 for **x**, and 113 for **q**. Use a lower-case letter if you want to use the same key with EasyPlot for DOS; the DOS version requires <Caps Lock> for upper-case hot-keys to work.
- //radians** Set trig-functions to operate in radians.
 usage: **//radians**
 Use **//degrees** to switch to degrees. Available in the **Curve fit**, **Transform**, and **Open** (in equation mode – §2.1) dialogs.

<code>//redef</code>	<p>Redefine data columns. Use in batch files to change column definition of data already on a graph.</p> <p>usage: //redef <i>string</i></p>
<code>//rev_curves</code>	<p>Reverse the numbering order of curves in pick boxes.</p> <p>usage: //rev_curves</p> <p>When EasyPlot prompts you to select a curve, it lists them in numbered boxes. By default, EasyPlot numbers curves in reverse order from the way they were added to the graph; the first curve placed on the graph has the highest number and the last, or newest, curve has the number 1. The //rev_curves command makes the oldest curve number 1 and counts up to the newest. Available interactively in System Preferences.</p>
<code>//rot_x</code>	<p>Rotate <i>x</i>-axis tic-mark labels by 90°. Also in Axis Setup dialog.</p> <p>usage: //rot_x</p>
<code>//scale</code>	<p>Sets axis against which next curve is plotted.</p> <p>usage: //scale <i>X Y [c]</i></p> <p>where <i>X</i> and <i>Y</i> are either x / y (primary axis), x2 / y2 (secondary axis), or xp<i>N</i> / yp<i>N</i> (private axis #<i>N</i>, where <i>N</i>=1 is the private axis closest to the primary axis). The <i>X</i> specifies the <i>x</i>-axis and <i>Y</i> the <i>y</i>. To see examples, look at save files for multiple-axis graphs.</p>
<code>//scrn</code>	<p>Turn the screen on or off for quicker, cleaner batch processing.</p> <p>usage: //scrn ON/OFF</p> <p>Be sure to turn the screen back on before the batch file ends.</p>
<code>//sdat</code>	<p>Use same data set as in the next graph in the stack.</p> <p>usage: //sdat</p> <p>If you read exceedingly huge, multiple-column data sets and want to plot individual columns in different graphs, read the file into one graph, create a new graph, and type //sdat in the EasyPlot-standard Open dialog (§2.1). This feature is roughly implemented; trying to delete graphs that share data may crash the program. We don't recommend you use this feature but if you do, set up your graphs, print, and exit. A safer alternative is the Copy data option in File reading,2 Preferences.</p>

<code>//style</code>	<p>Set up styles for a data column.</p> <p>usage: <code>//style S C</code></p> <p><i>S</i> is a string of style characters: C (connect), M (markers), D (dashed), B (bar), and X or Y (error bars), for example CM (solid lines & marks), or CMD (dashed lines & marks). <i>C</i> is the data column number.</p>
<code>//tag</code>	<p>Assign tag numbers to curves automatically for doing math on multiple curves (§4.2.2).</p> <p>usage: <code>//tag [A]</code></p>
<code>//td_scale</code>	<p>Scale distance between graph box and the graph and axis titles up or down by <i>n</i>. Interactive equivalent in Printing Preferences.</p> <p>usage: <code>//td_scale n</code></p>
<code>//udf_tol</code>	<p>Set tolerance factor for nonlinear curve fitting.</p> <p>usage: <code>//udf_tol n</code></p> <p>where <i>n</i> is any number. The default is $n = 10^{-8}$. Interactive equivalent in Curve fit Preferences.</p>
<code>//wait</code>	<p>Suspend batch processing until someone hits a key.</p> <p>usage: <code>//wait</code></p>
<code>//xform_fcn</code>	<p>Add a custom transformation equation to the list of pick boxes.</p> <p>usage: <code>//xform_fcn "your eqn"</code></p> <p>Interactive equivalent: Add >> button in Transform dialog.</p>

Index

Numerics

3D contour lines 84
3D data 82

A

adding
 annotations 28–30
 axes 42
 curves 11
 grid lines 33–34
 legend 41–42
 lines & arrows 32–33
 tic marks 36–39
 titles 28
 curve/legend 41–42
 graph & axis 28
 x - & y -axes 42
annotation color 29
annotations 28–30
 attaching to screen coordinates 29
 deleting 28
 editing 28
 frame around 29
 rotating 29
 saving in screen coordinates 30
arrows for pointing 32–33
arrows, sketching on graph 32
ASCII data files 11, 14, 51
aspect ratio of printed graph 118
asymmetric error bars 22
autorange 45
averaging ranges 77–78
axes on 3D plots 85
axis
 autorange 45
 labeling 36–39

 linking & mirroring 43
 placement 40
 private 42
 reverse-direction 41
 scale (log/linear) 40
 second x - or y -axes 42
 setting range 35, 45
 titles 28

Axis Setup dialog 34–41

B

backslash - extend-line character 51
Bar chart preferences 25
bar charts 17, 24–25
 darkness of fill patterns 25
 fill patterns 18
baseline subtraction on integrations 76
batch commands
 adding tic marks with 38
 case sensitivity 91
 concatenating strings 97
 defining columns with 57
 in files 53, 55
 macros 92
 menu 89–91
 non-menu 91–94
 setting defaults 94
binary data files 13–14
bold text 30
breaking curves at missing values 21
Browsing Files 12

C

caliper feature 32
case sensitivity of batch commands 91
clearing behind open markers 20
Clipboard Plotting 13

- color
 - as 3rd or 4th dimension 88
 - map 19, 87
 - of annotations 29
 - of curves 18
 - on 3D surface plots 84
 - printing 117
 - color palettes 112
 - Color Setup dialog 112
 - column naming 53–54
 - columns of data
 - defining 23, 55–57
 - hiding/ignoring 55
 - maximum number of 51, 87
 - comma as decimal marker 17
 - command-line parameters 103
 - comments in data files 59
 - computing error bars 67
 - concatenating string in batch files 97
 - connect style 20
 - connecting line thickness 26
 - contour lines, on 3D plots 84
 - contour plots 86–87
 - color maps 19
 - defining data for 55–57
 - max # of contour lines 87
 - placing contours 38, 86
 - positioning contour lines 36
 - x*- & *y*-ranges 92
 - converting xyz triplets to surfaces 87
 - Copying to the Clipboard 120
 - cross-hair tool 80
 - cubic spline 73
 - Current directory, remembering for next session 106
 - curve fitting 70–72
 - cubic spline 73
 - exponential 70–72
 - logarithmic 70–72
 - nonlinear 70–72
 - numeric stability 72
 - polynomial 70–72
 - residual error 72
 - weighted 71
 - Curve Toolbar 65
 - curves
 - default line weight 21
 - deleting 50
 - differentiating by linewidth 26
 - math on multiple 67–69
 - number of points in computed 63
 - resolution 71
 - restyling 17
 - restyling all 27
 - titles for legend 41–42
 - transforming 65–69
 - CXMAX, CXMIN 93
 - cyclical data 21, 114
- D**
- darkness of bar fill patterns 25
 - dashed lines 17, 20
 - have EasyPlot generate 22
 - selecting pattern 18
 - dast & dot length on printouts 119
 - data
 - columns, max # of 15, 51
 - columns, min # of 16
 - date & time 58–59
 - editing 108–109
 - entering 12, 109
 - evenly spaced 78
 - interpolating evenly spaced 78
 - missing values 91
 - numerical range 53
 - quoted 15
 - representing infinity 53, 92
 - restyling 17
 - transformations 65–69
 - data columns
 - ignoring extra 15
 - maximum number of 15, 87
 - data files
 - batch commands in 53, 55
 - binary 13–14
 - comments in 59
 - creating 51–53
 - defining columns 23, 55–57
 - for 3D plots 82
 - format 51–53
 - missing values in 91
 - reading 55
 - spreadsheets 51, 55
 - data markers 17, 18
 - clearing area behind open marks 20
 - marking every Nth point 18, 26
 - passing connecting through 20
 - selecting 18
 - size 19
 - data smoothing 77–78

Data Style dialog 26
 date & time data 58–59
 date & time stamps on printouts 105
 decimating 13, 77–78, 92
//def_str 57
 default
 column-definition string 57
 settings 94
 styles 18, 94
 defining columns of data 55–57
 as error values 23
 to choose graph type 55–56
 with batch commands 57
 with **Define Data** 55
**** key 32
 deleting
 annotations 28
 axes 43
 curves 50
 graphs 47
 legend 41
 rows/columns in Data Table 110
 dependent axis 55–56
 derivatives 76–77
 difference equations 69
 differentiation 76–77
 digital filtering 69
 digitizing from screen to file 80
 directory for scratch files 113
 directory, remembering for next session 106
 discontinuities in data 21, 92
 discontinuity checker 21
 distance between points, measuring 32
 drawing lines/arrows 32–33
 dual-axes 42

E

EasyPlot-standard menu 1, 107
 editing
 annotations 28
 data 108–109
 entering data 12, 109
 environment variables 113
ep.cal 76
epw.inn / **epw32.inn** file 107
 equations 51, 60–63
 number of points computed 63
 recomputing 62
 surface 60–63

transforming data 65
 error bars 22–24
 computing 67
 defining error columns 23
 entering error values 22
 transforming 67
 weighting fits with 71
 European characters 30
 European decimal format 17
 exiting the program 106
 exponential notation, importing 16
 exporting graphs 120
 expressions 105
 extend-line character 51
 extra data columns, ignoring 15
 extracting text from files 94

F

FFT 78–80
 file reading preferences 15
 fitting curves 70–72
//fkey 96
 F-key macros 96
 floating tic marks 37–39
 font & font size 25–26
 scaled by graph size 26
 formatting codes in text 31
 Fourier transform 78–79
 frame around annotations 29
 frame around graph & text 119
 frequency change over time 79
 function-key macros 96

G

generate all dashed lines 22
 graph
 deleting 47
 maximizing and minimizing 47
 moving 46
 popping 46
 saving 48–49
 Tiling 46
 title 28
 graphical data editor 108
 Greek characters 30
 grid 33–34
 Preferences 34
 ultra-fine 34
 gridding 3-D data 87

H

Hamming window 78–79
 header information
 in ASCII files 14
 in binary files 13
 help 115
 hidden-line removal 84
 hiding columns 55, 110
 histogram 74

I

ignoring line breaks 16
 ignoring text in ASCII data files 14
 ignoring/hiding columns 55, 110
 image plots 87
 Importing data from Clipboard 13
 independent axis 56–57
//inf+ & //inf- 53, 92
 infinity - as data point 53, 92
 inputting data 109
 inserting rows/columns in Data Table 110
 installing EasyPlot 1
 uninstalling 1
 integer data 16
 integration 75–76
 baseline subtraction 76
 interpolating evenly spaced data 78
 inverse FFT 79
 isometric surface 83
 italic text 30

K

Kaiser window 78–79

L

labeling axes 36–39
 landscape orientation 117
 legend 41–42
 creating 41
 deleting 41
 distance between entries 42
 moving 41
 rearranging entries 41
 titling curves 27
 line breaks, ignoring 16
Line button 32–33
 Line chart preferences 21
 line plots 17, 20

line weight
 default for curves 21
 line weight on printouts 117
 line, dashed & solid 20
 Line/Arrow Properties 33
 linear regression 70–72
 linear scales 40
 linking axes 43
 locking cursor to a curve 80
 log scales 40

M

//m, missing data 91
//macro 92
 macros
 assigning to F-Keys 96
 in batch files 92
 referencing curve statistics 67
 major tics 37
Mark Pts 17
 marking every Nth point 18, 26
 math on multiple curves 67–69
 math symbols 30
 math transformations 65–68
//max_conts 87
 maximizing graphs 47
 maximum # of data columns 15
 mean of the data 74
 measuring distance between points 32
 menu system 107
 menu, EP- or Windows-standard 1, 107
 minimizing graphs 47
 minor tics 37
 mirroring axes 43
 missing data points 91
 missing values
 breaking curves at 21
 in Data Table 111
 moving
 annotations 28
 axes 40
 data points 108
 graphs 46
 legend 41
 polar axes 40
 multiple data-page printing 120
 multiple graphs on a screen 45–47
 multiple-axis graphs 42
 multiple-curve math 67–69

N

named ranges in spreadsheet files 55
 naming columns 53–54
 new-curve (**//nc**) command 92
 needle style 25
 network, running EasyPlot on 113
 nonlinear axis scales 38–39

O

on-line help 115
 open markers, clearing behind 20
 overlaying data on a graph 12

P

Page Setup dialog 117–118
 page setup, saving 118
 pen plotters 117
 period on X 21
 periodic data 21, 114
 perspective - with 3D graphs 83
//pick_cols 54
 plotting equations 11, 51, 60–63
 point size 30
 of legend text 42
 selecting 25–26
 polar plots 27–28
 angular range 40
 defining data for 52, 55
 grid 34
 labeling one axis 37
 radial range 35
 switching from xy to polar 56
 polynomial curve fitting 70–72
 portrait orientation 117
 positioning tic marks 36
 Preferences
 Bar chart 25
 file reading 15
 Grid 34
 Line charts 21
 Saving 49
 previewing a printout 119
 Print area 118
 print preview 119
 printers supported 117
 Printing 117–122
 printing
 aspect ratio 118
 color 117

date & time stamps 105
 differences between what-you-see
 & what-you-get 121
 legend 122
 line weights 117
 multiple data pages 120
 polar plots 118
 save page setup 118
 size of graphs 118
 sizing the graph 118
 tic marks inside graph 121

private axes 42

Programming Interface 97–103

Q

Quick ASCII reads 16
 quitting the program 106
 quoted data 15

R

random xyz data, converting to a surface
 87
 range
 autorange 45
 of spreadsheet to read 55
 range averaging 77–78
 range of data values 53
 range on axes 35
 reading coordinates from graph 80
 reading data files 55
 rearranging legend entries 41
 recomputing equations 62
 refreshing the screen 107
 regression analysis 70–72
 removing EasyPlot from disk 1
 residual error of curve fit 72
 resolution of computed curves 63
Restyle button 17
 restyling all curves 27
 restyling data 17
 reverse-direction axes 41
 rotating annotations 29
 rotating in 3D 83
 row length 53
 rows of data, variable length 53

S

Save button 48
 saving annotations in screen coordinates 30
 saving graphs 48–49, 89
 saving preferences 49
 scaling text size with graph 26
 scatter plots 17
 scratch-file directory 113
 screen colors 112
 scrolling 44–45
 selecting
 dash patterns 18
 data marks 18
 font 25
 marker, dash, & color 18
 selectively viewing columns 55, 110
 semilog plots 40
Set Locs button 36
 //setup_dir 113
 shadowing xyz data onto 3D axis planes 85
 singularity in data 53
 size
 of data markers 19
 of printed graph 118
 of text 25–26
 size of printed graphs 118
 slope of a line 32
 smoothing data 77–78
 solid connecting lines 20
 Splines
 surfaces 73
 splines 73
 splitting a data file - the **//nc** command 92
 spot diagrams 75
 spreadsheet Data Table 109
 spreadsheets 11, 51–55
 reading ranges of 55
 standard deviation 74
 statistics 74
 referencing in equations 67
Stats button 74
 strings, concatenating in batch files 97

STYLE menu 17–25

 bars 24–25
 connecting lines 20
 defaults 18
 error bars 22–24
 three d 24
 subscripts 30
 superscripts 30
 surface grid, creating from random xyz data 87
 surface plots 83–87
 surface splines 73
 system requirements 1

T

tagging for multiple-curve math 67
 temp directory 113
 template files 38
 text formatting codes 31
 Text Toolbar 30–31
Three D button 24
 three D data 82
 tic marks 36–39
 adding floating tics 37–39
 with batch commands 38
 direction 39
 flipping onscreen 39
 nonlinear scales 38–39
 placement 36
 positioning with **Set Locs** button 36
 printing inside graph 121
 removing 37
 size 39
 Tiling graphs 46
 time & date data 58–59
 time & date stamps on printouts 105
 titles
 curve/legend 41–42
 batch command for 41
 editing 28
 graph & axis 28
 titling curves (for legend) 27
 //tmp_dir 113
 Toolbar, curve 65
 transformation equations 65–69
 transforming data 65–67
 transposing a graph by redefining columns 57

U

- ultra-fine grid lines 34
- underlined text 30
- undo 106
- uninstalling EasyPlot 1
- unzooming 44

V

- variance 74
- viewing data in 3D 83–85

W

- weighted curve fits 71
- wind rose 25
- Window menu 48
- Windows menu system 1, 107

X

- Xform** button 65–69
- X-hair tool
 - lock to curve 80
- x-hair tool 80

Y

- y-axis, adding a second 42

Z

- zooming 44

Spiral Software Sales

7909 Charleston Court, Bethesda, MD 20817

Tel: 301 767-1675 Fax: 301 767-1675

Email: carol@spiralsoftware.com WebSite: www.spiralsoftware.com

Spiral Software Support

57 Baker Hill Road, Lyme, NH 03768

Tel: 603 795-4004

Email: easyplot@spiralsoftware.com WebSite: www.spiralsoftware.com

