NOESYS

Plot User's Guide and Reference Manual

RESEARCH SYSTEMS
Software ≡ Vision™

Plot Version 1.2 Mac/1.3 Win
May, 1999 Edition

Acknowledgments

Fortner Software and its logo are trademarks of Fortner Software LLC

Plot, Noesys, Transform, T3D and the HDF Browser are trademarks of Fortner Software LLC

Contributors: National Center for Supercomputing Applications (NCSA) at the University of Illinois, Fortner Software, Unidata Program Center (netCDF), The Independent JPEG Group (JPEG), Jean-loup Gailly and Mark Adler (gzip), and Digital Equipment Corporation (DEC).

Research Systems, Inc. documentation is printed on recycled paper. Our paper has a minimum 20% post-consumer waste content and meets all EPA guidelines.

Contents

Part IV: Appendices

Part I: Introduction

Chapter 1:
Welcome

Welcome to Plot. Whether you are a scientist, engineer, or anyone with a large amount of data, Plot will help you plot your data quickly and conveniently.

Plot is unique among plotting packages in that it can import and work with enormous datasets. You can put a million numbers in a column if you have that much memory (4 bytes is required for each number). Plot organizes data into columns of floating point numbers. The spreadsheet-like data window lets you have up to 32,000 columns. The number of rows is limited only by your system's memory.

Plot can read ASCII files where columns are delimited by spaces, tabs, or non-numeric punctuation. Plot also reads non-column files, such as arrays saved in Transform, and places them in the spreadsheet as if they were column files.

Hour	Temp(F)
01	18.37
02	19.88
03	20.10
04	20.89
05	22.93
06	24.50
07	27.62
08	27.07
09	30.28
10	32.05
11	32.86
12	30.32
13	28.65
14	30.21
15	28.54
16	30.41
17	28.25
18	28.12
19	26.68
20	24.52
21	22.23
.	.
.	.

Figure 1-1: Data Organized into Columns

Plot lets you create line, double-y, scatter, number scatter, and error bar plots by simply selecting from the plot gallery. You may also "build" advanced plots in layers, selecting X,Y plot pairs and assigning the desired attributes for each.

The binary file format used for all Fortner Software LLC's software is the Hierarchical Data File (HDF) format, a public-domain standard format for the storage of scientific data and ancillary information. It is an object-oriented format capable of storing many different kinds of data in one file. Plot uses HDF Vset to save column data files. HDF Vset is discussed in Appendix C.

Plot uses a macro language that allows you to save scripts to automate data import/export and plotting tasks. Because Plot supports a text formatting language, you can even control text formatting with macros. Plot for Power Macintosh also lets you send macro scripts to Plot from other programs using AppleEvents.

Plot Lets You...

- Import extremely large data sets—the number of rows is limited only by system memory
- Display data in a spreadsheet-like data window
- Enter and apply formulas in the data window
- Generate line, scatter, double-Y and error bar plots from an easy-to-use plot gallery
- Apply built-in functions
- Build advanced plots applying different attributes to different X,Y pairs in layers
- Save macro scripts of plots and apply them to other data
- Set axis data range, label range, labels, and numeric format
- Add tick marks and grid lines
- Edit lines and symbols
- Add error bars
- Create and edit text labels and legends; control text using text formatting language
- Do curve fitting
- Export data as ASCII text files
- Print plots to any Windows- or Macintosh-compatible printer
- Save or copy plot images for use in presentation programs

Getting Started with Plot

This section describes how to install and upgrade Plot.

Installing Plot

Plot is installed with Noesys from the Noesys CD-ROM. For more information, see *Installation Guide for Windows and Macintosh* included with your Noesys CD-ROM.

Upgrading Plot (Windows)

When upgrading Plot, an additional step is recommended. Besides performing the normal installation mentioned above, we recommend you import the preferences settings and custom macros from your previous version of Plot. Importing preferences settings and custom macros enables you to upgrade Plot and retain this information.

Note

If you decide not to import the preferences settings and custom macros, the new version of Plot will use factory default settings.

To import these settings, start the upgrade version of Plot. Once the upgrade version of Plot is running, follow the procedures below:

1. Select **Preferences** from the Edit menu.
2. Select the **Import** button from the Preferences Settings window.
3. In the Import Preferences window, change the directory to where the previous version of Plot can be found.
4. If the previous version of Plot is from Spyglass, select the file ‘Prefs.spy.’ If the previous version of Plot is from Fortner Software, select the file ‘Prefs.frl.’
5. Select **Open**.

After step 3, you will be prompted if you would like to import your custom macros. By selecting yes, all of your custom macros from the previous version of Plot will be transferred to the new version of Plot.

Updating Custom Macros (Power Macintosh)

The preference file for storing custom macros has changed with the upgraded version of Plot. The old preference file is called ‘Spyglass Settings’ and it stores custom macros from Spyglass Plot. The new preference file is ‘Plot Prefs’ and is located in the Preferences:Fortner Software folder. This file stores Plot custom macros separately.

To copy the custom macros from the old Spyglass tools (Plot and Transform) into the upgraded tools, simply follow the steps below. The file ‘Spyglass Settings’ can be read by the new tools and therefore, following these steps will avoid the tedious task of exporting the custom macros to text files, then reading them back in.

Warning

These steps must be performed before creating any new custom macros, otherwise the new macros will be replaced.

1. Create a folder called ‘Fortner Software’ inside the Preferences folder of the System Folder.
2. Drag the ‘Spyglass Settings’ file (which is located inside the Preferences folder) into the ‘Fortner Software’ folder.
3. Rename the ‘Spyglass Settings’ file to ‘Plot Prefs’.

Note

To copy both sets of custom macros into the new VDA tools, simply duplicate the ‘Spyglass Settings’ file (after it is placed in the ‘Fortner Software’ folder) and rename them to ‘Plot Prefs’ and ‘Transform Prefs’.

Part II: Tours

Chapter 2:

Simple Line Plots

After installing Plot according to the directions in Chapter 1, double-click the Plot icon (Power Macintosh) or select **Plot** from the Program Menu (Windows) to start the program.)

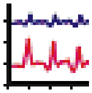


Figure 2-1: Plot Icon

The first time you open Plot, you will be prompted for your name, organization, and registration number. Enter these and click **OK**.

A Simple Line Plot

This chapter will briefly introduce you to most of the major features of Plot. Here you will enter a series of numbers in the data window, plot the data, do a curve fit, and change the appearance of the plot.

The Data Window

A new data window is created each time Plot is started. To open a new data window, choose **New** from the File menu. A Plot data window consists of a series of named columns which are, by default, Column1, Column2, and so on. Click on **Column1**, then enter **y** for the column name and press **Enter**.

Now, with the cursor in the first cell of the same column, enter a number between 1 and 10. Press **Enter** and enter a series of numbers between 1 and 10 in the column. Your data window should now look similar to the figure below.

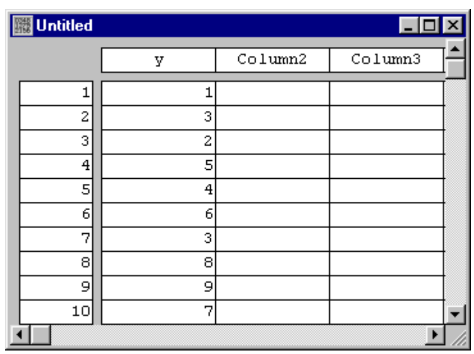


Figure 2-2: New Data Window

Selecting Columns

Next choose **Gallery** from the Graph menu, and then select **Line** on the Gallery hierarchical menu to generate a line plot.

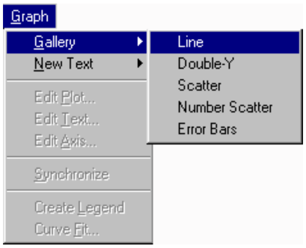


Figure 2-3: Gallery Menu

Now in the Select X,Y Pairs dialog, you can select which columns you wish to plot. Here select **Row Numbers** as the X column and **y** as the Y column, as shown in the figure below.

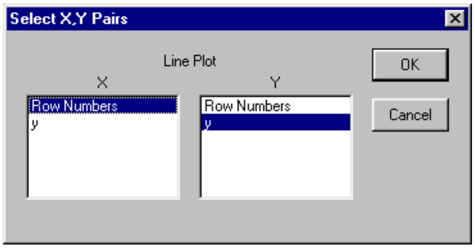


Figure 2-4: Select X, Y Pairs Dialog

Click on **OK**, and a simple line plot will appear.

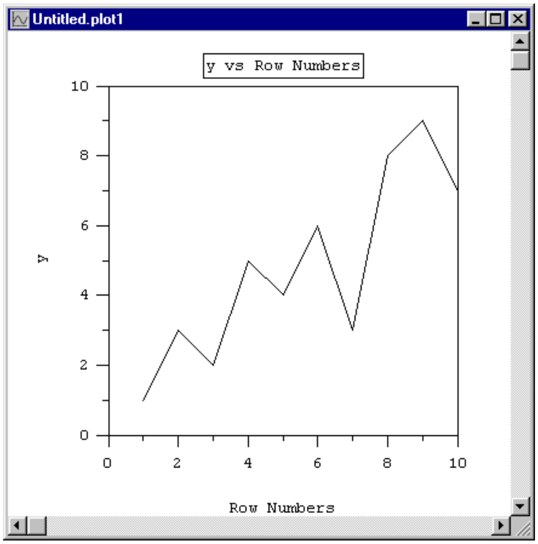


Figure 2-5: Plot Window

Curve Fit

Next we will do a simple linear curve fit on this data. Select **Curve Fit......** from the Graph menu in the Plot window. In the Curve Fit dialog, make sure that **Linear** is selected. Then click on **Calculate** to generate a curve fit equation, which will be displayed in the box in the lower half of the dialog.

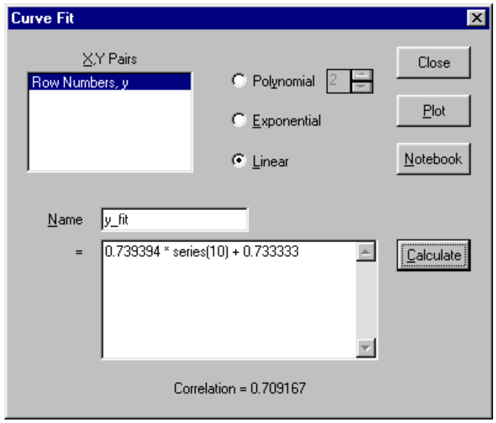


Figure 2-6: Curve Fit Dialog

The curve fit equation is used to create a column of values. These values are then plotted to display the curve fit. Click the **Plot** button to plot the curve fit equation. Then choose **Close** (Windows) or **OK** (Macintosh) to return to the plot window.

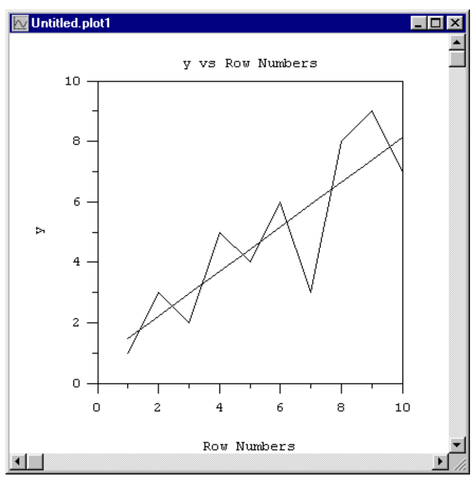
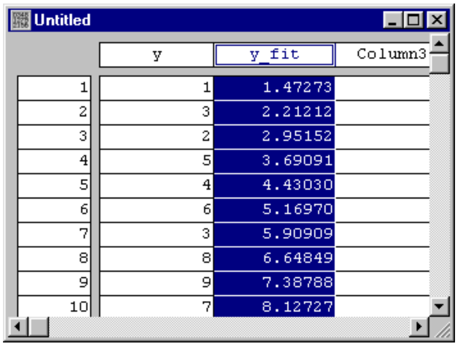


Figure 2-7: Plot Window with Curve Fit

Curve Fit Data Column

Now bring the data window to the front. You can do this either by clicking on its title bar or by selecting **Untitled** from the Windows menu. Note how the data used to generate the curve fit is stored in a new column, called y_fit.



	y	y_fit	Column3
1	1	1.47273	
2	3	2.21212	
3	2	2.95152	
4	5	3.69091	
5	4	4.43030	
6	6	5.16970	
7	3	5.90909	
8	8	6.64849	
9	9	7.38788	
10	7	8.12727	

Figure 2-8: Data Window with Curve Fit

Plot Appearance

Next we change the plot's appearance by modifying its text fonts and sizes, labels, and lines.

Changing the Text

Begin by bringing the plot window back to the front. Choose **Select All** from the Edit menu, then select **Font...** from the Edit menu (Windows) or choose the fonts and styles from the Format menu (Power Macintosh). For this example, select **Arial** font, **Regular/Plain** style, and **12** points. Note that in Windows, a sample of the font with the selected attributes is displayed in the Sample box. Click **OK** to save your changes. All text should now be displayed using the new font.

Next, we'll change just the plot title, 'y vs Row Numbers'. Click on the title to select it. Then select **Font...** from the Edit menu (Windows) or Format menu (Power Macintosh). Change the font size to **14** point. Click on **OK** to save the change. You can also change the axes labels, 'y' and 'Row Numbers' text the same way.

To change the text itself, double-click on any of the text items to open the Edit Text dialog (or select **Edit Text...** from the Graph menu).

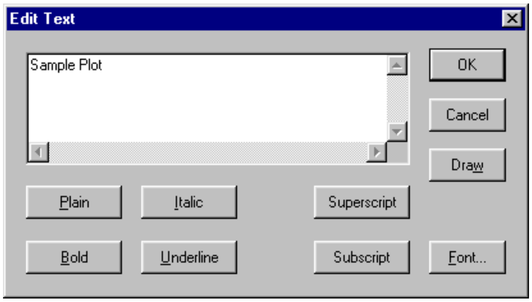


Figure 2-9: Edit Text Dialog

For now, enter a new title for the plot (here 'Sample Plot') and click **OK** to update. Your plot should now resemble the following figure.

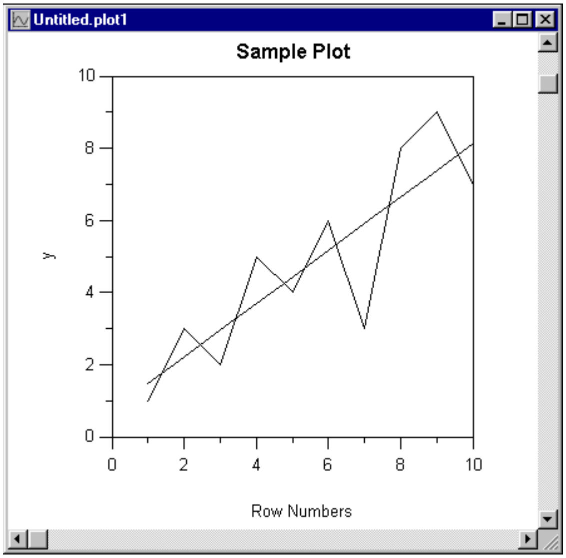


Figure 2-10: Plot Window with Text Changes

Moving and Resizing the Plot

To move the plot, click and hold anywhere within the plot boundaries and drag the plot to the desired location. To interactively resize the plot, click once within the plot boundaries to select it. In Windows, solid black resize boxes will appear at each corner and on each side of the plot; grab one of the resize boxes and drag to resize the plot. In Power Macintosh, click the box in the lower right corner and drag the window to a new size.

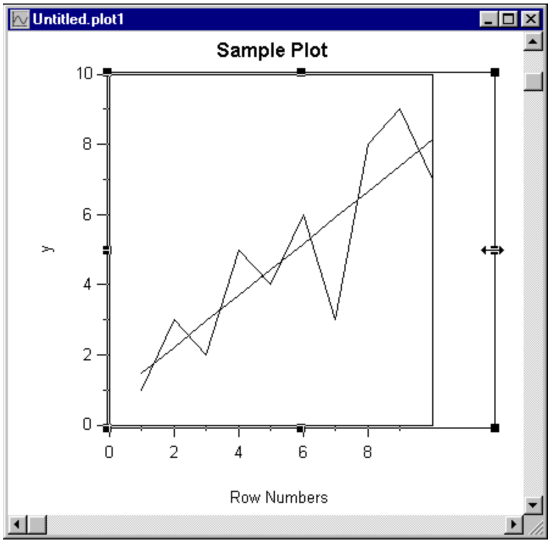


Figure 2-11: Resizing a Plot

Changing the Axes

Next click anywhere on the horizontal axis of the plot to select it separately from the plot itself. Then select **Edit Axis...** from the Graph menu. (Alternatively, you can double-click on the axis.)

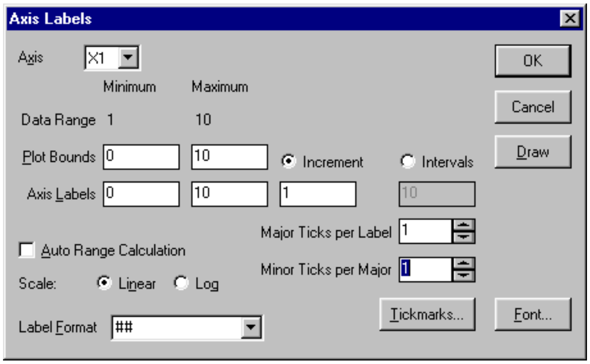


Figure 2-12: Axis Labels Dialog

In the Axis Labels dialog, the Plot Bounds fields set the data range of the plot. The Axis Labels fields set the range of the plot that will be labeled. These are usually, but not always, the same as the Plot Bounds fields. The Increment and Intervals fields are used to determine how many labels are displayed.

In Windows, select the **Increment** (as opposed to Intervals) radio button; in Power Macintosh, click the double arrow once to signify that you wish to specify an Increment value (as opposed to an Intervals value).

Enter **1** as the increment value.

Note

Auto Range Calculation must be deselected before Increment can be selected.

Next enter **1** for the Minor Ticks per Major field. Your dialog should look identical to the one shown above.

The Axis list (in the upper left corner of the dialog in Windows; in the upper right hand corner of the dialog in Macintosh) is used to switch between axes. Open the Axis list and select the **Y1** axis. Make the same changes for this axis as you did for the **X1** axis (Increment of 1, Minor Ticks per Major of 1). Choose the **OK** button to apply these settings. The resulting plot should look similar to the figure below.

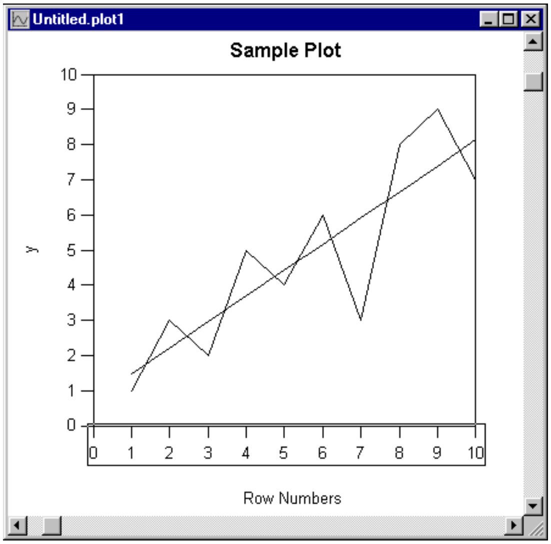


Figure 2-13: Plot window with New Axes

Changing the Lines, Styles, and Colors

Select the plot by clicking within its boundaries then select **Edit Plot...** from the Graph menu (alternatively, you can double-click on the plot). You will see the Edit Plot dialog shown in the figure below, where you can change the attributes of each line in the plot.

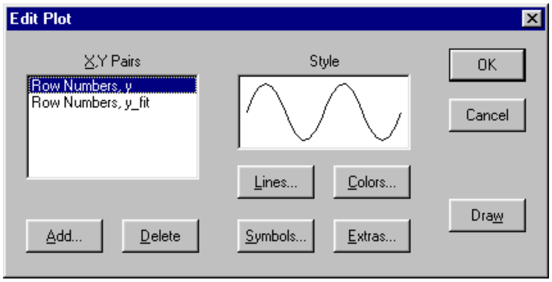


Figure 2-14: Edit Plot dialog

The Pairs list shows you all of the X,Y data pairs that are available for the current plot. The Style/preview box shows you how the selected X,Y pair will be plotted. The Lines..., Symbols..., Colors..., and Extras... buttons let you change the way the selected X,Y pair is plotted. The Add... and Delete buttons at the bottom of the dialog let you add and delete X,Y pairs to the plot.

With the 'Row Numbers,y' X,Y pair selected, click the **Lines...** button. Select **None** as the Style, then click **OK**. Note that the Style/preview box reflects the change.

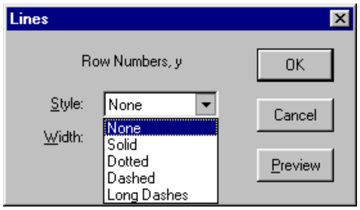


Figure 2-15: Lines dialog

Next, click the **Symbols...** button and select **Draw** as the symbol type. Open the Draw list and select **Circle** (on Power Macintosh this is the open circle symbol which is the 6th symbol from the left). Click **OK** to close the Symbols dialog.

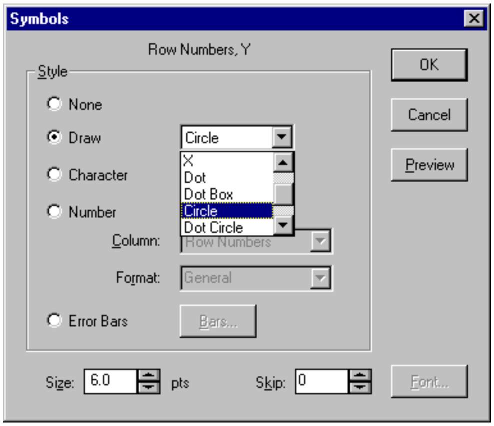


Figure 2-16: Symbols dialog

Now click on the **Row Numbers,y_fit** X,Y pair. Click on **Lines...** and check that the line style is set to **Solid**. Click **OK** to close all dialogs. Your plot should now look like the one below.

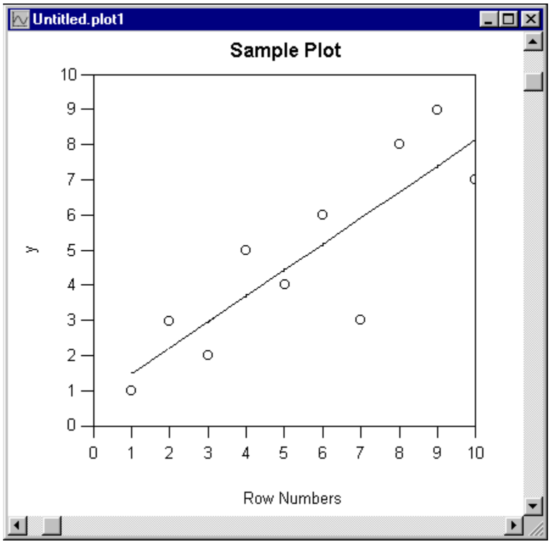


Figure 2-17: Plot Window with New Lines

Adding Error Bars

Our final change to this plot is to add error bars. Select **Edit Plot...** from the Graph menu and choose **Add...**. Select **Row Numbers** for the X column, and **y** for the Y column as shown below.

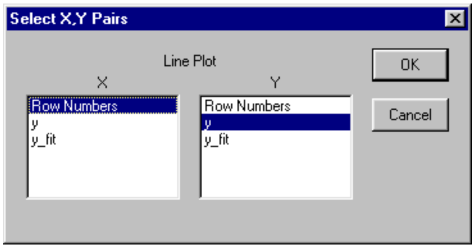


Figure 2-18: Adding another Plot Pair

Click **OK** to add the selected pair and to return to the Edit Plot dialog. You should now see three X,Y pairs, with the first and last being identical. The first 'Row Numbers,y' X,Y pair is needed to display the open circle symbols we added earlier. The second 'Row Numbers,y' X,Y pair is needed to display the error bars.

With the last X,Y pair selected, click **Lines...** and set the line Style to **None**, then click **OK**. Next, click **Symbols...** to bring up the Symbols dialog. Here, click the **Error Bars** radio button, then click on the **Bars...** button to enable error bars.

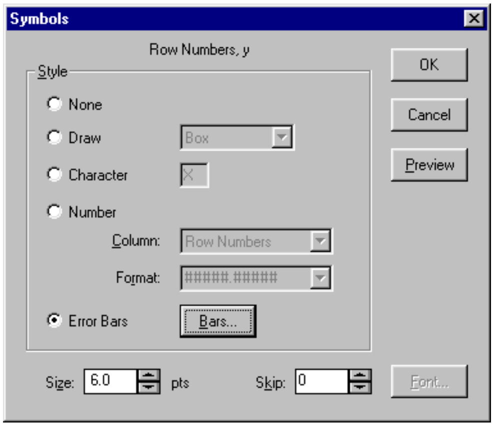


Figure 2-19: Selecting Error Bars

In the Error Bars dialog that now appears, in Windows choose **Sigma** and enter **1** as the number of standard deviations (in Power Macintosh select the **Number of Std. Dev's** radio button and enter a **1**). Click **OK** in this dialog, in the Symbols dialog, and in the Edit Plot dialog.

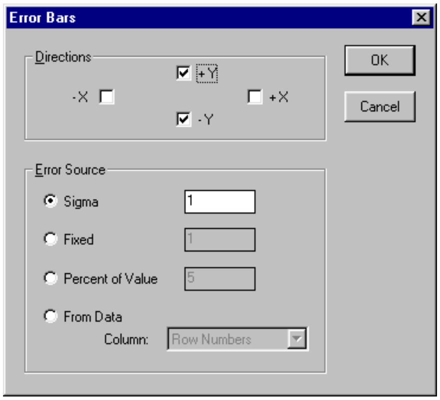


Figure 2-20: Error Bar dialog

The resulting plot should look similar to the figure below. Note that by default, the error bars are symmetric.

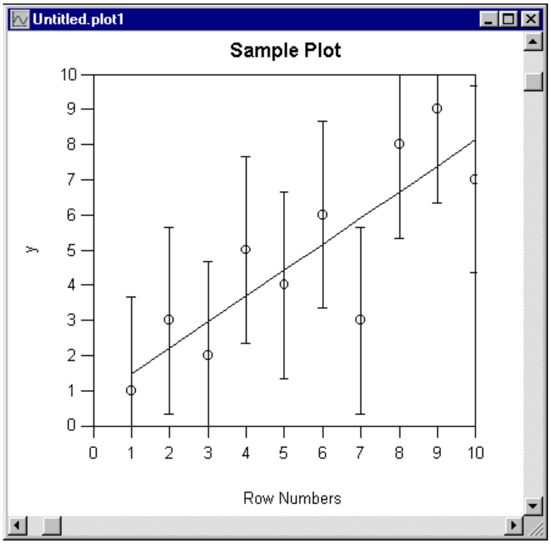


Figure 2-21: Plot Window with Error Bars

Notebook Calculations

The example you are working on brings up a question: What exactly is the standard deviation used for the error bars? We can answer that question by using the notebook to do a few calculations.

Calculating Standard Deviation

To open the notebook, select **See Notebook** from the Data menu (Windows) or select **Untitled.notebook** from the Windows menu (Power Macintosh). Click in the text box and type **print sdev(y)**. With the cursor still on the line you just typed, select **Calculate Now** from the Data menu to evaluate that line. The result is displayed on the next line. The exact value you see will probably be different from the one shown below, since it depends on the values you typed in at the beginning.

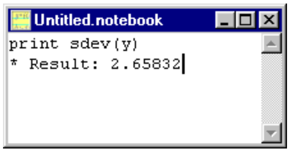


Figure 2-22: Notebook Window

The ‘sdev’ function uses the following expression to calculate the standard deviation of a data column:

$$sd = \sqrt{\frac{\sum (y_i - y_{ave})^2}{N-1}}$$

Figure 2-23: Standard Deviation Calculation

Where the elements of y are y_i, the average of all y values is y_{ave}, and the number of y values is N. We can verify that this is the equation used by sdev by again using the notebook.

Click in the notebook again and enter the following three lines:

```
sum=sum((y-mean(y))**2)
n=rows(y)
print sqrt(sum/(n-1))
```

Select all three lines, then choose **Calculate Now** from the Data menu. The result should be identical to that shown for the 'sdev' function, as shown below.

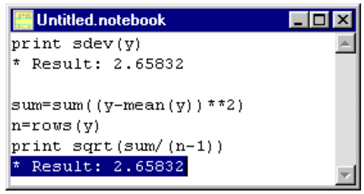


Figure 2-24: More Notebook Calculations

Calculating RMS Error

Although the standard deviation is useful, a more interesting number would be the deviation of the data values from the fitted straight line, called the RMS error. The equation for this is almost identical to that for the standard deviation, except that y_{ave} (a single value) is replaced by $y_{fit,i}$ (a column of values).

$$rms = \sqrt{\frac{\sum (y_i - y_{fit,i})^2}{N-1}}$$

Figure 2-25: RMS Error Calculation

We can calculate this value by entering the following three lines:.

```
diff=y-y_fit
sum=sum(diff**2)
print sqrt(sum/(n-1))
```

Again, select all three lines and choose **Calculate Now**. The value should be significantly smaller than the standard deviation value, as you would expect.

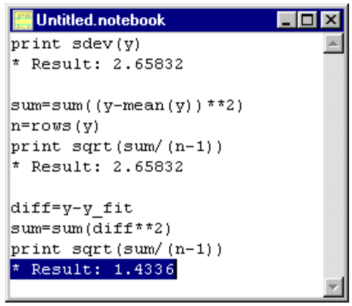
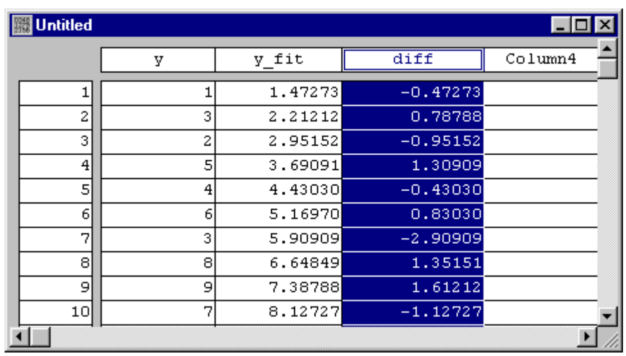


Figure 2-26: Using the Notebook to Calculate RMS Error

Click on the data window and note that it now contains another column called ‘diff.’



	y	y_fit	diff	Column4
1	1	1.47273	-0.47273	
2	3	2.21212	0.78788	
3	2	2.95152	-0.95152	
4	5	3.69091	1.30909	
5	4	4.43030	-0.43030	
6	6	5.16970	0.83030	
7	3	5.90909	-2.90909	
8	8	6.64849	1.35151	
9	9	7.38788	1.61212	
10	7	8.12727	-1.12727	

Figure 2-27: ‘diff’ Column Added to Data Window

Any time the result of a notebook expression is a column of values (as in the case of ‘diff’), a new column is created in the data window. No column is created, however, when the result of an expression is just a single value (as in the case of ‘sum’ and ‘n’).

As our final step in this chapter, we will assign the error bar length to be equal to the RMS error. To do so, bring the Plot window back to the front. Select the plot, then choose **Edit Plot...** from the Graph menu.

Select the last X,Y pair listed (the one we used previously for the error bars). Then click on **Symbols...**. In the Symbols dialog, click on **Bars...** to get to the Error Bars dialog.

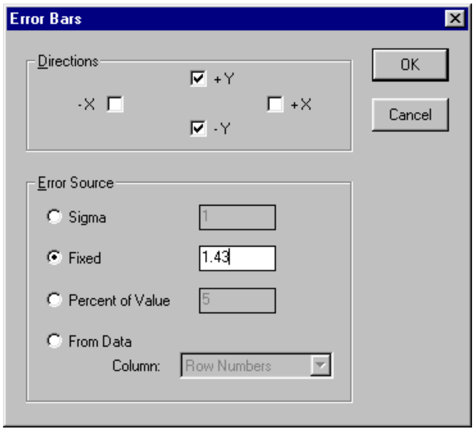


Figure 2-28: Error Bars dialog

Click on **Fixed**, then enter the RMS error that you calculated in the notebook (shown in Figure 2-26) in the corresponding box. Click **OK** in this dialog, in the Symbols dialog, and in the Edit Plot dialog to return to the plot window. You should see a plot similar that shown below.

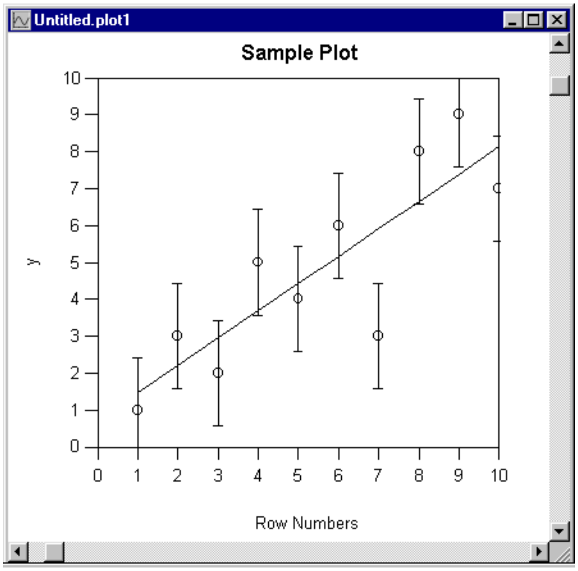


Figure 2-29: Plot Window with RMS Error Bars

Chapter 3:

Double-Y Plots

In this chapter we will read in data from a computer simulation, graph it as a double-y plot, add text annotations, and use macros. If you have not already done so, select **Close All** from the File menu to close all of the windows that you opened in Chapter 2.

Reading a Data File

We will start by importing a text file. Select **Open...** from the File menu, then select the file 'ls753.txt' in the 'Plot\Samples' folder and click **OK**. This data is from a time-dependent computer simulation of the flow of material around a compact star (Nature, 342, 775 (1989)).

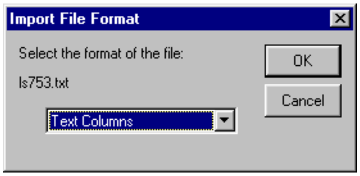


Figure 3-1: Selecting Text Columns Import File Format

If Plot does not automatically recognize the file, it prompts you for more information. In this case, the data file is a text column file, so select that option and click **OK**. The Text Columns dialog will appear.

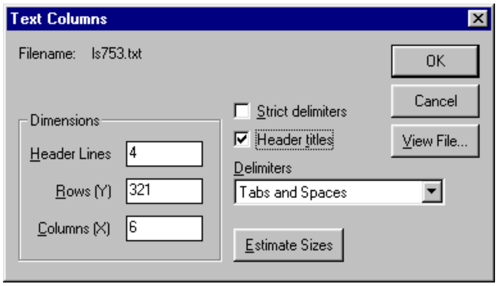


Figure 3-2: Text Columns dialog

In this dialog, make sure that 'Header titles' (Windows) or ‘Last header line contains column titles’ (Power Macintosh) is selected. Note that Plot has pre-scanned the data file, finding that there are 4 header lines, 321 rows, and 6 columns of data. To verify this guess, click on the **View File...** button.

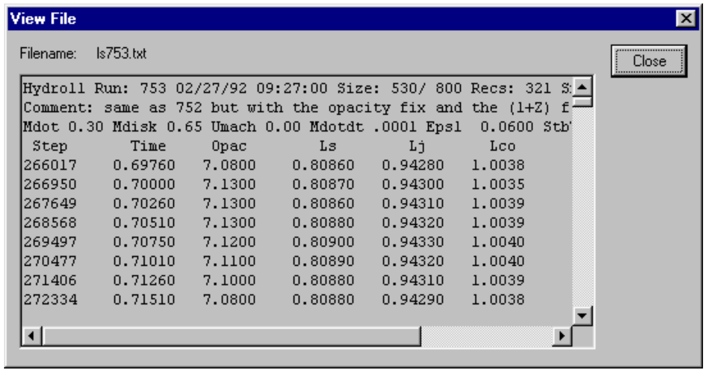


Figure 3-3: View File dialog

The file should look like the one shown above. Note that the fourth line of the file does contain names for each of the columns. Click **Close/Done** to return to the Text Columns dialog. Click **OK** to read the file.

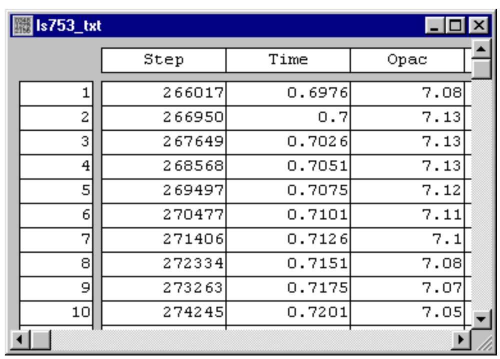


Figure 3-4: The Data Window

The resulting data window should look like the one above. If the names of the columns are Column1, Column2, and so on, then the 'Header titles'/'Last header line contains column titles' check box was not selected. Close this data window and try again.

Creating a Double-Y Plot

Now we will prepare to generate a plot from the imported data.

Entering Plot Labels

We first need to enter new plot labels for some of the columns. Click the **Time** column name, and then select **Column Settings...** from the Data menu (alternatively, simply double-click on the column title).

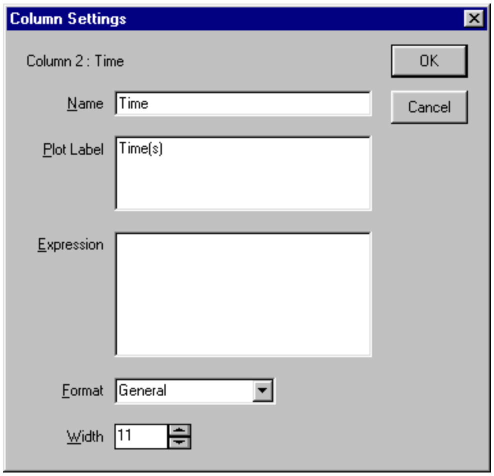


Figure 3-5: Specifying Plot Label for Time Column

In the Column Settings dialog for this column, enter 'Time(s)' for the Plot Label. This name will be used for display whenever the column is plotted. Now click **OK**.

Next, select the **Opac** column and change the Plot Label for this column to **Opacity**. Click **OK**, then open the **Lj** Column Settings dialog and enter **L/L_E** for the Plot Label. Click **OK** to close the dialog.

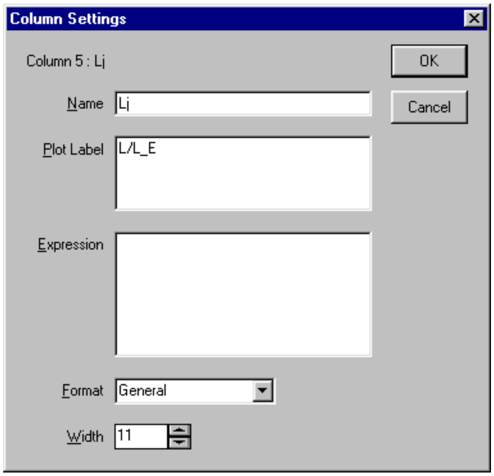


Figure 3-6: Specifying Plot Label for Lj

The underscore (_) character specifies that the character that follows will be subscripted, like L_E. Later in this chapter we will discuss the text formatting codes in more detail.

Selecting Columns

The next step is to select the columns to plot. Select **Double-Y** from the Gallery submenu on the Graph menu.

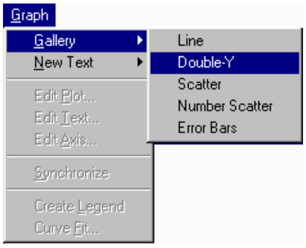


Figure 3-7: Graph Submenu

The first dialog you see is for specifying columns that are plotted on the left Y axis. Select **Time** for the X axis and **Opac** for the Y axis, then click **OK**.

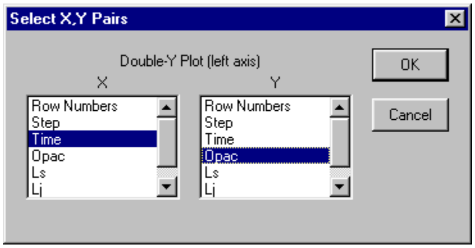


Figure 3-8: Select X, Y Pairs Dialog for Left Axis

The second dialog is for specifying columns that are plotted on the right Y axis. Again specify **Time** as the X axis and **Lj** for the Y axis, and click **OK**.

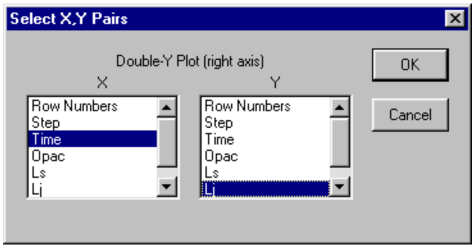


Figure 3-9: Select X,Y Pairs Dialog for Right Axis

You should now see a Double-Y line plot that looks similar to the figure below.

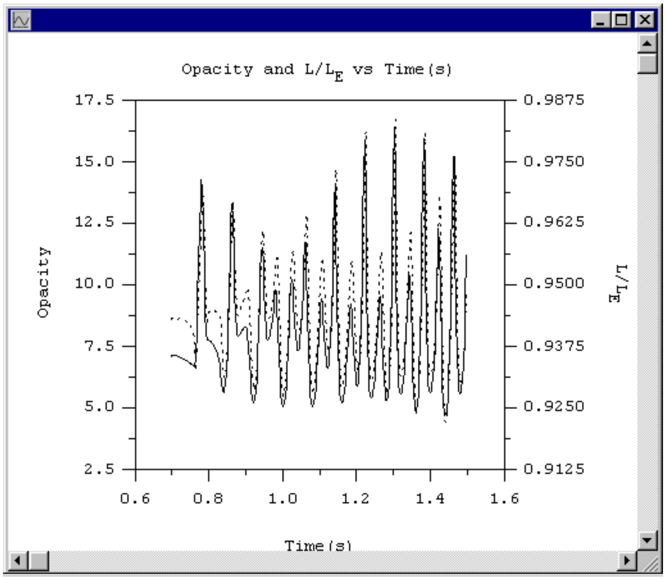


Figure 3-10: Double-Y Plot

Changing the Plot Appearance

To improve the appearance of the plot, double click on the X axis to open the Axis Labels dialog. Select a Plot Bounds and an Axis Labels range of **0.7** to **1.5**, an Increment of **0.1**, and a Label Format of **#####.#** (Windows) or **####.#** (Macintosh).

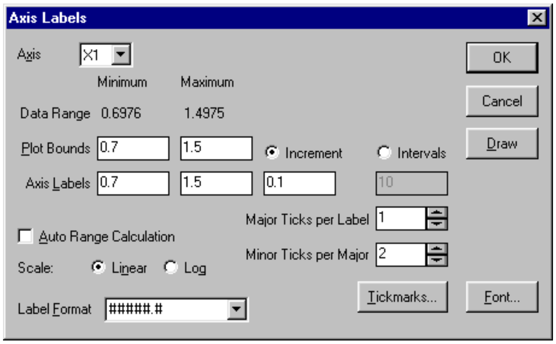


Figure 3-11: X Axis Labels Dialog

Next, click the **Tickmarks...** button to bring up the Tickmarks dialog. Click **In** to select inward directed tickmarks, and **Out** to deselect outward directed tickmarks. Click **OK** when done. In the Axis Labels dialog, choose **OK**.

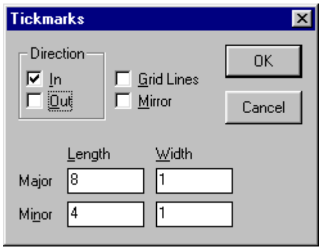


Figure 3-12: Tickmarks Dialog

Double-click the left Y axis to open the Axis Labels dialog. Enter a Plot Bounds and an Axis Labels range of **0** to **30**, an Increment of **5**, a Label Format of **#####** (Windows) or **####** (Macintosh), and **5** Minor Ticks per Major. Click on **Tickmarks...** to make these tickmarks point inward. Click **OK** in the Tickmarks dialog, and in the Axis Labels dialog.

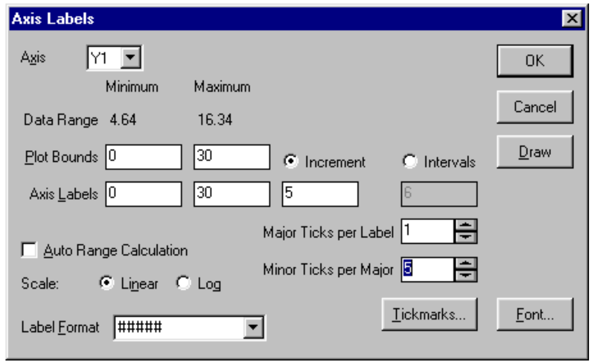


Figure 3-13: Left Y Axis Labels Dialog

Finally, double-click the right Y axis to open the Axis Labels dialog. Enter a Plot Bounds of **0.63** to **1.08**, an Axis Labels range of **0.7** to **1.3**, an Increment of **0.1**, a Label Format of **#####.##** (Windows) or **####.##** (Power Macintosh), and **2** Minor Ticks per Major. Also click **Tickmarks...** and make these tickmarks point inward. Click **OK** in the Tickmarks dialog and the Axis Labels dialog to return to the plot.

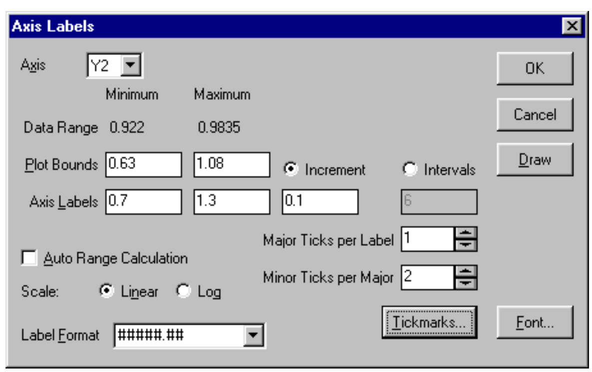


Figure 3-14: Right Y Axis Labels Dialog

In this case, we wanted the axis labels to start at 0.7, a nice round number, instead of 0.63. Therefore, our Axis Labels range was different from the Plot Bounds range.

Your resulting graph should look similar to the one in the figure below.

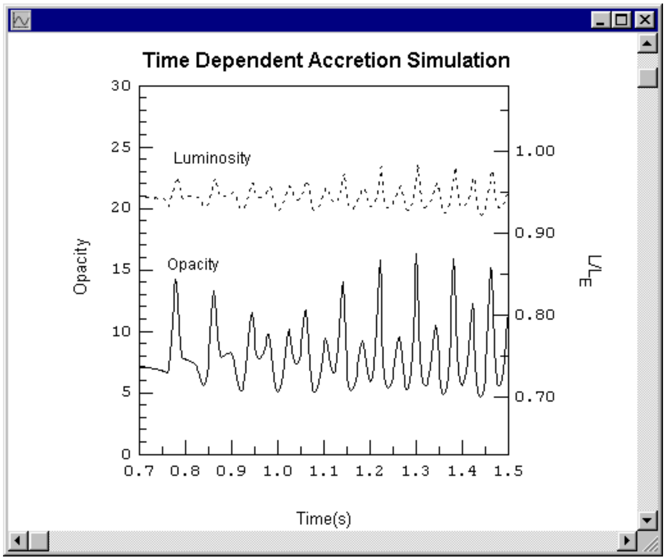


Figure 3-15: Data Window with Improved Double-Y Plot

In the figure above, we also changed the shape of the graph, changed all text to Arial, and added two new text labels (Luminosity and Opacity), using the **New Text...** command from the Graph menu. We also changed the title of the graph using the **Edit Text...** dialog. We will show how to add text labels next.

Adding Text Annotations

Plot provides two methods of formatting text. You can specify size, font, and style, from the menus within the Edit text dialog (Windows) or from the Format menu (Power Macintosh), or you can use Plot's text formatting language. In the next example, we will use both methods to illustrate their use.

The main elements of annotation formatting are the \ character, which signifies the beginning of a command, the _ and ^ characters, which signify subscripts and superscripts, and the { } characters, which signify groupings.

To see an example, in Windows create a new text annotation by selecting **Center** from the **New Text...** submenu in the Graph menu (in Power Macintosh, just select **New Text...** from the Graph menu). Click in the Edit Text dialog and type **rout=30R**.

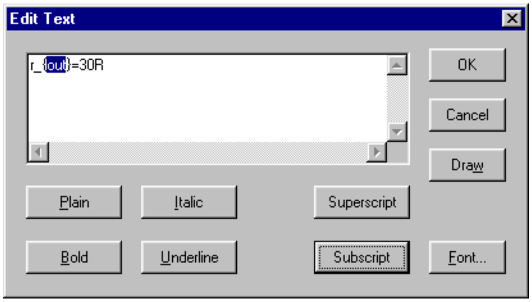


Figure 3-16: Edit Text Dialog with Selected Text

Next, highlight 'out' and click the **Subscript button** (in Power Macintosh, select **Subscript** from the Style submenu located on the Format menu). The text string now appears as r_{out}=30R.

Normally, only the first character after the _ character is subscripted. However, if the first character is an open bracket ({) then everything up until the closing bracket (}) is subscripted.

The ^ character is used to signify superscripts in the same way. To actually display either an _ or a ^ character, preface them with a \.

Click **OK** to see how the text is displayed. The formatted text string is shown below. Note that we have changed the font to Arial.

$$r_{out}=30R$$

Figure 3-17: Text String with Subscript

Now select the same text annotation again by double-clicking it. On the second line, enter **\epsilon =0.05, \mu_r=0.30**. The \epsilon and \mu commands are used to display the Greek epsilon and mu characters, respectively. The space and underscore characters that follow epsilon and mu are important for command termination. See Chapter 12 for more details on text editing.

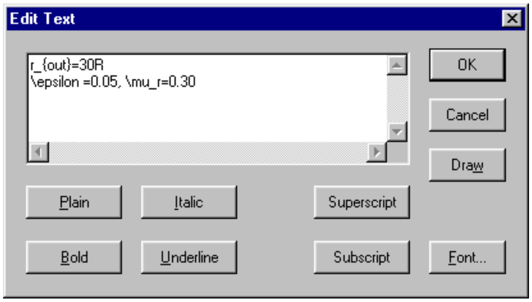


Figure 3-18: Complete Text Annotation

Click **OK** to display the text annotation and plot. The completed graph should now look similar to the figure below. You can move any of the text blocks by clicking on and dragging them.

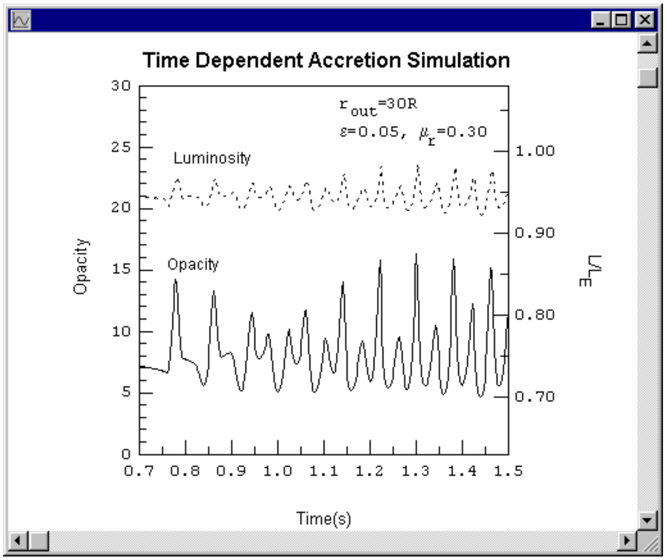


Figure 3-19: Completed Double-Y Plot

The formatting language can be very useful, particularly because it permits formatting information in your macros. Keep the plot window open, as we will use it and the related data window to create import and plot macros.

Creating and Using Macros

Plot's macro capabilities let you save, import, or plot parameters to automate the process of importing data and creating plots. Macros use exactly the same language that is used in the notebook and in the Expression field of the Column Settings dialog.

Creating an Import Macro

Recall that at the beginning of this chapter, we made several choices in the process of importing the sample data file ‘ls753.txt.’ Here we will save those choices and apply them to a new data file.

First, click on the data window to make sure it is the frontmost window. Then select **Create Macro...** from the Macros menu. In the dialog that appears, enter the name “Import Macro.”

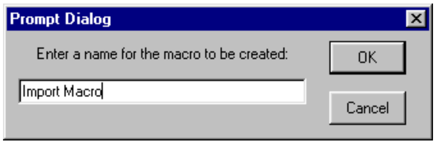


Figure 3-20: Specify Macro Name

Click **OK.** This creates a macro script that reads data files with the same parameters we chose interactively to import ‘ls753.txt.’

Note

Because periods and other non-alphanumeric cannot be used in macros, Plot replaces all such characters with underscores.

Creating a Plot Macro

Next, we will create a plot macro. Click on the plot window to make sure it is the frontmost window. Now, again, select **Create Macro...** from the Macros menu. Enter the name “Double Y Plot.”

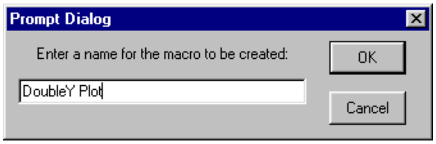


Figure 3-21: Saving a Plot Macro

Now, from the Macros menu, select the import macro we created (Import_Macro).

Figure 3-22: Selecting an Import Macro

You will be prompted to specify a data file. Select the file ‘ls746.txt’ from the Plot\Samples folder, then click **Open**. Plot will read this new file, using the parameters stored in the ‘Import_Macro’ macro. The resulting data window should look like the figure below.

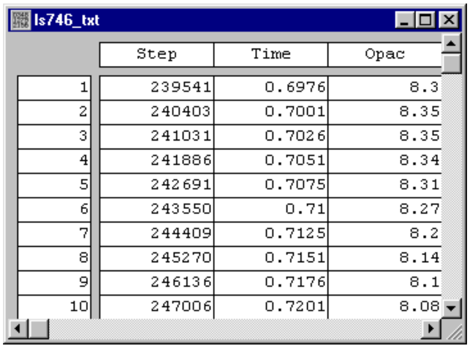


Figure 3-23: Results of Executing the Import Macro

Now, from the Macros menu, select the plot macro **Double Y_Plot.**

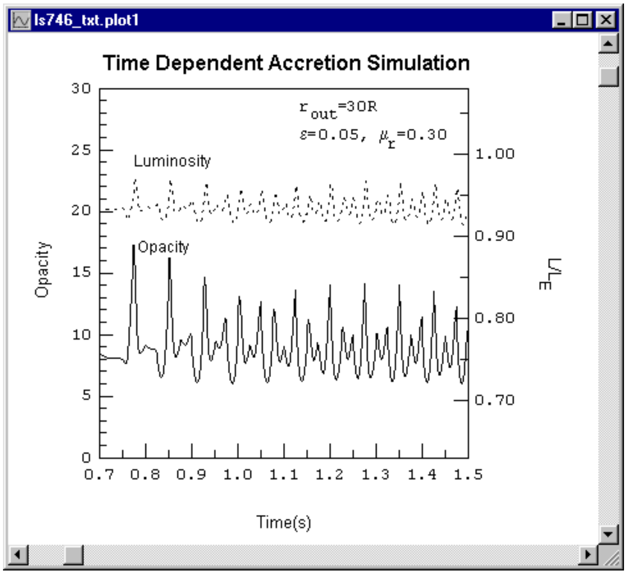


Figure 3-24: Results of Executing a Plot Macro

With minor exceptions, your plot should look like the figure shown above. As you can see, even if you do not write or edit macros directly, they can save you a great deal of time and effort.

Chapter 4:

Analytic Line Plots

In the previous two chapters, we showed you how to use Plot to graph data. This chapter describes how to use Plot to graph equations. Here you will plot sine, cosine, and parametric curves.

If you have not already done so, select **Close All** from the File menu to close all windows from the previous chapters.

Plotting Sine, Cosine Curves

Sine and cosine curves require index fields, so first we create those fields, then we create and plot the sine and cosine curves.

Creating Columns

Select **New** from the File menu to open a new data window. In Windows, select **See Notebook** from the Data menu to open the notebook for the new data window. In Power Macintosh, select **Untitled.notebook** from the Windows menu to open the notebook for the new data window.

In the notebook, type **t=series(200)** and, with the cursor on the same line, select **Calculate Now** from the Data menu.

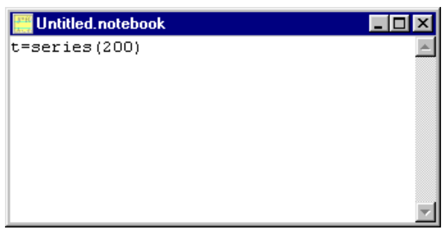


Figure 4-1: Notebook Expression for t

Next enter the lines **inc=6** then **print inc** in the notebook. Select both lines, then choose **Calculate Now** from the Data menu.

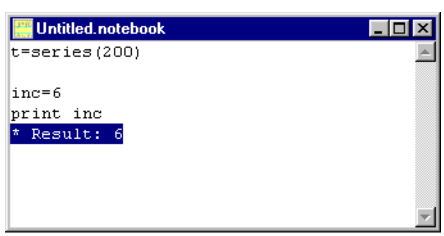


Figure 4-2: Scalar Variable Verified

The print command verifies that the scalar variable inc does contain 6. We will be using this variable in our equations.

Type **s=sin(t/inc)** and, again, select **Calculate Now**. Note how we use the inc variable to scale the sine curve. The function sin works with radians, where 2π is a full cycle. So with inc=6, the s column will go through 200/(6*2*π), or about 5, cycles.

Finally, type **c=cos(t/inc)** in the notebook, and select **Calculate Now**.

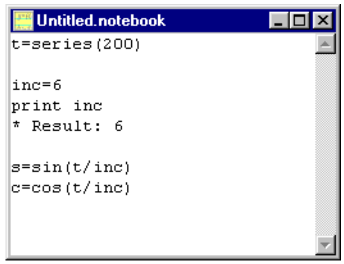
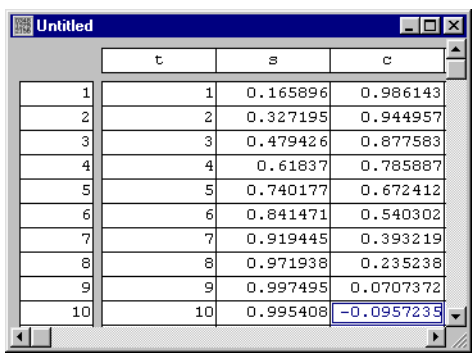


Figure 4-3: Notebook with Expressions for s and c Added

The end result (shown below) should be three columns: a t column with a series of index numbers, an s column with sine values, and a c column with cosine values.



	t	s	c
1	1	0.165896	0.986143
2	2	0.327195	0.944957
3	3	0.479426	0.877583
4	4	0.61837	0.785887
5	5	0.740177	0.672412
6	6	0.841471	0.540302
7	7	0.919445	0.393219
8	8	0.971938	0.235238
9	9	0.997495	0.0707372
10	10	0.995408	-0.0957235

Figure 4-4: Data Window with t, s, and c Columns

Plotting Columns

Now we can plot these curves. Select **Line** from the **Gallery** submenu on the Graph menu. Select **t** as the X column and both **s** and **c** as the Y columns.

To select both these columns, select the first one (here **s**), then hold down the shift key and select the last column (here **c**). Click **OK**.

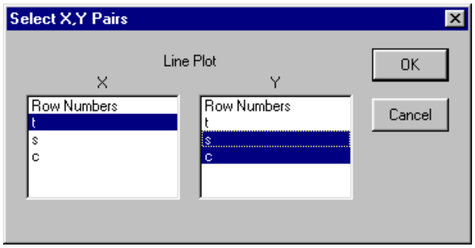


Figure 4-5: Select Data Columns

The resulting graph should look like the figure below.

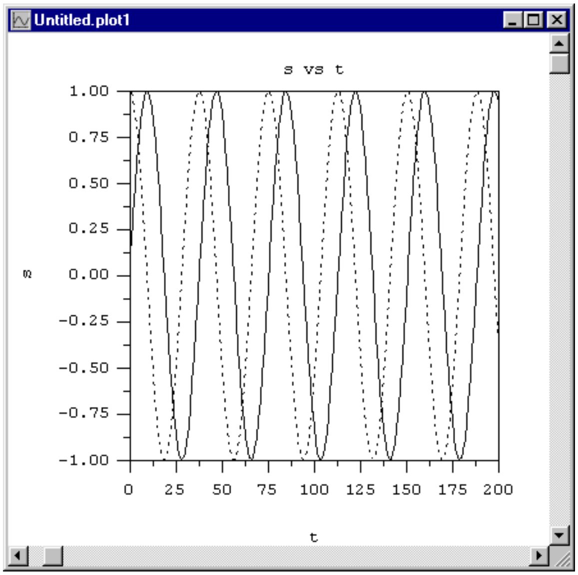


Figure 4-6: Plot Window with s,c Curves

The plot appearance can be improved by changing the size, font, axes labels, and some of the text. An improved plot is shown below.

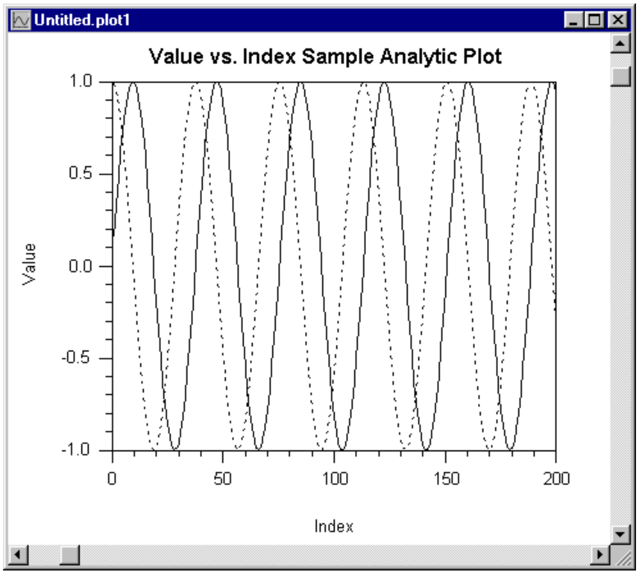


Figure 4-7: Improved s,c Plot

Note that we have changed the font for all text to Arial, modified the axes, changed the axes text, and changed the title.

Parametric Plots

Instead of plotting the sine and cosine curves against an index column, we could instead plot them against each other. This would produce a parametric plot. Unfortunately, plotting c versus s would produce a fairly boring parametric plot: a circle. To make it more interesting, we will create more columns.

Return to the notebook and enter **ts=t*s** and **tc=t*c**. Make sure both lines are selected then click on **Calculate Now**.

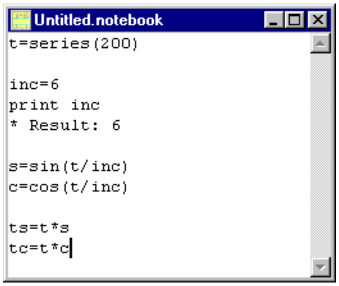


Figure 4-8: Notebook with New Expressions for New Data Columns

Select **Line** from the **Gallery** submenu on the Graph menu. Select **ts** as the X column and **tc** as the Y column.

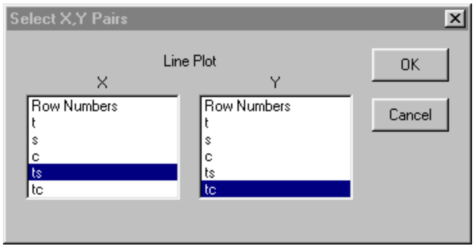


Figure 4-9: Select X,Y Pairs Dialog

Click **OK** in this dialog, and you should see the following parametric plot.

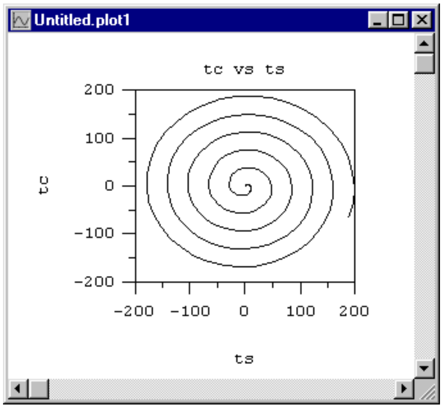


Figure 4-10: Parametric Plot

Adding Color

This plot can be improved. As before, you can modify the font, labels, and so on. You can also enhance the plot by using color. To do so, double-click anywhere within the plot boundaries to bring up the Edit Plot dialog.

In the Edit Plot dialog, click on the **Lines...** button and change the line width to **3** points (pt.). Return to the Edit Plot dialog then click on the **Colors...** button. The Colors dialog will appear.

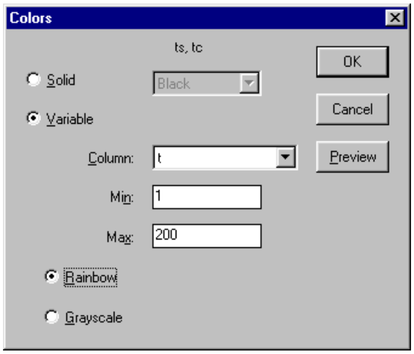


Figure 4-11: Colors Dialog

Click on the **Variable** radio button. This tells Plot to map colors to a data column.

Next, open the Column menu and select the **t** column. Finally, click on **Rainbow**. Click **OK** in the Color and Edit Plot dialogs. Your plot should now look much like Figure below, in color (we have also adjusted the font and labels).

Note that with **Rainbow** selected, Plot maps colors evenly from blue to red as values increase. Since we colored the parametric line by the t column, your actual plot should run from blue in the center to red in the end.

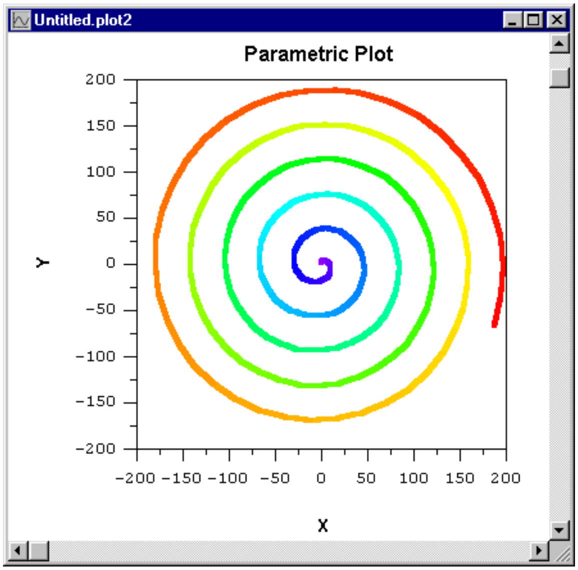


Figure 4-12: Parametric Plot with Color Added and Labels Adjusted

Updating Calculations

Note that everything we have done depends on the ‘inc’ variable that we defined at the beginning of this chapter. Let us see what happens when we change its value. Bring up the notebook again. On a new line, type **inc=12**. Then select **Calculate Now**.

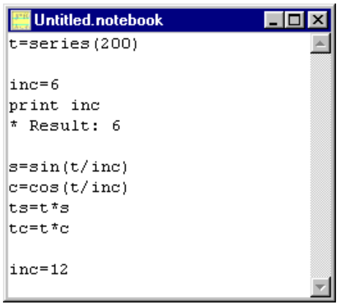


Figure 4-13: Notebook Window with the New 'inc' Calculation

Select the **s**, **c**, **ts**, and **tc** lines. Then select **Calculate Now** from the Data menu to update all the columns. This is necessary because the ‘inc’ variable that the column equations depend upon has changed.

If you looked closely, you may have noticed the column values in the data window change. Now bring the parametric plot window to the front. It should replot using the new column values.

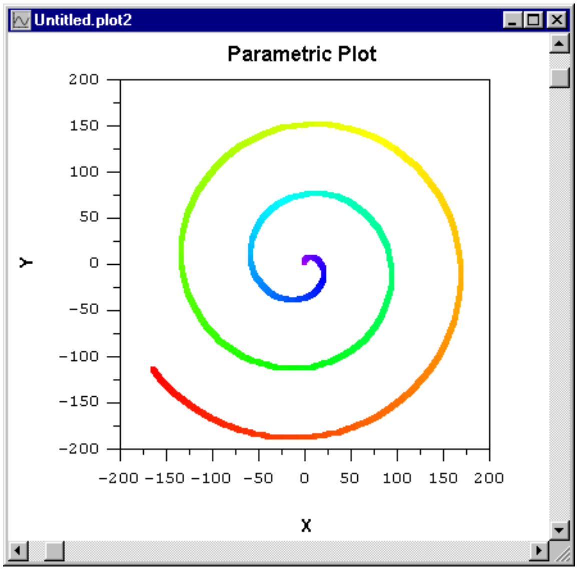


Figure 4-14: New Parametric Plot

By doubling the 'inc' variable, we halved the number of cycles that the sine and cosine curves go through. So instead of five cycles, you see about two-and-one-half cycles.

Chapter 5:

Color Scatter Plots

In the previous chapters, we plotted one variable (Y) against another (X). In this chapter, we will plot a value against both X and Y to produce a color scatter plot.

If you have not already done so, close all the Plot windows from the previous chapters.

Reading a Text File

Choose **Open...** from the File menu and select the file ‘weather.txt’ from the ‘Samples’ directory. This data is from the U.S. Weather Service and consists of temperature measurements from January 2, 1991.

In the dialog that appears, select Text Columns as the import file type. You will now see the Text Columns dialog shown in Figure 5-1.

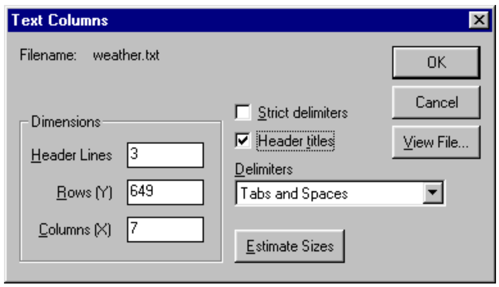


Figure 5-1: Text Columns Dialog

Make sure that the 'Header titles'/‘Last header line contains column titles’ check box is selected before clicking **OK**. You may also want to click **View File...** to see the import file before continuing.

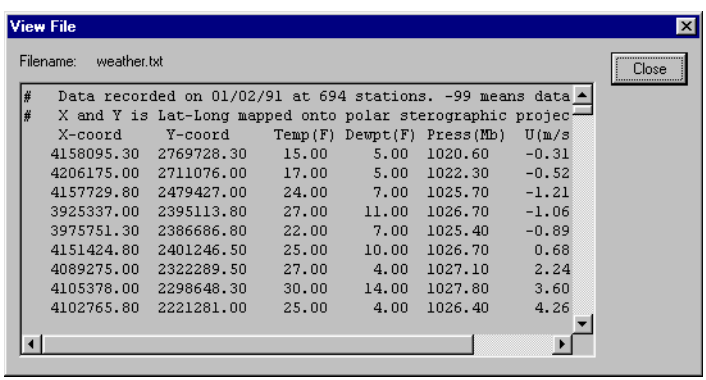
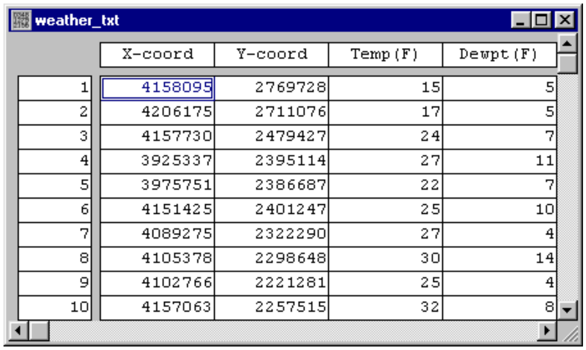


Figure 5-2: View File Dialog

The resulting data window should look similar to the next figure.



weather_txt

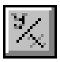
	X-coord	Y-coord	Temp(F)	Dewpt(F)
1	4158095	2769728	15	5
2	4206175	2711076	17	5
3	4157730	2479427	24	7
4	3925337	2395114	27	11
5	3975751	2386687	22	7
6	4151425	2401247	25	10
7	4089275	2322290	27	4
8	4105378	2298648	30	14
9	4102766	2221281	25	4
10	4157063	2257515	32	8

Figure 5-3: Data Window for weather.txt

The first two columns consist of map locations for the measurements. The rest of the columns are measured values for temperature, dewpoint, pressure, and wind speed. In this chapter we will be working with only the temperature column, Temp(F) (Windows) or Temp_F_ (Power Macintosh).

Creating a Scatter Plot

Select **Scatter** from the **Gallery** submenu on the Graph menu. Select **X-coord** as the X coordinate and **Y-coord** as the Y coordinate, then click **OK**.



Note how the data values outline the shape of the United States, but the shape is a bit compressed. In Windows, click the **Aspect Ratio** button to make the plot look more realistic

Note

In order for Aspect Ratio button to appear at top of the screen, you must select **Tool Bar** from the Edit menu.

On Power Macintosh, drag the image wider until it looks more natural.

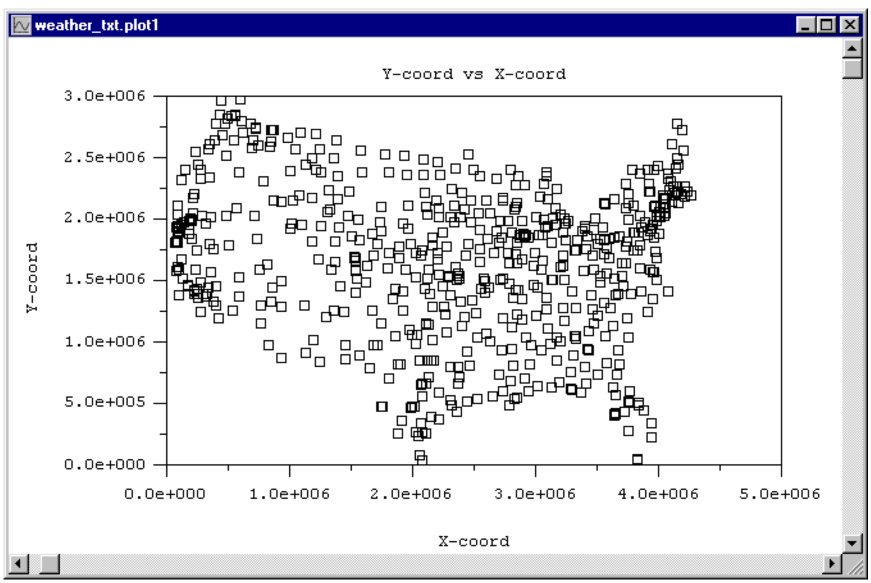


Figure 5-4: Scatter Plot for weather.txt

In addition, modify the axes labels to improve the image. As shown in the image below, we changed the number display in the Axis Labels dialog from exponential to an integer, truncated the right side of the plot at 4,500,000, the top at 3,000,000 and made reasonable increment and tickmark choices.

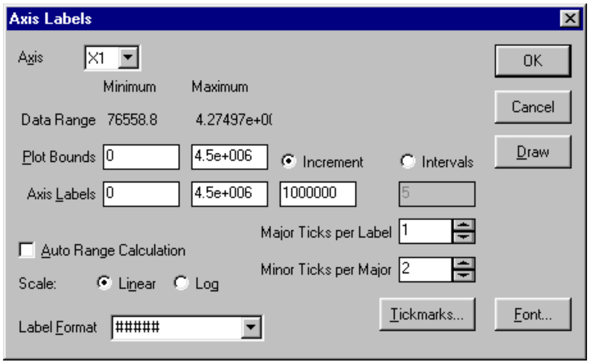


Figure 5-5: X Axis Labels Dialog

The improved scatter plot should look similar to the figure below.

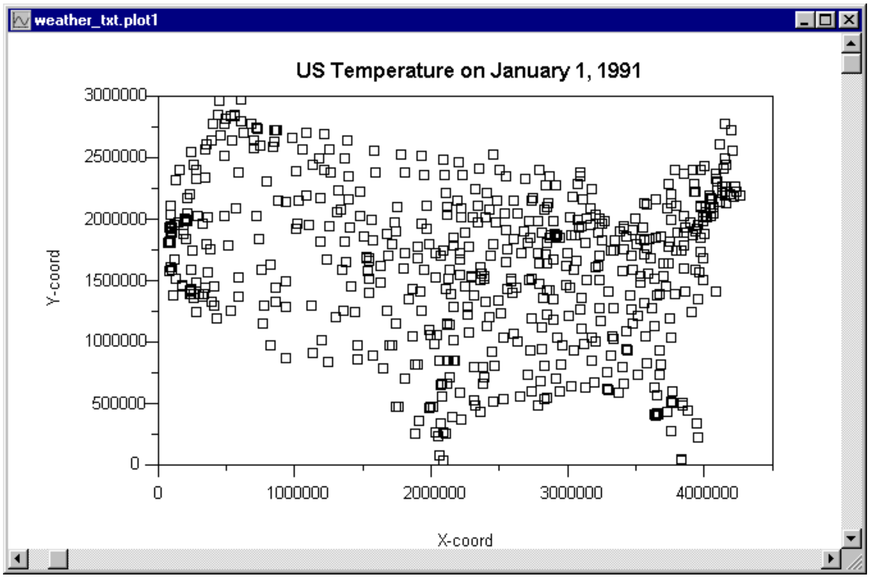


Figure 5-6: Improved Scatter Plot for weather.txt

Adding Color to the Scatter Plot

Now we will add color to these symbols. Double-click on the plot to open the Edit Plot dialog. Click on **Symbols...** to open the Symbols dialog. Open the Draw list and select **Filled Circle** (on Macintosh, this is the last symbol on the right).

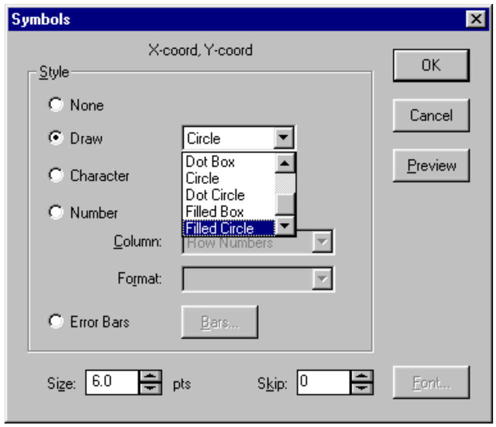


Figure 5-7: Symbols Dialog

Click **OK** to return to the Edit Plot dialog. Now click on **Colors...** to open the Colors dialog. Choose **Variable**. Then open the Column list and select **Temp(F)**. Set the Min/Max range from **-15** to **76**. Also, select **Rainbow** as the color mapping.

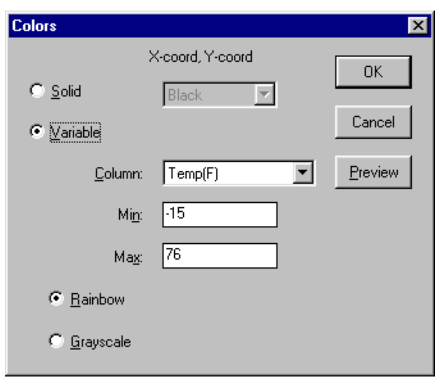


Figure 5-8: Color Dialog

Click **OK** on this dialog and on the Edit Plot dialog. The resulting color plot should look something like the figure below.

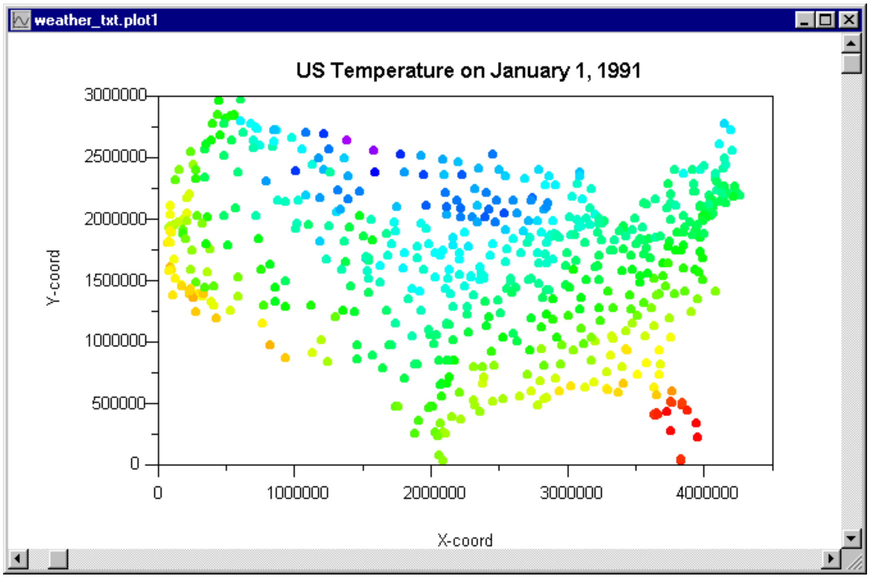


Figure 5-9: Weather Color Scatter Plot

Note that this color plot is different from the color plots in the previous chapters. Here, there is one data value, Temp(F), shown as a function of two columns (X-coord, Y-coord).

Creating a Number Scatter Plot

Instead of displaying the temperature readings as colors, we can display the actual readings.

To do so, open the Edit Plot dialog again. Click **Symbols...** to open the Symbols dialog. Select **Number** as the symbol style. Open the Column list and select the **Temp(F)** column. Open the Format list and select the ##### (Windows) or #### (Power Macintosh) number format.

In Windows, click the **Font** button and select the Arial font with a size of 8 points; close the Font dialog (in Power Macintosh, enter **8** in the Size box near the bottom of the dialog). Then, enter **6** in the Skip box.

The Skip value tells Plot to put a symbol at every sixth X-Y location. Skipping values creates space between numbers, making them more legible.

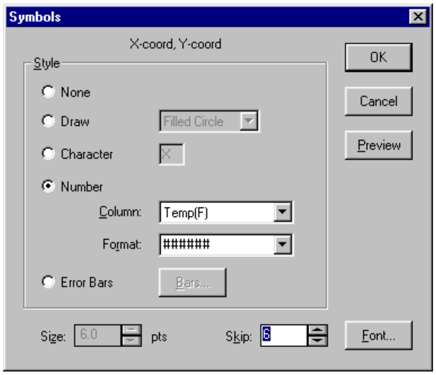


Figure 5-10: Symbols Dialog for Number Scatter Plot

Choose **OK** to close the Symbols dialog. In the Edit Plot dialog, click on **Colors...** and set the color to **solid black**. Click on **OK** to return to the plot. Your plot should now look like the one shown below.

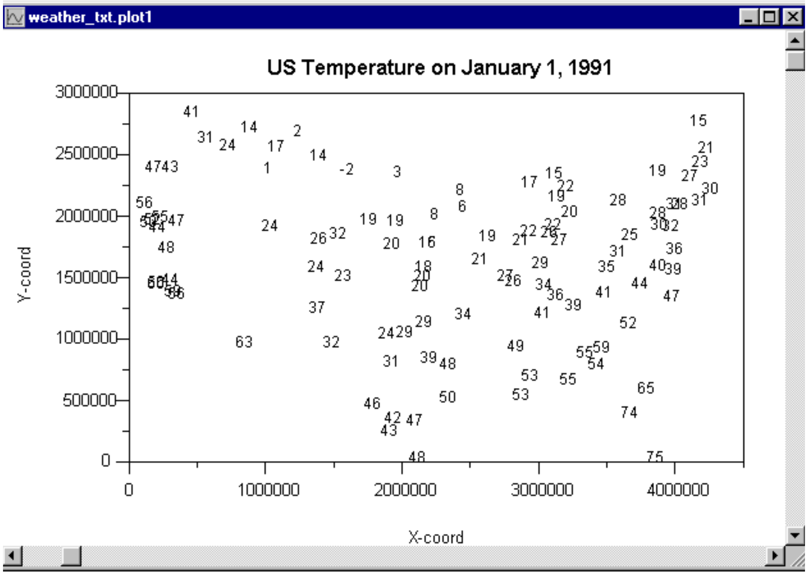


Figure 5-11: Finished Number Scatter Plot

Synchronizing Plot and Data

You can easily compare points on a plot with their values in the data window. First, be sure that the plot window is active. Select **Synchronize** from the Graph menu (Windows) or Edit menu (Power Macintosh).

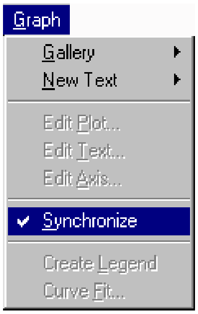


Figure 5-12: Synchronize Command

Now, click on any point along the graph. Note how the corresponding row of data values is selected in the data window.

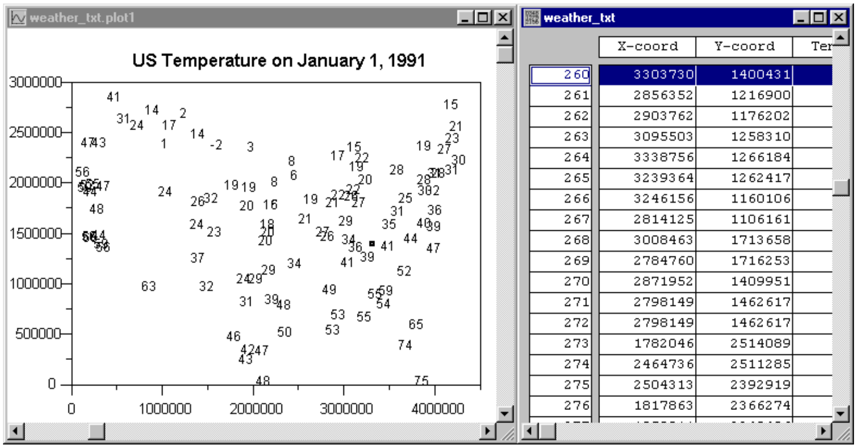


Figure 5-13: Corresponding Points on the Plot and in the Data Window Selected using Synchronize

This works the other way also. You can click on a value in the data window and see it selected in the plot window.

Part III: Reference Manual

Chapter 6:

The Data Window

A blank data window appears each time you launch Plot. You can also create a new data window by selecting **New** from the File menu. A new data window is shown in Figure 6-1.

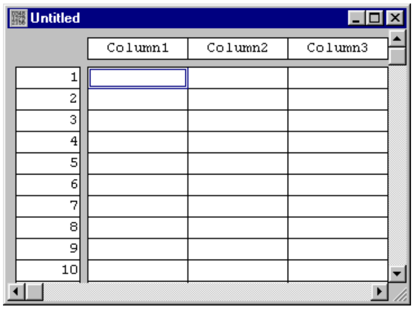


Figure 6-1: An Empty Data Window

This chapter describes how to manipulate data and the data window's settings.

Entering Data Manually

When a new data window appears, the first cell of column one is automatically highlighted. To start entering data, simply type a value for this cell. Note that the entry also appears in the edit box at the top of the screen.

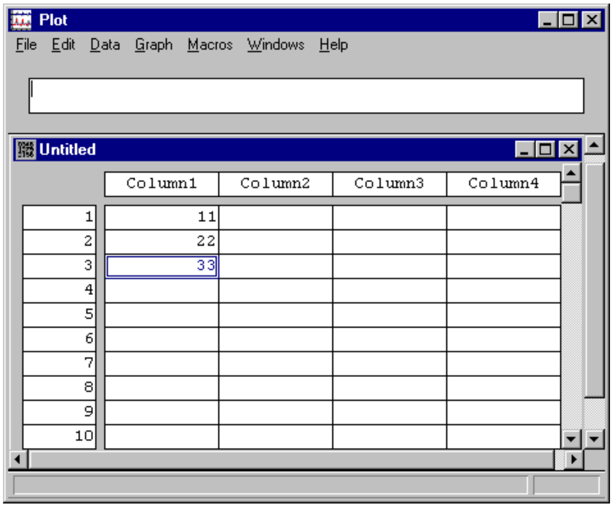


Figure 6-2: Entering Data

Enter Data by Column

To enter the value in the cell and move down to the next cell in the column, press Enter. The value appears in the cell, and the next cell down is automatically highlighted, ready for the next entry.

Enter Data by Row

To enter a value in the cell and move right to the next cell in the row, press the Tab key. The value will be entered in the cell, and the next cell to the right is highlighted, ready for the next entry.

Moving Around in the Data Window

Besides using the mouse, you can also use the **GoTo...** command from the Data menu or the keyboard to move around in the data window.

GoTo... Command

To move directly to a specific cell, select **GoTo...** from the Data menu. The Go To Cell dialog appears.

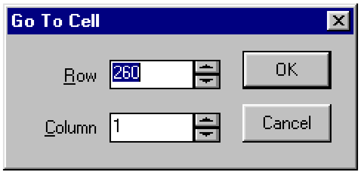


Figure 6-3: Using the Go To Cell Dialog

Enter the row and column number for the destination cell, and click **OK**. That cell and its contents will be displayed and highlighted.

Keyboard (Windows)

Keyboard keys generally work the same way in Plot as any other Windows application.

Tab, the arrow keys, Home, End, Page Up, and Page Down are used for moving, selecting, and scrolling. Enter is used to confirm an action.

Shift is used to extend the selection or perform the opposite action. For example, Tab moves right; Shift+Tab moves left.

Ctrl is used to perform the same action on a different scale. For example, Page Up move up one window; Ctrl+Page Up moves to the top row.

Keyboard (Power Macintosh)

To Move...	Press
Down one cell	Return, Enter, or the down arrow
Up one cell	Shift-Return or up arrow
Right one cell	Tab or right arrow
Left one cell	Shift-Tab or left arrow

Table 6-1: Power Macintosh Navigation Keys

Selecting Data

The data window provides a variety of methods for selecting data.

Select a Cell

Click on a cell to select it, or use the keyboard or **GoTo...** command described above. The contents of the selected cell are displayed in the edit box at the top of the screen.

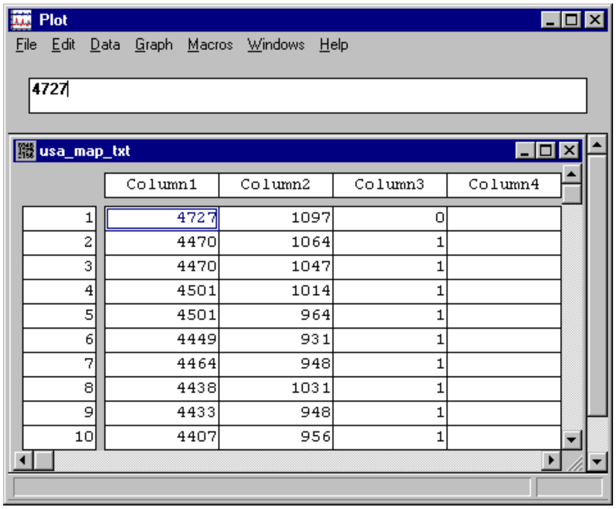


Figure 6-4: Selecting a Cell

Select a Region of Data

To select a region of data points in a window, click on one corner of the desired region, drag to the opposite corner, and release the mouse button. The contents of the upper left cell of the selected region is displayed in the edit box.

To extend a selection of data, hold down the Shift key while selecting the desired data by using the mouse or arrow keys.

Select Entire Row/Column

To select an entire column, click on the column label box for that column (Figure 6-5). To select an entire row, click on the row number box for that row (Figure 6-6). You can also use the arrow keys to select a column label or row number box.

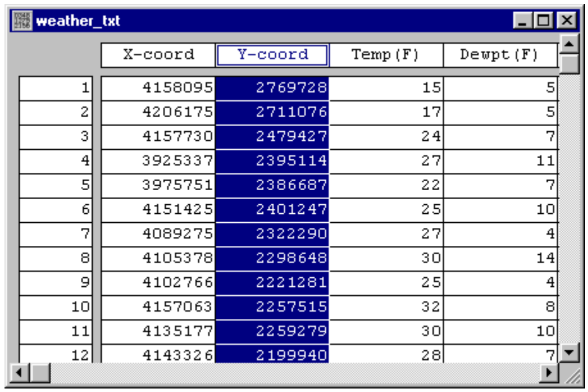


Figure 6-5: Selecting an Entire Column

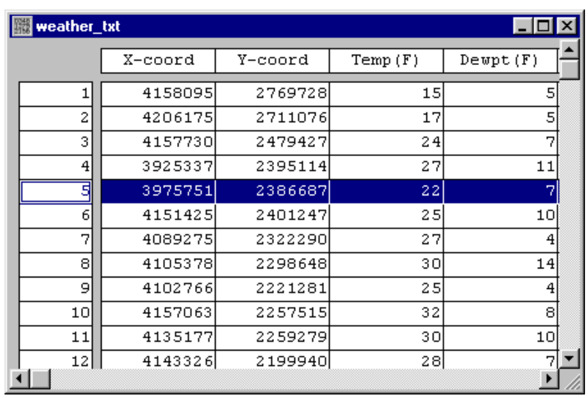


Figure 6-6: Selecting an Entire Row

Select Contents of Entire Window

To select all the rows and columns of the data window (including empty rows/columns), choose **Select All** from the Edit menu.

Insert Columns and Rows

The **Insert** command under the Data menu allows you to insert new columns or rows into the data window.

To make an insertion, select a column or row to indicate where the new column(s) or row(s) are to be inserted. In Windows, you can select multiple rows or columns to insert an equal number of new columns or rows. In Power Macintosh, you can insert one column at a time or multiple rows. Then select **Insert** from the Data menu.

Plot inserts the new columns, and the originally selected columns (as well as all columns to the right) shift to the right. Inserted rows shift all rows downward below the insertion point.

Delete Columns and Rows

The **Delete** command under the Data menu removes a selected column or row from the window.

To delete a column or row, click the column title or row number to highlight the entire column or row. (You may also select multiple rows or columns for deletion.) Then select **Delete** from the Data menu.

Columns to the right of the deleted column shift left to fill deleted columns. Rows below the deleted row shift upward.

Copying and Pasting Data

The **Copy** and **Paste** commands, under the Edit menu, provide a useful way to manipulate data within Plot, as well as a means of importing and exporting data.

Copy

The **Copy** command copies the current selection region from the data window. The selected numbers are taken as displayed in the selection region and copied as lines of numbers separated by tab characters. Copied data can be pasted elsewhere in the data window, in other data windows or notebooks, or into other applications, such as spreadsheet programs.

Paste

Text fields delimited by spaces or tabs may be brought into a data window using the **Paste** command. To paste data, select a cell as an insertion point, then select **Paste** from the Edit menu. Numbers are pasted into the data window starting at the insertion point, filling down and to the right.

Editing Data Window Columns

This section describes how to enter and change column names and editing columns.

Entering and Changing Column Names

To enter or change a column name, click on it, then enter the new name in the edit box as shown in Figure 6-7. You may also double-click on the column name to open the Edit Columns dialog, described below.

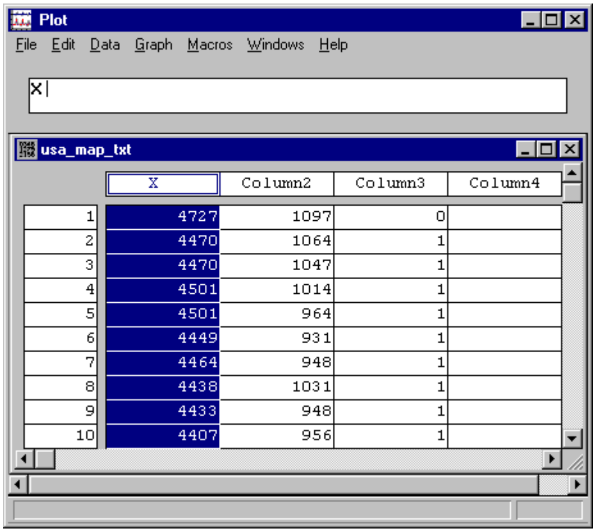


Figure 6-7: Entering New Column Name

Editing Columns

To bring up the Column Settings dialog, either double-click on the column name or select **Column Settings...** from the Data menu with the column selected. The dialog shown in Figure 6-8 appears.

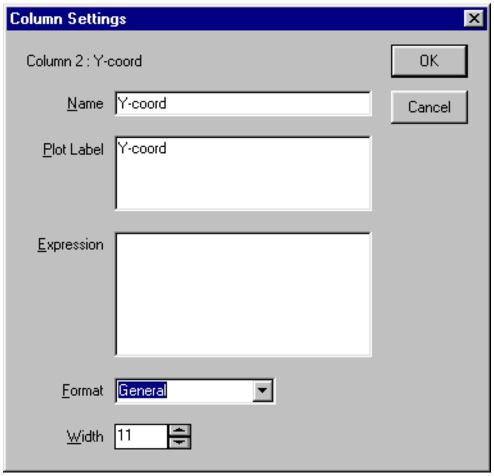


Figure 6-8: Column Settings Dialog

Column Number

The number of the column selected appears in the upper left corner.

Name

Displays the current name of the selected column and allows you to change it.

Plot Label

By default, column headings are used for plot axes labels. This box allows you to specify a more descriptive plot label with text formatting, such as super- and subscript. For more on text formatting, see Chapter 12.

Expression

You can calculate values for the selected column by entering an expression in this box. Plot executes expressions in this field and returns a column of numbers based on the entered function or expression. For more on calculations, see Chapter 14.

Format

This box lets you specify the numerical format for data displayed in the selected column. You can click on the arrow to the right of this box for a list of formats. To have Plot select a format automatically, select **General**.

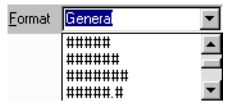


Figure 6-9: Numerical Format Pull-down Menu

You can also enter custom formats in the box. The pound sign (#) indicates the number of digits to be displayed to the right of the decimal point.

Width/Characters Wide

This box lets you set the column width in terms of characters. Note that you can also set column width interactively, as shown in Figure 6-10.

Note

If a number is wider than the column, the hidden portion of the number is represented by an ellipsis. To display the full number, widen the column or use a different number format.

Set Column Width Interactively

You can change a column's width using the Column Settings dialog, as described above, or you can change it interactively. To do so, move the cursor over the vertical rule. When the cursor changes, click and drag the rule to the desired width.

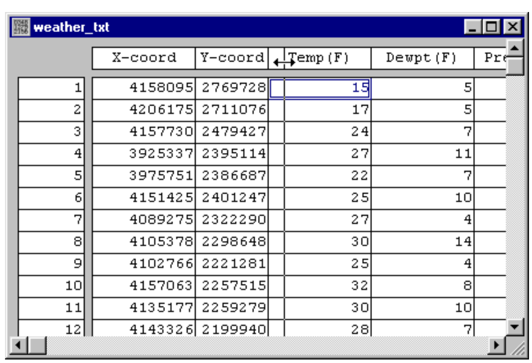


Figure 6-10: Changing Column Width Interactively

Data Specifications

Plot is designed to work with columns of IEEE floating point numbers. For each column, Plot saves:

- Name
- Plot label
- Printing format
- Column width
- Formula
- Column of 32-bit floating numbers

For more information, see the description of the Column Settings dialog, under Editing Columns, earlier in this chapter.

Number Ranges

The floating point values stored by Plot are 4-byte, 32-bit IEEE standard floating point numbers. 24 bits are used for the mantissa.

Significant/Mantissa precision:	7 to 8 digits
Maximum positive number:	3.4E+38
Minimum positive number:	1.2E-38
Maximum negative number:	-1.2E-38
Minimum negative number:	-3.4E+38
Infinity:	INF
Not a Number:	NaN (for example, log of a negative number)

Using the Notebook Window

The notebook window in Plot provides a place to add comments about your data. It also allows you to enter and execute expressions and macro commands, which are described in Chapters 14 and 15. The figure below shows a notebook window containing both comments and expressions for calculations to be performed.

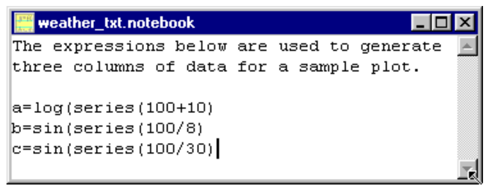


Figure 6-11: Notebook Window

A notebook window is automatically created for each data window and takes the name of the data window. To display the notebook in Windows, select **See Notebook** from the Data menu.To display the notebook in Power Macintosh, select **untitled.notebook** from the Windows menu.

Entries in the notebook window are automatically saved when you save the data window. When you subsequently reload the dataset, the notebook window will be loaded and available under the Data menu (Windows) or Windows menu (Power Macintosh).

The standard Windows and Macintosh text editing functions, such as selecting, copying, deleting, and pasting text, all work in the notebook window. In Windows, the text font for the window can be changed by selecting **Font** from the Edit menu; in Power Macintosh these commands can be found under the Format menu. Only one font and size can be used in the notebook.

Changing the Font in the Data Window

In Windows, you can change the font used in a dataset window by selecting **Font...** from the Edit menu. In Power Macintosh, the font, size and style can be modified using commands available from the Format menu.

Using a monospaced font, such as Courier New, is recommended so that the data values are aligned. The same font must be used for the entire data window.

Chapter 7:

Opening and Importing Files

To open or import a data file, select the **Open...** command under the File menu. By default, Plot displays all file types. To limit displayed files to a specific file type, use the pull-down menu and select the desired type.

Importing Text Column Data

Plot imports ASCII files of numbers delimited by spaces, tabs, or most non-numeric punctuation. Text fields, such as "Jan," "Feb," or "Mar," are accepted as column titles but not as data. Note that when text fields appear in column data, they are ignored, but may cause inaccurate column counts.

After using the **Open...** command to select the text file you wish to import, click on **OK** (Windows) or the **Open** button (Power Macintosh), or double-click on the file name. The Import File Format dialog shown in Figure 7-1 appears.

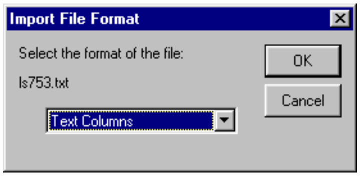


Figure 7-1: Import File Format Dialog (Windows and Power Macintosh)

Make sure that **Text Columns** is selected and click **OK**. The dialog shown in Figure 7-2 appears. The options on this dialog are described below.

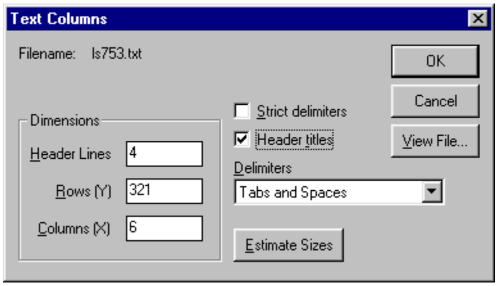


Figure 7-2: Text Columns Dialog

Delimiters

Select the desired text delimiter by selecting from the Delimiters list, shown in Figure 7-3.

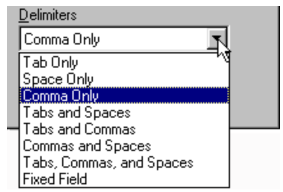


Figure 7-3: Text Columns Delimiters

Importing Fixed Field Column Files

Selecting **Fixed Fields** from the Delimiters pull-down menu changes the **View File...** button to **Set Columns....** Click this button to specify your fixed field columns in the dialog shown below.

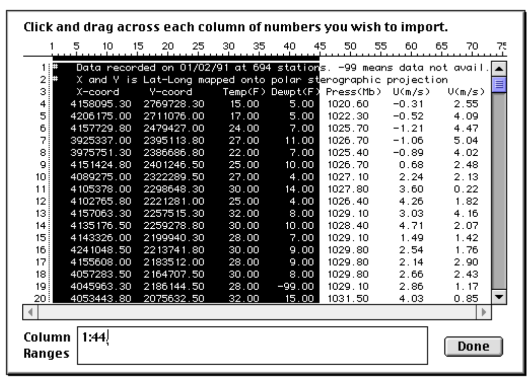


Figure 7-4: Fixed Columns Dialog

Click and drag anywhere in the data portion of the window to select each column of data to import. The listing at the bottom shows you the exact character positions of the columns you select. You may also enter column specifications directly in this listing. Use the form `first:last`, where `first, last` are the first and last character positions for that column. The column specifications are separated by commas. If you overlap a new column onto an existing one, the earlier one is replaced with the new entry.

Strict delimiters/Separate column for each delimiter

Select this check box if you have some data columns with missing data, indicated only by two consecutive tabs, for example. If you have a data value in each column, leave the box unchecked.

Header titles/Last header line contains column titles

Select this check box if your text columns have headings and those headings are separated by the same delimiters as the data.

Header Lines

Enter the number of lines of text for Plot to skip before import.

Estimate Sizes

Click here to have Plot estimate row and column sizes, given your choice for delimiters. Plot assumes that all of the data values for a row are on the same line. If this is not the case, override the incorrect estimates.

Rows (Y)/Number of rows
Columns (X)/Number of columns

Here enter the number of rows and columns of data to read from the file.

View File...

Click this button to view the data before you import it, as shown in Figure 7-7.

Importing Binary Column Files

Selecting **Binary Column** from the Import File Format dialog brings up the dialog shown in Figure 7-5. The options in this dialog are described below.

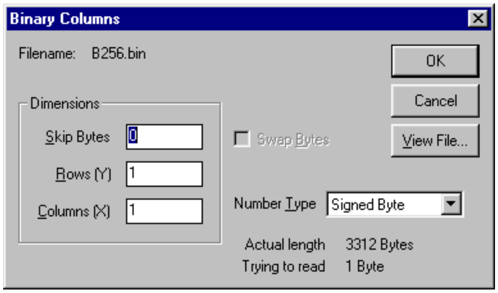


Figure 7-5: Binary Columns Dialog

Number Type

Open the Number Type menu and select the number type that represents your data.

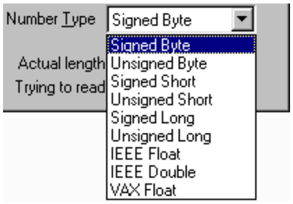


Figure 7-6: Number Type Menu

Swap Bytes

Select the Swap Bytes check box if your data is from a computer where binary data is stored in a different order.

Skip Bytes

Enter the number of bytes for Plot to skip before reading data.

Rows (Y)/Number of rows
Columns (X)/Number of columns

Enter the number of rows and columns of data to read from the file.

View File...

Click this button to view the data in hexadecimal before you import, as shown in Figure 7-7.

Actual length (Windows)

Size of file in bytes.

Trying to read (Windows)

Number of bytes that Plot will read, based on the current settings.

Importing Non-Column Data

Plot can import a variety of matrix and image file types, placing the data in the data window in column format.

Opening Transform Files (Windows)

Plot displays Transform matrix data as columns in the Plot data window. When an image is opened, Plot imports and displays the color mapping values for the image as columns of numbers.

Plot can also read files saved as ASCII Special from Transform, although the scale values will not be applicable. See the *Transform User's Guide and Reference* manual for information about ASCII Special.

Importing Text and Binary Matrix Data

To import text or binary matrix data, follow the same steps described above for text and binary columns. The data will be displayed as columns and rows in the Plot data window.

Two-Dimensional HDF Datasets (Windows)

Hierarchical Data Format (HDF) two-dimensional scientific datasets are opened and displayed as columns in the Plot data window.

HDF and TIFF Image Files (Windows)

Imported 8-bit image files are read as two-dimensional arrays of 8-bit data values, converted to floating point values, and displayed in the data window as columns and rows.

HDF Vset Files

HDF Vset is the normal storage format for data saved in Plot. The first Vgroup is read and each Vdata record is loaded as one column of data. See Appendix C for more on the HDF Vset standard.

FITS Files (Windows)

Plot can open FITS files containing two-dimensional or three-dimensional arrays of 8-bit unsigned, and 16-bit, and 32-bit signed integer values, as well as 32-bit and 64-bit floating point data. The data is scaled according to the BZERO and BSCALE keywords before import. The FITS ASCII header appears in the notebook window after importing.

GIF Files (Windows)

Graphics Interchange Format (GIF) files are read as an array of 8-bit data values, which are then displayed as columns of numbers. Plot supports 2-, 4-, 16-, 32-, 64-, and 256-color GIF plots, in '87a' or '89a' format.

PBM Files (Windows)

Portable Bitmap (PBM) files are read as 8-bit data values, which are then used to create a dataset of columns. 24-bit PBM plots are approximated as 8-bit images before being converted to data. PBM, PNM, PGM, and PPM file formats are supported.

Using View File...

In most import dialogs, Plot allows you to view the contents of your data file before importing the file. The View File dialog lists all the data in a file with row numbers. Plot for Power Macintosh also lists column numbers.

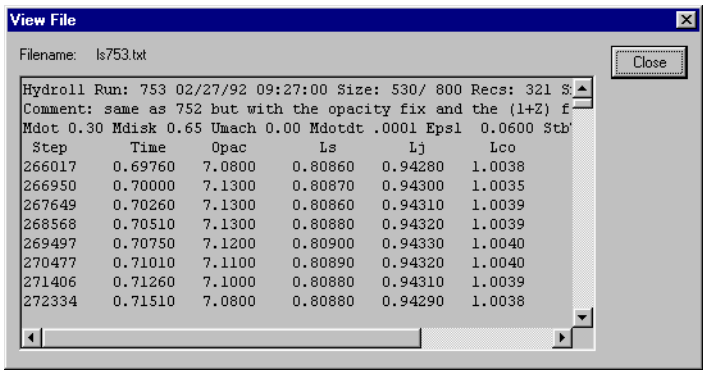


Figure 7-7: View File Dialog, Text Form

Figure 7-7 shows the text form of the View file dialog; Figure 7-8 shows the binary form of this dialog. Note that if the text form displays data that you cannot read, you may simply be trying to import binary data as text data.

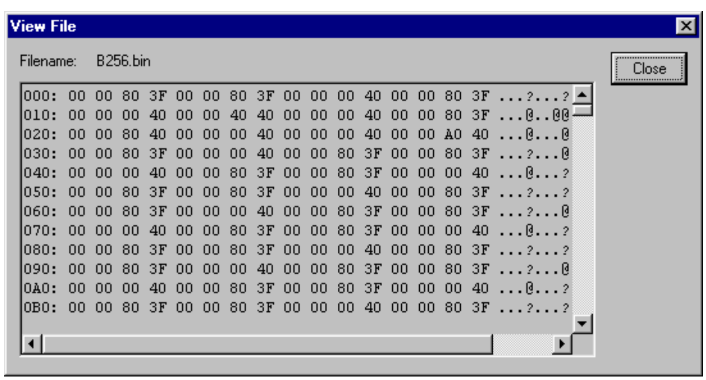


Figure 7-8: View File Dialog, Binary Form

Importing Multiple-Record Files (Windows)

Some Transform, Plot, MATLAB and HDF files contain multiple records. For these files, the Multiple Record dialog shown below appears. For HDF files, you see a list of HDF records and sizes, by name. For MATLAB files, you see a list of numbered records and their sizes.

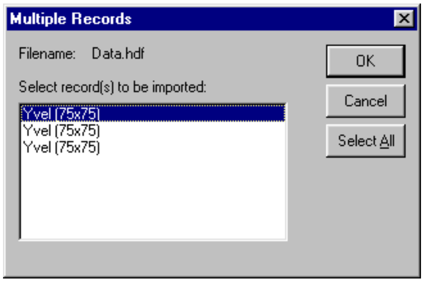


Figure 7-9: Choosing Record(s)

In the scrolling list, select any or all of the records listed. Use the **Select All** button to choose them all. Click **OK** to proceed. Plot imports each of the selected records into a separate data window.

Chapter 8:

The Plot Window

Each plot window is built on a similar foundation, so that fundamental features and operations are the same, regardless of plot type.

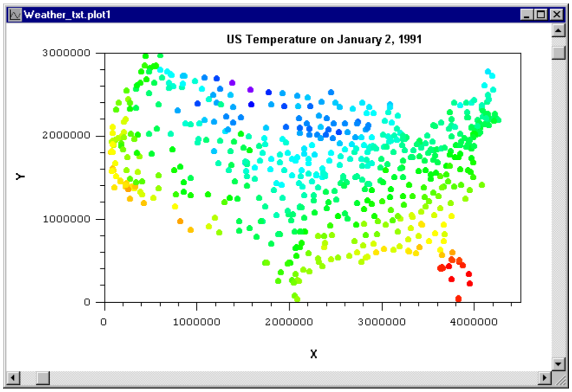


Figure 8-1: The Plot Window

The window name in the title bar displays the name of the dataset that was used to create the plot, followed by the plot number, which indicates when, in a sequence of plots created from this dataset, this plot was created.

Under the plot is the canvas—the imaginary surface the plot is drawn on, corresponding to the blank paper used for printing a plot. Scroll bars control the area of the canvas to be displayed.

In Windows, the right and bottom borders of the printable page are also indicated by rules representing the page borders.

In Power Macintosh, the mouse pointer is the only cursor tool used in Plot. The mouse pointer allows you to resize a plot, to select a plot or plot component for editing it, in conjunction with the **Synchronize** command, to select a point of interest on the plot to highlight corresponding values in the data window.

Resizing Plots

Preference Settings Dialog (Windows)

In Windows, the initial size of new plots is set in the Preferences Settings dialog. You can resize a plot by specifying an exact size or interactively by using the mouse.

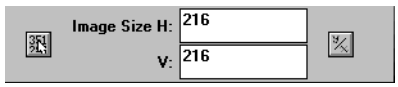
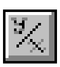


Figure 8-2: Toolbar with Resize Tool Activated

If the Tool Bar does not appear when a plot window is active, select **Tool Bar** from the Edit menu.

Using the Resize Tool (Windows)

To draw a plot at a specific size, type the height and width (in points) in the Image Size fields and press Enter.



To adjust the image size so that the aspect ratio of the image matches that of the dataset, click the **Aspect Ratio** button. The aspect ratio is calculated from the scale ranges of the dataset.

Resizing Plots Interactively

To resize a plot, position the pointer within the rectangular boundaries of the plot, and click once. The boundaries of the plot will be highlighted, and in Windows a solid box will appear in each corner and on each side of the plot (in Power Macintosh a solid box appears in the lower right corner).

To resize a plot in two dimensions at once, click on a corner box, and drag to expand or shrink the plot to the desired size (Figure 8-3), and release.

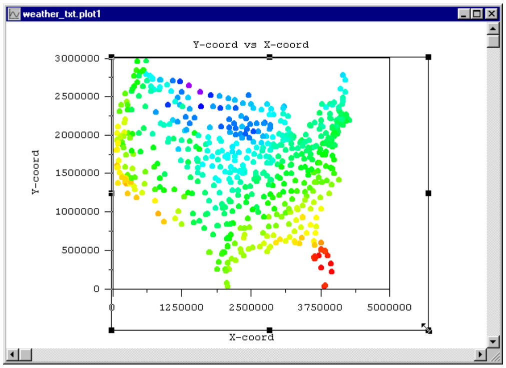


Figure 8-3: Resizing a Plot in Two Dimensions

In Windows, to resize a plot in only one dimension, click on a box along the plot's side, drag to the desired size (Figure 8-4), and release. To resize a plot using the keyboard, press Tab until the plot is selected. Then, pressing Ctrl+Shift and an arrow key to change the image size in the indicated direction. The image size is changed in increments of one point.

In Power Macintosh, to grow or shrink the plot proportionately, press the Shift key before dragging.

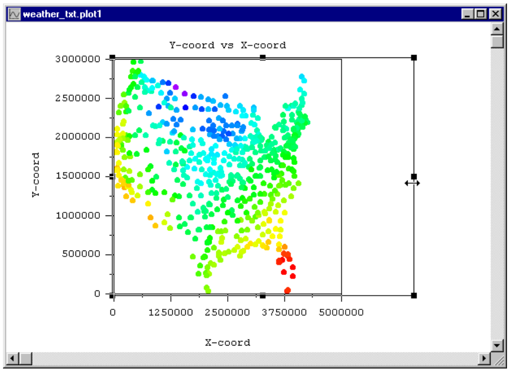


Figure 8-4: Resizing a Plot in One Dimension

Positioning Plots on the Canvas

To move a plot around on the canvas, click anywhere within the plot boundaries and drag the plot to the desired location. If you move the plot too high or too far to the left, Plot adjusts to make the plot visible.

In Windows, to move a plot using the keyboard, press Tab until the Plot is selected. Then, press Shift and an arrow key to move the plot in the indicated direction. The plot is moved in one-point increments.

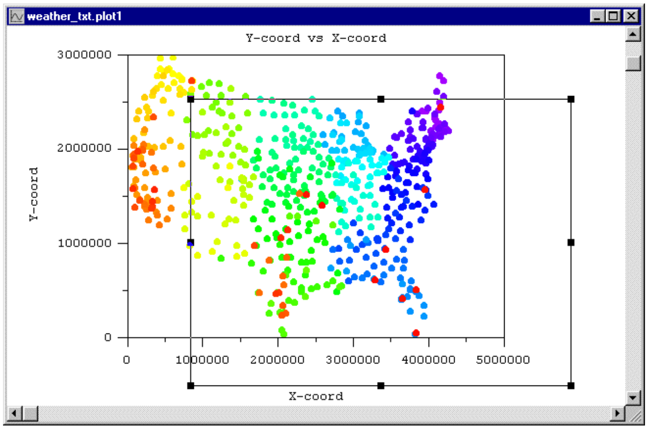


Figure 8-5: Moving a Plot on the Canvas

In Windows, the keyboard can also be used to scroll the plot window, as you would use a mouse to move the scroll bars. To scroll in one-page increments, press Home (left), End (right), Page Up (up), or Page Down (down). To scroll in smaller increments, press an arrow key in the indicated direction.

Selecting Plot Components

The mouse pointer also lets you select individual components for editing. You may select each axis and corresponding scale, each axis label, the plot title and other text labels, as well as the graph itself. See sections later in the manual for more on editing each of these components.

In Windows, to select plot components using the keyboard, press Ctrl+Tab to select the plot window. Then press Tab or Shift+Tab to select the different components. To edit the selected component, press Enter.

Synchronize

The **Synchronize** feature is an especially useful analysis tool, as it allows you to interactively display numeric values of plotted points. When **Synchronize** is enabled, Plot automatically updates all windows associated with a dataset each time a selection is made on a plot or in the data window.

Figure 8-6: Toolbar with Synchronize Activated (Windows)

To enable the Synchronize feature, make the plot window active. In Windows, click on the **Synchronize** button in the tool bar. If the plot contains more than one X,Y pair, open the Synchronize On menu list to switch between X,Y pairs (shown above). In Power Macintosh, select the **Synchronize** command from the Edit menu. Now, when you click on a point along the plot, the corresponding data value is automatically selected in the data window (Figure 8-7). Likewise, if a value in the data window is selected, the corresponding point is selected in the plot window.

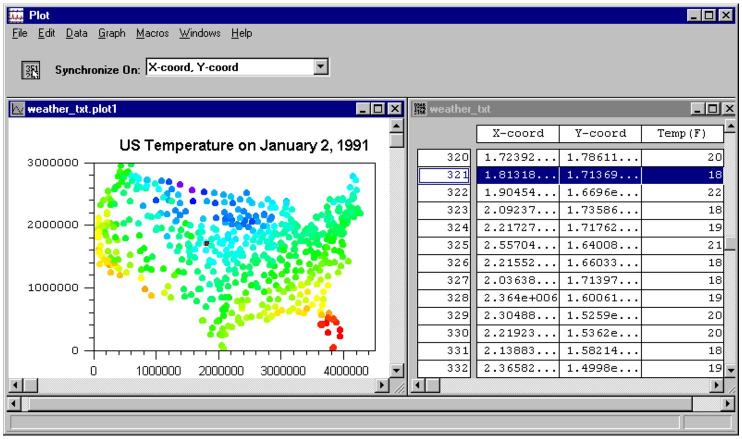


Figure 8-7: Plot and Corresponding Row of Data Selected using Synchronize

You may also select a value in the data window; Plot automatically selects the point in all plots that are drawn from that dataset which have Synchronize turned on. You must enable Synchronize for each plot in turn.

In Windows, to use the Synchronize feature with a keyboard, press Ctrl+Tab to select the plot window. Then select **Synchronize** from the Graph menu. Press Space to select the top row. Then press Shift and the up- or down-arrow key to move through the data. To synchronize on a different X,Y pair, press Tab or Shift+Tab.

Copy and Paste Plots (Power Macintosh)

One way to export plots created in Plot is to use the **Copy** and **Paste** commands under the Edit menu.

To copy a plot, click anywhere within the plot bounds, then select **Copy** from the Edit menu. The plot and axes, as well as all labels and legends will be copied to the clipboard. The copied plot can then be placed into another presentation program, using the **Paste** command.

Hold down the Option key and select **Copy** from the Edit menu to copy just the plot without axes and titles. This can be useful for creating overlays in Transform.

Chapter 9:
Creating Plots

Once data has been entered or imported into a data window, creating a plot is a matter of selecting a plot type from the Gallery under the Graph menu, then selecting the data to be plotted. This chapter first discusses the Gallery, then the X,Y Pairs dialog used for selecting data.

Note that the Gallery is only a starting point for creating plots. Plot is designed to make it easy to go on to change plot parameters, and change plot type, allowing you to create exactly the plot you need. For more information on changing plots, see Chapter 10: Axes, Labels, and Grids; Chapter 11: Editing Plots; and Chapter 12: Text and Legends. In addition, Chapter 13: Building Plots, illustrates one of Plot's most powerful features, the ability to plot pairs in layers to create advanced plots.

The Gallery

To create a plot from the Gallery select **Gallery** from the Graph menu. The Plot Gallery offers five commands and plot types: **Line, Double-Y, Scatter, Number Scatter,** and **Error Bars**. Each selection applies to a set of preselected parameters for a new plot.

The plot title is taken from the names of the data columns plotted. Axes names are taken from the names of plotted columns, or from the Plot Label entry in the Column Settings dialog for the plotted columns. Plot estimates reasonable axes bounds and intervals. Each Gallery choice is described below.

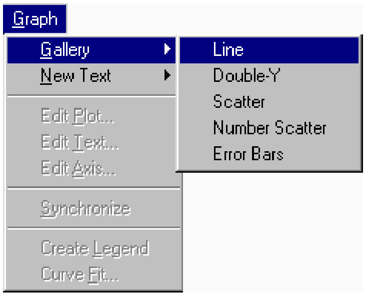


Figure 9-1: Plot Gallery Commands

Line Plot

Line graphs are useful for showing trends in an X,Y pair or for comparing multiple X,Y pairs. Selecting **Line** will plot a simple solid line connecting the data points for a selected X,Y data pair. If multiple X,Y pairs are selected, each uses a different dashed line style.

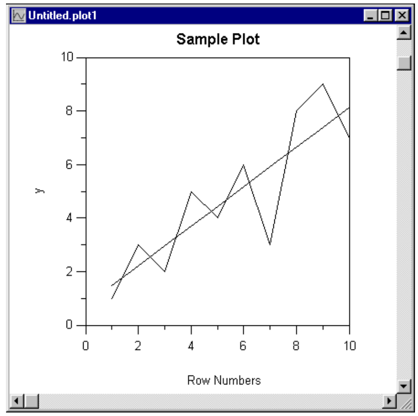


Figure 9-2: Line Plot

Double-Y

Double-Y plots allow comparison of trends where data is measured in different units. Selecting **Double-Y** is similar to selecting **Line**, except that it plots two independent Y axes. The first column is plotted as a solid line, the second as a dashed line. Double-y plots require the selection of two X,Y data pairs (see “Selecting X,Y Pairs” later in this chapter), and the horizontal axis should be equivalent for both pairs.

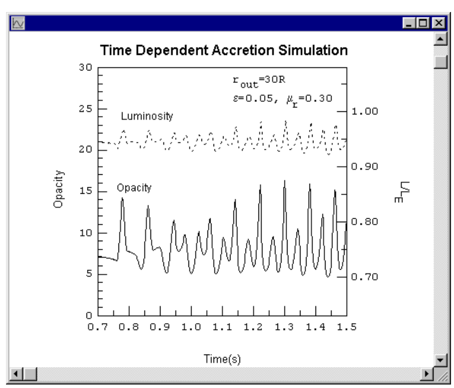


Figure 9-3: Double-Y Plot

Scatter

Scatter plots are useful for displaying especially large datasets and allow the display of a third variable at each X,Y location. Selecting **Scatter** will plot a symbol for the selected X,Y data pair to produce a scatter plot. By default, Plot simply draws a box at each X,Y location.

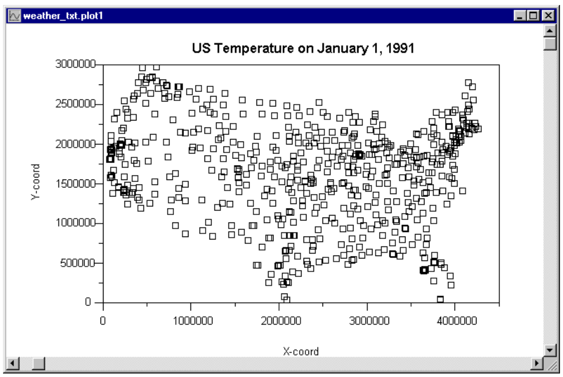


Figure 9-4: Scatter Plot

Number Scatter

Number Scatter works similar to **Scatter**, except that it prints a number from a data column, centered at each X,Y location. By default, row numbers are plotted initially.

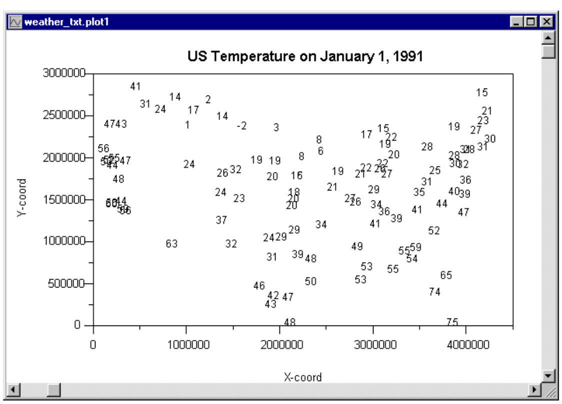


Figure 9-5: Number Scatter Plot

Error Bars

Error bars display some level of error along an X,Y data pair. Selecting **Error Bars** will plot a simple line with error bar symbols for the selected X,Y pair. By default, Plot sets the error at one standard deviation for the Y error. The error level can also be a percentage of plotted data, or it can be derived from a data column.

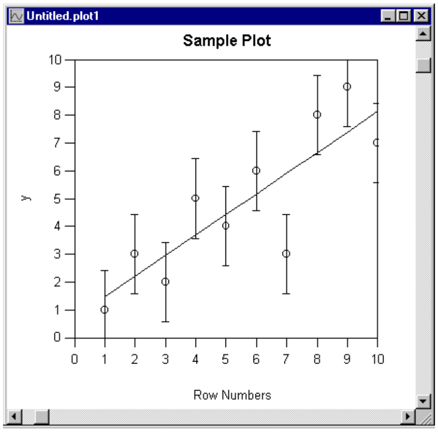


Figure 9-6: Error Bar Plot

Select X,Y Pairs Dialog

Choosing any plot type using the **Gallery** command opens the Select X,Y Pairs dialog shown in Figure 9-7. This dialog prompts you to select the data columns to be plotted.

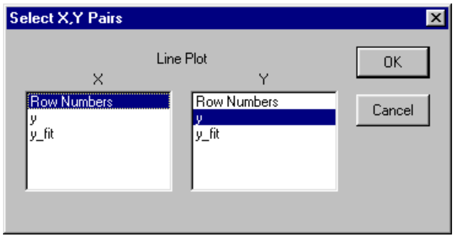


Figure 9-7: Select X,Y Pairs Dialog

The current plot type is displayed just below the "Select X,Y Pairs" title. The rest of the dialog consists of two lists, one for the X (horizontal) axis, the other for the Y (vertical) axis. Each list displays all columns from the active data window. A column selected under the X list will be plotted as the horizontal axis. A column selected under the Y list will be plotted as the vertical axis.

You can select only one X column, but you can select multiple Y columns. To select a range of columns in the Y list to plot against a single horizontal axis, select the first column, hold down the Shift key, then select the last column. To select non-contiguous columns, hold down the Control key (Command key in Power Macintosh) instead of the Shift key while selecting additional columns.

Clicking **OK** adds the selected data and draws the plot according to the Gallery submenu selection.

Select X,Y Pairs for Double-Y Plots

Selecting **Double-Y** from the Gallery will prompt you twice with the Select X,Y Pairs dialog.

The first dialog prompts you to select columns to be plotted against the left vertical axis (Y1). After a selection is made, a second dialog prompts you for the data to be plotted against the right vertical axis (Y2).

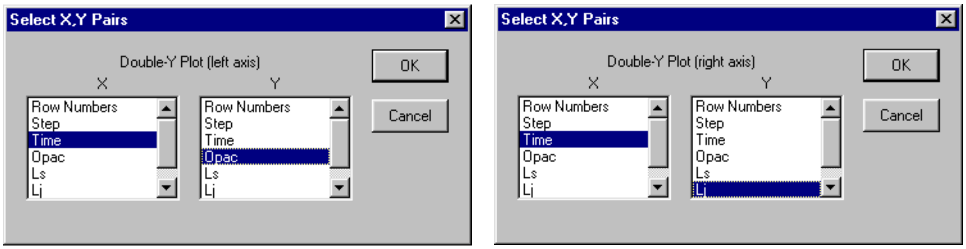


Figure 9-8: Select X,Y Pairs Dialog for Left and Right Axis of a Double-Y Plot

After selecting data columns for the right vertical axis, click **OK** to draw the Double-Y plot.

Select X,Y Pairs from the Data Window

You can also preselect X,Y pairs in the data window before selecting a plot type from the Gallery. Column(s) selected in the data window are automatically highlighted in the Select X,Y Pairs dialog.

When a single column is selected in the data window, that column is preselected for Y in the Select X,Y Pairs dialog. Plot automatically selects Row Numbers for the X column.

When two or more contiguous columns are selected in the data window, the leftmost column is preselected as X. The remaining column(s) are preselected for Y.

Chapter 10:

Axes, Labels, and Grids

Plot provides full control over axes, axes labels, tickmarks, grid lines, and scales. The procedure for editing these components, described below, is the same for all plot types.

Axis Labels Dialog

To open the Axis Labels dialog (Figure 10-1), double-click on the axis you wish to edit, or click once to select it, then select **Edit Axis...** from the Graph menu. A cut-away of the axis that would be drawn from the values shown below appears beneath the dialog.

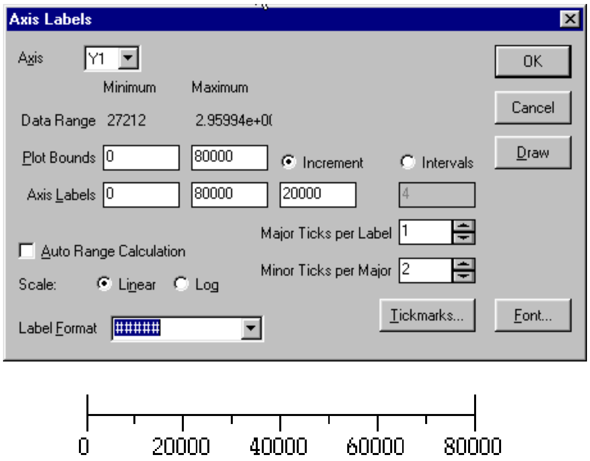


Figure 10-1: Axis Labels Dialog and Resulting Axis

Axis

Open the Axis list in the top corner of the dialog to select another axis for editing without closing this dialog.

Data Range (Windows)

Displays the actual range of data values, which is calculated from all the X,Y pairs associated with the axis.

Plot Bounds

Specify the range of values to be plotted in the Minimum and Maximum fields. You may plot only a portion of the actual data range or extend the boundaries of the plot beyond the actual data range.

Axis Labels

These boxes list the starting and ending values for axis labels. You can change these values to start and stop labels at points other than the plot boundaries.

Axis Label Increment and Intervals

You can choose to set spacing for labels by using the value listed under Increment or by setting intervals in the Intervals box. The Auto Range Calculation check box must be deselected before choosing Increment. When a new value is entered for one, the other is automatically adjusted. For example, in the dialog shown in Figure 10-1, if the Intervals value was 2 instead of 4, the increment would become 40000.

Auto Range Calculation

Check this box if you want Plot to calculate the increment, and minimum and maximum label values for you. Plot makes a “best guess” at boundaries based on the values entered for Plot Bounds, and Intervals.

Major Ticks per Label

This value sets the number of major tickmarks to be drawn per label interval. In the dialog shown in Figure 10-1, if you set the Major Ticks per Label to 5, then 5 major tickmarks will be set at intervals of 20,000 (i.e., 0, 20000, 40000, 60000 and 80000). Note that the first major tickmark is drawn at the label.

Minor Ticks per Major

This value specifies the number of minor tickmarks between major tickmarks. Note that the first minor tickmark is drawn at the major tick, and therefore is not visible. For example, if set to 2, one minor tick will appear between each pair of major tickmarks.

Scale

Plot draws plots with either linear or logarithmic scales. Click on the desired scale type here.

Label Format

Enter your own format for numerical labels or open the Label Format list and selected a predefined format.For more information about formats, see Chapter 6.

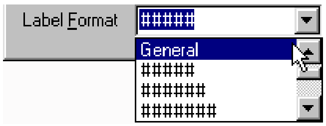


Figure 10-2: Menu of Numerical Formats for Labels

Tickmarks...

Click this button to open a dialog that provides control of tickmarks and grid lines.

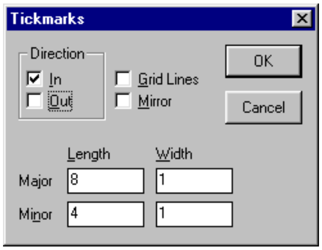


Figure 10-3: Tickmarks Dialog

Direction

Plot will draw tickmarks two ways, depending on the option selected here.

In—This option draws ticks from the axis toward the center of the plot (Figure 10-4).

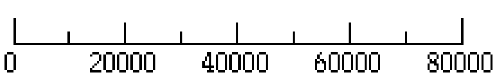


Figure 10-4: Tickmarks Drawn In

Out—This option draws ticks from the axis away from the plot (Figure 10-5).

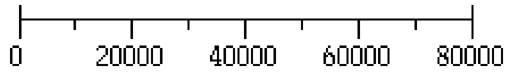


Figure 10-5: Tickmarks Drawn Out

You may also check both boxes to draw tickmarks both inward and outward.

Grid Lines

This option draws a line across the entire plot at each major tick.

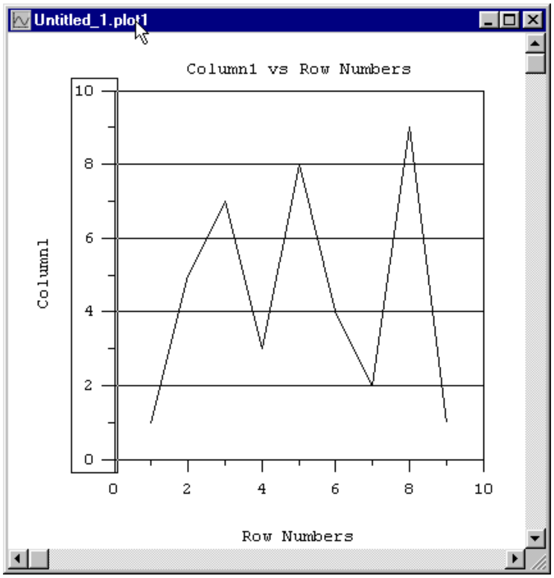


Figure 10-6: Grid Lines

Mirror

Selecting this option will draw identical tickmarks on the opposite axis of the plot.

Length and Width

You may enter length and width (in points) for both major and minor tickmarks using these boxes. The width values also apply to grid lines. The values specify point sizes, and you may enter fractions (e.g., 0.5).

Chapter 11:

Editing Plots

This chapter describes how to use the Edit Plot dialog and related dialogs to edit a plot.

Edit Plot Dialog

The Edit Plot dialog, shown below, provides control over all attributes of every plot type, as well as which data columns to plot for which axis.

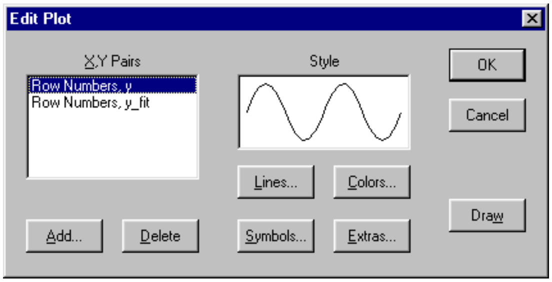


Figure 11-1: Edit Plot Dialog

To open the Edit Plot dialog, you can double-click the plot to be edited, or select **Edit Plot...** from the Graph menu.

Adding and Deleting Plot Pairs

The X,Y Pairs list displays the names of X,Y pairs available for the current plot, with the X data column listed first and the Y column listed second. To add an X,Y pair to be plotted, click on the **Add...** button. The X,Y Pairs dialog appears.

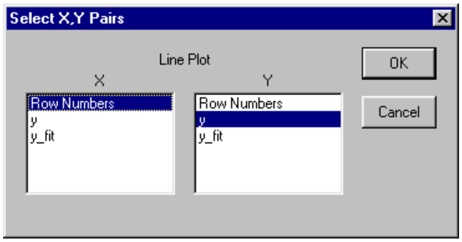


Figure 11-2: Select X,Y Pairs Dialog

All columns from the corresponding data window will by listed by column title in two lists. Clicking a column title in the 'X' list assigns that column to X; clicking a column title in the 'Y' list assigns that column to Y.

When you click **OK**, the new X,Y pair is added to the pairs list in the Edit Plot dialog. The axes are recalculated when you click on **Draw** or **OK**.

The Edit Plot dialog also allows you to delete X,Y data pairs. To do so, select the pair to be deleted, and click on **Delete**. The deleted pair is removed from the X,Y Pairs list in the Edit Plot dialog.

Style Box (Preview)

The Style box (unmarked in Power Macintosh) represents the current attributes for the selected X,Y pair. When you edit lines, symbols, colors, and other attributes of an X,Y pair, your changes will be applied to this graphic immediately before you return to the Edit Plot dialog. This allows you to preview the effects of your changes without redrawing the entire plot.

Lines

To edit the line associated with any X,Y pair, select the pair from the X,Y Pairs list and click on **Lines...** in the Edit Plot dialog. The dialog below will appear.

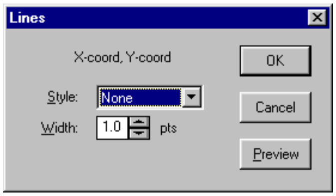


Figure 11-3: Lines Dialog

The name of the X,Y data pair associated edited line appears above the Style field.

Style

The list in Figure 11-4 provides five choices for line style.

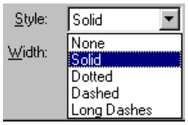


Figure 11-4: Line Styles Menu

Width

Specify the line width (in points) of the current X,Y pair.

Preview (Windows)

Choose **Preview** to update the Style box in the Edit Plot dialog.

Symbols

Clicking on the **Symbols...** button in the Edit Plot dialog opens the Symbols dialog which lets you add or change symbols plotted along an X,Y data pair. The selected pair is displayed just below the title bar.

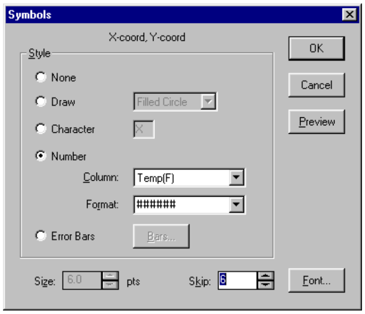


Figure 11-5: Symbols Dialog

None

Choose **None** to disable all symbols.

Draw

Choose **Draw** to plot symbols along the selected X,Y pair. You may choose from any of the nine symbols to the right of the **Draw** button by selecting the symbol type from the popup menu. Specify the size of the symbol in the Size field.

Character

Choose **Character** to plot a keyboard character along the selected X,Y pair. Enter the character to be used in the corresponding box. To set the attributes for this character, click on **Font...** or **Size**.

Number

Choose **Number** to plot actual numerical values along the selected X,Y pair.

To select a column from the corresponding data window, open the Column menu. Open the Format menu to select a numerical display format, or type your own format in the box provided. To set the font attributes of the numbers, click on **Font...**.or **Size**.

Error Bars

Choose Error Bars, then click on **Bars...** to open the Error Bars dialog.

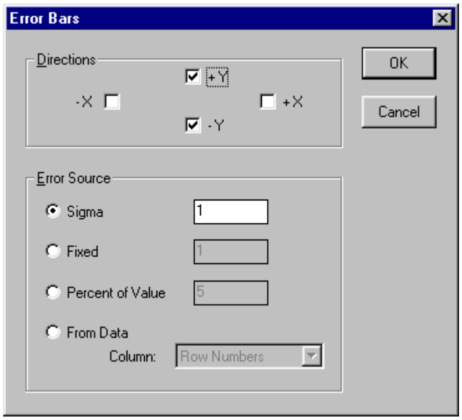


Figure 11-6: Error Bars Dialog

The first group of check boxes allow you to control how the error bars are drawn. Note that you can select any combination of choices.

Positive Y—Choose this option to draw a vertical bar for the positive error of the Y value.

Negative Y—Choose this option to draw a vertical bar for the negative error of the Y value.

Positive X—Choose this option to draw a horizontal bar for the positive error of the X value.

Negative X—Choose this option to draw a horizontal bar for the negative error of the X value.

The remaining buttons provide a choice of error sources.

Sigma—Error bars will be drawn according to the number of standard deviations from the data, as entered in the accompanying box.

Fixed—Error bars will be drawn at a fixed length specified in the accompanying box, scaled to the axis.

Percent of Value—Error bars will be drawn as a percentage of the data value.

From Data—Error bars will be scaled to a column of data, as selected from the accompanying list.

The remaining items in the Symbols dialog control character font, symbol size, and how many data values to plot.

Size

Enter the point size for drawn symbols in this box.

Skip

The Skip box lets you display symbols at intervals, rather than at every data point, to prevent symbols from overlapping. The default value of zero draws a symbol at every data point. Enter the number of data points to skip between symbols.

Font

Opens the Font dialog used to select the font for character or number symbols.

Preview (Windows)

Choose **Preview** to update the Style box in the Edit Plot dialog.

Colors

Plot lets you display lines or symbols in solid color, variable color, or grayscale. Use this feature to highlight features of an X,Y plot or to display a third column of data along an X,Y pair plot. To apply or edit colors, select **Colors...** from the Edit Plot dialog. The name of the currently selected plot pair is displayed just below the title bar.

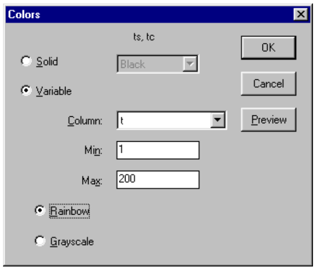


Figure 11-7: Colors Dialog

Solid

Choose **Solid** to display lines, symbols, characters, or numbers in a solid color. Open its menu to select a color (black, yellow, magenta, red, cyan, green, blue, or white).

Variable

Choose **Variable** to map color hue or grayscale along the selected X,Y plot to the column displayed in the Column field.

Column—Open the Column menu to select the column to which you want to apply color (all columns in the active data window are listed).

Min and Max—The Min and Max boxes let you set the range of values from the color control to be mapped to a color or grayscale range.

Rainbow

Choose **Rainbow** to map values to hues of blue, cyan, green, yellow, orange, and red.

Grayscale

Choose **Grayscale** to map values to shades of gray.

Preview (Windows)

Choose **Preview** to update the Style box in the Edit Plot dialog.

Extras

Clicking on the **Extras...** button in the Edit Plot dialog brings up the Extras dialog, which provides access to other features, including pen control and axis selection. The name of the currently selected plot pair is displayed just below the title bar.

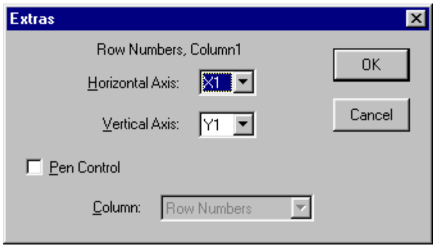


Figure 11-8: Extras Dialog

Horizontal Axis

This box displays the X axis that corresponds to the selected plot pair, where X1 is the bottom horizontal axis and X2 is the top axis. To assign a different axis to the plot pair, open the Horizontal Axis menu and select from the axes listed.

Vertical Axis

This box displays the Y axis that corresponds to the selected plot pair, where Y1 is the left vertical axis and Y2 is the right axis. To assign a different axis to the plot pair, open the Vertical Axis menu and select from the axes listed.

Pen Control

Pen control makes it possible to draw map outlines and other line drawings. It provides the equivalent of plotter pen control using three columns of data: one column for the X location, one column for the Y location, and a third for the Pen-Up/Pen-Down indicator.

Clicking the checkbox enables Pen Control. The currently selected plot pair provides the X and Y columns. The adjacent arrow button displays a list allowing you to choose a third column of data as the pen controller.

Numbers in this column that are greater than zero toggle the pen down, meaning Plot will draw the line. Numbers less than or equal to zero toggle the pen up, meaning Plot will move the pen without drawing.

To see an example of how pen control is used, open the ‘USA_Map.txt’ file as a text column file. Create a line graph using **Column 1** as X and **Column 2** as Y. You will see a map of the United States with crossed lines. Choose the **Edit Plot** command to open the Edit Plot dialog, then select **Extras...**. In the Extras dialog, select the **Pen Control** option, then choose **Column 3** from the pop-up menu. Click **OK** to close the Extras and Edit Plot dialogs. You will now see a map of the United States with state borders.

Chapter 12:

Text, Legends, and Curve Fitting

Plot lets you add and edit text labels and generate legends for your plots. In addition, Plot supports a text formatting language, which can be especially useful in macros.

Adding Text

To add text to a plot, select **New Text** from the Graph menu. In Windows, select the position where the new text should be placed in relation to the plot from the submenu. New text can be aligned with any side of a plot or with the center of the plot.

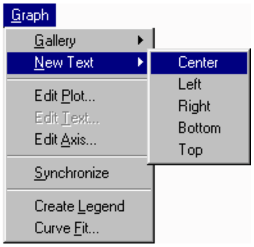


Figure 12-1: Selecting New Text and Text Position from the Graph Menu

Note

In Power Macintosh, you can change the text alignment with the macro language. Refer to Chapter 15: Macro Reference.

The Edit Text dialog that appears lets you enter text and use the standard text editing functions. When finished, click **Draw** to place the new text on the plot without leaving the dialog, or click **OK** to place the new text and exit the dialog.

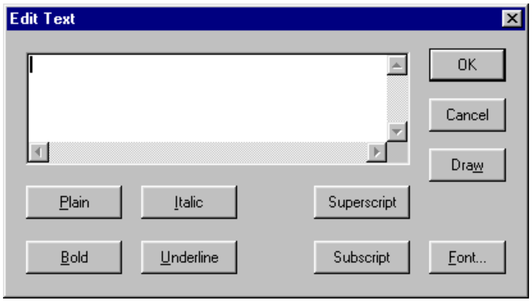


Figure 12-2: Edit Text Dialog

Moving Text

To move any text label, simply click to select it and hold (selected text in has a border). Then drag the text to the desired location.

Editing Text

To edit existing text, click once on the text and select **Edit Text...** from the Graph menu, or double-click on the text. The Edit Text dialog will appear, allowing you to edit the selected text.

Font, Size, Style

You can change text font, size, or style from within the Edit Text dialog. First select the text you wish to change. To change the font style (plain, italic, bold, underline) or position (superscript, subscript), highlight the text, then click on the appropriate button (Windows) or select the appropriate style from the Style submenu located on the Format menu (Power Macintosh).

In Windows, to change the font or font size, select the text you want to change and click on **Font...** to open the Font dialog.

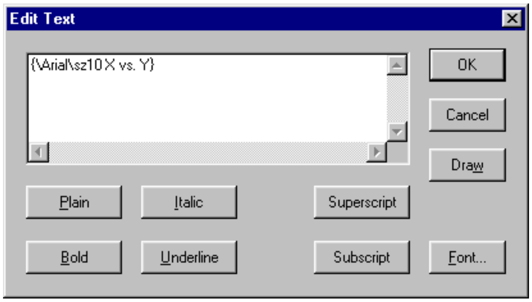


Figure 12-3: Changing Text Style

In Power Macintosh, simply select the available font and size from the Format menu.

Using the Text Formatting Language

This section describes how to use Plot's text-formatting language, which is based on a subset of the T_EX language.

The text formatting language consist of special codes that are inserted in your text. These codes are inserted automatically when you change font attributes using the Font dialog or use any of the style buttons in the Edit Text dialog. You can also type these codes directly into macros or text annotations. Macros, for example, can contain all formatting information for your text annotations. The same macro can be used to control text formatting on any platform supported by Plot.

Style attributes, such as subscript, are indicated by formatting codes in the Edit Text dialog, as shown below.

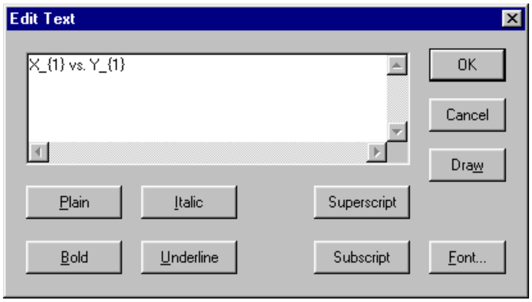


Figure 12-4: Formatted Text and Corresponding Codes in Edit Text Dialog

Using the example above, when you click **OK**, the text will appear as X_1 vs. Y_1. Other formatting codes are described later in this section.

Commands and Control Symbols

Functions and special characters are specified by commands (commands are also referred to as control sequences). A command is a backslash (\ ) followed by a name. An example of a command and its printed product follow:

`\alpha` α

Commands consisting of a single character, such as an ampersand

(&) are called control symbols. An example and its printed product follow:

`\&` &

Command Termination

A command consists of upper- or lower-case alpha characters only, never numbers. The command terminates with any non-alpha character. This character is usually a space, as shown below:

`hi \alpha there` hi αthere

Note that there is no space between 'a' and 'there'. That is because the command always absorbs the space that follows it. If it did not do this, then special characters would always be followed by a space. To add a space to the example above, the command would be as follows:

`hi \alpha  there` hi α there

Note also, in the example below, that commands absorb spaces only, not characters that follow it:

`hi \alpha\alpha there` hi $\alpha\alpha$ there

Here the backslash terminates the first \alpha, and a space terminates the second one.

Commands must be terminated to work correctly.

State Commands and Command Range

The examples above displayed specific words or characters. There are also commands that affect the state (or style) of the text, such as bold, italics, etc. Such a command, and its printed product, follow:

`hi \bf there` hi **there**

In this example, `\bf` (for bold font) affects everything that follows. This is true of all state commands. To limit the range of a state command, use brackets (`{}`), as illustrated in the next example:

`This is a {\sz18 BIG} Word` This is a BIG Word

Here, `\sz18` (specifying 18 pt. type) affects only the text within the brackets.

Text Formatting Codes

Listed below are all the codes currently available in Plot for formatting text. They can be entered in the Edit Text dialog or within macros you create using Plot.

Command Symbol

`\` The backslash character must be used to start any command.

Super- and Subscript

`^`	Superscript next character.
`_`	Subscript next character.
`^{text}`	Superscript all text within brackets.
`_{text}`	Subscript all text within brackets.

State Commands

The following commands must begin with a backslash, and affect all text that follows the command. Brackets (`{}`) can be used to limit the range of the effect of a command, as described earlier.

`\bf`	Boldface
`\bg`	Fill text background with white (opaque)
`\black`	Display/print black
`\blue`	Display/print blue
`\cyan`	Display/print cyan

`\fontname`	Change font to `fontname`. For example, `\courier-new` or `\symbol`. (Note that if a font name contains spaces, hyphens are used instead.)
`\green`	Display/print green
`\frame`	Place a frame around text
`\it`	Italics
`\legend#`	Display sample of attributes for X,Y pair referenced by #, where # is the sequence in which data appears in the Pairs list in the Edit Plot dialog
`\magenta`	Display/print magenta
`\n`	line feed
`\red`	Display/print red
`\rm`	Cancel all font and font style changes (revert to defaults)
`\sy`	Change to symbol font
`\sznn`	Change to `nn` point font
`\white`	Display/print white
`\yellow`	Display/print yellow

Characters and Symbols

To print uppercase Greek letters, capitalize the command used to print lowercase letters (see below). For example, for uppercase alpha (α), type `\Alpha`

`\\`	Backslash
`\^`	Caret
`\_`	Underscore
`\{`	Open bracket
`\}`	Close bracket
`\alpha`	α
`\beta`	β

`\chi`	χ
`\delta`	δ
`\Delta`	Δ
`\epsilon`	ε
`\eta`	η
`\gamma`	γ
`\Gamma`	Γ
`\iota`	ι
`\kappa`	κ
`\lambda`	λ
`\Lambda`	Λ
`\mu`	μ
`\nu`	ν
`\omega`	ω

`\Omega`	Ω
`\phi`	ϕ
`\Phi`	Φ
`\varphi`	φ
`\pi`	π
`\Pi`	Π
`\varpi`	ϖ
`\psi`	ψ
`\Pse`	Ψ
`\rho`	ρ
`\varrho`	Ρ
`\sigma`	σ
`\Sigma`	Σ
`\varsigma`	ς

\tau	τ
\theta	θ
\Theta	Θ
\vartheta	ϑ
\upsilon	υ
\Upsilon	Υ
\xi	ξ
\Xi	Ξ
\zeta	ζ

Creating Legends

The **Create Legend** command in the Graph menu displays the name of each plotted data column, along with a sample of the attributes for each plot pair, including line style, symbol, and color. The legend is drawn with a frame and is placed to the right of the plot.

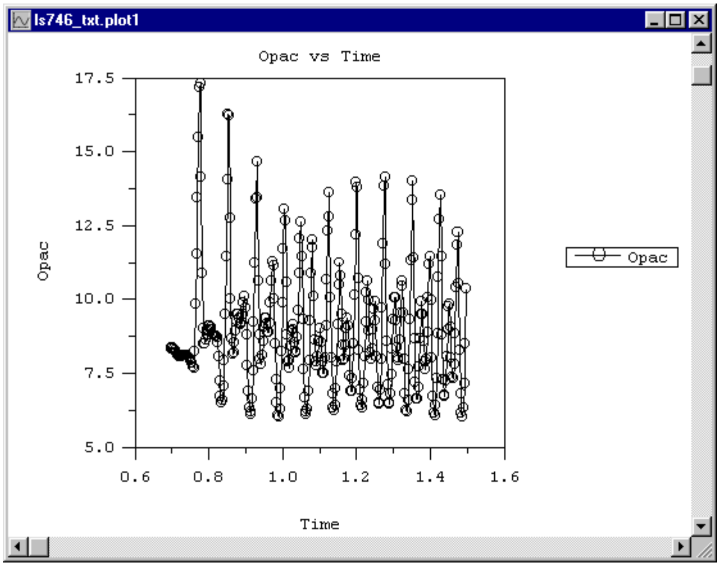


Figure 12-5: Legend Placed on Plot

You can move the legend by clicking and dragging it to the desired location. You can edit the legend by clicking on it once and selecting **Edit Text...** from the Graph menu or by double-clicking on the legend. The Edit Text dialog will appear, displaying the text and codes for other components of the legend.

Curve Fitting

Curve fitting is a method of fitting an equation to your data using an interpolation technique. In Plot, this equation may be linear, exponential, or a polynomial expression of specified degree. This section describes how to fit your data and extract information from the fit.

Fitting a mathematically-definable curve to a set of data points can be extremely useful for extrapolation and prediction or interpolation and understanding. In some cases it is merely desirable to find what curve shape best fits a set of data, since this information can reveal a great deal about the nature of the process or system being studied.

Curve Fitting a Plot

To fit your data to an equation, you must already have generated a plot from that data. With the plot window selected, choose **Curve Fit...** from the Graph menu. The Curve Fit dialog shown in Figure 12-6 appears. This dialog is described below.

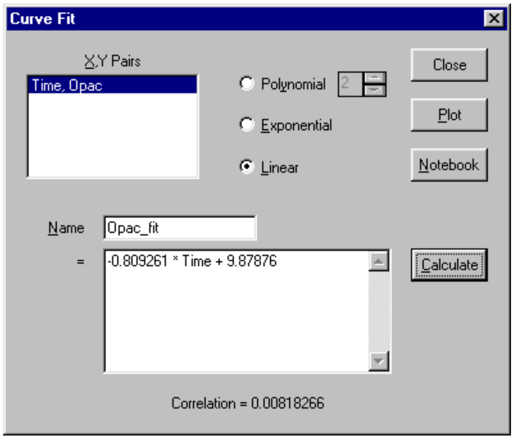


Figure 12-6: Curve Fit Dialog

Pairs

The X,Y pairs used for the current plot appear in the Pairs list, allowing you to select the pair to be fitted.

Name

This box displays the name of the data column that will result from the curve fit. By default, the name is taken from the y column of the X,Y pair selected, with the suffix `_fit` attached, but you can edit the name as desired.

Linear Curve Fits

Select this option to model your data as a straight line. Plot will use linear regression to produce the best fit through your data, yielding a slope (`m`) and y-intercept (`b`) in the form `y = mx+b`.

Exponential

Select this option to model your data as an exponential function of the form $Y = ae^{bx}$.

Polynomial

Select this option for a polynomial fit. Specify the degree of the polynomial by entering a value in the Order box. Plot fits polynomials of degree one through twelve. If you are uncertain what degree polynomial will best fit the shape of your graph, choose several options and compare their relative correlation through the use of the notebook (see the discussion later in this section). Note that higher order polynomials do not necessarily produce better fits, especially for small datasets.

Calculate

After selecting the type of curve fit, click on the Calculate button. Plot will produce an equation with the best fit through your data and display it in the text box.

Notebook

To copy the equation in the text box to the notebook for future reference, click on the Notebook button. The equation is placed in the notebook as a macro expression. You may calculate and place in the notebook as many different fitted curves as you like. (You may need to place lengthy equations in the notebook in order to view them in their entirety.)

From the notebook it is also possible to get additional information from your curve. See the notebook section later in this chapter for more information on using the notebook with curve fitting.

Plot

To plot your fitted curve on the same graph as your data, click the Plot button. By default, a curve fit is plotted as a solid line. You can, however, edit the curve fit plot as you would any other plot. Simply select the curve fit plot pair in the Edit Plot dialog and apply the desired attributes.

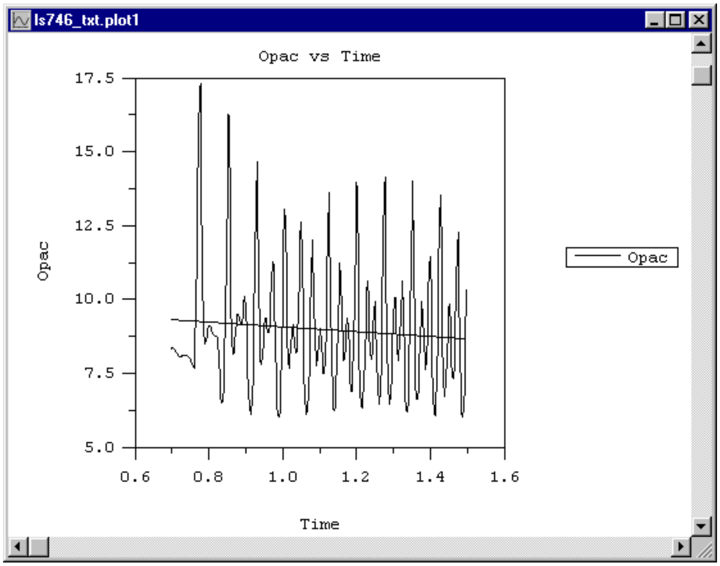
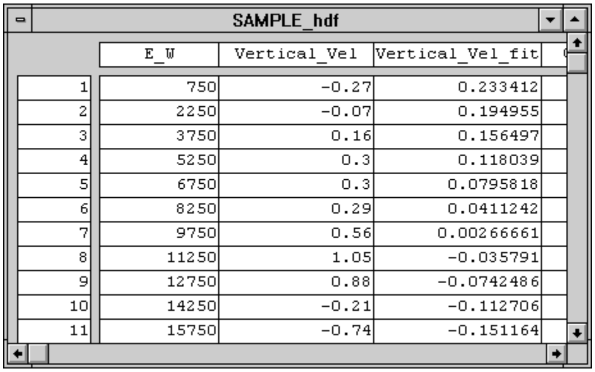


Figure 12-7: Plotted Curve Fit

When a fitted curve is plotted, the corresponding column of fitted data is placed in the data window, allowing you to compare actual data points (y-values) to those generated by the fitted equation.



	E_W	Vertical_Vel	Vertical_Vel_fit
1	750	-0.27	0.233412
2	2250	-0.07	0.194955
3	3750	0.16	0.156497
4	5250	0.3	0.118039
5	6750	0.3	0.0795818
6	8250	0.29	0.0411242
7	9750	0.56	0.00266661
8	11250	1.05	-0.035791
9	12750	0.88	-0.0742486
10	14250	-0.21	-0.112706
11	15750	-0.74	-0.151164

Figure 12-8: Curve Fit Data Column Added to Data Window

To exit the curve fit dialog at any time, click on **OK**.

Curve Fitting and the Notebook

It is often useful to try several curve fits to find the best possible fit. Often, however, it is impossible to visually determine which method is best.

By placing a curve-fit equation in the notebook, you can do a more rigorous evaluation of a fit, as illustrated by the following example.

The two plots in Figure 12-9 are the products of two curve fits taken from the same data. The left plot was produced by an exponential curve fit, the right plot was produced by a sixth-order polynomial curve fit. We can use the notebook to ascertain which produces the best fit.

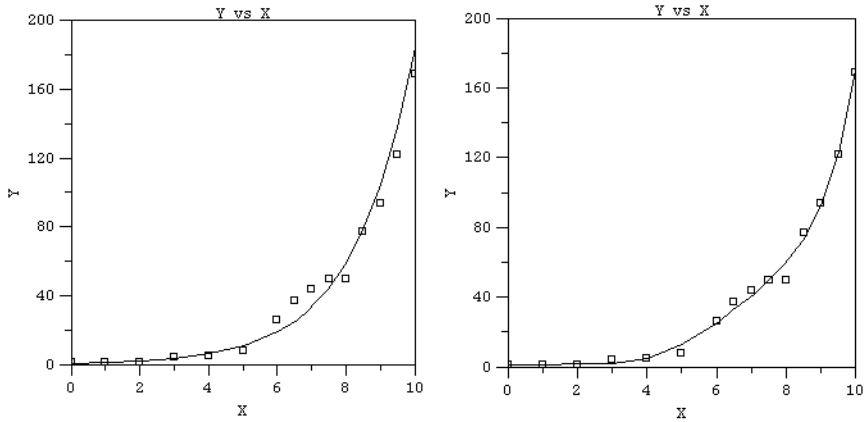


Figure 12-9: Exponential Fit (left) and Polynomial Fit (right) from the Same Data

To do so, we can calculate the correlation of each and compare. First, we enter the equation for the exponential fit in the notebook from the Curve Fit dialog as described earlier and calculate. Next, in the notebook, we enter the following expression on the line after the first equation:

```
print _corr
```

When we select **Calculate Now** from the Data menu, this expression calculates and displays the correlation coefficient for the first exponential fit. (Note that in Plot, a value of 1 represents a perfect correlation.)

The result is shown in Figure 12-10.

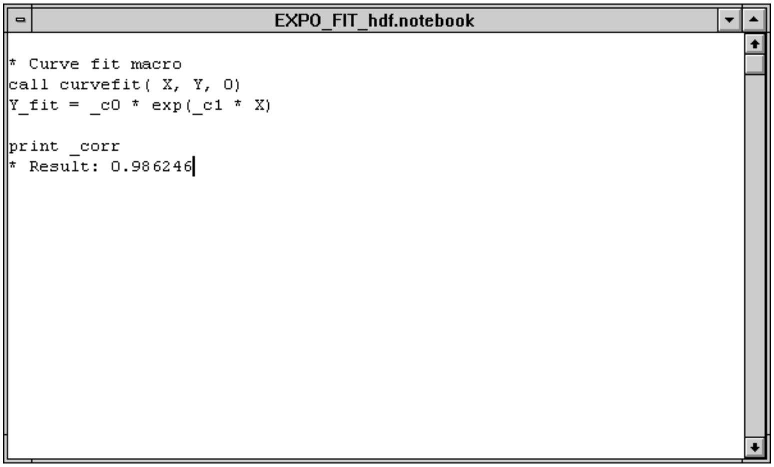


Figure 12-10: Correlation for Exponential Curve Fit Calculated in the Notebook

Next, we enter and calculate the same expression for the second curve fit plot and equation. The result is shown in Figure 12-11.

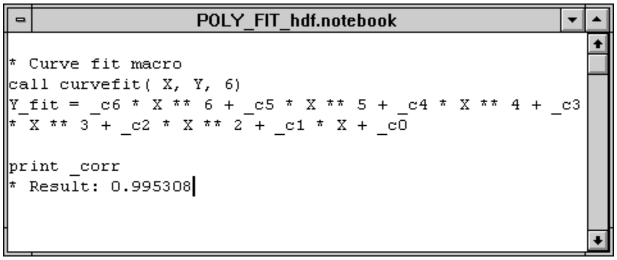


Figure 12-11: Correlation for Polynomial Curve Fit Calculated in the Notebook

As you can see, the polynomial fit is the better of the two.

Chapter 13:

Building Plots

Chapter 9 explained how to create basic plots using the **Gallery** command under the Graph menu. This chapter introduces the concept of editing and building plots in layers, and provides several layering examples that may be used to create more intricate plots.

To understand layering, it is important to note that each plot type listed in the Gallery is created using a set of parameters that have been preselected from the Edit Plot dialog.

Once a plot has been created from the Gallery, the Edit Plot dialog provides explicit control over all plot attributes for any given X,Y data pair. In addition, the Edit Plot dialog lets you use duplicate X,Y pairs. Duplicate pairs make it possible to plot the same data in separate layers, with each layer having different attributes.

Layering

Combining lines and symbols—a basic example of layering—is simply a matter of enabling both lines and symbols for the same X,Y data pair. Below, the left plot shows a solid line plotted for one Y axis and an open-circle symbol plotted for another Y axis. The right plot shows the result of enabling both a line and a circle for the same X,Y pair.

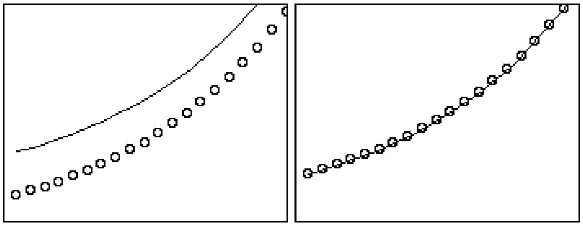


Figure 13-1: Simple Layering of Two Symbol Types for the Same Plot Pair

Although easily accomplished, this method of layering has limitations. For example, turning on color would affect all attributes for the X,Y pair, making it impossible to color the symbols without coloring the line in the example above.

Creating a Layer

By using duplicate X,Y pairs to create layers, it is possible to plot color symbols on a black line. In Figure 13-2, the left plot shows a solid black line plotted for one Y axis, and color symbols plotted for the other. The plot on the right uses duplicate X,Y pairs and layering to combine these components for the same X and Y data columns.

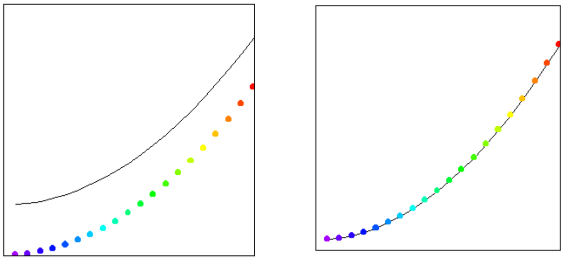


Figure 13-2: Two Different Symbol Types Layered for the Same Data

To create a similar, layered plot, start with a simple line plot created using the **Gallery** command. Then select the **Edit Plot...** command from the Graph menu to bring up the dialog shown in Figure 13-3.

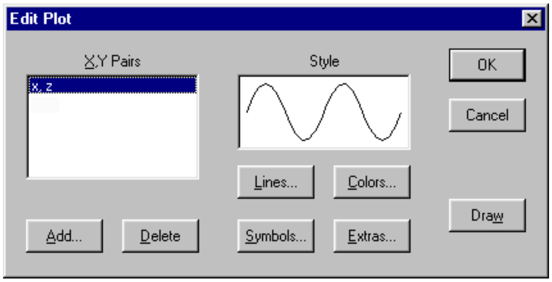


Figure 13-3: Edit Plot Dialog Listing One Plot Pair

Now click the **Add...** button and select the same X and Y columns that are already on your X,Y pair list. You should now have two matching X,Y pair entries in the list as shown in Figure 13-4. Though they represent the same data columns, each pair can now be treated as a separate plot with its own characteristics.

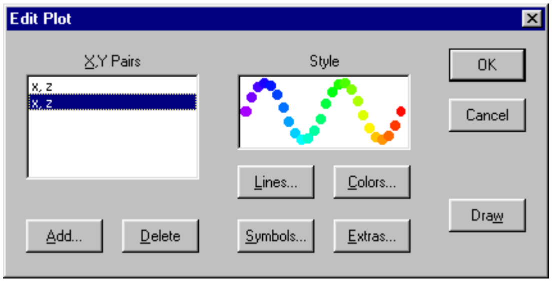


Figure 13-4: Edit Plot Dialog with Duplicate Pair Added

Now select the first pair, click on **Lines...**, and select **Solid** for line style in the Lines dialog. Click **OK** to return to the Edit Plot dialog. Next, select the second X-Y pair, click on **Lines** and select **None**. Click **OK** then click on **Symbols...**, and choose **Draw**. Open the Draw list and select **Filled Circle** (the last icon on the right for Power Macintosh). Click **OK**.

With the second pair still selected, click on **Colors...** In the Colors dialog, choose **Variable**, then choose **Rainbow**. Choose **OK**. In the Edit Plot dialog, choose **OK**.

Your plot should be drawn with colored symbols and a black line, similar to Figure 13-2, except in color. You may want to experiment by adding more layers to your plot. As you click on each X,Y pair, note that the Style box displays the appearance of that particular layer. When the plot is drawn, the layers are drawn in order and will appear as a combination of what you see in the preview.

More Layer Examples

The best way to add characteristics to a layer depends on the type of plot you are trying to create. Sometimes it is best to create all layers from duplicates of the same X,Y pair; in other cases a combination of X,Y pairs is necessary. The example combinations below illustrate several useful techniques.

For example, you can construct a unique symbol from smaller parts. Figure 13-5 shows a circle and a plus sign plotted separately, then layered using a duplicate pair to create a bull's-eye symbol.

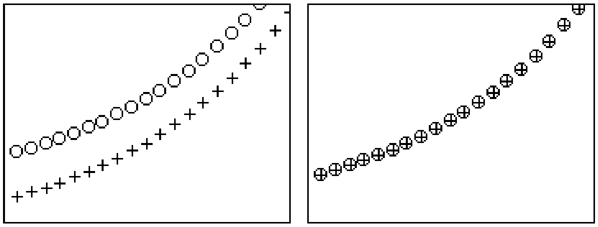


Figure 13-5: Two Symbols Layered to Create a New Symbol

Using different symbol sizes and character symbols, it is possible to produce distinctive symbols that can make complex plots much easier to understand. Figure 13-6 is an example of a 6 point @ character inside a 9 point square.

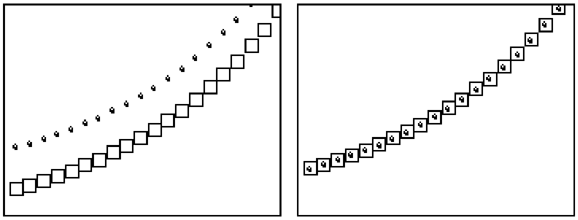


Figure 13-6: A Symbol and a Character Layered to Create a Unique Symbol

Sometimes only a few symbols are needed to make a line unique, as shown here.

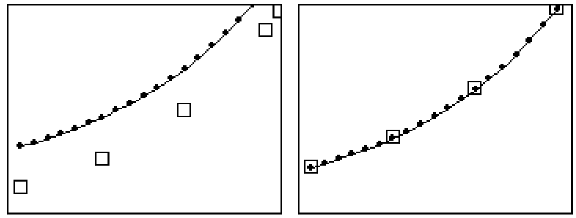


Figure 13-7: Layering Multiple Symbols

Also, numerical labels on points can be effective when layered on top of symbols. (Note in Figure 13-7 that the Skip feature was used to increase the space between the square symbols.)

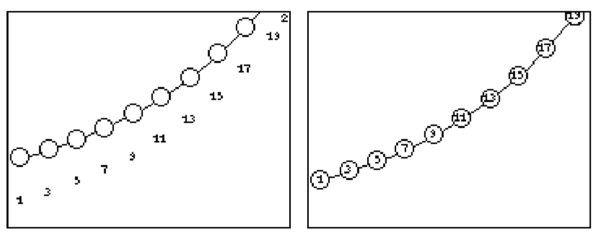


Figure 13-8: Symbols and Numerical Labels Layered

Error Bars

Using error bars effectively often requires multiple layers. This example takes the standard line with error bars and adds a circle to mark each point.

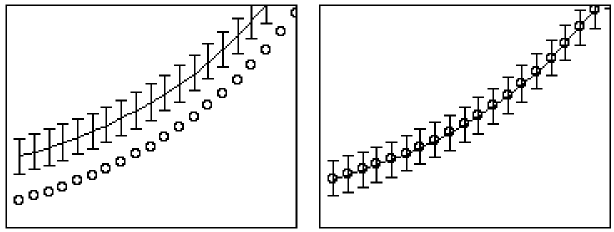


Figure 13-9: Error Bars and Symbols Layered using the Same X,Y Pair

When you have both X and Y error columns, one layer should plot the X error bars and the next layer should plot the Y error bars, as shown in Figure 13-10.

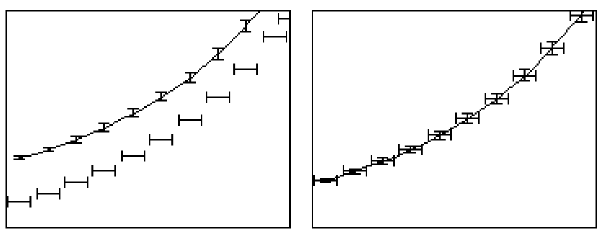


Figure 13-10: Using Separate Layers to Plot the X and Y Errors

Sometimes error bars are too small to be significant. In Figure 13-11, pen control was used to remove all of the error bars smaller than a certain level. The error bars were then layered on top of a basic line.

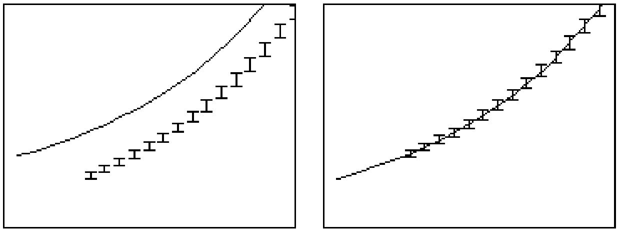


Figure 13-11: Error Bars Layered on a Line Plot

As a final example, note the following six-layer plot. Each error direction is showing a different type of error. A filled, black square is placed on the line, with a color-controlled filled circle on top of that.

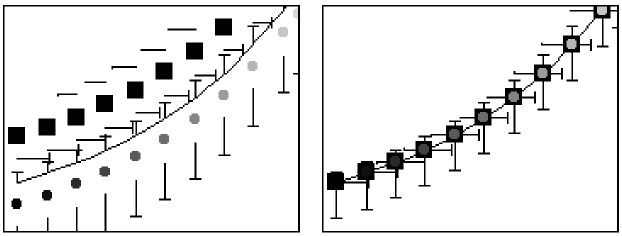


Figure 13-12: Six Separate Layers Combined

Map Outline

Layering is important even for data from different sources. In Figure 13-13, for example, a scatter plot is created from groundwater data. Using pen control, a map outline is created from a different data column. Finally, the two are layered.

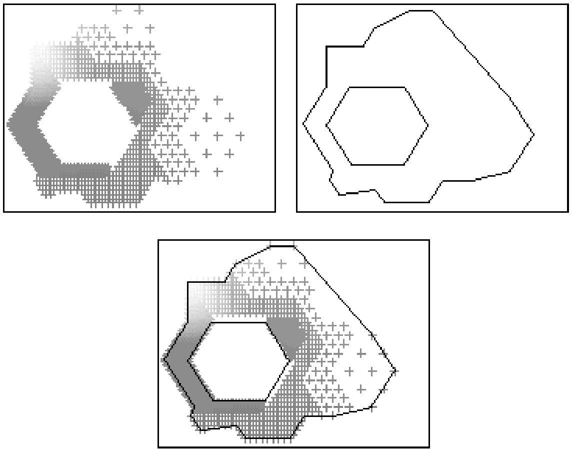


Figure 13-13: Scatter Plot and Map Outline Layered

Macros

Layering techniques are easily identified when you work in Plot's macro language. Create a plot using any of the layering techniques just described and use the **Create Macro...** command. Now use **Edit Macros...** to examine this macro.

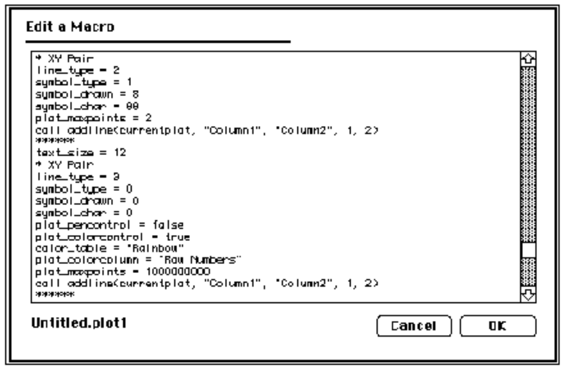


Figure 13-14: Edit Macro Dialog

Note that each separate layer is created by a call to the `addline( )` subroutine. The `line_type` and `symbol_type` variables (et al.) are set before each `addline( )` call. If you ever need explicit control over the drawing order, or the layering of the plot is confusing, refer to the macro for the step-by-step construction of the plot.

Chapter 14:

Using Macros

Plot has a built in macro language that is very similar to Fortran. The macro language is procedural and consists of three basic elements: functions (i.e., y=3*x), subroutine calls (i.e., call open("file") and reserved variables (i.e., axis_auto=true). In Plot the function and subroutine elements are primarily used to perform calculations, manipulate data, and generate images. The reserved variables give you the capability to set the preferences used by the subroutine elements.

Most of the following example macros are included in your Plot\Macros directory. To use these macros without typing them in, either copy and paste the text into an empty notebook window or simply select **Edit Macros...** from the Macros menu. Then select **Import**; in Windows you will type in the name of the file. In Power Macintosh, selecting Import adds the macro to the list; you can then "call" the macro in a notebook window. The name of each macro described in this chapter is in the parentheses by each figure number.

To run the macros, simply highlight all the text in the notebook window and select **Calculate Now** from the Data menu.

The Notebook Window

All of the three basic elements mentioned above contain a variety of macros which can be accessed through the notebook window. To open the notebook window in Windows, select **See Notebook** under the Data menu. In Power Macintosh, select the appropriate notebook under the Windows menu. The notebook window consists of the following popup menus: Columns, Functions, and Scalars (Windows only). These menus are designed to expedite entries into the notebook. When a selection is made from one of the menus, it is entered into the notebook.

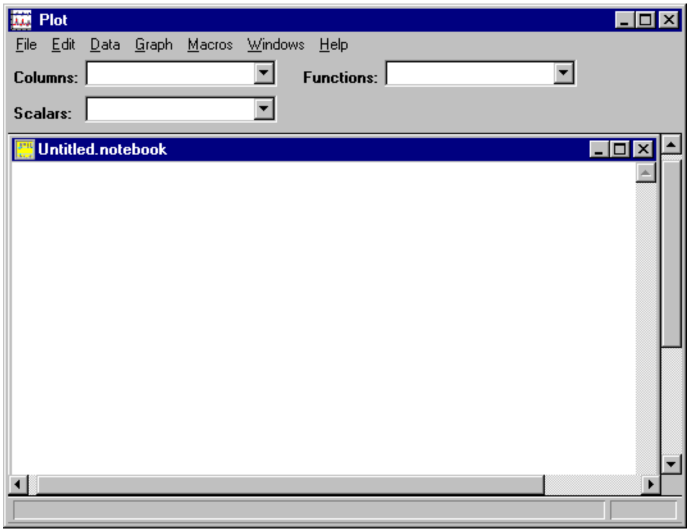


Figure 14-1: Dataset Window

Columns Pop-Up Menu

The Columns pop-up menu contains a list of all available columns currently in the active dataset. For example, the following dataset was imported into Plot and the following columns are currently available.

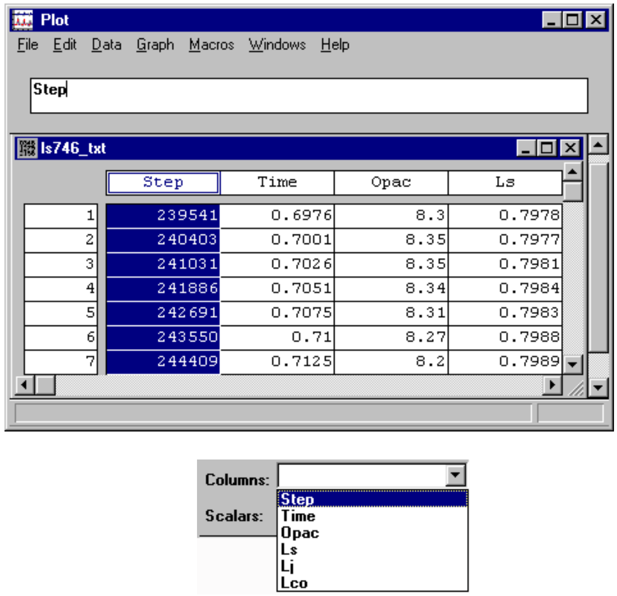


Figure 14-2: Columns Menu

Selecting a column name and pressing Enter/Return on the keyboard will paste it into the notebook at the current cursor location.

Scalars Menu (Windows)

The Scalars menu lists all of the reserved variables used in Plot. Selecting a scalar name and pressing Enter/Return on the keyboard will paste it into the notebook at the current cursor location.

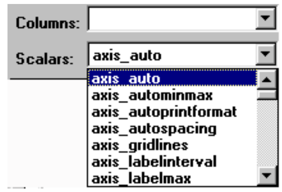


Figure 14-3: Scalars Menu

Functions Menu

The Functions menu contains a list of all the macro functions built into Plot.

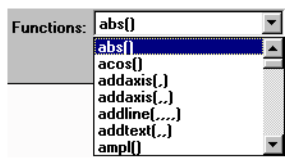


Figure 14-4: Functions Menu

When you select a function and press Enter/Return on the keyboard, the name is pasted into the notebook, followed by a set of parentheses (). The function name that is initially highlighted is always the last function selected from the menu.

Note

Each individual mathematical macro is described in greater detail in Chapter 15.

Executing Macro Commands in the Notebook

Once the notebook window is open, it takes three steps to execute a macro command. These steps are:

1. Enter either a single macro command or a series of macro commands.
2. Highlight the command or commands to be executed.
3. Select **Calculate Now** from the Data menu.

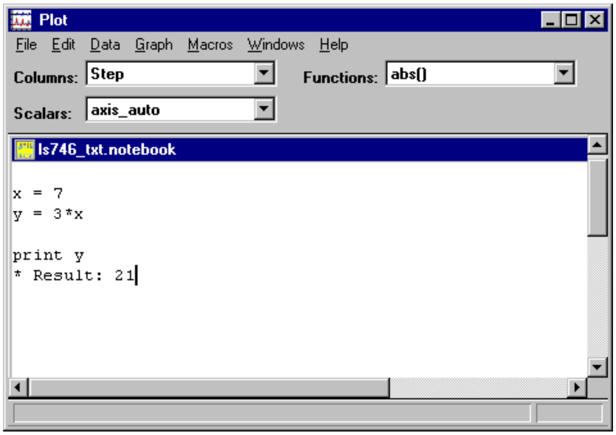


Figure 14-5: Multiply Macro (multiply.txt)

Note

Each macro command must occupy one and only one line.

Tip

If you only want to execute one macro command you do not have to highlight it. You can simply put the cursor at the end of the macro you want to execute and select **Calculate Now**.

The selection of **Calculate Now** causes Plot to execute every highlighted macro command in the notebook window starting from the top and ending at the bottom. This is very convenient for executing a series of commands in a specific sequence.

As in the example below, you may want to open a file, perform some data manipulation, and then save the file. In this example, we are using the file ‘ls746.txt’. Make sure that when you import this file, you activate the box associated with ‘Header Titles’ (Windows) or ‘Last header line contains column titles’ (Power Macintosh). This will ensure the column titles are imported.

In the example below, you would enter text into the notebook window (stored as ‘flipped.txt’ in the ‘Macros’ directory), select the entire text, and then select **Calculate Now** from the Data menu.

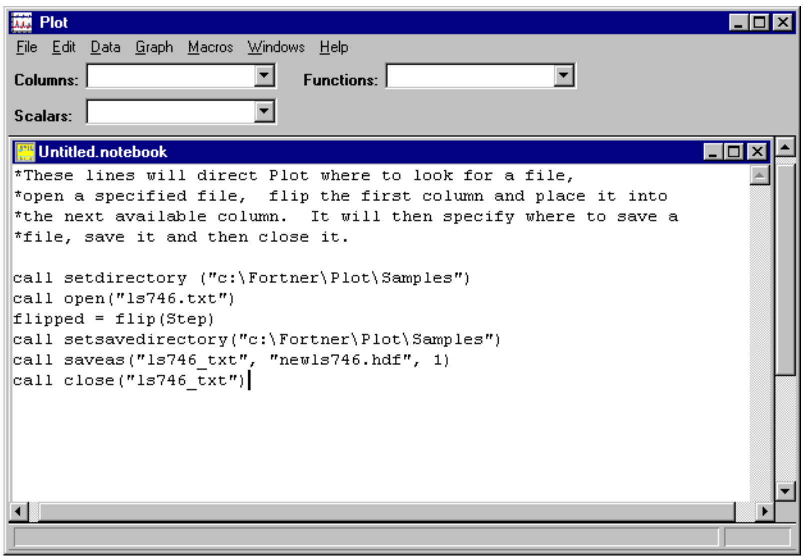


Figure 14-6: Flipped Macro (flipped.txt)

Note

Plot for Power Macintosh: in the example above, instead of ‘call setdirectory’ and ‘call setsavedirectory’ you would use the line ‘call setfolder’ and ‘call setsavefolder’. See the flipped.txt example in the ‘Macros’ directory for more information.

Comments

It is also possible to enter comments in the notebook window. Comments allow you to document your work so that it is easily understood at a later date. To enter a comment at the beginning of the line, place an asterisk (*) symbol before your text. To enter a comment following a macro command you want to execute, place an exclamation (!) symbol before your text. Any characters entered after an asterisk or exclamation symbol will be ignored by Plot when **Calculate Now** is selected, as shown in Figure 14-7.

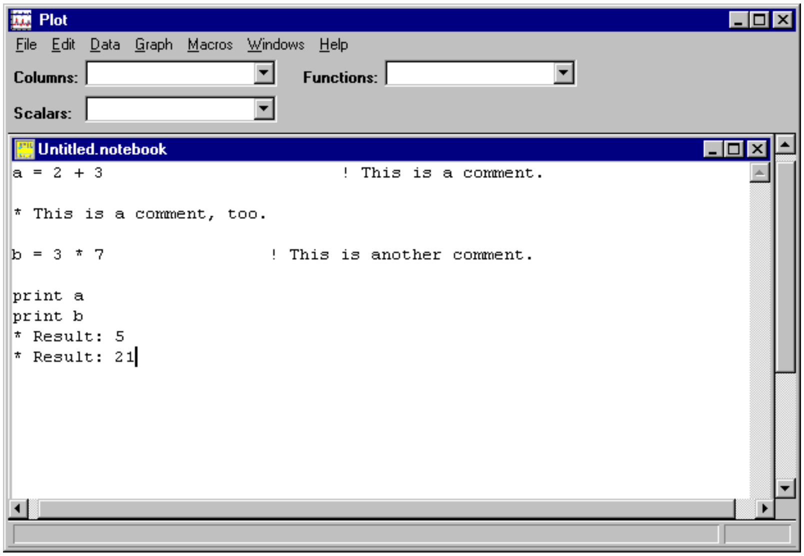


Figure 14-7: Comments Macro (comments.txt)

Macro Command Types

There are three types of macro commands: assignment statements, print commands, and subroutine calls. Each macro command must occupy one and only one line. The format of the commands is very much like Fortran.

Assignment Statements

Assignment statements evaluate an expression and assign the result to a variable or a column. For example:

```
x = (3.14/2)*y
```

If the expression returns a scalar value or a string, the variable being assigned is defined as a scalar variable. If the expression returns a column, a new column is created with the assigned name.

In the example above, if `y` is a scalar variable, then `x` will be a scalar variable. If `y` is a column, then a new column called `x` will be created when the macro command is executed.

Print Commands

Use the `print` command to display the result of a scalar calculation in the notebook. For example:

```
print 27.3*sqrt(54)
* Result: 200.613
```

The `print` command works only for scalar variables and expressions. (If you try to print a column or a column expression you will get an error dialog.)

Note

Output from the `print` command is preceded by an asterisk to distinguish it as a comment, rather than as a command to be executed.

Subroutine Calls

The `call` command is used to invoke either Plot's built-in subroutines or a custom macro. An example of a Plot subroutine is:

```
call open("ls753_txt")
```

A full listing of Plot subroutines is given in Chapter 15. An example of a custom macro call is:

```
call ls753_txt_plot1_macro
```

where "ls753_txt_plot1_macro" is the name of a macro that you have previously created. The custom macros that you can call are listed in the Macros menu. Note that you cannot pass parameters to the custom macros. You can instead set a scalar variable to a value that is then used in your custom macro. All variables in Plot are global. See the section earlier in this chapter for more information on custom macros.

Error Messages

Plot will display an error dialog box any time an illegal operation is attempted, like incorrectly typing in the macro or attempting to perform an impossible procedure. The Warning dialog box will display the macro in question and give general information about the problem. Below is a list of common error messages you may encounter. Please keep in mind that this list is not all inclusive and you may get an error message not listed below.

1. Warning, Incorrect or Misplaced Parameters in the Macro Function
2. Warning, Window Not Found
3. Error, Identifier Not Found. Undefined symbol or array
4. Syntax Error in line, ## Characters From the Beginning of Line

Some tips for avoiding error messages:

1. Ensure the macros are typed correctly.
2. Ensure CAPITAL letters are used when they refer to a name that has capital letters.
3. Ensure that double quotes (“”) are used around names when using the subroutine macros.
4. Ensure that the word `call` is placed before a subroutine macro name.

Components of Macro Commands

Before discussing how to use the macro commands, it is important to identify the components of a command. A macro command consists of any number of Mathematical Operators, Constants, Scalar Variables, Column Arrays, and Plot’s internal macros. Each component is defined below.

Mathematical Operators

Mathematical operators are listed below in order of precedence:

-	negation (unary minus)
**	exponentiation
*	multiplication
/	division
+	addition
-	subtraction
//	concatenation (of strings)

To override the precedence, use parentheses `()`.

An example of how mathematical operators are used is shown in Figure 14-8.

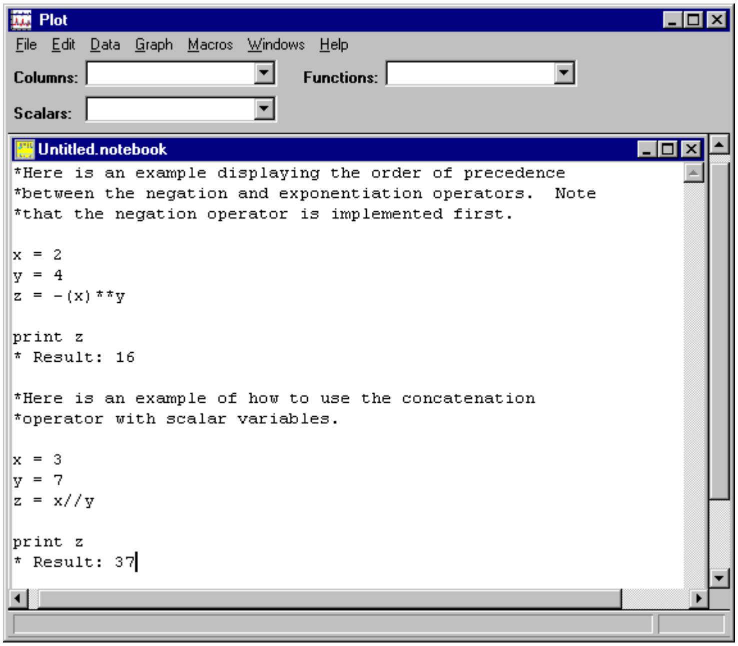


Figure 14-8: Precedence Macro (preceden.txt)

Constant

Numerical constants are entered with periods (.) for decimal points and with `e` representing exponentiation. Integers are converted to floating point values before being used in calculations. Examples of numerical constants are:

```
3.14159
1.0669e+23
```

String constants are always delimited with double quotes ("). Some examples are:

```
"ls753_txt"
"3.14159"
```

Scalar Variable

Scalar variables are created from assignment statements. For example,

```
pi = 3.14159
```

creates the scalar variable `pi`, which has the value "`3.14159`". All variables are global and may be referred to any time after they are created.

The values of all variables are stored as strings, so they may represent names as well as numbers. Whenever a variable is used in an expression that requires a number, it is converted to a number before being used. The maximum length of a variable value is 255 characters. The examples below are all legal expressions:

```
pi = "3.14159"
myfile = "ls753_txt"
print 1 + "3.14159"
* Result: 4.14159
```

In particular, the pi assignment statement given above and the one given previously are exactly equivalent.

The legal characters for a variable name include all of the alpha numerics and the underscore (_). Names cannot start with a number. Periods, slashes, asterisks, and plus or minus signs (`. / * + –`) are not allowed in variable names. For example, `pressure/dens`, `ke.hdf`, and `v*u` are invalid variable names, but `xv102` and `K_Energy` are valid.

Values for all variables are stored as ASCII strings which may be up to 255 characters in length. When a numeric value is required by Plot, the string is converted to a number. A value of zero is returned if the field does not start with a number.

An example showing values of scalar variables is show in Figure 14-9.

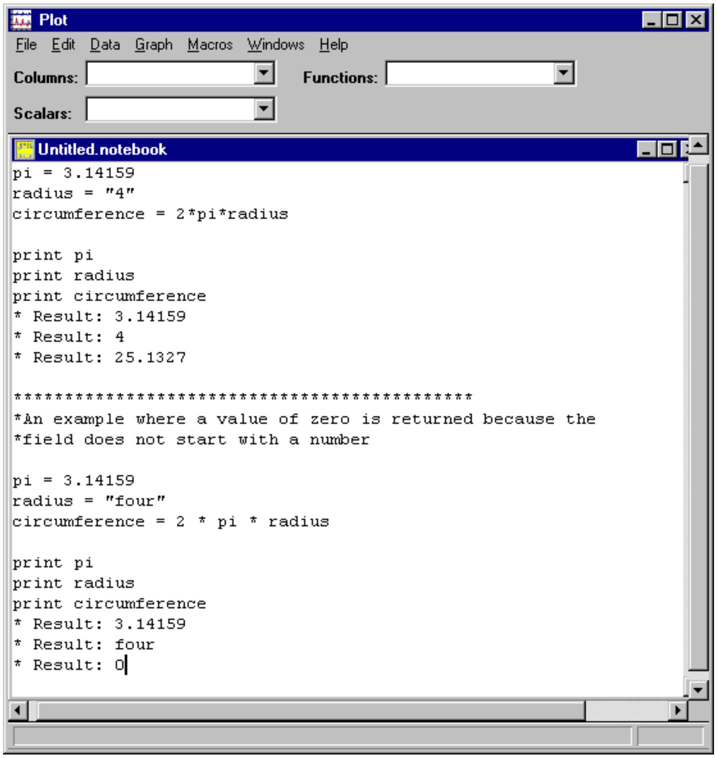


Figure 14-9: Pi Macro (pi.txt)

Dataset Column

Any column from the current dataset may be used in a calculation as you would use a scalar variable. For example, if A is a column, then executing the macro command:

```
B=log(A)
```

generates a new column called B, where each element is equal to the log of the corresponding element in A.

The following expressions are also legal:

```
B=log(A)+1
C=A*A+2
```

In the first example above, each element in B is set to the log of the corresponding element of A, plus 1. Note that columns and scalar variables can be combined in the same expression: the scalar variables are treated as columns with every element set to the scalar value.

In the second example, each element in B is set to the square of the corresponding element in A, plus 2. If you use more than one column in an expression, they must all have the same number of data values.

The new column will have the number of data values and attributes of the first column in the expression. Legal characters for column names used in expressions are the same as those for scalar variables.

Function Macro Commands

Function commands are used to calculate mathematical expressions and assign the results to either scalar variables or dataset columns. If the function returns a scalar value or a scalar string it is defined as a scalar variable. The best way to display the value of a scalar variable is to use the `Print` command, as shown in Figure 14-10. In most calculations all of the columns must have the same dimensions. Note that a mathematical expression can contain any number of these functions.

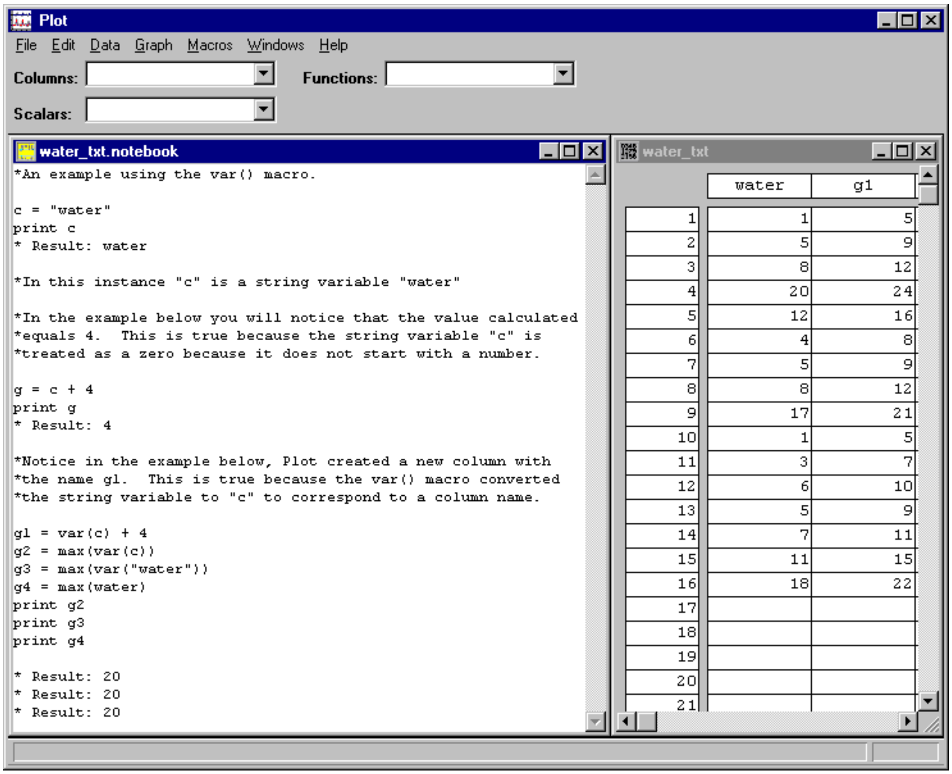


Figure 14-10: Water Macro (water.txt)

In order to run this macro, the data for the "water" column must first be entered in a new dataset window.

Function Macro Examples

The following pages provide examples of how to use function macro commands.

Mathematical Trigonometric Functions that Create Arrays or Return Scalars

If the function returns a column, then a new column with the variable name is created in the next vacant column. In the example below, you can see that since x is a scalar variable, y is also a scalar variable. Since ‘Time’ is a column, a new column called ‘Sine_Time’ is created.

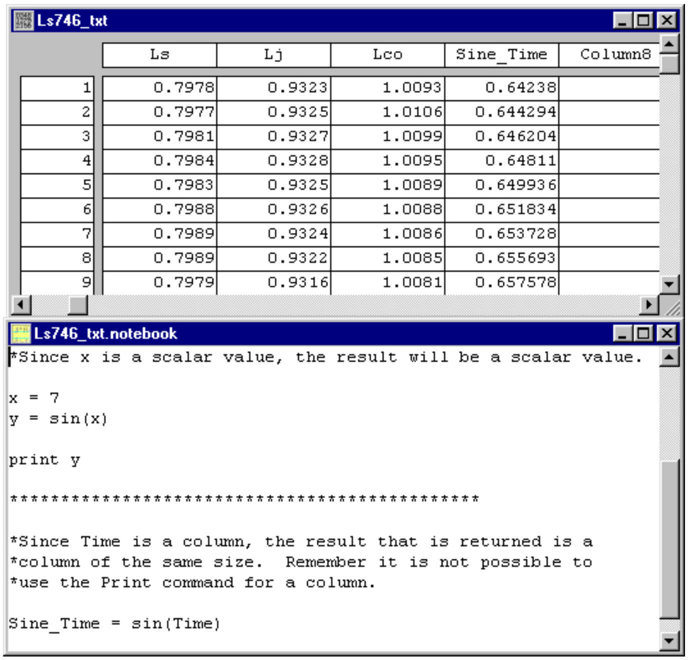


Figure 14-11: Sine Macro (sine.txt)

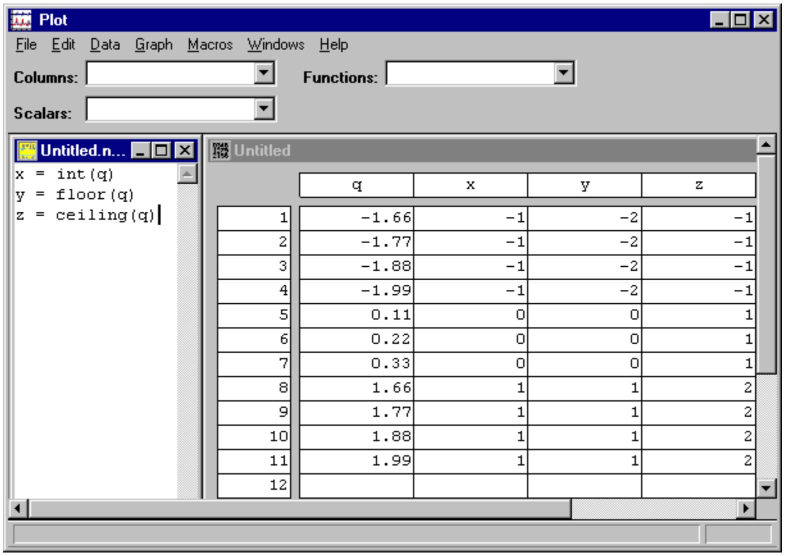


Figure 14-12: Round Macro (round.txt)

As you can see, when given a column q as an argument, the x column truncated the decimals, the y column rounded down, and the z column rounded up. This macro requires the column q to be entered in a new dataset window before running.

Mathematical Functions that Require a Column and Return a

Scalar Value

Plot contains a number of functions which return a scalar value when using a column as an argument.

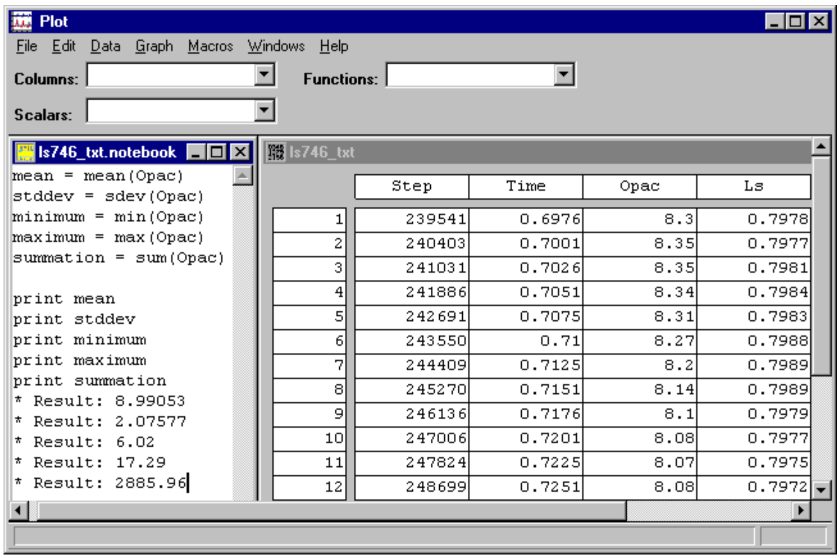


Figure 14-13: Means Macro (means.txt)

This example shows the mean, standard deviation, minimum value, maximum value, and sum of column "Opac" as scalar values. The file "ls746.txt" in the Plot\Samples directory must be open in order to run this macro.

Data Manipulation Functions

These functions require at least one data column as an argument and return a new data column. In all cases, the returned column has the same dimensions as the first column used in the argument.

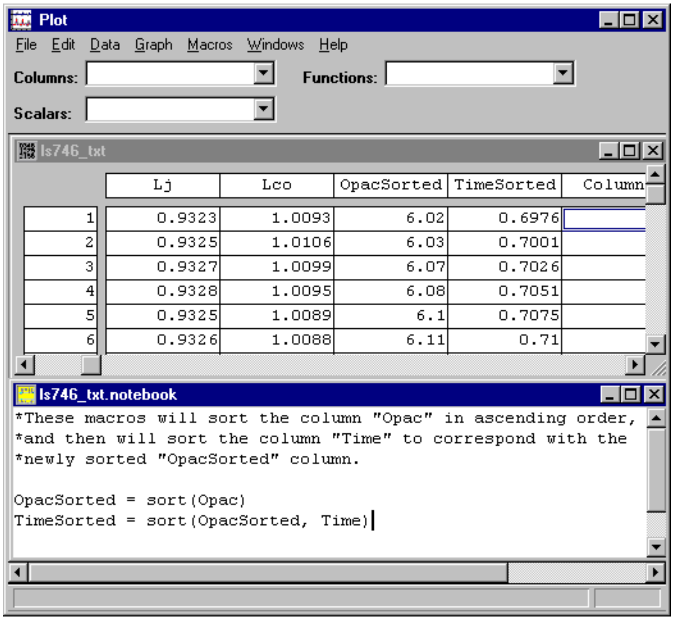


Figure 14-14: Sorting Macro (sorting.txt)

In this example, the column "Opac" will be sorted in ascending order and stored in the column "OpacSorted". The column "Time" is then sorted to correspond with the newly sorted "OpacSorted" column. The file "ls746.txt" in the Plot\Samples directory must be open in order to run this macro.

Data Manipulation Functions that Require a Column and Return a Scalar Value

These functions use a single data column as an argument and return a scalar value.

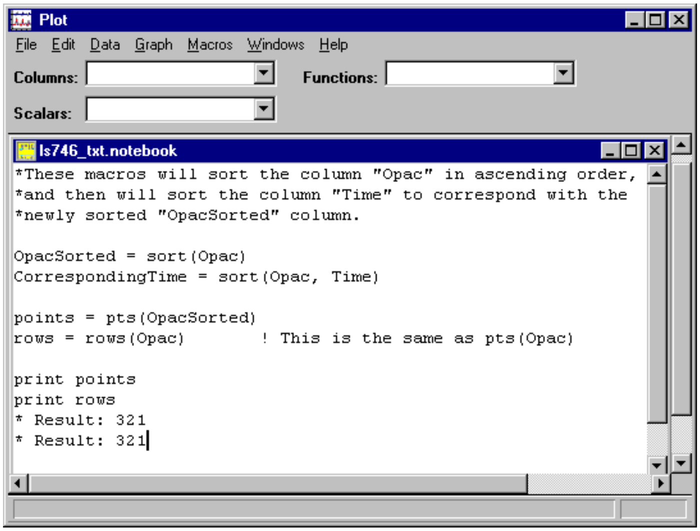


Figure 14-15: Row Macro (row.txt)

This example proves that the column "OpacSorted" has the same dimensions as the original column "Opac". In order to run this macro, the file "ls746.txt" in the Plot\Samples directory must first be opened.

Data Manipulation Functions that Generate Columns Based on Analytic Expressions

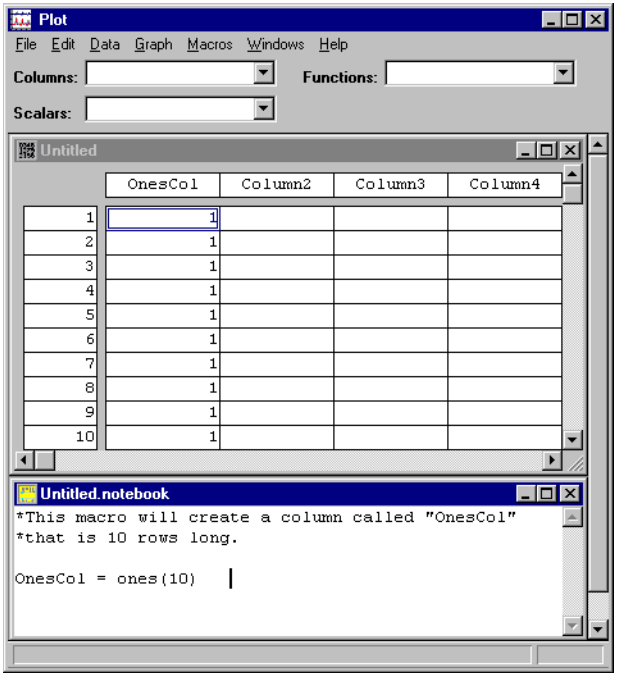


Figure 14-16: Ones Macro (ones.txt)

This example creates a 10-row column where every element has the value of one. These types of functions are useful for generating new columns based in analytic expressions.

Advanced Functions

Plot includes a variety of more complex data manipulation functions. As you can see in the following example, Plot inserts the frequency of the column's data values into the appropriate location. There are three values ranging from 1 to 2. There is one value ranging from 3 to 4. There are two values ranging from 5 to 6.

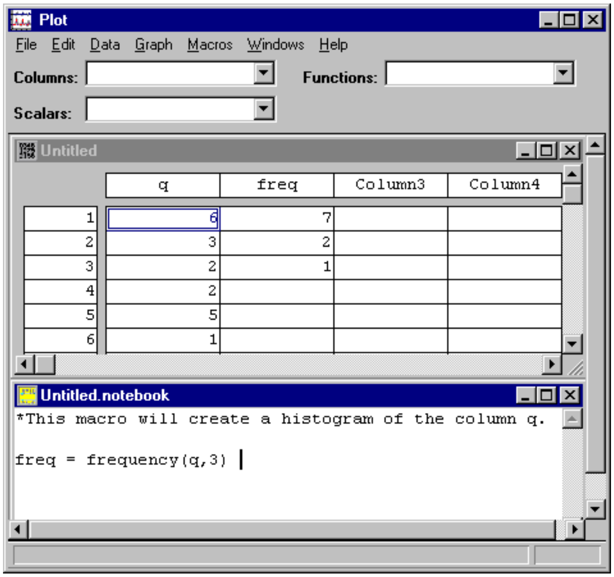


Figure 14-17: Histogram Macro (histogra.txt)

In order to run this example macro, the column q must first be entered into an empty dataset window.

Subroutine Macro Commands

Plot’s macro language includes subroutines for performing various tasks, such as creating images, processing data, and running custom macros. In order to use most subroutines, the macro command requires the word “call” to be placed in front of the macro. However, some of the subroutines have the form of functions. These subroutines DO NOT require the word “call” to be placed in front of the macro.

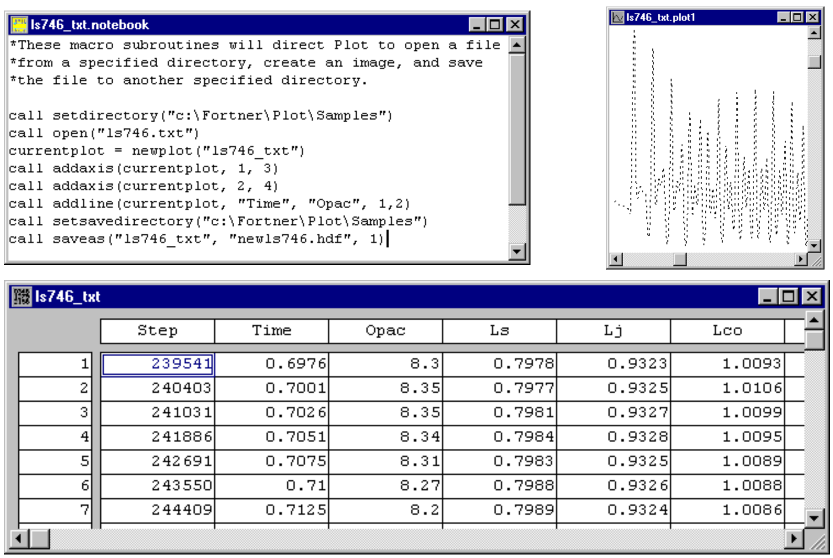


Figure 14-18: Plot Macro (plot.txt)

Tip

You can run custom macros from the notebook window by using the “call” statement.

It is also important to note that most subroutines require either the datasetname, windowname, filename, or macroname to be entered after the macro. Plot uses the following definitions:

Datasetname - Name of a dataset window currently open in Plot.

Windowname - Name of an image window currently open in Plot.

Filename - Name of a file as it is saved on disk.

Macroname - Name of macro listed under the Macros menu.

Note

In Plot, images that are created from your datasets have the datasetname with the image type and number added to it. The number specifies the order in which the image is created. This is helpful if you create several images from one dataset.

Subroutine Macro Examples

Various subroutine macros are included with Plot which allow users to open files, create plots, process data, and run other macros, as shown in the example below.

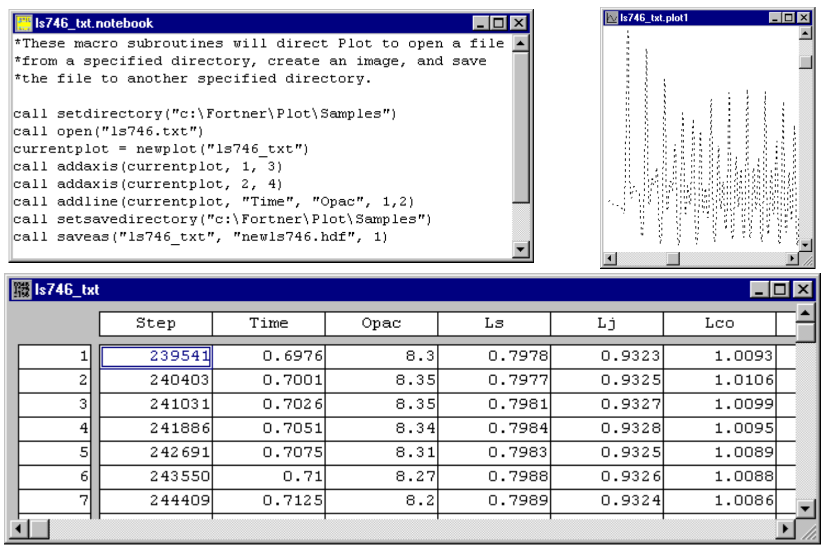


Figure 14-19: Plot Macro (plot.txt)

Note

Plot for Power Macintosh: in the examples above, instead of ‘call setdirectory’ and ‘call setsavedirectory’ you would use the line ‘call setfolder’ and ‘call setsave-folder’. See the plot.txt example in the ‘Macros’ directory for more information.

Reserved Variable Macro Commands

In Plot the reserved variable macros are primarily used to set the preferences of the subroutines. In other words, reserved variables provide the details necessary for a subroutine to carry out it's function. For example if you attempt to create an image using the subroutine:

```
currentplot = newplot(currentdataset)
```

All details about how to build that image are controlled by the reserved variable macros. The reserved variable macros define everything from image size to adding the axes. They can be used from the notebook window, but are primarily used when editing custom macros.

Reserved Variable Macro Examples

Plot contains a list of reserved variable names for use in macros.

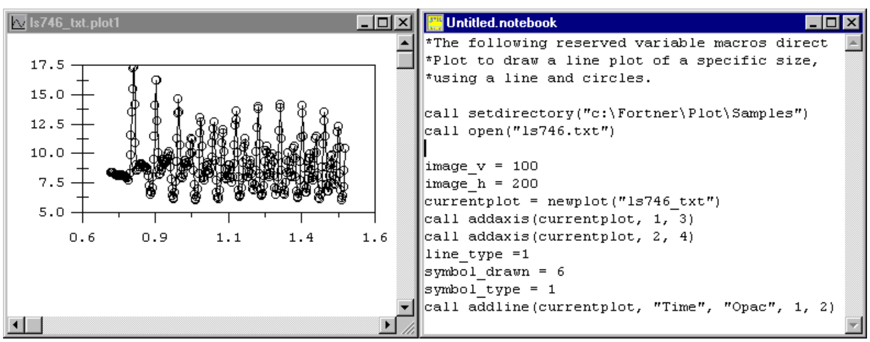


Figure 14-20: Line Plot Macro (lineplot.txt)

The reserved variables define the graph's vertical and horizontal dimensions, as well as the line type, symbol type, and number of symbols to be drawn.

Note

Plot for Power Macintosh: in the example above, instead of 'call setdirectory' and 'call setsavedirectory' you would use the line 'call setfolder' and 'call setsavefolder'. See the lineplot.txt example in the 'Macros' directory for more information.

Custom Macros

In addition to executing macros from the notebook, you can also execute macros from a stored menu.

Custom macros can be created in one of three ways, depending on whether a notebook window, a dataset array window, or an image window is currently active.

Custom Macros: Notebook Window

To save macro commands that you have entered in the notebook, make the notebook window the active window on your screen and select **Create Macro...** from the Macros menu. Enter a name for the macro, then click **OK**. The custom macro will be added and stored in the Macros menu for future use. If you choose the same name as an existing macro, the new macro will replace it. To use the custom macro at a later date, you can either select it from the Macros menu or you can type “call” in front of it on the notebook window.

Note

Custom macros never accept parameters. Any information needed by a macro should be stored in variables accessed by the macro before calling it.

The saved custom macro will work exactly as it did in the notebook. Note that every line of the notebook is brought into the custom macro, so make sure that any comments either begin with an asterisk (*) or an exclamation (!).

Custom Macros: Dataset Window

Custom import macros can also be saved for any non-HDF files that you open. To save an import macro, simply open a file and select **Create Macro...** from the Macros menu. Enter a name for the macro, then click **OK**. Creating an import macro is a good way to automate the loading of files that have exactly the same format.

Note

The dataset window must be the active window before selecting **Create Macro** from the Macros menu.

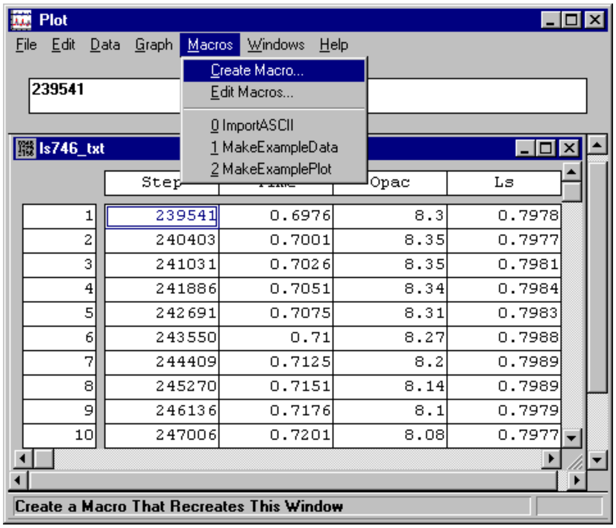


Figure 14-21: Creating a Data Macro

Note

If your dataset window was an imported HDF file, then the **Create Macro...** option in the macros menu will be grayed out.

Custom Macros: Plot Window

Custom plot macros can also be saved for any plot that you create. To save a plot macro, simply create a plot and select **Create Macro...** from the Macros menu. Enter a name for the macro, then click **OK**.

Note

The plot window must be the active window before selecting **Create Macro...** from the Macros menu.

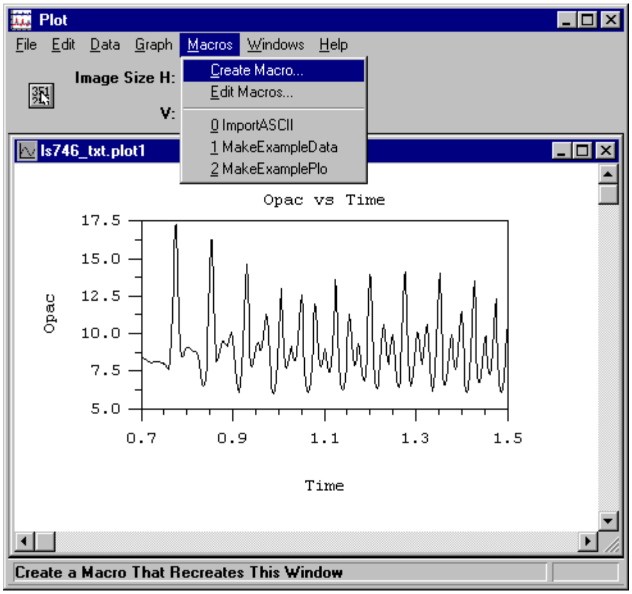


Figure 14-22: Creating a Plot Macro

Editing Custom Macros

To edit a custom macro, select **Edit Macros...** from the Macros menu.

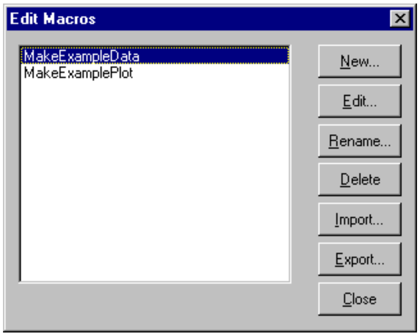


Figure 14-23: Edit Macros Dialog

This dialog lets you create a new macro or select a macro from a scrolling list and edit, rename, delete, or export it. You can also import macros from text files. Each of these options are described below. When you are finished, click **Close/Done** to return.

Note

It is always a good idea to make a back-up of all your custom macros using the **Export...** button.

New Macro

Choose **New** to create a new macro from scratch. In the dialog that appears, type the new name and choose **OK**. The Macro Editor, described next, appears, allowing you to enter a new macro.

Edit Macro

Click **Edit...** to display the Macro Editor dialog.

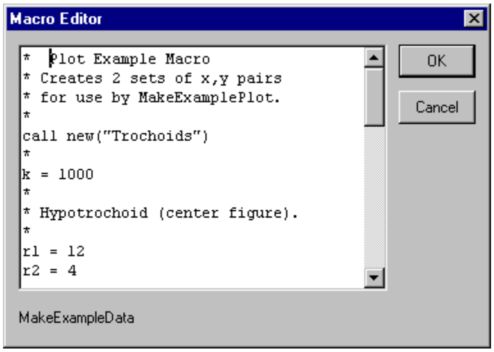


Figure 14-24: Macro Editor Dialog

You can now edit your macro. When done, click **OK** to save your changes or Cancel to leave your macro unchanged.

Rename Macro

Click on **Rename...** to rename the selected macro. Enter the new name and click **OK**; otherwise click **Cancel**.

Delete Macro

Click **Delete** to remove the selected macro. As soon as you confirm the deletion, the macro is removed from the list and cannot be recovered.

Import Macro

Click **Import...** to select a text file and save it in Plot as a custom macro. It will now appear on the Macros menu and in the dialog list.

Export Macro

Click on **Export...** to save the selected macro as a text file. You can edit this file with any text editor. Use **Import...** to read the file back in.

Custom Macro Examples

Examples of macro expressions, with explanatory notes, are given below. These examples can be executed in the notebook window.

Analytic Functions

You can use Plot to produce plots of analytic functions:

```
z = (series(201)-1)/100-1
fcn = exp(-3*z**2)
```

The expression above is equivalent to plotting the function exp(-3*x**2) from -1 to 1 at intervals of 0.01. This function is a bell curve and represents a Gaussian distribution.

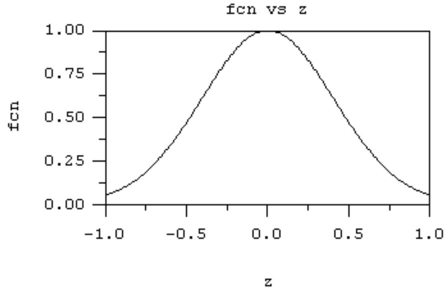


Figure 14-25: Analytic Expression Example

Notice that the function requires the use of series(z) instead of using RowNumbers. This is because Plot has an unlimited number of rows in a dataset. By using series(z), we tell Plot to create 201 points when evaluating the expression.

Fast Fourier Transform

You can use Plot to take FFTs of your data. To run the example, you need to have a column named A, which has a number of elements equal to a power of 2 (128, 256, 512,...). We used a column of numbers from 1 to 128.

```
A=series(128)                !create original column
c_A = complex(A,0*A)         !create complex column
c_freq = fft(c_A,1)          !perform forward fft
log_amp = log(ampl(c_freq))!log of the amplitude
```

The `complex()` function takes a real column and an imaginary column (here just zeros) and creates a complex column, with each set of two data elements representing the real and imaginary parts.

The `fft()` function does the Fast Fourier Transform in the forward (+1) direction. The `ampl()` function calculates the real amplitude of a complex column, here the complex frequency column. We took the log of the amplitude so we could make a better plot.

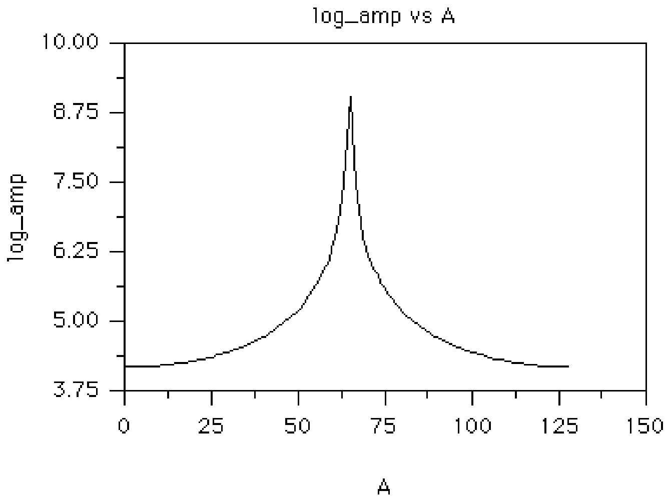


Figure 14-26: FFT Example

Length of a Line

This macro brings up a dialog that displays the length of a line. The column referred to is the first column of the active dataset. The length calculated is the length of the plotted line when this data column is plotted versus RowNumbers. There must already be a dataset open for this macro to function properly. The calculation is based on the following equation:

$$\int_{x_1}^{x_2} \sqrt{1+\left(\frac{df}{dx}\right)^2}\, dx$$

Figure 14-27: Length of Line Equation

```
user_interactive=false
myname=getcolumnname(1)
q=var(myname)
total_length=sum(sqrt(1 + ddx(q)**2))
user_interactive=true
call prompt("The line length of "//myname//
" =",total_length)
```

Setting `user_interactive` to false disables all future dialogs. The next line stores the name of the column in the variable myname for later reference.

`q = var(myname)` creates a column named `q` identical to `myname`. This is done because `myname` is a variable that contains a string, while mathematical functions require either a column or scalar as their arguments.

The next line calculates the line length and, finally, the last line displays the calculated line length.

Sorting Data

Plot preserves the order in which your data was stored. Sometimes, however, it is necessary to organize data columns in the order of their numeric values, rather than their position in the original data file. For example, if your data consists of two columns, A and B, where pairs of values in A and B are ordered in an arbitrary manner, you can sort the data with the following commands.

```
x = sort(A)
y = sort(A,B)
```

These two commands will create two new columns of data, `x` and `y`. Column `x` will have all the points that were in column `A` sorted from lowest to highest values. The second command creates a column named `y`, which contains all the data that was originally in column `B` but sorted by the numerical values stored in column `A`. In other words, every point `(x,y)` is identical to every point `(A,B)`, referenced in order of ascending values of `A`.

This method of reorganizing data is most helpful in producing a line graph of `B` vs. `A`, where `B` is a function of `A` when the data is not organized by ascending values of `A`. In the above example, a scatter plot of y vs. x is identical to a scatter plot of `B` vs. `A`. A line graph of y vs. x will move steadily from left to right, while the corresponding line graph of `B` vs. `A` will not.

Chapter 15:

Macro Reference

Plot has mathematical and data manipulation functions which return scalar values and/or create data columns. If the mathematical expression contains a column, the result will be a column of the same size. If there are no arrays in the mathematical expression, the result will be a scalar value. In all calculations, all of the columns must have the same dimensions.

Mathematical Functions

Note that a mathematical expression can contain any number of these functions.

Mathematical Trigonometric Functions that Create Columns or Return Scalars

The following trigonometric functions are available. They will either create a new column or return a scalar value, depending on the argument in the mathematical expression.

`sin(q)`	Sine of the argument (q) in radians
`cos(q)`	Cosine of the argument (q) in radians
`tan(q)`	Tangent of the argument (q), in radians
`asin(q)`	Arcsine of the argument (q), returns radians
`acos(q)`	Arcosine of the argument (q), returns radians
`atan(q)`	Arctangent of (q), returns radians between -p/2 and p/2
`atan2(q,p)`	Arctangent of (q)/(p), returns radians between 0 and 2p
`sinh(q)`	Hyperbolic sine of the argument (q) in radians
`cosh(q)`	Hyperbolic cosine of the argument (q) in radians
`tanh(q)`	Hyperbolic tangent of the argument (q) in radians
`dtor(q)`	Converts the argument (q) from degrees to radians
`rtod(q)`	Converts the argument (q) from radians to degrees

More Mathematical Functions that Create Columns or Return Scalars

`mod(q,p)`	Integer remainder of q/p
`log(q)`	Natural logarithm of the argument (q)
`exp(q)`	Exponential function of the argument (q)
`log10(q)`	Base 10 logarithm of the argument (q)
`pow(q,p)`	Take the argument (q) to the power of p
`sqrt(q)`	Square root of the argument (q)
`abs(q)`	Absolute value of the argument (q)
`int(q)`	Integer truncation of the argument (q)
`floor(q)`	Round down to next integer
`ceiling(q)`	Round up to next integer

Mathematical Functions that Require a Column, and Return a Scalar

These functions require a single data column as an argument, and always return a scalar value.

`mean(q)`	Arithmetic mean of the column (q)
`sdev(q)`	Standard deviation of the column (q)
`min(q)`	Minimum data value of the column (q)
`max(q)`	Maximum data value of the column (q)
`pts(q)`	Number of points in column (q)
`rows(q)`	Number of rows in column (q), same as pts (q)
`elem(q,#)`	Returns the value of the # number in column (q)
`sum(q)`	Sum of all the data values of the column (q)

Data Manipulation Functions

These functions require at least one data column as an argument, and create a new data array. The resultant columns always has the same dimensions as the first array used in the argument.

`flip(q)`	Exchanges the rows (flips top for bottom) of column `q`.
`shu(q)`	Creates a new column where the data has been shifted up one row. The bottom row will be set equal to the previous bottom row.
`shd(q)`	Creates a new column where the data has been shifted down one row. The top row will be set equal to the previous top row.
`rotrows(q,#rows)`	Rotates the rows in `q` from top to bottom by `#rows`. Copies the bottom rows to the top as they fall off. Negative numbers move the data up; positive numbers move the data down.
`recode(q,n)`	Replaces all zero values in column `q` with floating point value `n`.
`zapnan(q,n)`	Replaces all `NAN` (Not-A-Number) values in column `q` with floating point value `n`.
`LEmask(q,#)`	Returns a column where every element is set to `1` if the corresponding element in `q` is less than or equal to `#`, and set to `0` if it is greater than `#`.
`LTmask(q,#)`	Same as `LEmask()`, but with less than.
`GEmask(q,#)`	Same as `LEmask()`, but with greater than or equal to.
`GTmask(q,#)`	Same as `LEmask()`, but with greater than.
`EQmask(q,#)`	Same as `LEmask()`, but with equal to.
`NEmask(q,#)`	Same as `LEmask()`, but with not equal to.

Data Manipulation Functions that Generate Columns Based on Analytic Expressions

These functions are useful for generating new arrays based on analytic expressions.

`zeros(n)`	Creates a column of n rows, where every element in the column is equal to `0`.
`series(n)`	Creates a list of numbers ranging from `1` to `n` at intervals of 1. Since Plot has an infinite number of rows, use this expression when you would need to reference the row number in a calculation.
`rand(n)`	Creates a column of n rows, where every element is a random number between `0` and `1`.
`ones(n)`	Creates a column of n rows, where every element in the column is equal to `1`.

Advanced Functions

Plot allows you to use these more complex data manipulation features in macros.

`fillmissing(q)`	Fills missing data in column q. All `fill_ reserved` variables must be set appropriately before calling this function. See `fill_equal`, `fill_method`, and `fill_value` in the “Macro Variables” section.
`frequency(q,n)`	Creates a histogram of column `q`. The function first divides the number range into `n` equally spaced bins, and then counts the number of data points in the column `q` which fall in each bin.
`movingavg(q,n)`	Calculates moving average over `n` cells.
`smooth(q,n)`	Smooths column `q` using `n` passes.
`sort(q)`	Sorts the column `q` in ascending order.
`sort(q,p)`	Sorts column `p` in ascending order of `q`. Only data from c `p` is returned, but the resulting order is the same as if `q` is sorted and numbers in `p` are arranged in the same order.
`ddx(q)`	Calculates a numerical approximation to the derivative of `q`. Each element of the result, D_i is calculated: D_i=(qi+1-qi-1)/2.

Fast Fourier Transforms

Plot provides a set of functions for dealing with complex variables and Fast Fourier Transforms.

Complex Numbers

You can create complex columns using the `complex(r,i)` function. Here `r` is a column containing real values, and `i` is a column containing imaginary values. The resulting complex column has twice as many elements as either the real or imaginary columns (which must be the same size). The columns alternate real and imaginary numbers. For instance, the real column:

```
r1
r2
r3
```

and the imaginary column:

```
i1
i2
i3
```

become the combined complex column:

```
r1
i1
r2
i2
r3
i3
```

In the list below, `c` refers to complex columns (that is, double length columns that alternate real and imaginary components).

`complex(r,i)`	Combines the real columns `r` and `i` (both of the same dimension) into a complex column.
`real(c)`	Extracts the real part of complex column `c` and makes a new column with only the real values.
`imag(c)`	Extracts the imaginary part of complex column `c` and makes a new column of only the imaginary values.
`complexap(a,p)`	Forms a complex column, given two columns representing the amplitude and phase of those complex numbers. The phase is assumed to be in radians.
`ampl(c)`	Calculates the amplitude of each complex number in the complex column c of real and imaginary parts, and returns a real column with only the amplitude values.
`phase(c)`	Calculates the phase of each complex number in the complex column `c` of real and imaginary parts, and returns a real column with only the phase values, in radians.
`fft(c,n)`	Performs a one-dimensional Fast Fourier Transform on complex column `c`. The parameter `n=1` is for a forward transform and `n=-1` is for an inverse transform. The number of rows must be a power of 2.

Macro Subroutines Reference

Plot's macro language includes subroutines for opening files, creating plots, processing data, and running other macros.

Calling Macro Subroutines

To use one of these subroutines, put it in a macro with a call statement. For example:

```
call open("ls753.txt")
```

or equivalently,

```
myfile = "ls753.txt"
call open(myfile)
```

Some of the subroutines return a value, so you can use them in expressions as you would a function. For example:

```
name = getdatasetname(currentplot)
```

Custom Macro Subroutines

You also access your custom macro subroutines (the macros on the Macros menu) using the call statement:

```
call mycustommacro
```

Note that custom macros never accept parameters. Any information needed by the macro should be stored in variables before calling the macro.

Window Naming Conventions

Several macro subroutines either accept or return window names. The data window name is the name of the dataset. The plot window name is the dataset name appended with 'plot' and a number specifying the order in which the window was created (for example, `ls753_txt_plot1`). If you specify a window name without a number (for example, `ls753_txt_plot`), the window with the highest number will be used.

Listing of Macro Subroutines

In the listing of macro subroutines, the following conventions are used; each of these four names must be enclosed in quotation marks when used in macros:

`datasetname`	Name of a dataset currently open in Plot [e.g., ls753.txt (Macintosh), ls753_txt (Windows)].
`windowname`	Name of a currently open window (e.g., ls753_txt_plot).
`filename`	Name of a disk file (e.g., ls753.txt).
`macroname`	Name of macro listed under the Macros menu.

call addaxis(windowname, whichaxis, whichside **)**

Adds axis `whichaxis` to window `windowname`. `whichaxis` is represented by a number from 0 to 255. `whichside` indicates on which side of the plot the axis should appear:

`=1` Top (north)

`=2` Right (east)

`=3` Bottom (south)

`=4` Left (west)

call addline(windowname, X_column, Y_column, xaxis, yaxis **)**

Adds a new X,Y plot pair to the plot in the window `windowname`. `X_column` and `Y_column` are the names of the columns to plot. Reserved variables are used to indicate what type(s) of lines and/or symbols to plot for this data. New X-Y pairs must have an existing X axis (`xaxis`) and an existing Y axis (`yaxis`) in order to be drawn. These can be any axis numbers from 0-255 that have been created with call a`ddaxis( )`.

call addtext(windowname, text_string, whichside **)**

Adds the text in `text_string` to window `windowname`. When the plot is resized or modified, text objects stay in relative position to the side of the plot indicated by `whichside`. `whichside` indicates the relationship to the plot.

=0 Relative to the origin of the plot, not rotated

=1 Above the plot, not rotated

=2 Right of plot, rotated 90 degree clockwise

=3 Below the plot, not rotated

=4 Left of plot, rotated 90 degrees counterclockwise

call beep

Beeps the speaker once.

call close(windowname **)**

Closes window `windowname`. If you close the dataset window, all windows associated with that dataset are closed. An error will occur if you try to close a dataset from the notebook window associated with that dataset.

call closeall

Same as **Close All** from the File menu. If `user_interactive=false`, every window will be closed without being saved.

call copy(windowname **)**

Copies the highlighted data in the data window windowname to the Clipboard. Same as **Copy** from the Edit menu.

currentplot = newplot(currentdataset **)**

Creates a plot window from the dataset `datasetname`, using the current reserved variable values, and returns the name of the new window. The new plot is empty until axes and lines are added to the plot.

call curvefit(a, b, function_number **)**

Creates coefficients `_c0`, `_c1`, etc. to construct a fit of `b` vs. `a` when these coefficients are applied to the terms of the function number specified. The function number is:

`0` = Exponential

`1` = Linear

`>2` = Order of polynomial

Also sets the `variable _corr` to correlation factor of the fit.

call deletecolumn(column_name **)**

Deletes the `column column_name`.

call deletemacro(macroname **)**

Deletes the macro `macroname` from the Macros menu. Does nothing if the named macro is not found.

call exechdfmacro(filename, macroname **)**

Executes the macro `macroname` from the HDF file `filename`.

call execmacrofile(filename **)**

Executes a macro from the text file `filename`. The entire contents of the text file are executed as a single macro.

s = getcolumnname(column_number **)**

Returns the column name for column `column_number` in current dataset.

s = getdatasetname(windowname **)**

Returns the dataset name for the window `windowname`.

n = getfilecount( )

Counts the number of files in the current directory. Used in scripts or for processing sequences of files with the loop command.

f = getnthfilename(filenumber **)**

Returns the filename for file number `filenumber` in the current directory. The parameter `filenumber` is the number of the file.

s = getvalidname(datasetname **)** (Power Macintosh only)

Returns a unique name for the given `datasetname`. Useful in macros to avoid conflicts with existing dataset names.

call hide(windowname **)** (Windows only)

Hides `windowname`; same as selecting **Minimize** from the Control menu to the left of the title bar in a window.

call loadhdfvset(filename, reference# **)**

Loads the dataset record with reference number `reference#` from the HDF file `filename`. Used internally for opening HDF files.

call loadhdfnotes(filename, datasetname, reference# **)**

Given filename, a dataset name, and an HDF reference number, read an annotation record and attach it as notebook contents to the given dataset. Used internally for opening notebook records.

call loop(start, stop, increment, macroname **)**

Executes the macro `macroname` as the variable `loop_index` goes from the value start to the value stop in increments of increment. The macro `macroname` is executed (stop-start)/increment+1 times. The variable `loop_index` can be used to reference files to be opened or saved.

call new(text_string **)**

Creates a new blank data window named `text_string`. Same as selecting **New** from the File menu.

call open(filename **)**

Opens the disk file `filename`, opening any datasets and windows stored in the file. Used both for Plot-saved files and imported files. Looks for the file in the current directory. If no file name is given, a dialog prompts for the information.

call paste(windowname **)**

Pastes data from the Clipboard into the data window `windowname`. Same as **Paste** from the Edit menu.

s = prompt(promptstring, defaultstring **)**

Prompts with the text string `promptstring` and waits for text input. Returns the string typed in. If no changes are made, it returns the string `default-string`. Syntax example:

```
s=prompt(“How many rows?”, “5”)
```

`user_interactive` must be set to true prior to using this function and should be set back to false immediately after.

call saveas(windowname, filename, save_type **)**

Saves the window `windowname` to the disk file filename in the current directory. If `user_interactive=false`, it will overwrite any file of the same name without prompting. The parameter `save_type` is:

`=0` Prompts the user

`=1` Saves data, plots, and notebook to HDF file

`=2` Saves data only to HDF file

`=3` Saves data to text file

`=4` Saves plot to HDF file (Windows only)

=5 Saves plot to PICT file (Power Macintosh only)

`=6` Saves plot to TIFF file (Windows only)

`=7` Appends data, notebook to HDF file (Windows only)

`=8` Appends plot to HDF file (Windows only)

=10 Saves plot to EPSF file (Power Macintosh only)

`=13` Appends plot to HDF file, same size as image(s) already in file—useful for animations (Windows only)

=14 Saves plot to Windows bitmap (Windows only)

call savemacro(windowname, macroname **)**

Creates macro `macroname` from the plot window `windowname` and adds it to the Macros menu. Same as **Create Macro...** on the Macros menu.

call setcolumnname(column_number, column_name **)**

Specifies the name `column_name` for column `column_number` in the current dataset. Example: `call setcolumnname (5, “new name”)`

call setdirectory(directoryname **)** (Windows only)

Specifies the directory name and path in `directoryname` for opening files. You can use a full or partial path name with slash separators. Example:
`call setdirectory (“C:\Fortner\Plot”)`

call setexpression(column_name, text_string **)**

Assigns the expression in `text_string` to the column `column_name`.

call setfolder(foldername **)** (Power Macintosh only)

Specifies the folder name and path in `foldername` for opening files. You can use a full or partial path name with slash separators. Example:
`call setfolder (“Harddrive:Fortner:Plot”)`

call setformat(column_name, format_string **)**

Specifies numerical display format for `column_name`. Example:
`call setformat (“opac”, “###.##”)`

call setlabel(column_name, column_label **)**

Changes the plot label of column `column_name` to `column_label`. The label is used for axes and plot titles on newly created plots.

call setlocation(windowname, left, top, right, bottom, hide **)** (Windows only)

Sets the location for windowname. The parameters `left`, `top`, `right`, and `bottom` refer to pixel locations. When `hide=0` window is hidden; when `hide=1` window is brought to front.

call setsavedirectory(directoryname **)** (Windows only)

Specifies the directory name and path in `directoryname` for files saved with `saveas`. You can use a full or partial path name with slash separators. Example: `call setdirectory (“C:\Fortner\Save”)`

call setsavefolder(foldername **)** (Power Macintosh only)

Specifies the folder name and path in `foldername` for files saved with `saveas`. You can use a full or partial path name with slash separators. Example: `call setfolder (“Harddrive:Save”)`

call setselection(datasetname, left, top, right, bottom **)**

Sets the selection region for dataset `datasetname`. The parameters `left`, `top`, `right`, and `bottom` refer to the row and column index numbers of the dataset, starting with 0,0 for the upper left corner.

call setwidth(column_name, column_width **)**

Sets the column width of column `column_name`. The width is set to the number of characters in `column_width`.

call show(windowname **)** (Windows only)

Brings `windowname` to front.

d = var(columnname **)**

Returns a column variable from a column name. Example: `g=max(var("e"))` is equivalent to `g=max(e)`. Useful for applying mathematical functions when only the column name is available.

Macro Variables Reference

Listed here are the reserved variable names for use in Plot macros. These reserved variables are used by the macro subroutine calls, listed earlier in this chapter.

Values for all variables are stored as ASCII strings that may be up to 255 characters in length. When a numeric value is required by Plot, the string is converted into a number. A value of 0 is returned if the field does not start with a number. For variables that require the string "true" or "false", you can use the reserved variables `true` and `false` instead.

Plot Reserved Variables

axis_auto=true

When `=true`, Plot automatically calculates the axis labels, format, and spacing, and overrides `axis_labelspacing`, `axis_min`, `axis_max`, `axis_labelmin`, `axis_labelmax`, and `text_format`. When `=false`, those same parameters are used to calculate the axis labels.

axis_autominmax=true

When`=true`, Plot automatically calculates the minimum and maximum for the attached X,Y pairs.

axis_autoprintformat=true

When `=true`, Plot automatically calculates the printing format for axis labels, and overrides `text_formatdata`. When `=false`, text_formatdata is used to set the printing format.

axis_autospacing=true

When `=true`, Plot automatically calculates the label spacing (increment), and overrides `axis_labelspacing`. When `=false`, `axis_labelspacing` is used to calculate the label spacing.

axis_gridlines=true

When `=true`, Plot draws gridlines on the plot. Gridlines are drawn from each major tick to the opposite side of the plot. When `=false`, no gridlines are drawn.

axis_labelinterval=5

Number of intervals between labels on the axis. Labels are spaced to form `axis_labelinterval` intervals between `axis_labelmin` and `axis_labelmax`.

axis_labelmax=3.14

End of label range, in data or scale units. Labels are drawn starting at `axis_labelmin` and ending at `axis_labelmax`. They may be clipped by `axis_min` and `axis_max`. If `axis_labelmax` is less than `axis_labelmin` the labels decrease in value along the axis.

axis_labelmin=0.0

Start of label range, in data or scale units. Labels are drawn starting at `axis_labelmin` and ending at `axis_labelmax`. They may be clipped by `axis_min` and `axis_max`. If `axis_labelmax` is less than `axis_labelmin` the labels decrease in value along the axis.

axis_labelspacing=0.5

Distance between axis labels, in data units.

axis_logscale=false

When `=true`, the axis created will use logarithmic scaling. When `=false`, the axis will use linear scaling.

axis_majorlength=8

Length of a major tickmark, in points.

axis_majorticks=2

Number of major tickmarks per label interval. If `axis_majorticks=1`, then a major tick is placed at every axis label.

axis_majorwidth=1

Width of a major tickmark, in points.

axis_max=3.14

End of the axis in scale units. No tickmarks and labels appear beyond this value.

axis_min=0.0

Start of the axis in scale units. No tickmarks and labels appear before this value.

axis_minorlength=8

Length of a minor tickmark, in points.

axis_minorticks=2

Number of minor tickmark intervals per major interval. If `axis_minorticks=5`, then four minor ticks are visible between each two major tickmarks. If `axis_minorticks=1`, then each minor tick is obscured by a major tickmark.

axis_minorwidth=1

Width of a minor tickmark, in points.

axis_mirror=false (Windows)

When `=true`, two sets of axes are drawn. The axes are drawn on both sides of the plot when vertical axes are being drawn, or above and below the plot when horizontal axes are being drawn.

axis_tickinout=2

Direction to draw both major and minor tickmarks:

`=1` Inward ticks

`=2` Outward ticks

`=3` Both inward and outward ticks

color_calculateminmax=true

When `=true`, overrides `color_scalemin` and `color_scalemax` and calculates min/max for the color control data column.

color_rgb=“0, 65535, 0"

Color of a plot object, in red/green/blue components. Each component ranges from 0 to 65535.

color_scalemax=30

Maximum value to be used from the color control data column.

color_scalemin=100

Minimum value to be used from the color control data column.

color_table="Rainbow"

Maps values to colors from blue for low values to red for high values. If `color_table="Grayscale"`, shades of gray are used instead.

column_width=9

Specifies the column width, in characters.

currentdataset

Name of the most recently created dataset. Macros often use this to refer to the file that was just opened.

currentfile

Name of the file being opened. Used internally for opening files. It contains the filename during the open process.

currentplot

Name of the most recently created plot. In macros saved with **Create Macro...**, this variable is used to store the name of the new plot.

data_max=3.14 (Power Macintosh only)
data_min=0.0 (Power Macintosh only)

The highest and lowest data values. These are used for converting data values to colors.

file_directory (Windows only)

Name of the current directory for opening files. See the `setdirectory` subroutine for details.

file_folder (Power Macintosh only)

Name of the current folder for opening files. See the `setfolder` subroutine for details.

file_savedirectory (Windows only)

Name of the current directory for saving files. See the `setsavedirectory` subroutine for details.

file_savefolder (Power Macintosh only)

Name of the current folder for saving files. See the `setsavefolder` subroutine for details.

fill_equal=1

Parameter used for defining missing data values:

`=1` if missing data values are equal to `fill_value`

`=2` if missing data values are greater than `fill_value`

`=3` if missing data values are less than `fill_value`

fill_method=1

Parameter for selecting fill missing data method. Used for the `fillmissing()` notebook function:

`=1` for nearest neighbor interpolation

`=3` for linear interpolation

fill_value=-99

Missing data value. Along with `fill_equal`, specifies the range of missing data values (and, therefore, the range of valid data values).

image_canvash=2048 (Windows) or **=640** (Power Macintosh)
image_canvasv=2048 (Windows) or **=480** (Power Macintosh)

Size of the canvas in pixels. The default value is the size of the main display.

image_frame=true (Windows)

When `=true`, the plot is enclosed in a frame.

image_v=300
image_h=300

The plot size, in points.

image_marginleft=0
image_margintop=0

The position of the upper left corner of the plot, relative to the upper left corner of the canvas, in points.

import_3d=true

When `=true`, the file being imported is flagged as a three-dimensional file. When `=false`, the file is flagged as a two-dimensional file.

import_3daxis=1

Parameter used when importing 3D files to specify axis direction. Ignored unless `import_3d=true`:

`=1` for importing slice in XY plane

`=2` for slice in XZ plane

`=3` for slice in YZ plane

import_3dslices="20,22,23"

List of two-dimensional slices to import from a three-dimensional file. Ignored unless `import_3d=true`.

import_byteswap=true

When `=true`, Plot swaps bytes when reading data. See the data import section for more information.

import_coldelim=false

When`=true`, Plot treats every delimiter as a new column. When `=false`, you can have multiple delimiters (spaces, commas, tabs) between columns.

import_colfixedchar="1:4,5:10,11:15"

Specifies the character positions of the columns when `import_delimiter=7` (fixed character positions). Each column is specified by a starting character and ending character position, separated by a colon.

import_coltitles=true

When `=true`, the last line of the header contains the column names (for text column import only).

import_delimiter=3

Specifies how text columns are delimited:

`=0` for columns separated by tabs only

`=1` for columns separated by spaces only

`=2` for columns separated by commas only

`=3` for columns separated by tabs or spaces

`=4` for columns separated by tabs or commas

`=5` for columns separated by spaces or commas

`=6` for columns separated by tabs, spaces, or commas

`=7` for columns in fixed locations specified by `import_colfixedchar`

import_dimcolumns=100
import_dimrows=80
import_dimlayers=70

Specifies the size of the two-dimensional or three-dimensional matrix or column file to be imported. The parameter `import_dimlayers` is ignored unless `import_3d=true`.

import_filetype=1

Parameter specifying the type of import file:

`=0` for file type not specified (the default case)

`=1` for HDF files

`=2` for HDF files saved with Plot

`=3` for HDF VSET files

`=6` for TIFF files (Windows only)

`=7` for FITS files (Windows only)

`=8` for binary matrix files

`=9` for binary column files

`=10` for binary PBM files (Power Macintosh only)

`=12` for MATLAB files (Windows only)

`=14` for GIF files (Windows only)

`=15` for ASCII Text matrix files

`=16` for ASCII Special files

`=17` for ASCII Text matrix files with scale information

`=18` for ASCII column files

import_numtype=1

Parameter specifying the number type in a binary file:

=1 for signed 8-bit byte

=2 for unsigned 8-bit byte

=3 for signed 16-bit integer

=4 for unsigned 16-bit integer

=5 for signed 32-bit integer

=6 for unsigned 32-bit integer

=7 for IEEE 32-bit floating point

=8 for IEEE 64-bit floating point

=9 for VAX/VMS 32-bit floating point

import_record="5,6,7,10"

Specifies which records to read from a multi-record file.

import_skip=0

For text files, specifies the number of text lines to skip before reading data. For binary files, specifies the number of bytes to skip before reading data.

line_dashlength=5 (Power Macintosh only)

Length of dashes in a dashed line, in pixels. Used only when drawing dashed lines.

line_type=1

Specifies what type of line to draw between X,Y pair locations.

=0 No line

=1 Solid line

=2,3 Dotted line

=4,5,6 Dashed line

=7,8 Long dashes

line_width=1

Width of drawn lines, in points. Note that you can set `line_width` to fractional values such as `0.4`, to produce very fine lines when printing to high-resolution output devices.

loop_index

Current loop counter. See the `loop()` subroutine call for details.

offset_h=0
offset_v=0

Adjustment distance for axis labels, text objects, and plot number symbols, in points. Used for manually adjusting the positioning of objects in the plot.

plot_colorcolumn="colorcol"

Specifies the name of the column to use when `plot_colorcontrol=true`.

plot_colorcontrol=false

When `=true`, data from `plot_colorcolumn` is used to set the color of each line segment or symbol in the line. The color depends on the data value and the current `color_scalemin`, `color_scalemax` range. Used for color scatterplots. When `=false`, all lines and symbols are drawn in the same color.

plot_errorcolumn="errcol"

Specifies the name of the column to use for error bars when `plot_errortype=4`.

plot_errordir=1

When drawing error bars, specifies which direction(s) to draw the error bars. The directions drawn are indicated by the sum of any combination of the following values.

`=1` Positive X

`=2` Negative X

`=4` Positive Y

`=8` Negative Y

plot_errorfixed=5.0

When drawing error bars and `plot_errortype=2`, indicates what fixed length (in data units) to use for the error bar length.

plot_errorpercent=8.0

When drawing error bars and `plot_errortype=3`, indicates what percentage of the data value to use for the error bar.

plot_errorstd=1.0

When drawing error bars and `plot_errortype=1`, indicates how many standard deviations to use for the error bar length.

plot_errortype=1

When `symbol_type=4`, specify the method to calculate the length error bars.

=1 Standard deviation

=2 Fixed error

=3 Percent of value

=4 From data specified by `plot_errorcolumn`

plot_maxpoints=100000

For a given set of X,Y pairs, limits the number of symbols displayed. If `plot_maxpoints` is greater than the total number of points available, all points are drawn. If `plot_maxpoints` is less than the total, points are skipped while drawing.

plot_pencolumn="pencol"

Specifies the name of the column to use for `plot_pencontrol`.

plot_pencontrol=false

When `=true`, `plot_pencolumn` is used as a series of pen control commands. When the data value is > 0.0, the pen draws a line; when the data value is <= 0.0, the pen moves without drawing. When `=false`, no pen control is active.

pref_columnwidth=9

Specifies the column width, in characters, to use as a default. New datasets and all empty columns have the default column width.

pref_maxnumber=INF (Power Macintosh only)
pref_minnumber=-INF (Power Macintosh only)

Specifies the largest and smallest legal numbers. All values outside this range are assumed invalid, so they are left out of max/min calculations, etc. Note: `INF` represents infinity.

pref_printoutline=true (Windows only)

When `=true` an outline of printable area of the page is displayed, when `=false` no page outline is displayed.

pref_startwithnew=true (Windows only) or **=1** (Power Macintosh only)

When `=true` (Windows) or `=1` (Power Macintosh) a new data window will be created when Plot is started, when `=false` (Windows) or `=0` (Power Macintosh) no new window is created at startup.

pref_textface=0 (Windows only)

Specifies default font style. Valid codes are:

`=0` normal

`=1` bold

`=2` italic

`=4` underline

These codes may be added together to achieve the desired style; for example, `pref_textfont=7` specifies a bold, italic, and underlined version of the font named by `pref_textfont`.

pref_textfont="Courier New"

Default font for new windows.

pref_textformatdata="F8.1" (Windows only)

Specifies the printing format used for the data values in all new data windows. Also found in the Preferences dialog.

pref_textsize=12

Default font size for new windows.

pref_windowh=200

Width of new windows in pixels.

pref_windowv=200

Height of new windows in pixels.

print_cropmarks=true (Windows only)

When `=true`, places cropmarks on multi-page printouts for alignment assistance.

print_marginleft=1 (Windows only)
print_marginright=1 (Windows only)
print_margintop=1 (Windows only)
print_marginbottom=1 (Windows only)

Specify page margins, in inches, as measured from the border of the printable area of a page.

print_pageheaderleft=" " (Windows only)
print_pageheaderright="&n" (Windows only)
print_pagefooterleft="&d" (Windows only)
print_pagefooterright="Page &p of &P" (Windows only)

Parameter specifying text page headers and page footers to be printed. The following special symbols will be expanded as each page is printed:

`&d`	date MMM DD YYYY
`&D`	date DD MMM YYYY
`&n or &N`	dataset name
`&p`	page number
`&P`	total number of pages
`&t or &T`	the total number of pages
`&w or &W`	name of the window
`&&`	&

symbol_char=65

When `symbol_type=2`, specifies which character to use for the symbol. The number is the ASCII value for the character to plot.

symbol_column=“columnname”

Specifies the name of the column of data to use for drawing number-scatter plots, where `symbol_type=3`.

symbol_drawn=1

When `symbol_type=1`, specifies the symbol to be drawn:

`=0` No symbol

`=1` Square

`=2` Plus

`=3` X

`=4` Dot (square)

`=5` Square with center dot

`=6` Circle

`=7` Circle with center dot

`=8` Filled square

`=9` Filled circle

symbol_size=8

Size of plot symbol in points (or fractions of a point). This is the size of caps on error bars and the size of drawn symbols.

symbol_type=1

Specifies how to draw the symbol at each X,Y pair location.

`=0` No symbol

`=1` Draw a symbol according to `symbol_drawn`

`=2` Draw a text character according to `symbol_char`

`=3` Draw a number from `symbol_column` at each point

`=4` Draw error bars

text_formatdata="#####.#"

Specifies the number printing format. Used for numbers printed as axis labels.

user_interactive=true

When `=true`, dialogs are allowed. When `=false`, no dialogs are displayed, and the keywords defined here are used to create the plots. The parameter `user_interactive` is usually set to `false` for complex macros.

Literals

Literals are terms that have a special meaning in Plot. Only use literals as variables in macro expressions.

true, false

These values are used to set other macro expressions to true or false.

INF

The value for infinity.

NAN

Not a number.

Chapter 16:

Printing

Plot provides extensive support for printing dataset windows, notebook windows, and plot windows. Plot windows are printed just as they are displayed on the screen.

This chapter describes the print commands and windows associated with the Windows and Macintosh platforms.

Printing in Plot for Windows

This section describes how to set up your printer, the options available in the Print dialog, page setup, the Print command, and details for printing each window type.

Printer Setup

Before printing, you must select and configure a printer from the list of printers installed on your Windows system.

To do so, select the **Printer Setup...** command from the File menu. You will see a dialog similar to the one shown below.

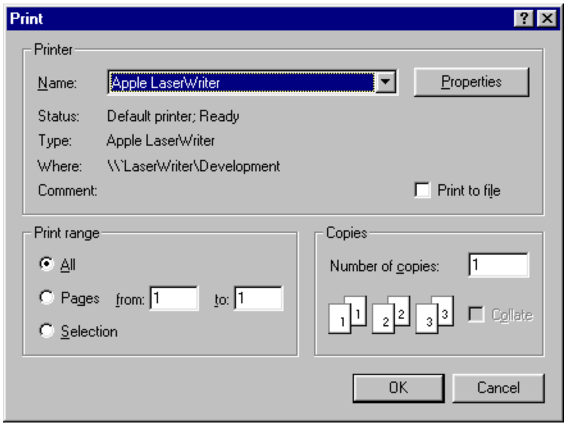


Figure 16-1: Print Dialog

Print Range

The buttons under Print Range let you choose to print all pages, to limit output to a range of pages or, for dataset and notebooks, to the selection region.

Other options in the Print dialog will vary by printer.

The first time that you open the Print dialog while using Plot, the default settings for the default system printer will appear. To change printers or paper attributes, choose **Properties....** You will see a Print Setup dialog similar to the one shown in the next figure.

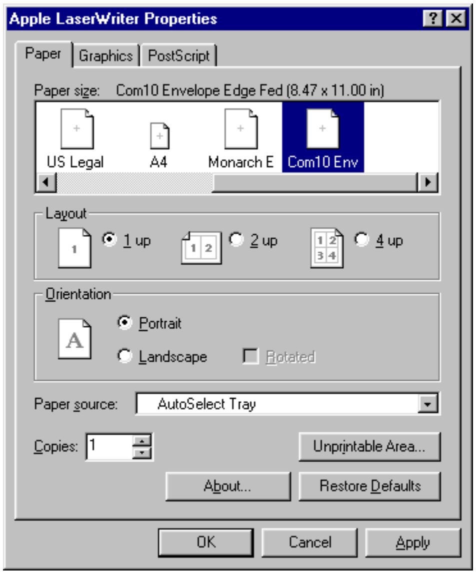


Figure 16-2: Print Setup Dialog

Printer

The currently selected printer is displayed in the Specific Printer field. To select a different printer, open the printer list and select one of the other printers configured on your system. (If this list is empty or does not include the printer you wish to use, install or configure the printer using the Printers program in the Windows Control Panel; please refer to the your Windows documentation for additional information.)

The remainder of the Print Setup dialog is tailored to the currently selected printer and to your version of Windows. Therefore, its content may vary considerably from that shown above. Additional dialogs may be available to you to specify more advanced printer settings, such as the scaling percentage, or to select network-connected printers.

Page Setup

When you print a Plot window, it will be paginated if necessary. The Page Setup dialog, available from the **Page Setup...** command on the File menu, lets you select page margins and optional header and footer text.

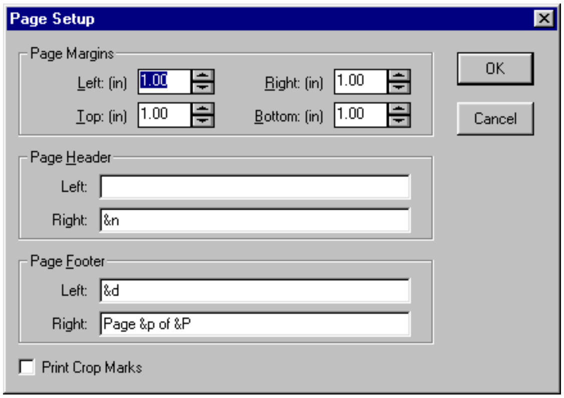


Figure 16-3: Page Setup Dialog

Page Margins

Use the Page Margin fields to control the size and placement of text and graphics on the printed page. Values are specified in inches, and are relative to the printable area of the page as reported by the printer driver. Therefore, if you have exact page placement requirements, you will need to decrease your margins to compensate for the unprintable region.

Page Headers and Footers

Plot also allows you to specify header and footer text to appear at the top and bottom of each page of your output. Vertically, headers and footers are placed just outside the margin; they are suppressed if the corresponding vertical margin is less than the height of the selected font. Horizontally, they are placed just inside the margin and are justified to the outside.

The header and footer text may be set to any text or phrase of your choosing, or may be empty. Plot allows you to specify the following special tokens that will be replaced as each page is printed:

Token...	**Replacement...**
&d	date MMM DD YYYY
&D	date DD MMM YYYY
&n or &N	dataset name
&p	page number
&P	total number of pages
&t or &T	the total number of pages
&w or &W	name of the window
&&	&

Use of any other character following an & is reserved for future use.

Print Crop Marks

When printing an oversized plot in sections, on more than one page, check this box to print crop marks on each page. Crop marks can help you assemble the pages to make a complete plot.

The Print Command

Selecting **Print** from the File menu prints the contents of the active window. The details of printing the various window types are given in the next section. While the window is being printed, a Print Progress dialog will be visible and will show progress on a page-by-page basis.

Interrupting or Aborting a Print

To abort the printing process, click on **Cancel** in this dialog. It may take a moment for the abort signal to terminate processing, since it is checked at the beginning and ending of each page.

(If the output was queued to Windows Print Manager (the default), it may be aborted or cancelled using Print Manager; see the Print Manager documentation that came with your Windows system.)

Printing Windows

Here we give specific details on printing each Plot window type.

Printing Data and Notebook Windows

The page size and margin settings from the **Page Setup...** command determine how many rows and columns of text will fit on each page. Plot then prints the numbers from top to bottom then left to right on each page. Row numbers and column names are printed on each page so that multiple pages may be aligned.

Note

It is strongly recommended that you select a fixed width font, such as ‘Courier New’, so that decimal points line up properly.

Unless you specified to print only a selected region or range of pages, Plot prints all numbers in the data window, not just the visible window or the selection region. This can easily add up to many pages, so double-check the size of your data window before printing.

The notebook window prints with line breaks set up to match the screen appearance. The contents of the notebook are paginated, and the pages are numbered. The font and size match the font and size set up for the window on the screen.

Printing Plot Windows

Plot windows are printed actual size in the same position relative to the paper as they are placed relative to the canvas. The entire plot is printed with axis labels and any text annotations that have been added.

Printing in Plot for Power Macintosh

Here we give an overview of printing from Plot for Power Macintosh to LaserWriters. Although printing to other printers is usually very similar, you may need to refer to your printer manual for assistance.

Page Setup

To set up your printer, select the **Page Setup...** command from the File menu. If you are using the LaserWriter 8 driver, you will see the dialog shown in Figure 16-4. If you are using an earlier version of the LaserWriter driver, or a different printer entirely, your dialog will be different.

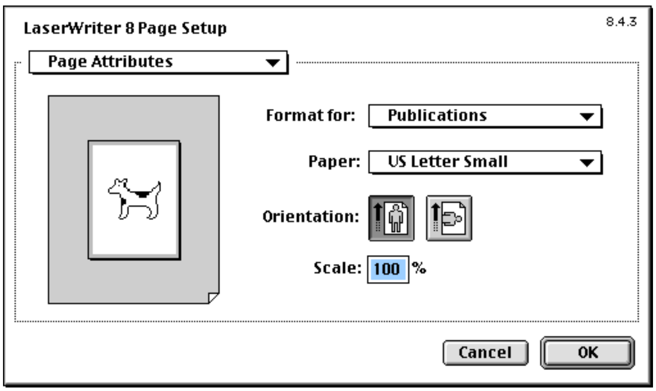


Figure 16-4: Page Setup...Dialog

Use this dialog to specify paper size, layout, and orientation. The Reduce or Enlarge box lets you specify the print size as a percentage of the size of the plot you see on your screen.

Print

Selecting **Print...** from the File menu prints the contents of the frontmost window. The details of printing the various window types are given in the next section.

If you are using Apple's LaserWriter 8 driver, you will see the dialog in Figure 16-5. The version number printed in the upper right corner of the dialog must be 6.0 or greater for grayscale printing. If you are not using a LaserWriter, your dialog may be different.

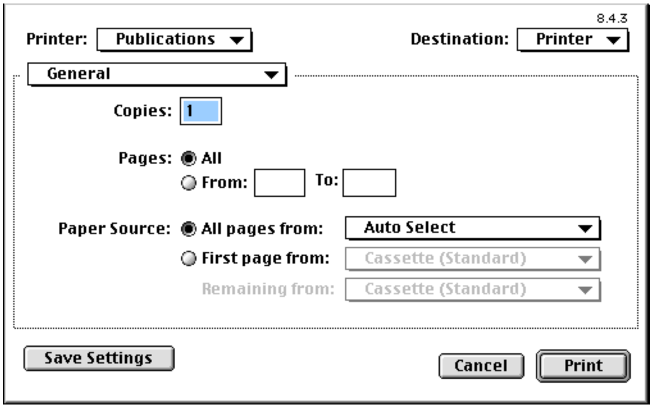


Figure 16-5: Print Dialog

Interrupting a Print

Command-period halts printing at the end of the page in process.

Choosing the Printing Font

The font for printing matches the font and size used in the window.

Positioning the Image on the Page

The plot canvas corresponds to the paper the plot is printed on. New plots always start out in the upper left corner of the canvas, so on hard copy they print in the upper left corner of the page. To position the plot on the canvas, click on it and drag the plot to the desired position.

If your monitor has 72 dots per inch (typical for Macintosh monitors), your printed image will be exactly the same size as the screen image, assuming that you have not changed the percentage enlargement in the Page Setup dialog. Plot always assumes that every pixel is 1/72 of an inch in size.

Setting Line Width

All plot types support the `line_width` setting to refine line thickness on printouts. Before you create your plot, set the reserved variable `line_width` to the desired line thickness, in fractions of a point. We have seen good results setting the thickness to 0.4 points.

Printing Data and Notebook Windows

The page size, orientation and percent reduction settings from the **Page Setup...** command determine how many rows and columns of text will fit on each page. Plot then prints the numbers from left to right, then top to bottom on each page. The pages are printed with the page number, and a borderline along the top and left of the data columns. The independent column names are printed on each page so that multiple pages may be aligned.

The **Print...** command prints the entire dataset contents, not just the visible part of the window or selection region. This can easily add up to many pages, so double-check the size of your dataset before printing.

To print only the contents of the selection region, first create a new data window using the **New** command from the File menu. Copy the desired selection region and paste it into the blank, new data window, then print from the new window.

The notebook window prints with line breaks set up to match the screen appearance. The contents of the notebook are paginated and the pages are numbered. The font and size match the font and size set up for the window on the screen.

Printing Plot Windows

All plot types are printed actual size in the same position relative to the paper as they are placed relative to the canvas. The entire plot is printed with axis labels and any text annotations that have been added.

Printing Color or Grayscale Plots

It is best to print color plots on a color printer, but black and white printers such as the LaserWriter can produce reasonable grayscale printouts of color plots.

If color or grayscale has been used in a plot, be sure that the color/grayscale is enabled for your printer drivers. For the LaserWriter 8 driver, Color/Grayscale is found under the Options dialog. For other printers, consult your manual.

Chapter 17:

Data Exchange and File Export

Plot provides four commands under the Edit menu that allow you to exchange data using the clipboard. The commands are **Copy, Copy As...**(Windows)**, Paste, and Paste As...**(Windows). The **Save As...** and **Save** commands let you save your plots and data in any of several formats. This chapter describes these commands.

Copy Commands

This section describes the **Copy** command and **Copy As...** (Windows) commands.

The Copy Command

The **Copy** command makes choices about what to copy, and about copy format, according to the context in which it is selected.

Using Copy in Dataset and Notebook Windows

When selected from a dataset window, the **Copy** command places whatever is currently selected on the clipboard as text. Data in the selection region is copied as lines of numbers separated by tabs. This is suitable for pasting into spreadsheet programs. In addition, text on the clipboard in this format may be pasted into another Plot dataset window.

When selected from a notebook window, the **Copy** command places the currently selected text on the clipboard. The text can be pasted into other applications, or into other Plot notebook windows.

Using Copy in Plot Windows

When selected from a plot, the **Copy** command places the plot on the clipboard as a bitmap/PICT file. This is suitable for pasting into presentation graphics programs. In Plot for Power Macintosh, hold down the Option key and select **Copy** to copy just the plot without axes sand titles. This can be useful for creating overlays.

The Copy As... Command (Windows only)

Copy As... allows more flexibility than the **Copy** command by opening a dialog with options. When selected from a dataset window, the **Copy As...** command produces the dialog shown in Figure 17-1.

This dialog always consists of two lists. The left list displays the choice of objects that may be copied. The right list contains the choices of what formats to use when storing the information on the clipboard.

The contents of the lists vary depending on what type of window is active, and the contents of that window.

Using Copy As... from Dataset Window

From a dataset window, the selection region is copied as text, just as it would be using the **Copy** command.

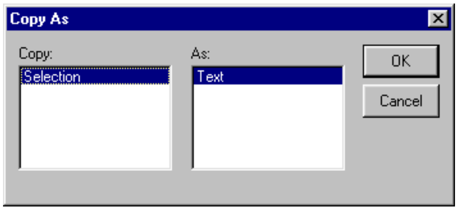


Figure 17-1: Using Copy As... from a Dataset Window

Note

From a notebook window, **Copy As...** works identically to **Copy**; the selection region is simply copied as text.

Using Copy As... from Plot Windows

Figure 17-2 shows the Copy As dialog when invoked from a plot.

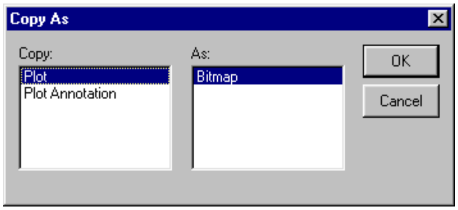


Figure 17-2: Copy As Dialog

To copy a plot, including any annotations, as a bitmap, select **Plot** from the Copy list. To copy only an annotation as text, select **Plot Annotation** from the Copy list.

Paste Commands

This section describes the **Paste** and **Paste As...** (Windows) commands.

The Paste Command

The **Paste** command, like the **Copy** command, is simple and allows no options. In fact, if the clipboard does not contain a format which is valid for the active window, the **Paste** command is grayed.

Using Paste in Data and Notebook Windows

For data windows, the **Paste** command requires a text object on the clipboard. The text must be text numbers, separated by tabs or spaces. Each row must be separated by a carriage return and linefeed.

The numbers on the clipboard are pasted into the dataset starting at the upper left corner of the selection region. If the clipboard is in the correct format, the numbers in the dataset are replaced by the numbers on the clipboard.

The **Paste** command is not supported in plot windows.

The Paste As... Command (Windows only)

The **Paste As...** command allows more flexibility in pasting objects from the clipboard. The Windows clipboard is capable of containing more than one kind of object at a time. Using the **Paste As...** command, it is possible to choose which item to paste and how to paste it.

Bitmap objects can only be pasted into applications other than Plot. Text objects, however, can be pasted into any plot window, as well as other applications.

In plot windows, a text object can be pasted as a new annotation, or as a replacement for a selected annotation. In dataset windows, text is pasted as new data.

When plot annotations that were copied as text are pasted into notebook windows, the formatting codes are also pasted. Text copied from dataset or other notebook windows is pasted into notebook windows simply as text.

Paste As Dialog

Shown below is the Paste As dialog. This dialog always displays two lists, described below.

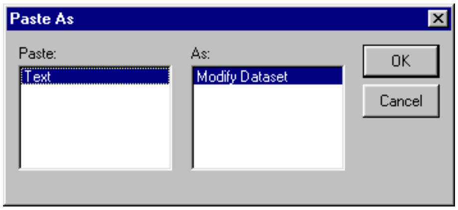


Figure 17-3: Paste As Dialog

Paste

This list shows the type of objects, text or bitmap, that is available on the clipboard for pasting.

As

This list shows the ways in which the object might be pasted into Plot, depending on the type of object under Paste, as well as on the type of the active window. Only formats appropriate for the active window are shown.

Exporting Files

This section describes how to export files using the **Save As...** command. Under the File menu, the **Save As...** command brings up the dialog shown below. The drop-down menu in the lower left corner of the dialog allows you to choose the file format to be used. The formats available depend on the active window type.

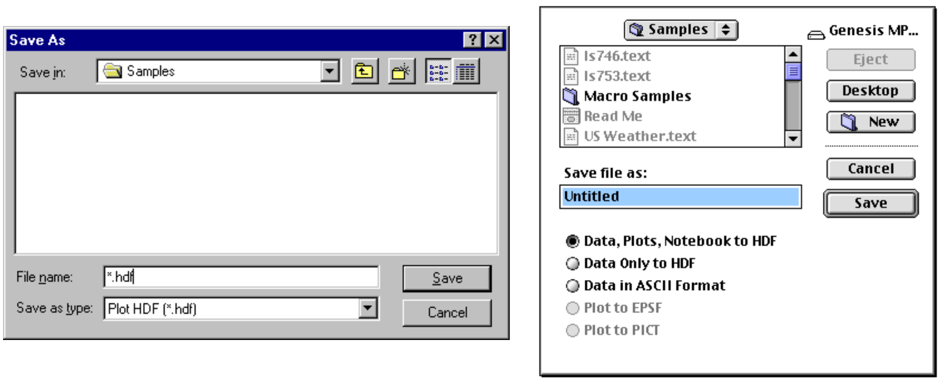


Figure 17-4: Save As Dialog (Windows and Power Macintosh)

Save As... for Data Windows

The following file formats are available when **Save As...** is chosen from a data window:

Plot HDF (*.hdf)/Data, Plots, Notebook to HDF

This option is available for both data and plot windows. It saves all your work as a complete HDF file. The data window, the notebook window, and every plot window associated with the dataset are stored in their own records in the HDF file. The HDF Vset standard is used to save and load column data, where one Vgroup is supported with a separate Vdata field for each column of numbers.

When you reopen and work with a file that has been saved under this option, you can simply select the **Save** command to save a complete file with no dialog prompts.

When saving files in this format you should use the “*.hdf” extension.

HDF Data (*.hdf)/Data Only to HDF

This option saves just the data window as an HDF Vset. This is especially useful for exporting datasets to other programs. When saving files in this format you should use the “*.hdf” extension.

ASCII Data (*.txt)/Data in ASCII Format

This option saves the contents of the data window as an ASCII text file. Each row is output on one line with each number separated by a tab. The numbers are saved in the same format as used in the data window. The first line contains the column headers, and each line starts with the row header for that row. This format is useful for exporting the data to spreadsheets or word processors.

Save As...From Plot Windows

The following options are available in the Save As dialog when the command is chosen from a plot window:

Plot HDF (*.hdf)/Data, Plots, Notebook to HDF

Same as described above for dataset windows.

HDF Image (*.hdf) (Windows only)

With this option, Plot creates an 8-bit raster image which contains the contents of the current window and saves it as an HDF file.

App. HDF Image (*.hdf) (Windows only)

This is the same as the HDF Image file described above, except that the image is appended to any existing HDF file.

TIFF Image (*.tif) (Windows only)

With this option, Plot creates an 8-bit raster image which contains the contents of the current window and saves it into a TIFF (Tagged Image File Format) file, one of the most commonly used standard formats for image data.

Bitmap (*.bmp) (Windows only)

With this option, Plot creates an 8-bit raster image which contains the contents of the current window and saves it into a Windows bitmap file. This file is a Device Independent Bitmap (BMP), which is appropriate for import into most graphics applications.

Plot to EPSF (Power Macintosh only)

This option saves the plot, with axes and all text annotations, to EPSF (also known as EPS) format. Briefly, EPSF is a format suited for storing single-page graphics that is commonly supported by presentation and page layout programs. It is a subset of the PostScript language.

Plot to PICT (Power Macintosh only)

This option saves the plot in a PICT file as one combined drawing. Plot embeds PostScript where necessary for highest possible print quality.

Part IV: Appendices

Appendix A:
Plot Menus

This appendix reviews each of the menus and commands that appear in Plot's menu bar.

File Menu

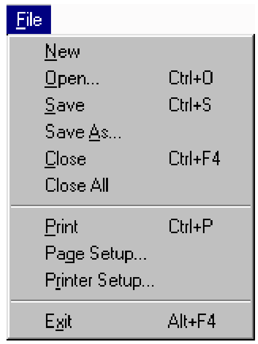


Figure A-1: File Menu

The **New** command creates a new, empty data window.

The **Open...** command accesses a file from disk. Use it to load datasets which were previously saved from Plot or to import data from a variety of file formats.

The **Save** command saves the current dataset, notebook, and any Plot windows.

The **Save As...** command saves a copy of the current dataset with the file name of your choice. Several export types are available.

The **Close** command closes the current window and removes it from the screen. When you close the dataset window, the data is removed from memory, and all dependent windows are also removed.

The **Close All** command closes all datasets and removes all windows from the screen.

The **Print** command prints the current window in a manner consistent with parameters set using **Page Setup...** and **Printer Setup...**.

The **Page Setup...** command opens a dialog that allows you to set page margins, and add header and footer text.

The **Printer Setup...** command (Windows) opens a dialog that allows you to select and configure a printer from the list of printers installed on your Windows system.

The **Exit/Quit** command closes all of the windows and exits the program.

Edit Menu

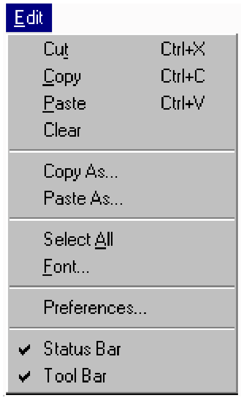


Figure A-2: Edit Menu

The **Cut** command removes the currently selected text and copies it to the clipboard.

The **Copy** command copies a bitmap representation of the current plot, or copies currently selected data as text, to the clipboard.

The **Paste** command adds the contents of the current clipboard to the current window wherever appropriate.

The **Clear** command removes the currently selected text from the currently selected notebook window.

The **Copy As...** command (Windows) opens a dialog that lets you select which components in a window to copy. When appropriate, this dialog also allows you to choose what format to copy to.

The **Paste As...** command (Windows) opens a dialog which lets you choose what to paste from the clipboard. When appropriate, this dialog also lets you choose a format for the pasted component.

The **Select All** command selects everything in the current window.

The **Synchronize** command (Power Macintosh) links a plot window with its corresponding windows so that when data points are selected in one window, the corresponding points will be selected in the other(s).

The **Font...** command (Windows) opens a dialog that allows you to select font and size for text in dataset and notebook windows, and for text annotations and axes labels in plot windows.

The **Preferences...** command (Windows) opens a dialog that allows you to edit the plot session settings and defaults, or restore factory settings. For descriptions of macro commands, see the Macro Variables Reference section in Chapter 15.

The **Status Bar** command (Windows) lets you display or hide the status bar. When enabled, the Status Bar appears along the bottom of the screen, and a checkmark appears in the menu next to the command.

The **Tool Bar** command (Windows) lets you display or hide the tool bar. When enabled, the tool bar appears below the menu bar, and a checkmark appears in the menu next to the command.

Data Menu

The Data menu allows you to navigate and manipulate the contents of a data window and perform notebook calculations.

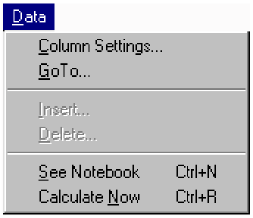


Figure A-3: Data Menu

The **Column Settings...** command opens a dialog that lets you set parameters for the currently selected column in the data window.

The **GoTo...** command opens a dialog that lets you specify and move to any cell in the data window.

The **Insert...** command lets you insert one or more blank columns or rows into the data window.

The **Delete...** command removes selected columns or rows from the data window.

The **See Notebook** command (Windows) opens the notebook for the current dataset.

The **Calculate Now** command evaluates the currently selected expression in the notebook, or evaluates the expression fields for all columns in the data window.

Graph Menu

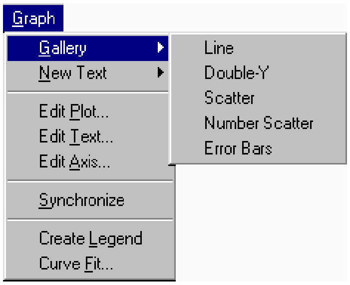


Figure A-4: Graph Menu

The **Gallery** command opens a submenu that lets you select and create one of five plot types from the current data window.

The **New Text** command in Windows opens a submenu that lets you select the orientation for new text. In Power Macintosh, selecting this command opens a dialog that lets you enter and edit text labels.

The **Edit Plot...** command opens a dialog that provides access to all plot parameters.

The **Edit Text...** command opens a dialog that lets you edit currently selected text object. It displays all text formatting codes used in the currently selected text.

The **Edit Axis...** command opens up a dialog that lets you edit parameters for the currently selected plot axis.

The **Synchronize** command (Windows) links a data window with corresponding plot windows so that when data points are selected in one window, the corresponding points will be selected in the other(s).

The **Create Legend** command generates a legend displaying the names of plotted columns, along with a sample of the attributes used to plot the X,Y data pair.

The **Curve Fit...** command opens a dialog that allows you to choose parameters for generating a curve fit data column to an X,Y data pair.

Format Menu (Power Macintosh)

The Format menu lets you choose fonts and sizes for currently selected text. It also includes the Style submenu, which lets you apply listed styles to selected text.

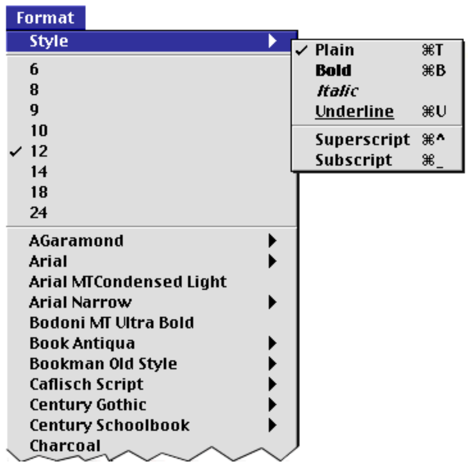


Figure A-5: Format Menu

Macros Menu

Figure A-6: Macros Menu

The **Create Macro...** command prompts for a macro name and saves a macro based on the characteristics of the current window.

The **Edit Macros...** command opens a dialog that allows you to create a new macro, or edit, rename, delete, import, and export existing macros.

Additional items on the Macros menu represent the list of currently saved macros. Selecting one of them executes the stored macro.

Some example macros, MakeExampleData and MakeExamplePlot, are provided to demonstrate how macros work.

Windows Menu

The Windows menu is used to arrange and switch between windows.

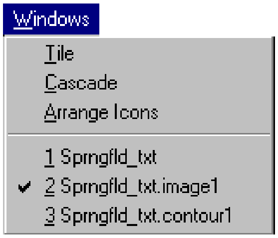


Figure A-7: Windows menu

The **Hide Window/Show Window** command (Power Macintosh) is used to hide and re-display windows.

The **Tile** command (Windows) resizes and arranges the open windows side by side.

The **Cascade** command (Windows) resizes and arranges the open windows so that each title bar is visible.

The **Arrange Icons** command (Windows) aligns all icons in a row near the bottom of the window.

Below the divider, the Windows menu also lists all windows that are currently open, including those that have been iconified in Windows. In Power Macintosh, hidden windows and the **untitled.notebook** are also displayed in this menu in italicized type.

Help Menu

Figure A-8: Help Menu

Windows

The **Macro Language Reference...** starts the on-line reference to the macro language.

The **About Plot...** command gives information on Plot, including the version number and copyright information (Note: in Power Macintosh, this command is available from the Apple menu.

Power Macintosh

The Help menu provides information on how to use Balloon Help and lets you turn this feature on and off.

Appendix B:

Startup Macros and Preferences

Plots default settings can be changed by creating a macro named `Startup_Macro`. This macro will automatically be executed each time Plot is started. The startup macro lets you set variables, perform calculations and use the macro subroutines just as you would in any other macro. See Chapters 14 and 15 for more information on creating macros.

Preferences, available only in Plot for Windows, are a collection of settings that specify the default behavior of Plot.

Creating a Startup Macro

To create a startup macro in Windows, select **Edit Macro** from the Macros menu. In the Edit Macros dialog, choose **New.** Type `Startup_Macro` as the name of the macro and choose **OK**. In the blank window that appears, enter any of the macro expressions described in Chapter 15. To save the macro, choose **OK**.

To create a startup macro in Power Macintosh, enter it into an empty notebook, then select **Create Macro** from the Macros menu. Name it `Startup_Macro`. The Macros menu will be updated to include the newly created macro.

Example 1: Plot Size

This `Startup_Macro` sets the initial plot size at 400 x 400 points.

```
image_h = 400
image_v = 400
```

Example 2: Setting Data Window Attributes

This `Startup_Macro` sets initial characteristics for data windows. It sets type size to 10 points, font to Arial, and column width to six characters.

```
pref_textsize=10
pref_textfont="Arial"
pref_columnwidth=6
```

Example 3: Setting Default Print Options (Windows)

This example sets initial printer and print options. It sets the top page margin, and disables page outline printing.

```
print_margintop=.5
print_marginbottom=.5
print_marginleft=.5
print_marginright=.5
pref_printoutline=0
```

Example 4: Setting Default Directories (Windows)

This macro sets initial directories for opening and saving files.

```
call setdirectory("C:\Plot\rawdata")
call setsavedirectory("C:\Plot\dataout")
```

Preferences... Command (Windows)

The **Preferences...** command can be used to set variables that determine the initial appearance of new plots and windows. Advanced users can set additional variables by creating a special macro, named `Startup_Macro`, which is executed automatically each time Plot is started.

To open the Preferences Settings dialog, select **Preferences...** from the Edit menu.

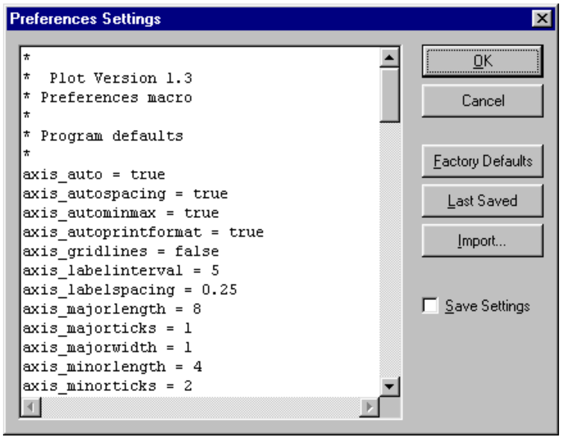


Figure B-1: Preferences Settings Dialog

This dialog consists of a text editing area which lists preference settings. When the dialog first appears, the values shown reflect the current state of the preference settings in Plot. Note that these settings may change during your use of the program.

Using the text editing area and the various options in this dialog, you may modify the current settings within the program and, optionally, save those settings to disk for future sessions. The options in this dialog are discussed below.

Factory Defaults

Choose **Factory Defaults** to restore all preferences to the default settings that were shipped with Plot. Remember to mark the **Save Settings** checkbox if you want to use the factory settings for future sessions.

Last Saved

Click this button to reset all the current settings to the last settings you saved.

Import

Click this button to bring up the Import Preference dialog.

If this box is checked at the time you click **OK**, the settings shown in the dialog will be saved on disk, and will be active the next time you use Plot as well.

Appendix C:

About the HDF Libraries

HDF is an extensible, binary, public domain file format specification for storing data and images. All Fortner Software products use HDF as their primary data storage format. HDF files can store floating point data, scaling information, color images, text, and other items. HDF originated at the National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign, where it was developed as a solution to the problem of sharing data among all of their different computers.

NCSA maintains and distributes a public domain software library for reading and writing HDF format files. The libraries are available on a variety of computers including Macintosh, Windows, Sun, VAX, Silicon Graphics, and Cray UNICOS. The software is written in C with both Fortran and C interfaces supported for making calls to the HDF libraries.

Source code and documentation for the HDF storage routines is available from the National Center for Supercomputing Applications (NCSA) at the University of Illinois. The libraries and documentation can be obtained from NCSA's HDF home page at http://hdf.ncsa.uiuc.edu.

Scientific Data Groups and Vsets

The most important HDF storage format supported by Plot is the Vertex Set (Vset) format. Columns of data are stored in VSETs where columns may represent x-location, y-location, density, temperature, etc. Plot reads and writes column data stored as VSET groups, also known as Vgroups. You may use the HDF Vset calling interface to write column data directly into HDF files to be read by Plot.

HDF is most commonly used for storing N-dimensional arrays of scientific data. The HDF storage type for arrays is the Scientific Data Group, or SDG. When Plot reads two-dimensional arrays stored in SDGs, each column in the array becomes a column in the data window. Plot cannot write SDGs.

In addition, Plot reads 2-dimensional arrays from SDG type data. In Plot for Windows, each column in the array becomes a column in the data window.

HDF Web Sites

Additional information about HDF and its application can be found on the World Wide Web at the following web sites:

- HDFinfo.com: A clearinghouse of HDF format information
  http://www.hdfinfo.com/
- NCSA's web page
  http://hdf.ncsa.uiuc.edu
- NASA Goddard Space Flight Center (GSFC) page on HDF
  http://daac.gsfc.nasa.gov/REFERENCE_DOCS/HDF/gdaac_hdf.html
- Earth Science Enterprise (Mission to Planet Earth) web page:
  http://www.hq.nasa.gov/office/mtpe/
- EOS Data Resources web page
  http://eospso.gsfc.nasa.gov/eos_homepage/

Other Data Web Sites

The following web sites let you access data other than HDF to be downloaded.

- TOMS (zipped):
  http://daac.gsfc.nasa.gov/CAMPAIGN_DOCS/FTP_SITE/readmes/toms_daily.html
- GOME:
  http://daac.gsfc.nasa.gov/CAMPAIGN_DOCS/FTP_SITE/readmes/gome_daily.html

- CZCS:
  http://daac.gsfc.nasa.gov/CAMPAIGN_DOCS/OCDST/czcs_readme.html
- SeaWiFS:
  http://daac.gsfc.nasa.gov/CAMPAIGN_DOCS/OCDST/OB_main.html
- SSMI:
  http://daac.gsfc.nasa.gov/CAMPAIGN_DOCS/hydrology/readme_html/ssmi_monthly_readme.html
- TRMM:
  http://lake.nascom.nasa.gov/DATA/TRMM/
- NSCAT:
  ftp://podaac.jpl.nasa.gov/pub/ocean_wind/nscat/
- DTED:
  http://164.214.2.59/geospatial/products/DTED/dted.html
  http://164.214.2.54/mel/data.html
- SDTS:
  http://edcwww.cr.usgs.gov/doc/edchome/ndcdb/ndcdb.html

Note that the SeaWiFS and TRMM sites require you to order the data for a fee. With the exception of the DTED and SDTS web sites, all data is stored in HDF. The Noesys technical data processing application supports import of DTED and SDTS to HDF. For more information on the data products that Noesys supports, see the Fortner Software web site at http://www.fortner.com.

Appendix D:
AppleEvents

In this appendix we describe Plot's support for AppleEvents, which may be used by AppleScript, Fortran, HyperCard and other programs to send commands to Plot. You must be running System 7 to use this feature.

Using AppleEvents to Control Plot

AppleEvents is a Macintosh System 7 feature which allows applications to exchange data and commands. AppleEvents generated by scripting systems and programming languages may be used to send macro commands to Plot to be executed. Since most program features are available through macros, these macro commands can manipulate data and create any plots which are available from the menus.

Supported Events

Plot supports the required AppleEvents: Open Application, Open Document, Print Document, and Quit Application. These give external programs the ability to make Plot open specific data files or even voluntarily quit. The additional two events which Plot supports are DoScript and Evaluate. These make it possible to send macro commands to Plot and retrieve variable values.

Note

Required events are in the event class ‘aevt’ and are of type ‘oapp’, ‘odoc’, ‘pdoc’, and ‘quit’. The other two events are in class ‘misc’ and are of type ‘dosc’ and ‘eval’. DoScript accepts a handle to a script of commands to be executed as notebook commands. Each command is separated by a hard return character. Evaluate accepts a variable name and returns a string with the value of that variable.

HyperCard Example

Claris HyperCard (version 2.1 and later) supports the send command, which lets you send and execute Plot commands. For example, the following send command will create a column of 50 numbers, assuming that Plot is already running and has a data window open. Internally HyperCard generates a DoScript AppleEvent.

```
send "x = series(50)" to program "Plot"
```

Note

If you do not have a name assigned to your Macintosh under System 7, the send command will not be able to find Plot. Use the Sharing Setup control panel to assign a name to your Macintosh.

To retrieve results and variable values from Plot, HyperCard supports the request command. For example, the following commands perform a simple calculation and retrieve the result from Plot. Note that any variable value from Plot's variable storage may be requested. For the request command, HyperCard generates an Evaluate AppleEvent.

```
send "q = 2+2" to program "Plot"
request "q" from program "Plot"
```

Suppressing User Interaction

To suppress user interaction when sending macro commands, set the `user_interactive` flag to false. To set it from an external script, send the following:

```
user_interactive=false
```

Index

A

C

D

M

N

P

R

S

T

U

V

W

X