Asymptote Demo Users Please Note: We have reduced the number of fonts used in this file and have modified its formatting to avoid problems when it is viewed or printed. You may still encounter problems, however. If these problems prevent you from going through the steps of this tutorial, please contact Brains Software for a printed copy. Thank You.

Chapter 4

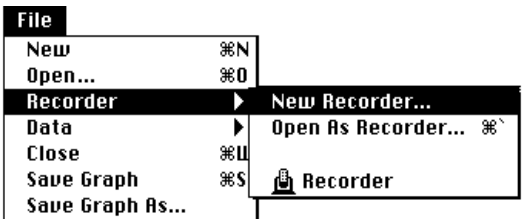
Getting Started

This chapter shows you how to make a simple graph. It describes how the Sampler script in Chapter 2 was created and it introduces you to some important scripting techniques.

Making a Simple Graph

Opening a New Recorder Window

To create a new graph in Asymptote, first open a new Recorder Window. The quickest way to do this is to select File ▸ Recorder ▸ New Recorder.

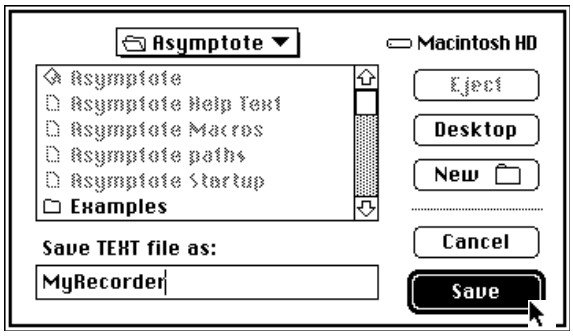


- *Note: If you have been working on another graph in Asymptote you probably want to save your work before starting a new graph. You should also select* Plot ▸ Clear *to clear the Graph Window before you start drawing a new graph.*

As you create a new graph, Asymptote, records the commands you issue in the Recorder Window. Whether you type the commands in the Command Window or use Asymptote's menus, all the commands you use to create your graph are recorded in a new Recorder Window.

• *Note: The current Recorder Window has the symbol*  *next to its name in the* Windows *menu.*

At this point the contents of your Recorder Window are saved only in memory. Normally you would construct your graph and save the finished script at the end. For the proposes of this tutorial, let's save the Recorder Window now so you can name it. Do this by clicking the Recorder Window (to make it the active window) and then select File ▸ Save File As. Give your new Recorder Window the name "MyRecorder" (as shown below) so you can refer to it later on.

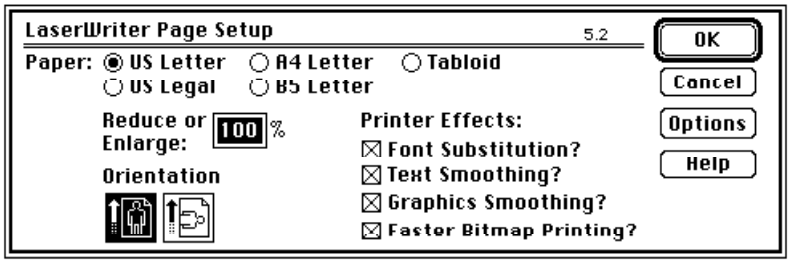


• *Note: Although it is not necessary, it is generally a good idea to save the Recorder Window with a single word name (no spaces) because you can execute scripts with single word names by typing their names in the Command Window. If the name has spaces in it, you have to use the* ***run*** *command instead.*

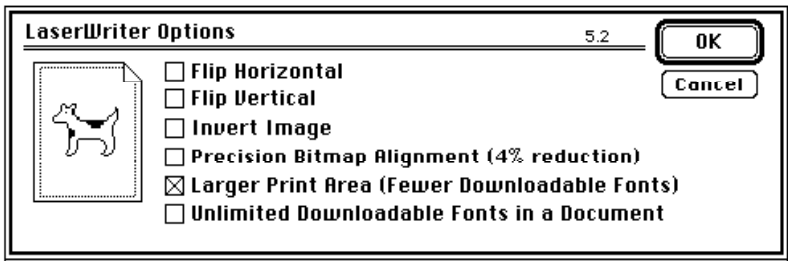
Choosing a Graph Page Setup

Before you start plotting your graph, you may want to change the Graph Page Setup. The default setup prints on 8.5 · 11 inch paper in portrait mode (the long edge of the paper is vertical). Select File ▸ Graph Page Setup ▸ New Page Setup if you want to print on a different paper stock or print in landscape mode.

After you select File ▸ Graph Page Setup ▸ New Page Setup Asymptote displays the page setup dialog box for the current printer specifications. For example, the LaserWriter page setup dialog box looks like this:



To increase the printable area on the LaserWriter page setup, click the Options button and make your selection as shown below:

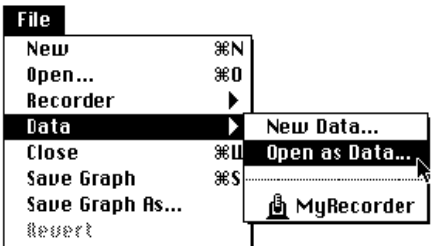


After you select this option, the margins Asymptote displays in the Graph Window will be much smaller.

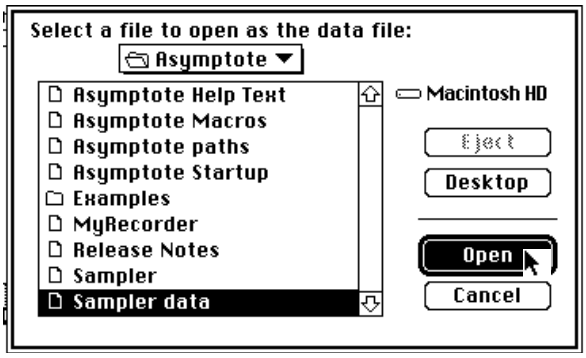
Important
For the highest quality graphics, select File ▸ Options ▸ Preferences, and select the preference "Use maximum printer resolution for highest quality graphics" in the Preferences dialog box.

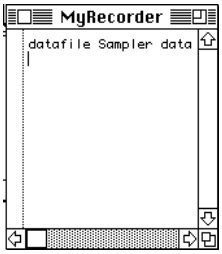
Opening the Data File

Now open the file that contains the data you want to plot. In this tutorial we are using the data file *Sampler data* used in Chapter 2. To open this data file, select File ▸ Data ▸ Open As Data shown below:



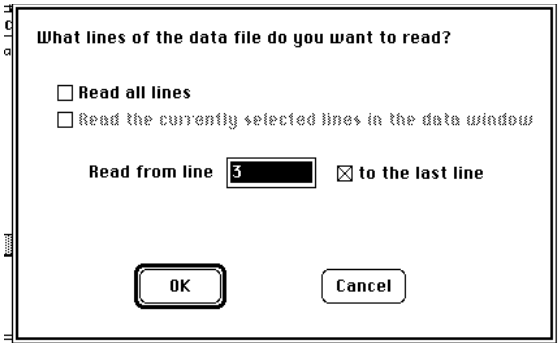
Next, click on the data file *Sampler data* as shown below:

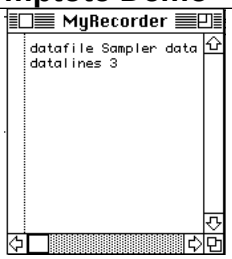




When you click the Open button, Asymptote opens *Sampler data* and records the command `datafile Sampler data` in your Recorder as shown on the left.

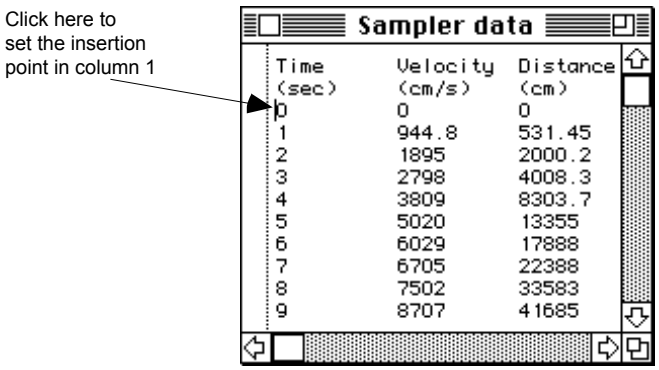
Now you are ready to start reading data. Notice that the data file *Sampler data* has values organized in columns. The first line is a header line that tells us what is in each column. Since the first two lines don't contain numbers, you must tell Asymptote to skip over them. To do this, select Data▸Lines, deselect the *Read all lines* option and specify that you want to read from line 3 to the last line in the file.



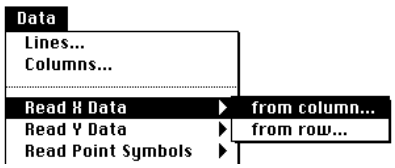


When you click OK, Asymptote adds the command `datalines 3` to your script.

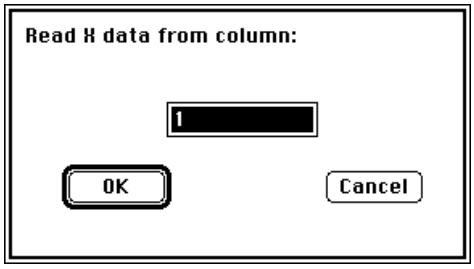
Next, read the *x* and *y* data from the file and store them in Asymptote's internal *x* and *y* vectors. (Refer to Chapter 2 for a discussion of how Asymptote stores data in memory.) You want to read the *x* data from the first column so click the window *Sampler data* to activate it. Now click a value in the first column under the heading *Time* to set the insertion point in that column as shown below.

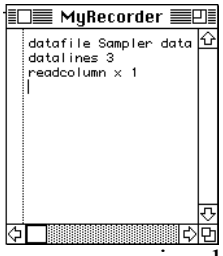


Now select Data ▸ Read X Data ▸ from column.



Asymptote displays a dialog box in which you can type the column number you want to read for the *x* data.





Because you previously clicked a value in the first column of the data file, Asymptote makes the default choice 1. Click OK to enter that choice and Asymptote reads the data from column 1, stores them in the *x* vector, and enters the command `readcolumn x 1` into *MyRecorder*.

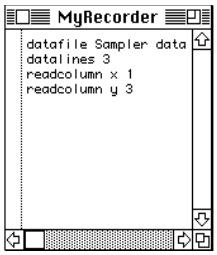
At this point you may want to make sure that the data were read properly. To do this, select Window ▸ Vectors to bring the Vectors Window to the front.

Vectors

index	X	Y	E	P	L
1	0.0000	0.0000	0.0000	0.0000	
2	1.0000	0.0000	0.0000	0.0000	
3	2.0000	0.0000	0.0000	0.0000	
4	3.0000	0.0000	0.0000	0.0000	
5	4.0000	0.0000	0.0000	0.0000	
6	5.0000	0.0000	0.0000	0.0000	
7	6.0000	0.0000	0.0000	0.0000	
8	7.0000	0.0000	0.0000	0.0000	
9	8.0000	0.0000	0.0000	0.0000	
10	9.0000	0.0000	0.0000	0.0000	
11	10.0000	0.0000	0.0000	0.0000	
12	11.0000	0.0000	0.0000	0.0000	
13	12.0000	0.0000	0.0000	0.0000	
14	13.0000	0.0000	0.0000	0.0000	

You should find the values 0 through 20 stored in the *x* vector.

Now let's do the same for the *y* data. But this time, click a value in column 3 of *Sampler data* first and select Data ▸ Read Y Data ▸ from column



Again, Asymptote provides the number of the selected column as the default so you can simply click OK. Asymptote reads *y* data from column 3 and adds the line `readcolumn y 3` to *MyRecorder*.

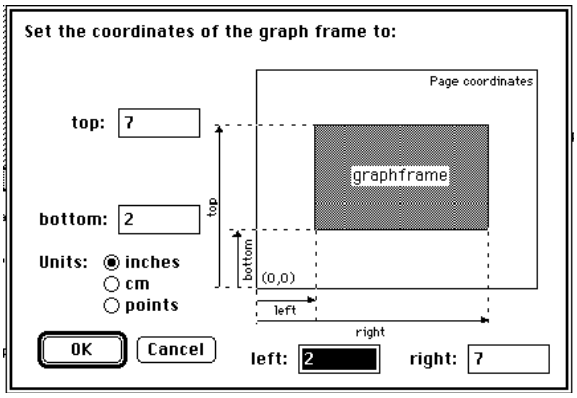
Now that you are finished reading the *x* and *y* data, the Vectors Window looks like this:

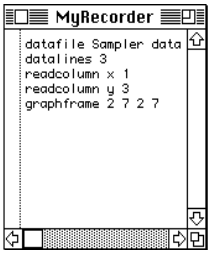
Vectors

index	X	Y	E	P	L
1	0.0000	0.0000	0.0000	0.0000	
2	1.0000	531.4500	0.0000	0.0000	
3	2.0000	2000.2000	0.0000	0.0000	
4	3.0000	4008.3000	0.0000	0.0000	
5	4.0000	8303.7000	0.0000	0.0000	
6	5.0000	13355.0000	0.0000	0.0000	
7	6.0000	17888.0000	0.0000	0.0000	
8	7.0000	22388.0000	0.0000	0.0000	
9	8.0000	33583.0000	0.0000	0.0000	
10	9.0000	41685.0000	0.0000	0.0000	
11	10.0000	50751.0000	0.0000	0.0000	
12	11.0000	64954.0000	0.0000	0.0000	
13	12.0000	71420.0000	0.0000	0.0000	
14	13.0000	80406.0000	0.0000	0.0000	

Setting the Graph Frame and Limits

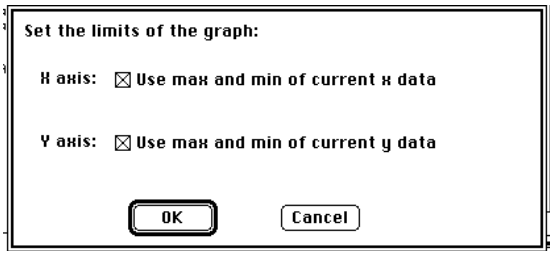
There are a couple of things you must do before you can actually plot the *x* and *y* data. First, you need to tell Asymptote where to position the plot on the page. The easiest way to do this is to select Set ▸Graph ▸Frame. The graph frame in the dialog box below has been set to plot a square graph, 5 inches on a side and 2 inches from the bottom left corner of the page. If you want to experiment a bit, feel free to specify something else.

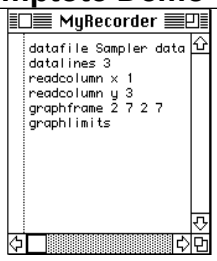




Click OK to add the command `graphframe 2 7 2 7` to *MyRecorder*.

Next, you need to set the limits of the graph by telling Asymptote the range of data values to plot on the *x* and *y* axes. Choose Set ▸Graph ▸Limits and select the options to use the current maximum and minimum of the *x* and *y* data.



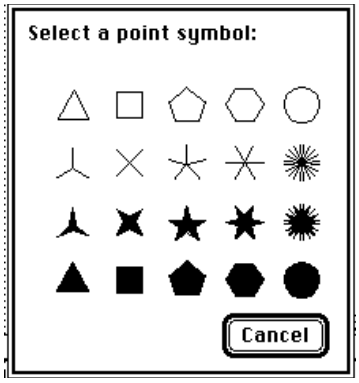


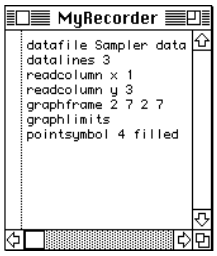
When you click OK, Asymptote scans through the *x* and *y* vectors, finds their maximum and minimum, and sets the limits of the graph accordingly. Lastly, Asymptote adds the command `graphlimits` to *MyRecorder*.

• *Note: If no arguments follow the **graphlimits** command, Asymptote automatically sets the limits to the maximum and minimum.*

Plotting Points

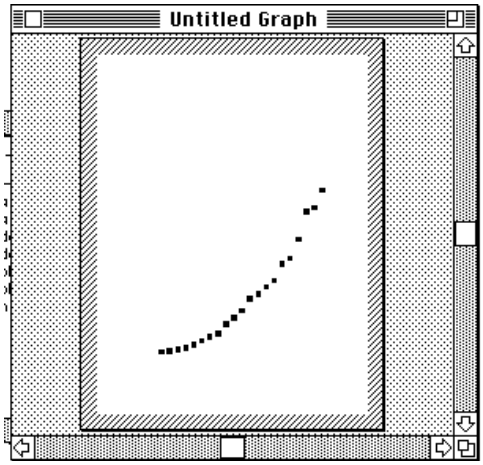
Now you are ready to plot your data. First let's choose a symbol for the points you want to plot by selecting Set ▸ Point Symbol.

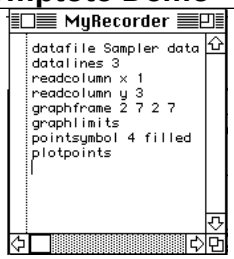




Now, click any point symbol with the mouse. If you click the filled square, for example, Asymptote sets it as the point symbol and enters the command `pointsymbol 4 filled` into *MyRecorder*.

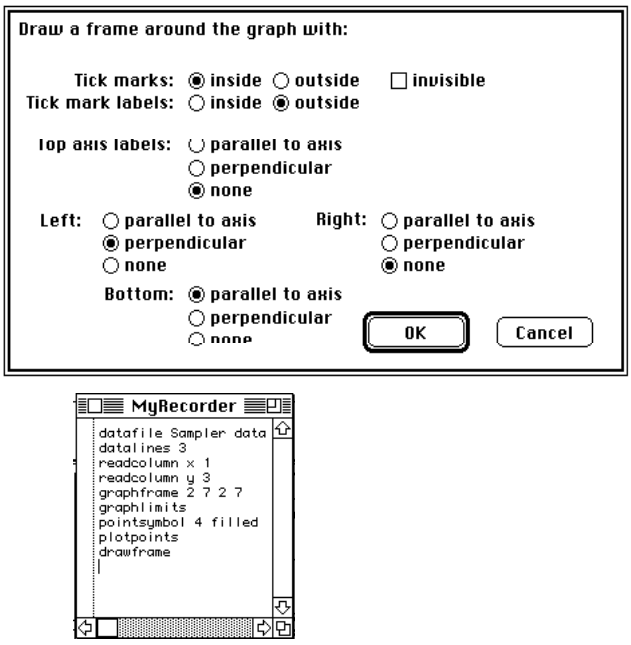
Next, select Plot ▸ Points to actually plot the points as filled squares (or whatever you have chosen). The points of your graph appear in the Graph Window as shown below. To conveniently view the entire graph, select Plot ▸ Show Graph.





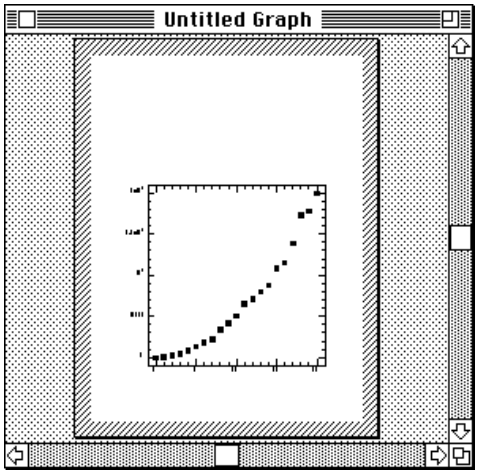
Asymptote adds the command `plotpoints` to *MyRecorder* after it draws the points.

Now let's add an axis frame around the graph with Draw▸Frame. For now, use the default settings and click OK.



Asymptote draws a frame with tickmarks around the graph and adds the command `drawframe` to *MyRecorder*.

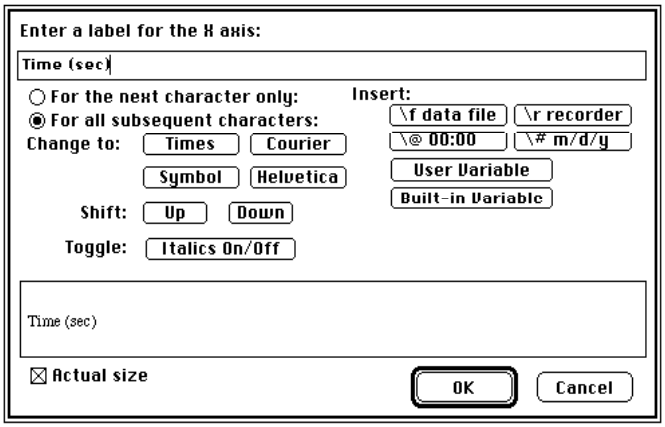
Congratulations! You have just produced a simple graph.



Remember, to view portions of your graph in greater detail, click the graph while holding down the command key. Click while holding down the command and option keys to demagnify the graph.

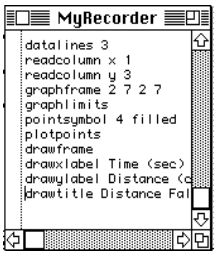
Adding Labels and a Title

Next, let's practice adding labels to the graph. To place a label under the *x* axis, select Draw ▸ X Label and type the label "Time (sec)" in the dialog box as shown below:



As you type, Asymptote shows you a preview of what the label looks like.

Now do the same for the *y* axis. Select Draw ▸ Y Label and type "Distance (cm)" for the label. For the title, select Draw ▸ Title.. and type "Distance Fallen by an Object" or choose your own title.

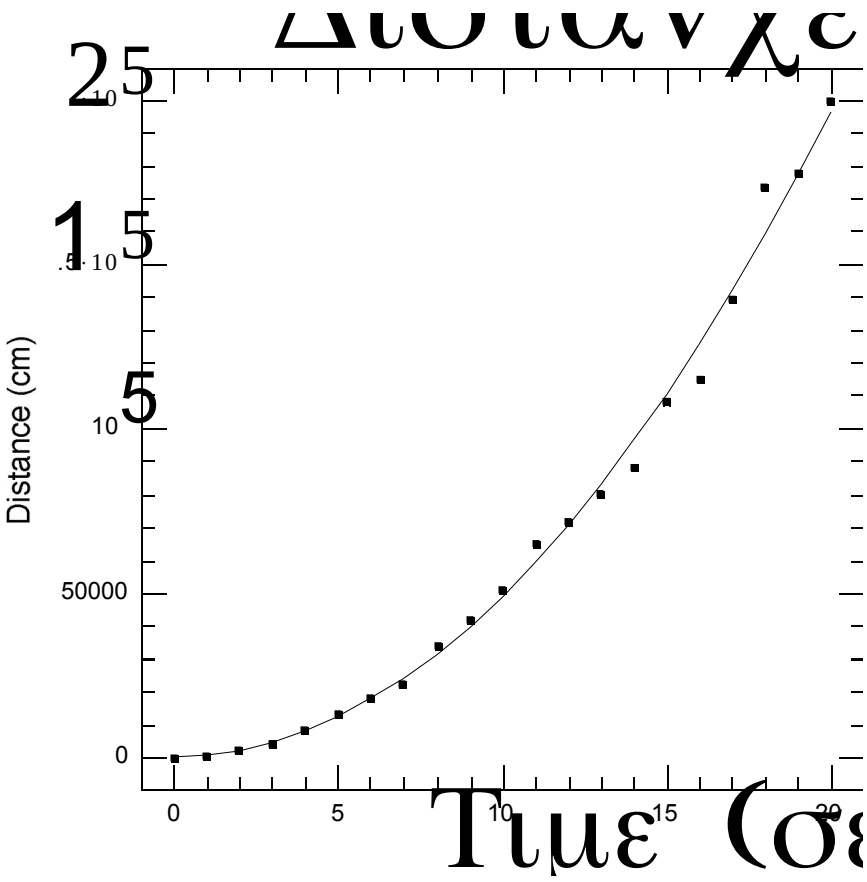


While you were adding the labels and title, Asymptote added the following commands to *MyRecorder*:

```
drawxlabel Time (sec)
drawylabel Distance (cm)
drawtitle Distance Fallen by an Object
```

Printing Your Graph

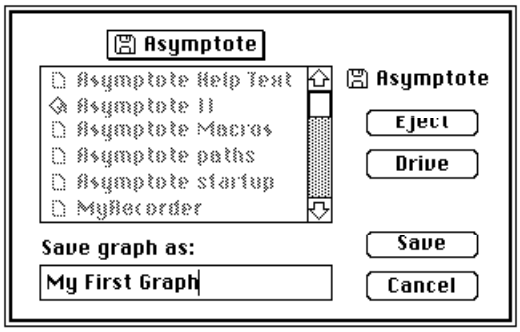
To print your graph simply select File ▸ Print Graph. Your printed graph should look something like this:



Saving Your Work

Now that you have finished the script, you should save it. To save the Recorder click it (to make it the active window) and select File ▸ Save File. Because you saved this file earlier and named it *MyRecorder*, Asymptote replaces the old copy with the current contents of the *MyRecorder* window.

MyRecorder actually contains all the information Asymptote needs to recreate your graph, so it isn't necessary to save your graph as a PICT file. If you would like to save your graph as a PICT file, simply click the Graph Window and select File ▸ Save Graph. Asymptote displays a file dialog box and asks you to name the graph. Let's name it *My First Graph*



Naming the Graph Window *My First Graph*

In the next section you learn to fine tune the graph and get some experience running the Recorder script. If you would like to take a break, select File▸Quit to end this session. To pick up where you left off, you can start Asymptote again by double clicking the icons for the files "My First Graph" or "My Recorder" in the Finder.

Fine Tuning Your Graph

Adjusting the Details

Once you've created a graph, you may want to make small changes to get it just right. For example, the default font and size for labels (unless you change them in the "Asymptote Startup Script") is Times 12 point. You might want to use Helvetica instead and draw the title of your graph in 18 point characters, for example. To make these refinements you need to modify the script in the "MyRecorder" window.

Right now your script should look something like this:

```
datafile Sampler data
datalines 3
readcolumn x 1
readcolumn y 3
graphframe 2 7 2 7
graphlimits
pointsymbol 4 filled
plotpoints
drawframe
drawxlabel Time (sec)
drawylabel Distance (cm)
drawtitle Distance Fallen by an Object
```

To change the label font to Helvetica, for example, simply type the command `fontname helvetica` before any of the commands that draw labels (such as **drawframe**, **drawxlabel** and **drawtitle**). Remember that Asymptote uses the font you set with the **fontname** command when it draws labels on your graph. If you want to draw the title in a larger point size than the other labels in your graph, add

the **scale** command just before the **drawtitle** command. When the *scale* factor is 1.0, Asymptote draws 12 point labels. To draw an 18 point title, for example, set **scale** to 1.5. After you make these changes to "MyRecorder" your script should look like this (changes appear in bold):

```
datafile Sampler data
datalines 3
readcolumn x 1
readcolumn y 3
graphframe 2 7 2 7
graphlimits
pointsymbol 4 filled
plotpoints
fontname Helvetica
drawframe
drawxlabel Time (sec)
drawylabel Distance (cm)
scale 1.5    % make the title 50% bigger
drawtitle Distance Fallen by an Object
```

• *Note: The % symbol indicates a comment. You can use comments to leave useful information to yourself or other users of your scripts. Although you don't have to include comments if you don't want to, they do help make your scripts more understandable and easier to debug. When Asymptote processes a command line, it ignores all of the characters following the % symbol.*

After you make your changes, select Script ▸ Run Recorder. Asymptote runs the Recorder script and draws your graph using the new label font and the new scale factor for the title. Notice that all of the labels following the **fontname** command (including the title) appear in the new font.

Try experimenting by making other changes to the script. For example, if you'd like to plot smaller point symbols, add a **scale** command before the **plotpoints** command. Remember that since **scale** controls the sizes of points *and* labels, if you don't want the graph labels drawn smaller, you must restore the *scale* factor after you plot the points. Try making even more changes to your script to get more experience fine tuning your graph. You may want to try the changes shown in bold below. After your done, select Script ▸ Run Recorder to view the results.

```
datafile Sampler data
datalines 3
readcolumn x 1
readcolumn y 3
graphframe 2 7 2 7
graphlimits
pointsymbol 4 filled
scale .75  % set scale before plotpoints
plotpoints
scale 1.0  % restore the scale factor for labels
fontname Helvetica
drawframe
drawxlabel Time (sec)
drawylabel Distance (cm)
scale 1.5    % make the title 50% bigger
drawtitle Distance Fallen by an Object
```

Now examine your graph closely and notice that the point symbols are indeed smaller.

These exercises have helped to show how you can control the fine details of your graph. In the next exercise, you are going to plot a theoretical function on top of the falling body data.

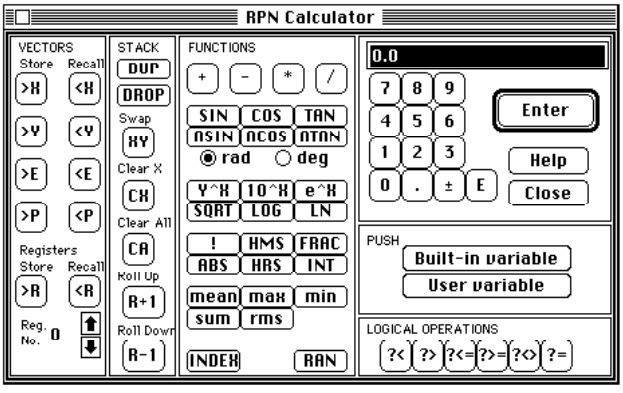
Plotting a Function

To plot a function, you can either calculate it using Asymptote's vector calculator, or you can read a 'theoretical curve' from a data file. The theoretical curve for a falling body is easy to calculate in Asymptote. The distance, d, fallen by an object is

$$d = .5\ a\ t^2$$

where t is the elapsed time and a is the acceleration of gravity.

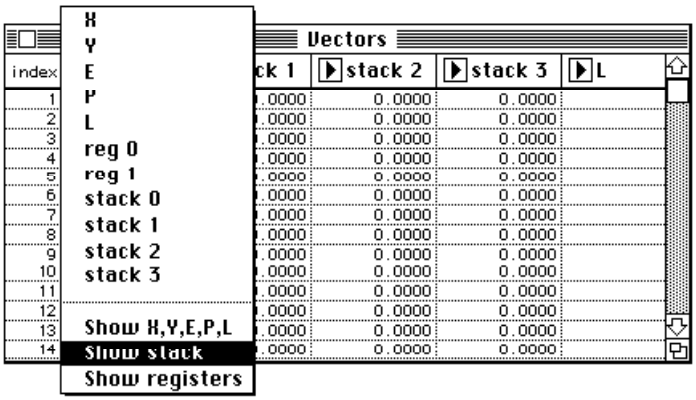
To calculate this function, select Window ▸ Calculator to bring the RPN Calculator to the front.



Asymptote's calculator uses Reverse Polish Notation (RPN). If you aren't familiar with RPN notation, read Appendix D.

Whether or not you are new to RPN, it may help to use the Vectors Window to check your calculations. Select Window ▸ Vectors to bring the Vectors Window to the

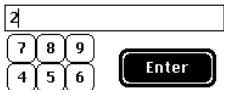
front. Click and scroll one of the ▸ symbols in the Vectors Window and select Show Stack (see below). As you click buttons in the Calculator Window you can view the results of each operation.



To calculate the falling body function follow these steps:

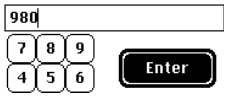
• *Note: As you go through these steps watch what is happening on the calculator stack using the Vectors Window.*

1. Click the Calculator Window to bring it to the front and click the <X button to push the current *x* values on the bottom of the stack.



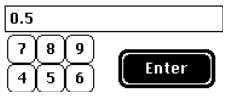
2. Type 2 in the Calculator Window and click the Enter button (or press the Enter key). This pushes the constant 2 on the bottom of the stack.

3. Click the Y^X button to square the *x* values on the bottom of the stack.



4. Type the acceleration of gravity on the surface of the Earth (980 dynes sec^{-1}) and click the Enter button.

5. Click the * button to multiply the bottom two values on the stack.



6. Type `0.5` and click Enter to push the constant 0.5 on the bottom of the stack.

7. Click the * button to calculate the result.

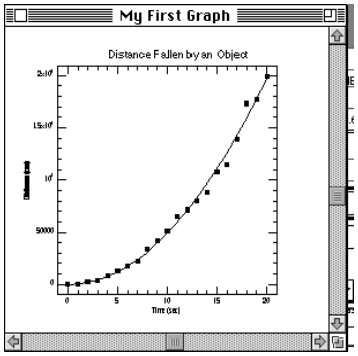
8. Click the >Y button to store the calculated value in the *y* vector.

As you performed these calculations, Asymptote added the appropriate **RPN** command to the end of your Recorder script as shown below:

```
rpn <x
rpn 2
rpn y^x
rpn 980
rpn *
rpn 0.5
rpn *
rpn >y
```

• *Note: You can put all the **rpn** commands on a single line. Instead of using eight lines, you could have used a single **rpn** command like this:* `rpn <x 2 y^x 980 * 0.5 * >y`

Now you have calculated the theoretical *y* values from the *x* values of the points. To plot the theoretical curve on top of the data, all you have to do is select Plot▸Line. Asymptote draws the curve on top of your graph as shown below:



Writing Interactive Scripts

Using ? as a Command Argument

As you work with Asymptote, you may find that you want to write a script that prompts you (or someone else running your script) for information while the script is running. For example, you might want your script ask you which data file to use, which columns of the data file to plot, and which point symbol to use for the points. Interactive scripts are essentially small plotting programs that help you automate your plotting tasks. Here you explore some of the ways you can make your scripts interactive. For a description of how to write more advanced scripts, see Chapter 6.

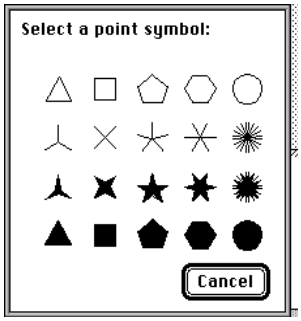
The easiest way to write an interactive script is to replace a command argument with a question mark. For example, to have the "MyRecorder" script ask you which point symbol to use for the points, simply change the command

```
pointsymbol 4 filled
```

to

```
pointsymbol ?
```

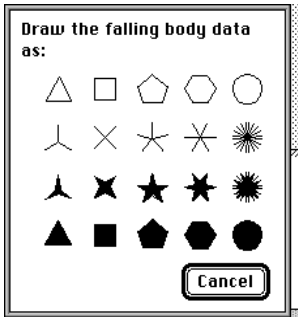
and select Script ▸ Run Recorder to see what happens.
The question mark tells Asymptote to process the **pointsymbol** script command as if you had selected the menu command Set ▸ Point Symbol instead. When Asymptote encounters the question mark after the **pointsymbol** command it displays the point symbol dialog box as shown below:



To help you (or someone else running your script) remember why the script is asking for the point symbol, you might want to add text to the prompt to be more specific. This is easy to do with Asymptote. Simply add the text after the question mark, like this:

```
pointsymbol ?Draw the falling body data as:
```

Make this change to "MyRecorder" and run it again. The dialog box now looks like this:



You can add text after any command that displays a dialog box prompting the user for input. Below are a few examples of these commands:

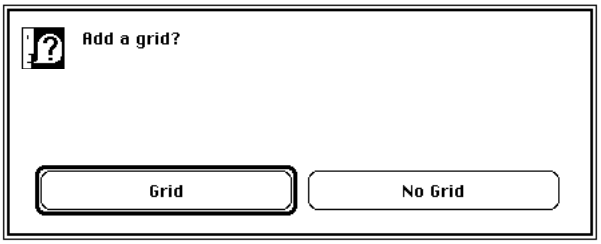
```
graphlimits ?Please specify new graph limits:
pendash ?What dash pattern for the X-ray data?
datafile ?Please open the January data file:
readcolumn y ?Enter the trial number to plot:
```

Using the 'ask' Command

Another way to write an interactive script is to use the **ask** command. For example, to have "MyRecorder" ask if you want to add a coordinate grid to the graph, add the three commands shown below in bold to "MyRecorder:"

```
datafile Sampler data
datalines 3
readcolumn x 1
readcolumn y 3
graphframe 2 7 2 7
graphlimits
pointsymbol ?Draw the falling body data as:
scale .75
plotpoints
scale 1.0
fontname Helvetica
set AddGrid yes
ask AddGrid "Grid" "No Grid" Add a grid?
if AddGrid then drawgrid
drawframe
drawxlabel Time (sec)
drawylabel Distance (cm)
scale 1.5
drawtitle Distance Fallen by an Object
rpn <x
rpn 2
rpn y^x
rpn 980
rpn *
rpn 0.5
rpn *
rpn >y
plotline
```

Now, select Script ▸ Run Recorder to run the new script. When Asymptote executes `set AddGrid yes`, it creates a new user variable called *AddGrid* and sets its value to 'yes.' Doing this also makes 'yes' the default choice for the **ask** command which follows. The **ask** command displays a dialog box that asks a 'yes or no' question.



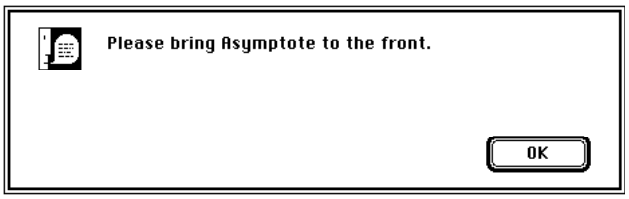
If you click Grid, Asymptote sets *AddGrid* to 'yes.' If you select No Grid, then Asymptote sets *AddGrid* to 'no.' Notice that the default choice (the option with the dark border around it) is the 'yes' response because it's the value *AddGrid* had when Asymptote executed the **ask** command. The third command added to the script was the **if** command. If *AddGrid* is 'yes,' the **if** command draws the coordinate grid with **drawgrid** and does nothing if *AddGrid* is false.

Running Scripts in the Background

The beach ball cursor looks like this.

If you are using Asymptote with MultiFinder or System 7, you can do other work while Asymptote runs your script in the background. To do this, start running your script. Then switch to your other application when the rolling beach ball cursor appears on the screen. While you are working in another application, Asymptote runs your script automatically in the background. To have Asymptote print your graph automatically after it finishes drawing the graph, add the **print** command at the end of your script.

Sometimes a script running in the background may need your attention. If this happens, Asymptote displays this message:



If you see this message, click OK to acknowledge the message. Then click one of Asymptote's windows to bring Asymptote to the front.

Summary

This chapter has introduced you to how graphs are created and refined with scripts. For a description of Asymptote's more advanced scripting techniques, see Chapter 6. In Chapter 5 we'll look at each of Asymptote's menu commands in detail and explore all of Asymptote's features.