Getting Started with Mathematica

This section will introduce you to notebooks, palettes and the *Mathematica* computing environment. Unless you already have a significant amount of experience using *Mathematica*, you will want to read and study the following material carefully. Topics to be covered here include working with a *Mathematica* notebook, entering and executing mathematical calculations, and learning shortcuts to typing in *Mathematica* commands using palettes. After covering these basics, you will learn about *Mathematica* algebra, *Mathematica* commands and functions, equation representation and how to create your own functions. Finally you will read about common mistakes new users of *Mathematica* make and how to correct them.

■ *Mathematica* Notebooks

A *notebook* is a document which allows you to interact with *Mathematica*. Each notebook is divided up into a sequence of individual units called *cells,* each containing a specific type of information such as text, graphics, input or output. For example, this file (*Intro.nb*) is an example of a *Mathematica* notebook and the text in this paragraph is contained in a *text* cell. Text cells contain information to be read by the user, but contain no executable *Mathematica* commands. The following cell, displaying `In[1] :=` 2^{20}, is an example of an *input* cell containing executable *Mathematica* commands. *Mathematica* computes the value of 2^{20} and the results of the calculation are displayed as Out[1] = 1048576 in an *output* cell.

In[1]:= 2^{20}

Out[1]= 1048576

Associated with each cell is a right cell bracket which can be seen in the right margin of this file (you will not see the brackets if you are viewing the file with Adobe Acrobat or Adobe Acrobat Reader). You can see that all of the text cells contain brackets of the same shape whereas the input cell and output cells have slightly different shaped cell brackets which distinguishes them from other types of cells. Also note that some cell brackets are grouped together within larger cell brackets and these larger cell brackets are in turn all contained within one large cell bracket. This is due to a hierarchy of cells which *Mathematica* automatically creates based upon cell types contained within the notebook. In this notebook, the cell at the top of the hierarchy is a *section* cell. This is the first cell in the file containing the text *Getting Started with Mathematica* and under this cell there are *text, input, output, subsection* and *subsubsection* cells. For example, after the section cell at the beginning of this notebook is a text cell followed by a subsection cell containing the title *Mathematica Notebooks*. All of the other cells below this cell are grouped together as a sequence of subcells of the subsection cell until another subsection cell, or a cell higher in the hierarchy, is encountered. For example, the subsection cell entitled *Palettes,* a few cells below this text cell, begins a new group of cells. To see a list of cell types, ordered according to their relative higherarchy, select **Format** and then **Style** from the Help menu.

Creating an Input Cell and Executing its Contents

Since most of your time will be spent working with input cells, *Mathematica* will assume the any new cell you create is an input cell unless you specify otherwise. Suppose, for example, you want to calculate the quotient `675 / 25` with *Mathematica* immediately below this text cell. To do this, either press the downarrow key (↓) on your keyboard or use your mouse and move the cursor to a point immediately below this cell until the cursor looks like a sideways "I" and then click the mouse button. A horizontal line should appear and this line will be replaced by an input cell as soon as you begin typing. Once you have entered `675 / 25`, you can the have *Mathematica* calculate the quotient by pressing **Shift-Enter** or by pressing **Enter** on the numeric key pad, depending on which platform you are using. *Mathematica* will calculate the result and place it in an output cell. The input and output cells are displayed below.

```
In[2]:= 675 / 25
Out[2]= 27
```

Creating Other Types of Cells

When you create a new cell, the default cell type is an *input* cell. Suppose instead, you want to create a text cell. To do this, use your mouse to click in an area where you want to create a new cell and a horizontal line will appear. Then from the Help menu, select **Format**, **Style** and then **Text**. A new text cell will then be created as soon as you begin typing. You can experiment with creating other types of cells by selecting a cell style of your choice, after first choosing **Help**, **Format**, and **Style** from the menu.

■ Palettes

Sometimes you will find it easier to enter information into a notebook using a *palette* instead of the keyboard. A palette is similar to a set of calculator buttons, providing shortcuts to entering commands and symbols into a notebook. The name of a useful palette is *BasicInput* and it can be found by selecting **File**, **Palettes**, and then **BasicInput** from the Help menu. After opening BasicInput, drag it to the right side of your screen and resize your notebook, if necessary, so that both the notebook and palette are visible in non-overlapping windows.

To demonstrate the usefulness of palettes, suppose you wish to calculate $\sqrt{804609}$. The *Mathematica* command for computing the square root of *n* is `Sqrt[n]`. The following input cell was created by typing in the information exclusively from the keyboard.

```
In[3]:= Sqrt[804609]
Out[3]= 897
```

A quicker and more natural way of entering $\sqrt{804609}$ can be accomplished by clicking on the square root button $\sqrt{■}$ in the palette and then entering 804609.

$$\text{In[4]:=}\quad \sqrt{804609}$$

```
Out[4]= 897
```

The remainder of this notebook will show you how to perform calculations using the keyboard and using palettes, where appropriate.

■ Help Menu

It is impossible to give you a complete overview of *Mathematica* in this short introduction, but more information about the *Mathematica* computing environment can be accessed through the Help menu. For more introductory information, select **Help**, then **Master Index**, and then click on the button entitled **The Mathematica Book**. Then choose **A Practical Introduction to Mathematica** in the first column of information below the Help menu buttons and then select **Practical Introduction** in the second column. You can then select any of the hyperlinks in the text window to read more about *Mathematica*.

■ *Mathematica* Arithmetic

Mathematica can be viewed as a powerful calculator when used to perform basic arithmetic calculations.

Addition and Subtraction

Addition and subtraction are performed using + and −, respectively. Here is an example.

```
In[5]:=  5479847844 - 84758974 + 8374874384567993749873
Out[5]=  8374874384573388838743
```

Remember, press **Shift-Enter** (or press **Enter** on the numeric key pad on some machines) to allow *Mathematica* to execute the contents of an input cell.

Multiplication and Division

Multiplication is performed by using either * or a space. To compute $x \div y$, enter x / y or $\frac{x}{y}$.

```
In[6]:=  25 * 25
Out[6]=  625

In[7]:=  25 25
Out[7]=  625

In[8]:=  1455 / 15
Out[8]=  97
```

To enter a fraction using the palette, click on $\frac{\blacksquare}{\square}$, then enter 1455, then press the **Tab** key and then enter 15.

In[9]:= $\frac{1455}{15}$

Out[9]= 97

Square Roots and Powers

Enter `Sqrt[x]`, or use click $\sqrt{\blacksquare}$ in the palette and then enter x, to compute $\sqrt{x}$.

In[10]:= `Sqrt[5000]`

Out[10]= $50\sqrt{2}$

In[11]:= $\sqrt{32364721}$

Out[11]= 5689

To compute x^y, enter `x^y`. In the following example, note that the integer in the output is so long that it requires more than one line to be displayed in its entirety.

In[12]:= `2^512`

Out[12]=
```
1340780792994259709957402499820584612747936582059239337772356144372176403007354697680187429816690342769003185818648605085375388281194656994643364900608409 6
```

Using the palette instead, click on ■□ , then enter 2, then press the **Tab** key (or use your mouse and click on the box in the power) and then enter 512.

In[13]:= 2^{512}

Out[13]=
```
134078079299425970995740249982058461274793658205923933777235614437217640300735469768018742981669034276900318581864860508537538828119465699464336490060840 96
```

Exact and Approximate Answers

When possible, *Mathematica* will compute and display exact solutions to arithmetic problems. For example, *Mathematica* will display the output $50\sqrt{2}$ when $\sqrt{5000}$ is evaluated. To view an approximate value of an expression *x*, use `N[x]` or place a decimal point after at least one of the numbers in the expression being evaluated.

In[14]:= $N[\sqrt{5000}]$

Out[14]= 70.7107

In[15]:= $\frac{58696.}{8888}$

Out[15]= 6.60396

If you omit the decimal point in the fraction above, the output will be reduced to lowest terms, but remain in exact, fractional form.

In[16]:= $\frac{58696}{8888}$

Out[16]= $\frac{667}{101}$

For more examples, go to the **Help** menu, then select **The Mathematica Book** and click on **Practical Introduction** in the first column, **Numerical Calculations** in the second column and then double click on **Arithmetic** in the third column.

■ Using Previous Results

You can use previous results in `Out[n]` by entering `%n`.

```
In[17]:= %16
Out[17]= 667/101
```

The following example shows you how to find the approximate value of the output in `Out[16]`.

```
In[18]:= N[%16]
Out[18]= 6.60396
```

■ Re-executing Commands

By clicking in any previously executed input cell, you can then modify the input cell and then press **Shift-Enter** to re-execute the contents input cell. Keep in mind that this will change the assigned output number `Out[n]` in the new output cell, even though the output appears in the same location as the previous output statement. You will therefore need to re-execute any other cells referring to this output, after first changing the value of *n* in `%n`.

Better yet, if you plan on using the output in another input cell, you should assign your own name to the expression instead of relying on the `%n` command. For example, suppose you want to determine the cost of placing a carpet in a circular room with a radius of 34.5 feet if the carpet costs $2.95 a square foot. You might first want to determine the area of the room and assign this value the name *area* as shown below.

In[19]:= **area = π 34.5**^{2}

Out[19]= 3739.28

Then, to compute cost, compute 2.95 times the area.

```
In[20]:= 2.95 area
Out[20]= 11030.9
```

■ *Mathematica* Algebra

Mathematica is an example of a Computer Algebra System in part because of its ability to work with symbolic mathematics. *Mathematica* can quickly perform algebraic tasks which are difficult, tedious or impossible to complete by hand.

Expand, Simplify & Factor

The *Mathematica* commands, `Factor` and `Expand`, can be used to factor or to multiply out an algebraic expression.

In[21]:= **Factor**$[x^5 - 1]$

Out[21]= $(-1 + x)\ (1 + x + x^2 + x^3 + x^4)$

In[22]:= **Expand**$\left[(x^3 + 2x^2 - 4x + 1)^4\ (x - 2)\right]$

Out[22]= $-2 + 33x - 224x^2 + 800x^3 - 1580x^4 + 1592x^5 - 436x^6 - 538x^7 + 368x^8 + 64x^9 - 76x^{10} - 8x^{11} + 6x^{12} + x^{13}$

The `Simplify` command is often useful in obtaining a simplified from of an algebraic expression. The output of the following expression, for example, is not simplified.

```
In[23]:=  (2 x - 2)^2 (x - 1)^5
Out[23]=  (-1 + x)^5 (-2 + 2 x)^2
```

Now the calculation is done again using the `Simplify` command.

```
In[24]:=  Simplify[(2 x - 2)^2 (x - 1)^5]
Out[24]=  4 (-1 + x)^7
```

For more information on algebraic calculations, choose **Help**, then **Master Index**, and then click on the button entitled **The Mathematica Book**. Then select **A Practical Introduction to Mathematica** in the first column of information below the Help menu buttons, then select **Algebraic Calculations** in the second column and select **Contents** in the third column. You can then explore the displayed hyperlinks to learn more about algebra and *Mathematica*.

■ *Mathematica* Constants, Commands and Functions

Mathematica contains hundreds of built-in constants, commands and functions. Some of the more common constants and commands will be introduced here.

Constants

The constant $\pi \approx 3.14159$ is entered by typing in *`Pi`* from the keyboard or by selecting π on the palette. As an example, the area of a circle of radius 3 is computed below using the area formula $A = \pi r^2$ (note that a space is placed between Pi and 3^2 for multiplication).

```
In[25]:=  Pi 3^2
Out[25]=  9 π
```

To obtain an approximate answer, use the ***N*** command.

```
In[26]:=  N[%23]
Out[26]=  (-1. + x)^5 (-2. + 2. x)^2
```

You can represent the constant *e* by typing in a capital *E* or by clicking on e in the palette.

```
In[27]:=  N[E]
Out[27]=  2.71828
```

Mathematica Commands and Functions

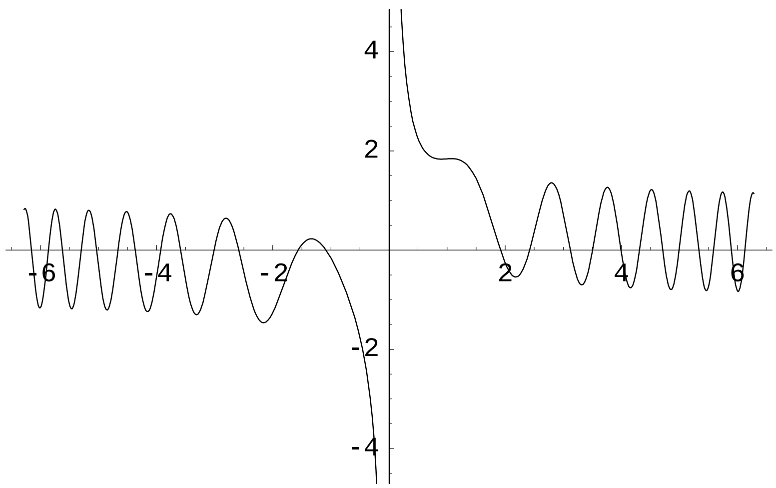
Mathematica commands and functions consist of a string of letters and numbers, beginning with a capital letter, followed by set of square brackets [...] containing a sequence of arguments. For example, *Mathematica* represents the function sin(x) by Sin[x]. In the next example, notice that the argument $\frac{\Pi}{12}$ is contained within square brackets instead of parenthesis (...).

In[28]:= **Sin[$\frac{\pi}{12}$]**

Out[28]= $\frac{-1+\sqrt{3}}{2\sqrt{2}}$

One of the more common commands you will use when working with *Mathematica* is the `Plot` command. Executing `Plot[f[x], {x, a, b}]` will command *Mathematica* to sketch a graph of $y = f(x)$ over the interval (a, b). For example, the plot of the function $f(x) = \sin(x^2) + \frac{1}{x}$ is displayed below.

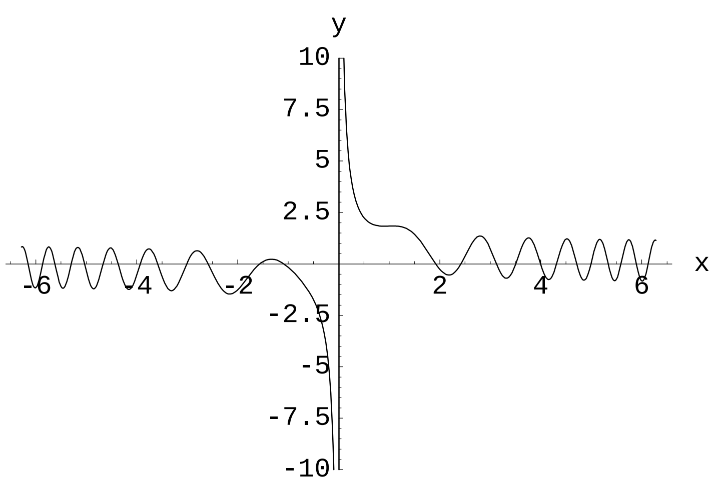
In[29]:= **Plot[Sin[x^2] + $\frac{1}{x}$, {x, -2 π, 2 π}]**



Out[29]= - Graphics -

Typically, Mathematica functions and commands come with an assortment of options of the form `name -> value`. For example, adding `PlotRange -> {-10, 10}` to the Plot function will result in a graph displaying the vertical axis from $y = -10$ to $y = 10$ and `AxesLabel->{x,y}` will add x and y labels to the x-axis and y-axis, respectively.

In[30]:= **Plot[Sin[x^2] + $\frac{1}{x}$, {x, -2 π, 2 π}, PlotRange -> {-10, 10}, AxesLabel → {x, y}]**



Out[30]= - Graphics -

To see a more complete list of options for the `Plot` command, click on **Help**, then select **Master Index** and then type in *Plot* in the text box immediately to the right of the **Go To** button. Then click on the **Go To** button or press **Enter**. In the text window, under **The Mathematica Book**, select the hyperlink next to **options for** to see a more complete list of options for the `Plot` command.

Representing Equations

Suppose you wanted to find the solution of $\sin(x^2) + \frac{1}{x} = 0$ lying near 2 as seen in the last output. When entering an equation as input, you must use a double equals == instead of an equals = statement. The following `FindRoot` command will compute the approximate solution to the equation lying in the interval (1.75, 2.25). The output of the following statement displays the approximate solution 1.92073.

In[31]:= $\text{FindRoot}\left[\text{Sin}[x^2] + \frac{1}{x} == 0, \{x, 1.75, 2.25\}\right]$

Out[31]= {x → 1.92073}

Using the Help Menu

It would be impossible to introduce the entire library of *Mathematica* constants, functions and commands in this short introduction. But by using the **Help** menu, you can usually find what you need. For example, suppose you want to compute the natural logarithm of 2, namely ln(2). Click **Help**, select **Master Index** and then start typing in *natural logarithm* until you see these words displayed and highlighted in the second column of text in the Help window. Click on the highlighted text and then select the hypertext labeled 1.1.3 to go to Section 1.1.3 of the online *Mathematica* book where ln(x) is introduced. From reading the text, you will see that the *Mathematica* function Log[x] represents the natural logarithmic function. From the same section in the online textbook, you will see that Log[b, x] represents $\log_b(x)$. Below, the values of ln(1000) and $\log_{10}(1000)$ are computed below. The decimal point is entered in the first input statement below so that an approximate solution is provided.

In[32]:= **Log[1000.]**

Out[32]= 6.90776

In[33]:= **Log[10, 1000]**

Out[33]= 3

If you have difficulty finding a particular command to suit your needs, then select **Help** and then click on **The Mathematica Book** button. Under the Help Browser buttons, click on **Contents** in the first column of text. You can then scroll down through the table of contents of The *Mathematica* Book to find the information you need. All of the text in the table of contents is hypertext and therefore all you need to do is click on the appropriate text to go straight to the corresponding section in the book.

Capitalization of *Mathematica* Constants, Functions and Commands

You may have noticed that all *Mathematica* constants, functions and commands are capitalized and there are no spaces between any of the letters and digits in the command or constant name. Sometimes a constant, function or command name will consist of two words in which case each word begins with a capital letter such as the command `FindRoot[...]` you saw earlier.

■ Standard Add-On Functions

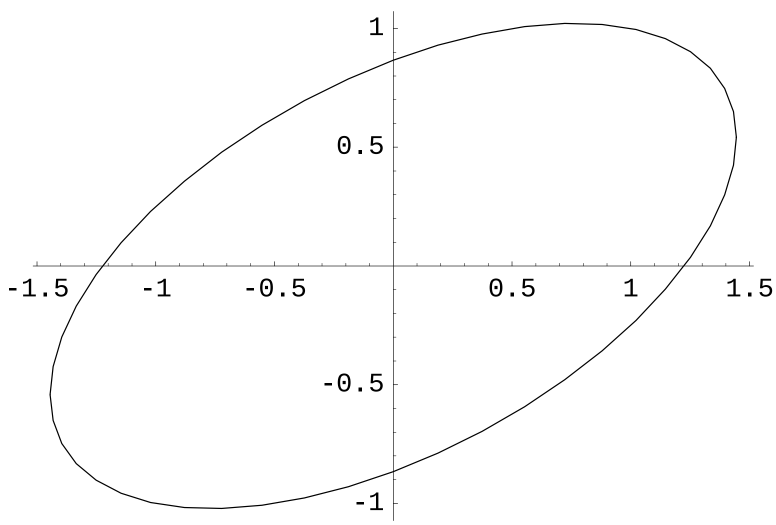
All of the constants, functions and commands introduced so far are *built-in* functions which means that they are loaded automatically into your computer's memory when you start performing computations with *Mathematica*. In addition to these built-in functions, there are several directories, each of which contains packages of additional functions, called *add-on* functions.

For example, the graph of the function $y = f(x)$ is obtained using the built-in command `Plot[f[x], {x, a, b}]` as discussed earlier. Suppose instead we want to plot the graph of the equation $2x^2 - 3xy + 4y^2 = 3$. One way to do this is to use the add-on function `ImplicitPlot` found in the package which in this case has the same name *ImplicitPlot*. To load the commands found in the ImplicitPlot package, you must enter `<< dir`pac`` where `dir` represents the directory or file where the package named `pac` is found. So, for this example, the `ImplicitPlot` package is found in the `Graphics` directory.

```
In[34]:= << Graphics`ImplicitPlot`
```

The command `ImplicitPlot[eqn, {x, a, b}]` can now be used to plot the graph an equation on the interval (a, b).

```
In[35]:= ImplicitPlot[2 x^2 - 3 x y + 4 y^2 == 3, {x, -1.5, 1.5}]
```




```
Out[35]= - Graphics -
```

To learn more about standard add-on functions, select **Help** and then click on the **Add-ons** button. From there, you might want to start by selecting **Standard Packages** in the left column, **Introduction in the middle column** and then click on **The Standard Add-on Packages** in the right column. You can then start reading more packages and explore those package which interest you.

■ Variables and Assignment Statements

We can assign names, called variables, to values using either = or :=. Suppose that you want to let the variable *w* represent the width of a rectangle, let *l* represent the length of a rectangle and suppose we want to assign the area of the rectangle the name area. So for a 10-by-20 rectangle, we can make the following assignment statements.

```
In[36]:= w = 10
Out[36]= 10
```

```
In[37]:= l = 20
Out[37]= 20
```

The assignment statement := is used to assign the *rule* *w* times *l* to the name `area` (the difference between = and := is a subtle, but important one and it will be explained in detail shortly).

```
In[38]:= area := w l
```

To compute the area of a 10-by-20 rectangle, just execute the following command.

```
In[39]:= area
Out[39]= 200
```

The command `Clear[var]` is used to clear the value currently assigned to a var.

```
In[40]:= Clear[w]
```

The value `area` is now `20 w` since a value is longer assigned to *w*.

```
In[41]:= area
Out[41]= 20 w
```

Entering Multiple Commands in a Single Input Cell

Up until now, only one *Mathematica* command or computation has been entered for each input cell. By using semicolons or by starting each new command or calculation on a new line in a cell, you can perform multiple tasks within a single cell.

```
In[42]:= Clear[l, w]
         l = 35; w = 25
         area
Out[43]= 25
Out[44]= 875
```

The semicolon also suppresses output from being displayed in the expression preceding the semicolon. Since a semicolon was not placed after the assignment statement for *w* on the second line, the value of 25, which was assigned to `w`, appears in an output statement. Place a semicolon after the assignment statement to prevent the output from this assignment statement from being displayed.

```
In[45]:= Clear[l, w]
         l = 35; w = 25;
         area
Out[47]= 875
```

Using := Instead of =

As you have already seen, assignment statements can also be made using := (called the `SetDelayed` command) instead of using =, but there is a subtle difference which you need to understand. Recall that the circumference of a circle is $2\pi r$.

In[48]:=
```
r = 3;
cir1 = 2 π r;
cir2 := 2 π r;
cir1
cir2
```
Out[51]= 6 π

Out[52]= 6 π

It appears up to this point that `cir1` and `cir2` both represent the circumference of a circle of radius *r*. The difference is that when the assignment statement `cir1 = 2 π r` is executed, the value of 2 π`r` is immediately calculated and assigned to `cir1`. Therefore the value of `cir1` is set equal to 6 π. However the assignment statement `cir2 := 2 π r` assigns the *rule* 2 π`r` to `cir2`, but the value associated with this rule is not calculated until `cir2` appears in another expression. So if you change the value of *r* to 5, the value of `cir1` remains unchanged while the value of `cir 2` is updated as seen in the following output.

In[53]:=
```
r = 5;
cir1
cir2
```
Out[54]= 6 π

Out[55]= 10 π

Protected Names

All *Mathematica* built-in and add-on commands are *protected* and attempting to redefine your own command or constant using a protected word will result in an error message. For example, attempting to assign the value of 5 to N will result in an error.

In[56]:=
```
N = 5
```
```
Set::wrsym : Symbol N is Protected.
```
Out[56]= 5

The letter *N* cannot be used because `N[x]` is a built-in *Mathematica* command which returns the numerical approximation of *x* as described earlier. To avoid using protected words, *always begin your own definitions with a lowercase letter*.

In[57]:=
```
n = 5
```
Out[57]= 5

To avoid protected names, you can still use capital letters just as long as the *first* letter is not capitalized.

In[58]:=
```
myFavoriteNumber = 7;
myFavoriteNumber^2
```
Out[59]= 49

■ Creating Functions

Suppose you want to define your own function $f(x) = \sin(x) - \cos(x^2)$. To do this, it is a good idea to first clear out the current definition of f by entering `Clear[f]`. Even if f was not previously defined, it is generally good to get into the habit of using the `Clear` command before defining functions. Then enter f[x_] = Sin[x] – Cos[x^2] and note that an underscore _ immediately follows the x on the left-hand side of the equation. Think of the underscore as a way of defining the value of x to be a variable or as a placeholder for a value to be specified later by the user.

```
In[60]:= Clear[f];
         f[x_] = Sin[x] - Cos[x^2]
Out[61]= -Cos[x^2] + Sin[x]
```

To compute the value of $f(2.1)$, for example, execute `f[2.1]`.

```
In[62]:= f[2.1]
Out[62]= 1.16101
```

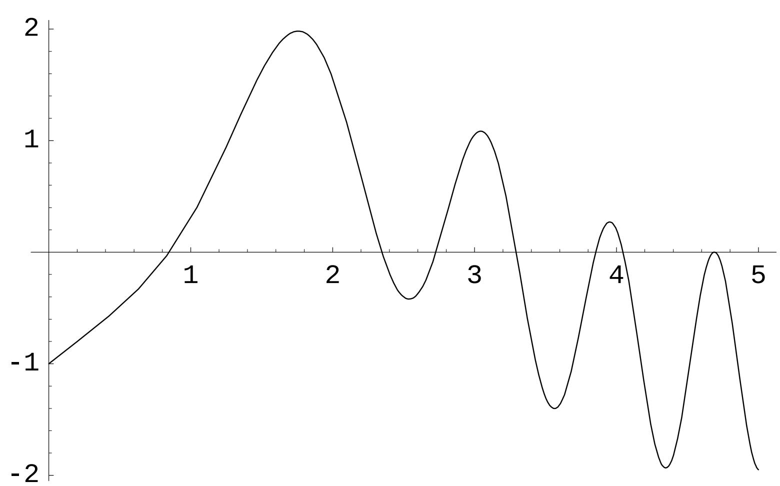
The variable x in the definition f[x_] = Sin[x] – Cos[x^2] above is just a placeholder for any value or variable. So we can replace x with any other value or variable and *Mathematica* will evaluate the function using this new value or variable.

```
In[63]:= f[ y]
Out[63]= -Cos[y^2] + Sin[y]
```

```
In[64]:= f[var]
Out[64]= -Cos[var^2] + Sin[var]
```

You can plot the function you created.

```
In[65]:= Plot[f[x], {x, 0, 5}]
```




```
Out[65]= - Graphics -
```

Your own function name can contain both letters and numbers, as long as you begin with a letter. *Avoid protected Mathematica commands by beginning with a lower case letter.* For example, `D` is a protected *Mathematica* command, so an error will result if you attempt to used this when creating a function.

In[66]:=
```
f[x_] = Sin[x - x^2];
D[x] = Cos[x] - f[x]
```

```
Set::write : Tag D in D[x] is Protected.
```

Out[67]= `Cos[x] - Sin[x - x^2]`

■ Lists

A list is a collection of objects. The following is a list of 4 numbers given the name `list1`.

In[68]:= `list1 = {1, 2, 3, 6};`

The following list contains three objects: two numbers followed by a list.

In[69]:= `list2 = {1, 5, {e, π, m}};`

The command `Table[f[n], {n, a, b}]` will create the list $\{f[a],\ f[a+1],\ f[a+2],\ \ldots,\ f[b]\}$.

In[70]:= `Table[n^2, {n, 1, 10}]`

Out[70]= `{1, 4, 9, 16, 25, 36, 49, 64, 81, 100}`

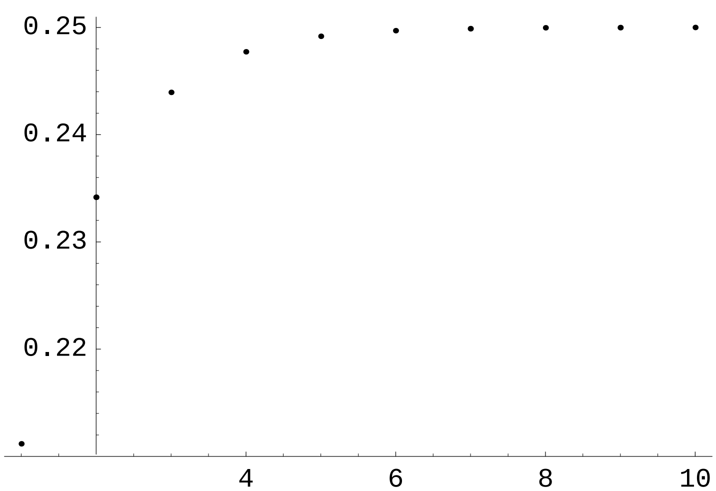
Suppose you want to plot the values given in the following output.

In[71]:= `list3 = Table[.5/(2 + e^-n), {n, 1, 10, 1}]`

Out[71]=
```
{0.211159, 0.234155, 0.243928, 0.247731, 0.249161,
 0.249691, 0.249886, 0.249958, 0.249985, 0.249994}
```

The *Mathematica* command `ListPlot` can be used to plot the values of a list as shown below. The values are plotted as the ordered pairs (1, .211159), (2, .234155), ..., (10, .249994).

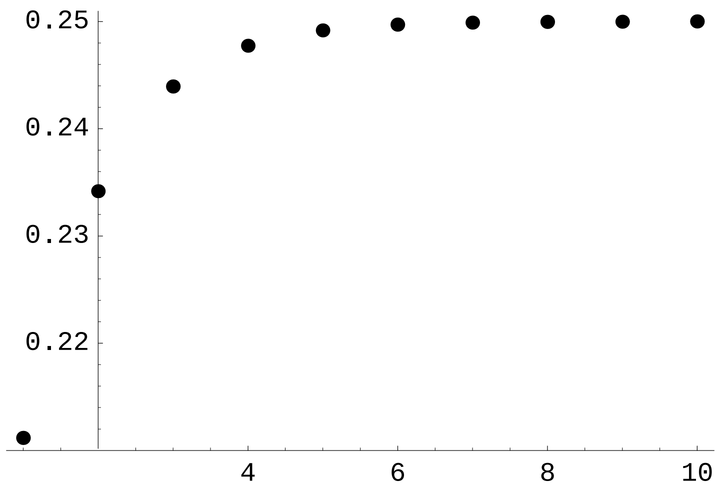
In[72]:= `ListPlot[list3]`



Out[72]= `- Graphics -`

To see the points better, you can add the following `PlotStyle` option.

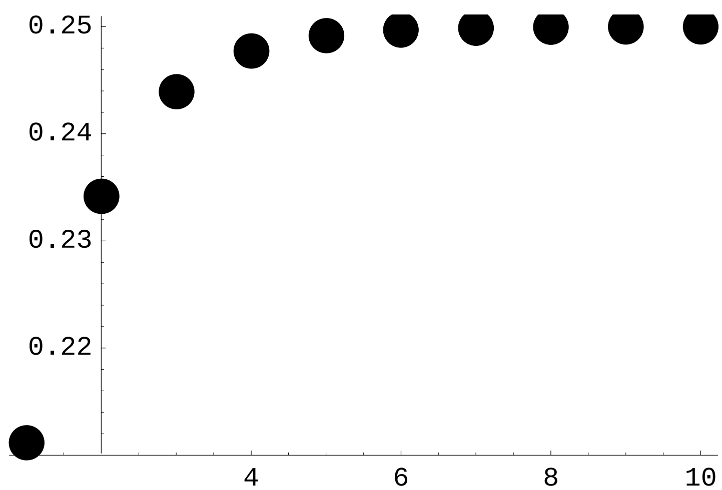
In[73]:= **ListPlot[list3, PlotStyle -> PointSize[.02]]**



Out[73]= - Graphics -

Changing the value of .02 to .05 in `PointSize` will increase the size of the points plotted.

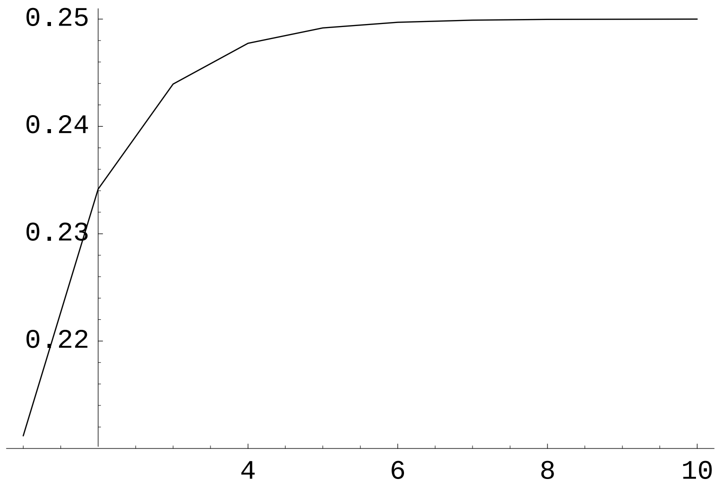
In[74]:= **ListPlot[list3, PlotStyle -> PointSize[.05]]**



Out[74]= - Graphics -

Use the option `PlotJoined -> True` if you want the points connected together.

In[75]:= **ListPlot[list3, PlotJoined -> True]**



Out[75]= - Graphics -

Extracting Parts of a List

Here is an example of a list containing the number 7 followed by two lists.

In[76]:= **newlist = {7, {2, 4, 6}, {5, 6, 9}}**

Out[76]= {7, {2, 4, 6}, {5, 6, 9}}

Double square brackets, immediately following the name of a list, are used to extract elements of a list. Executing `newlist[[i]]` will extract the ith object in the list.

In[77]:= **newlist[[1]]**

Out[77]= 7

In[78]:= **newlist[[2]]**

Out[78]= {2, 4, 6}

In[79]:= **newlist[[3]]**

Out[79]= {5, 6, 9}

Use `newlist[[i, j]]` to extract the jth item in the ith object of the list. For example, the number 6 is the 3rd item in 2nd object of `newlist`.

In[80]:= **newlist[[2, 3]]**

Out[80]= 6

As another example, study how the number 5 is extracted below.

In[81]:= **newlist[[3, 1]]**

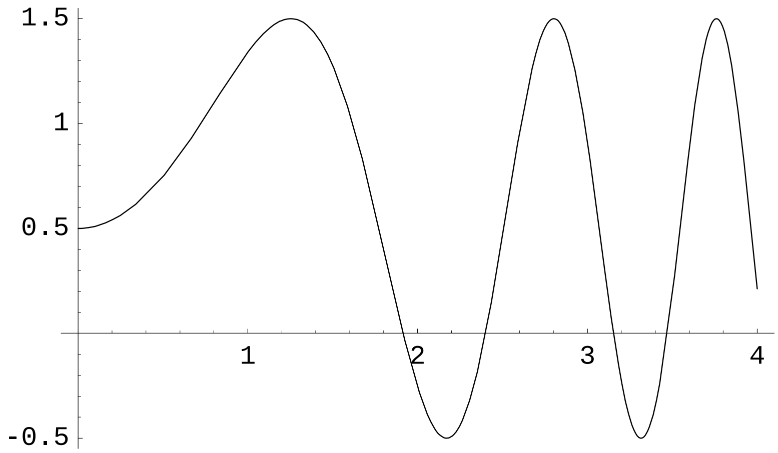
Out[81]= 5

As you will see in the next subsection, extracting objects from lists is very useful when solving equations.

■ Equation Solving

Suppose, for example, you wanted to find the solution to $\sin(x^2) + \frac{1}{2} = 0$ near $x = 0.5$. We could begin by graphing to function $f(x) = \sin(x^2) + \frac{1}{2}$ to see the roots of the equation.

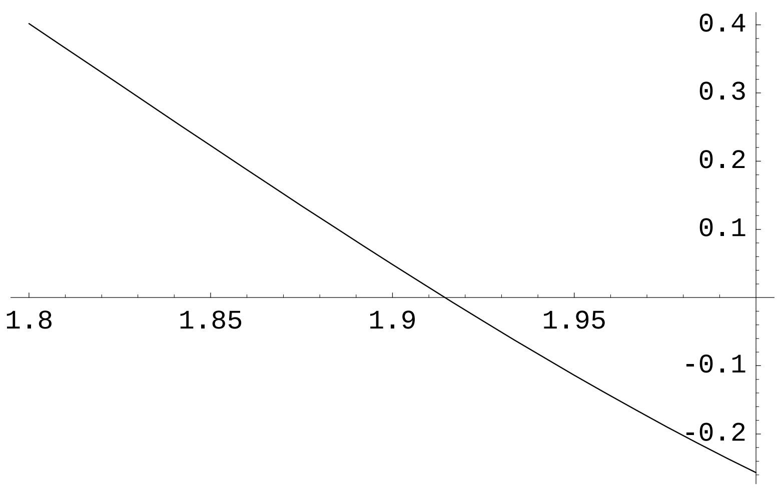
In[82]:=
```
Clear[f];
f[x_] = Sin[x^2] + .5;
Plot[f[x], {x, 0, 4}]
```



Out[84]= - Graphics -

We could zoom in on the root near 2 by changing the plot values.

In[85]:=
```
Plot[f[x], {x, 1.8, 2}]
```



Out[85]= - Graphics -

The *Mathematica* command `FindRoot[f[x] == 0, {x, x0}]` will attempt to find the approximate root of $f(x) = 0$ near $x = x_0$. Since *Mathematica* interprets an equals sign (=) as an assignment command, a double equals (==) must be used when representing an equation in an input statement.

```
In[86]:= sol = FindRoot[f[x] == 0, {x, 1.9}]
Out[86]= {x → 1.91447}
```

The output, assigned the name *sol*, is a list {x→1.91447} which is considered to contain one object with two parts, namely x and 1.91447. Entering sol[[1,1]] will extract x and sol[[1,2]] will extract the solution 1.91447.

```
In[87]:= sol[[1, 1]]
Out[87]= x

In[88]:= sol[[1, 2]]
Out[88]= 1.91447
```

The command NSolve can be used to find the approximate roots of a polynomial equations. Consider the following example.

```
In[89]:= newsol = NSolve[-x^3 + .7 x - .2 == 0, x]
Out[89]= {{x → -0.953778}, {x → 0.343732}, {x → 0.610046}}
```

To access the list containing the third solution, the following command is executed.

```
In[90]:= newsol[[3]]
Out[90]= {x → 0.610046}
```

To access the object contained in the third list, enter newsol[[3,1]].

```
In[91]:= newsol[[3, 1]]
Out[91]= x → 0.610046
```

Executing newsol[[3,1,2]] will execute the numerical value of the solution.

```
In[92]:= newsol[[3, 1, 2]]
Out[92]= 0.610046
```

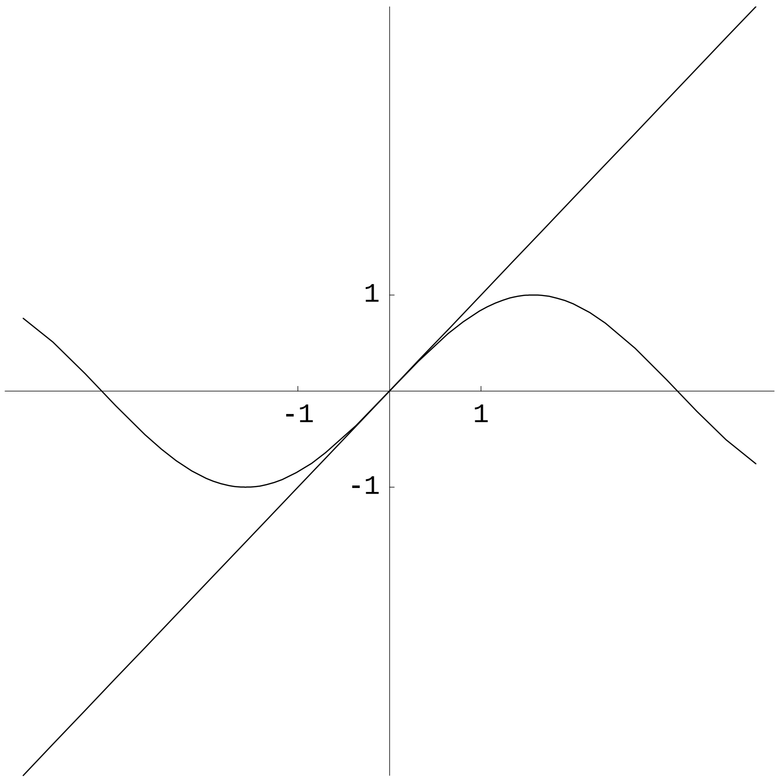
■ Animations

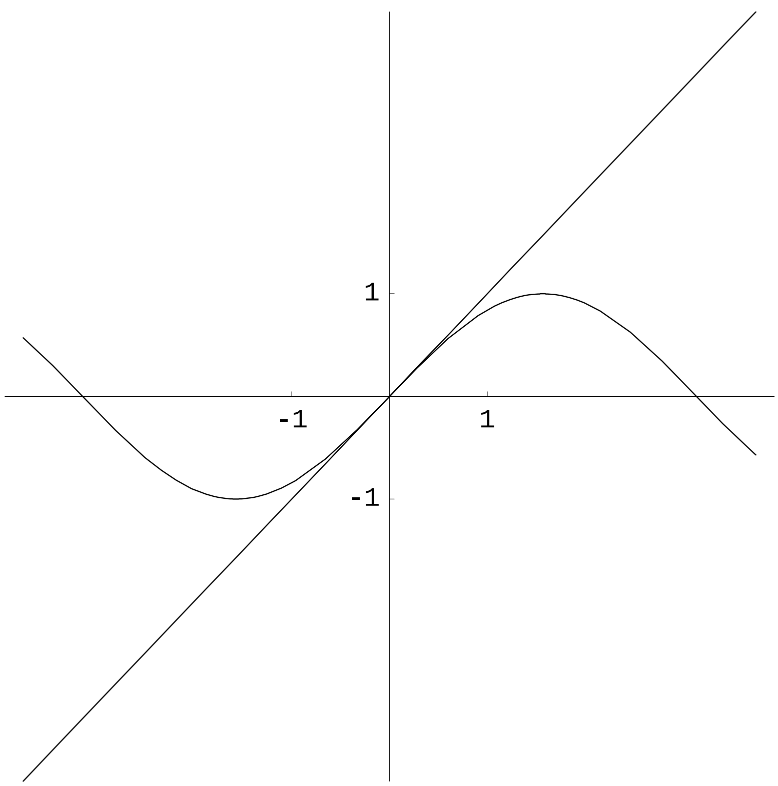
The Table and Plot commands introduced earlier can be used to form a sequence of frames which can then be animated. The command Plot[{f[x], g[x]},{x, a, b}] will simultaneously plot $y = f(x)$ and $y = g(x)$. The plot option Ticks -> {{-1, 0, 1}, {-1, 0, 1}} instructs *Mathematica* to only label the numbers $x = -1,\ 0,\ 1$ and $y = -1,\ 0,\ 1$ on the x- and y-axis, respectively and the option AspectRatio -> 1 specifies that the ratio of the height to width of the plot to be 1. In this example, the graphs of $y = x$ and $y = \sin(x)$ are simultaneously plotted for values of $x = -a$ to $x = a$. The Table command will execute the command

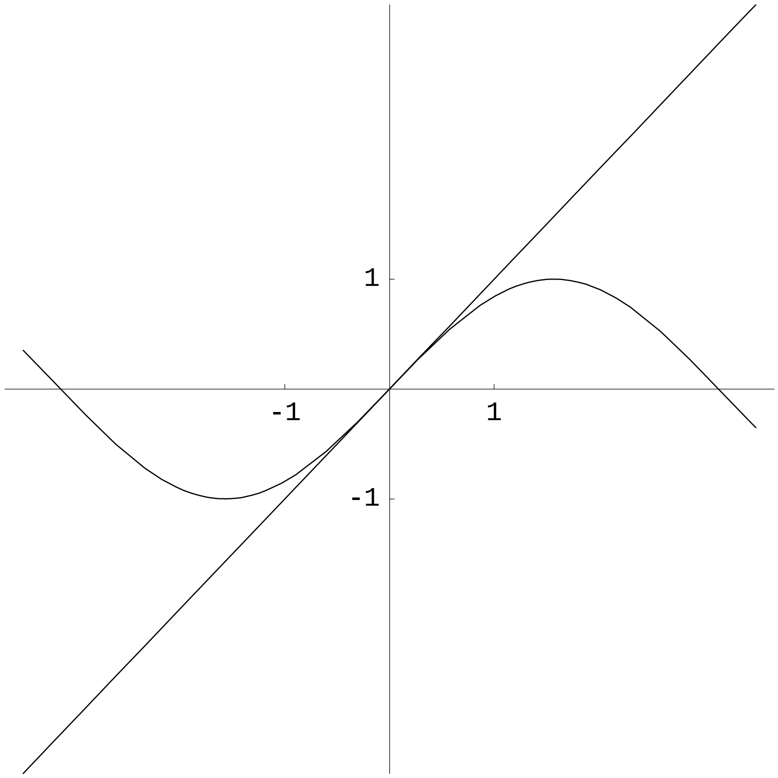
```
Plot[{x, Sin[x]}, {x, -a, a}, PlotRange → {-a, a},
 Ticks → {{-1, 0, 1}, {-1, 0, 1}}, AspectRatio → 1]
```

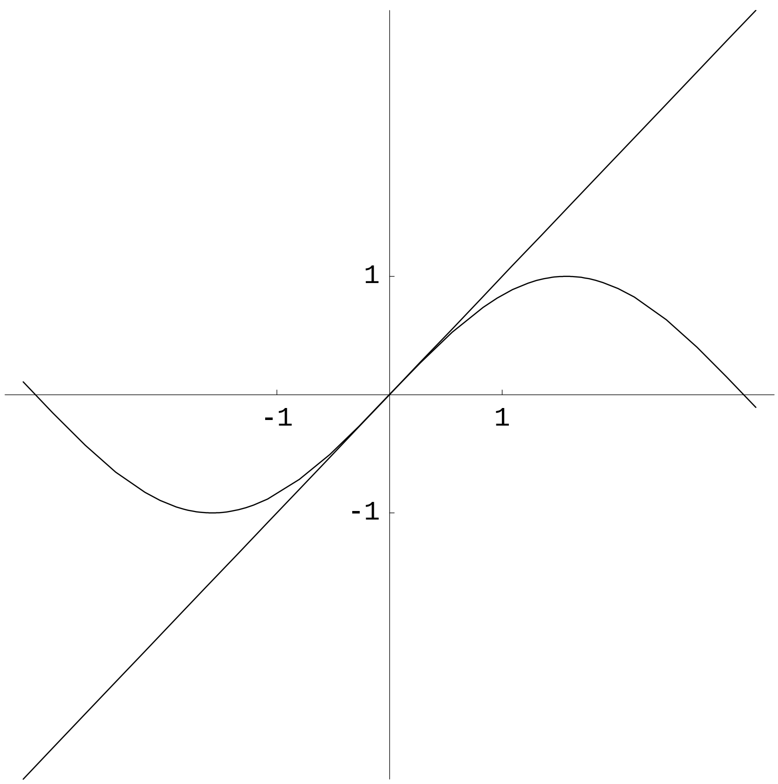
for a value of $a = 4$, then again for $a = 3.75$, and so forth all the way down to 1.

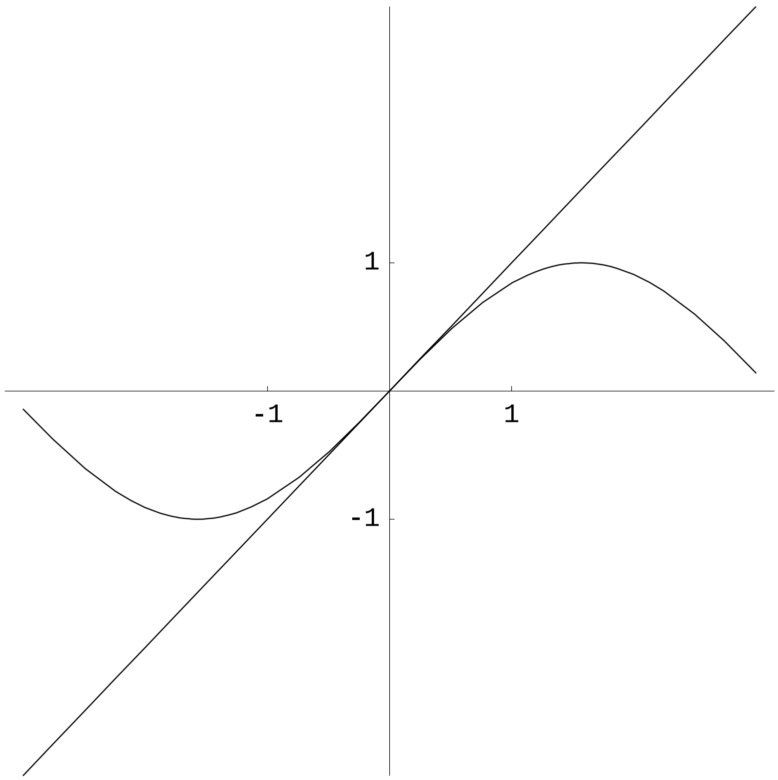
```
In[93]:= frames = Table[Plot[{x, Sin[x]}, {x, -a, a}, PlotRange → {-a, a},
            Ticks → {{-1, 0, 1}, {-1, 0, 1}}, AspectRatio → 1], {a, 4, 1, -.25}];
```

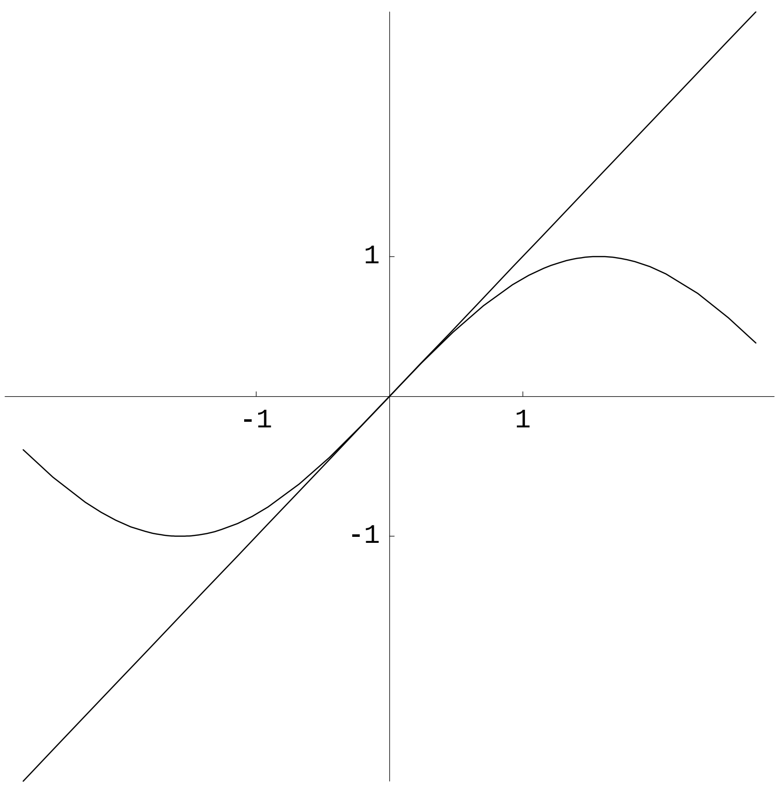

1
-1
1
-1

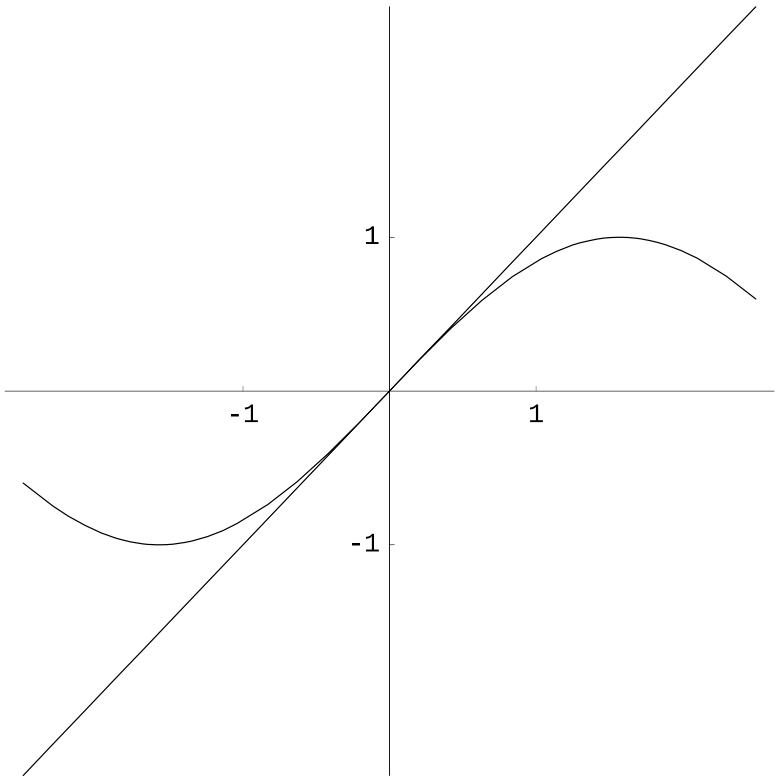

1
-1
1
-1

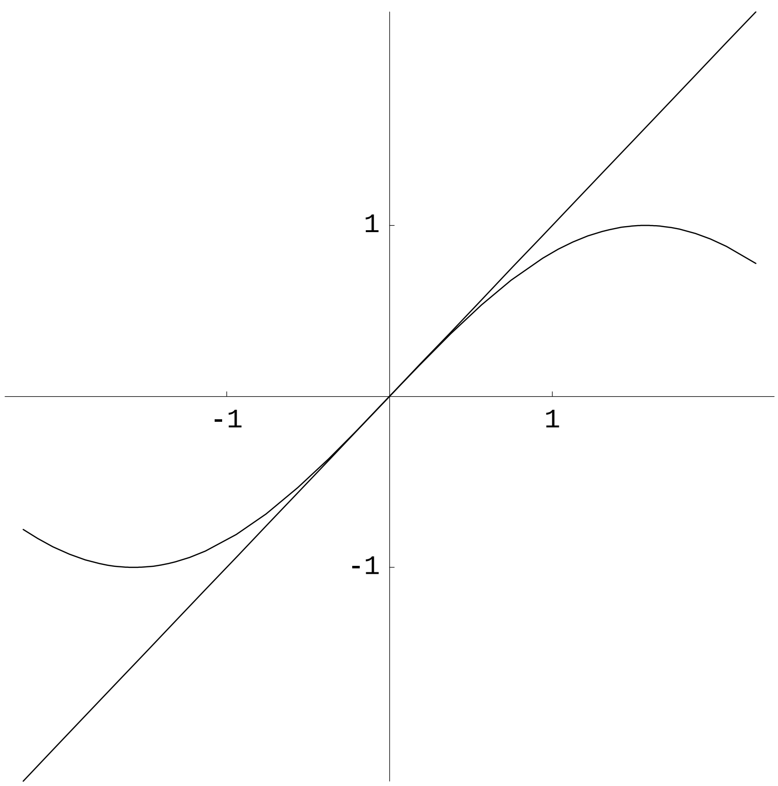

1
-1
1
-1

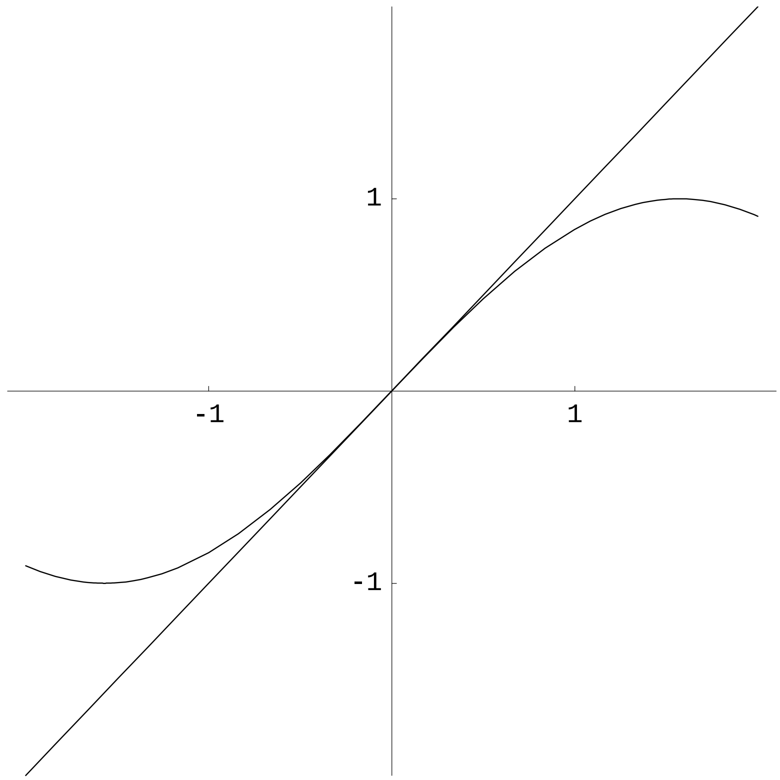

1
-1
1
-1

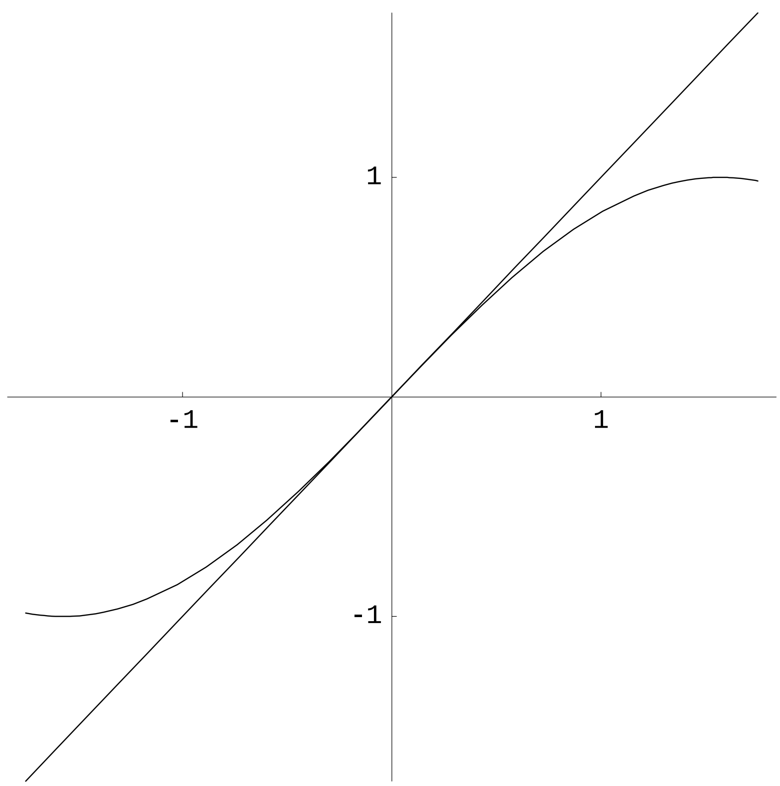

1
-1
1
-1

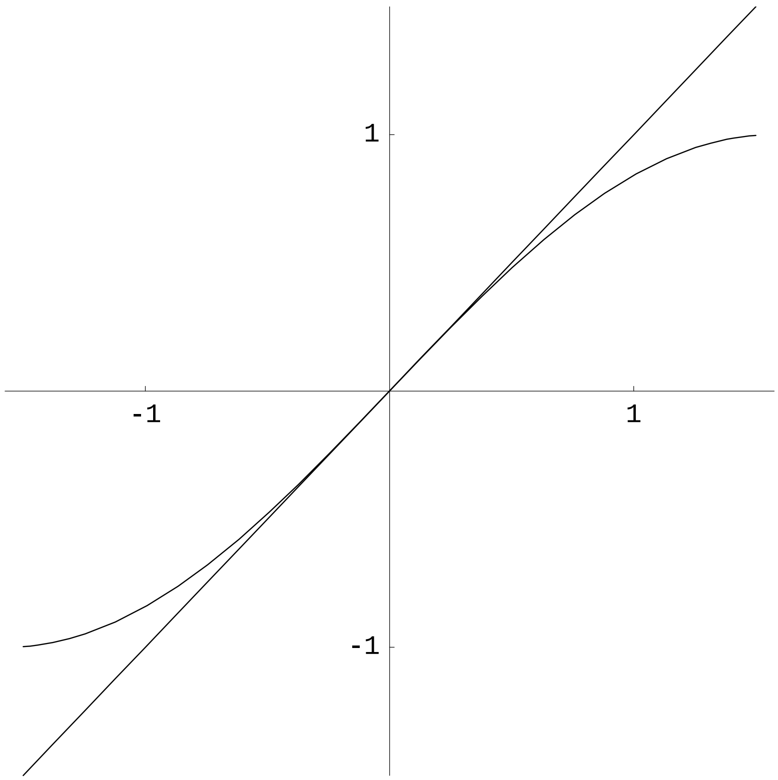

1
-1
1
-1

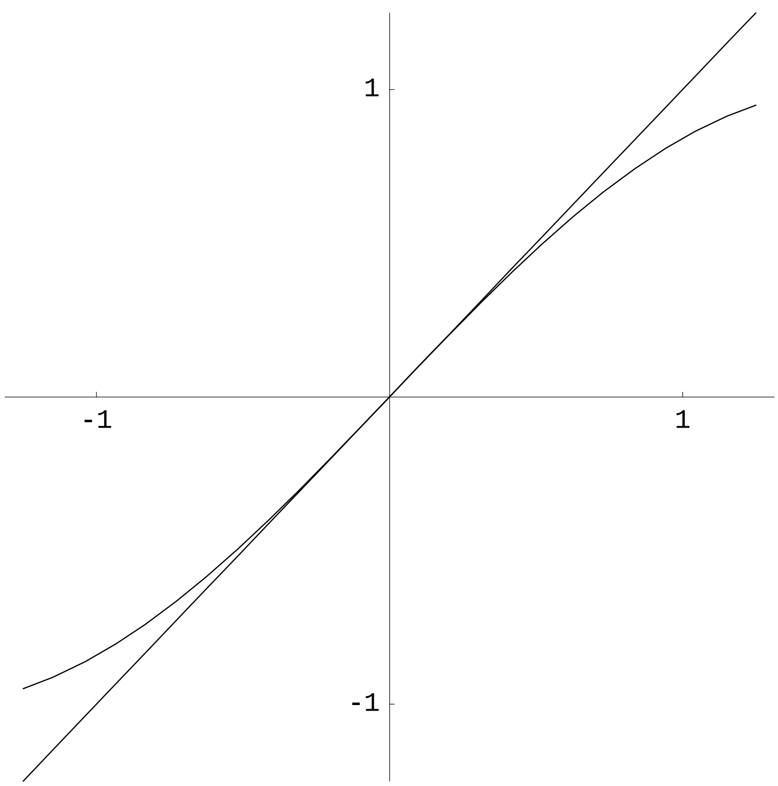

1
-1
1
-1

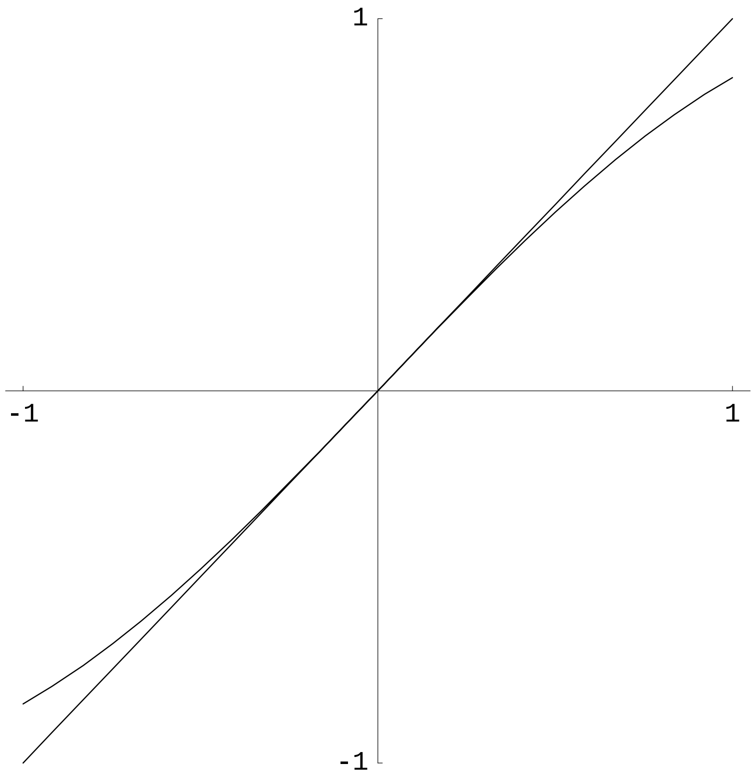

1
-1
1
-1


1
-1
1
-1


1
-1
1
-1


1
-1
1
-1


1
-1
1
-1



To animate the frames above, double-click on any one of the frames. When the frames are animated, you will notice some control buttons in the bottom left-hand corner of the screen. You might want to try out each of these buttons to perform such actions as reversing the direction of the animation and pausing the animation. To end the animation, click anywhere outside the animated frames. To learn more about animation, consult the Help menu.

■ Proper Use of Brackets { ...}, [...], [[...]] and (...)

Four types of brackets appear in the *Mathematica* computing environment: { ... }, [...], [[...]] and (...). Parenthesis (...) are used for grouping numerical or algebraic expressions.

```
In[94]:= 6000 - (5 6) * (10 70)

Out[94]= -15000
```

Square brackets [...] are used in functions.

```
In[95]:= Log[256]

Out[95]= Log[256]
```

Double square brackets [[...]] are used to extract objects from a list.

```
In[96]:= famousConstants = {0, 1, π, ⅇ};
         famousConstants[[3]]

Out[97]= π
```

Curly brackets { ...} are used in lists.

```
In[98]:= favoritenumbers = {7, 16, π}
Out[98]= {7, 16, π}
```

DO NOT interchange these symbols or else you will get an error message or undesired results. For example, *Mathematica* interprets the following input to mean a variable called Log times 256 instead of the natural logarithm of 256. Square brackets [...] should have been used here.

```
In[99]:= Log (256)
Out[99]= 256 Log
```

■ Common Mistakes

Forgetting a Space when Multiplying

In the following input cell, values are assigned to *x* and *y*, then the product *xy* is intended to be computed. But the product is not computed since a space is not placed between the *x* and the *y*. *Mathematica* interprets `xy` as an unassigned variable or undefined function.

```
In[100]:= x = 2; y = 3;
          xy
Out[101]= xy
```

By inserting a space between the x and the y, the correct output is obtained.

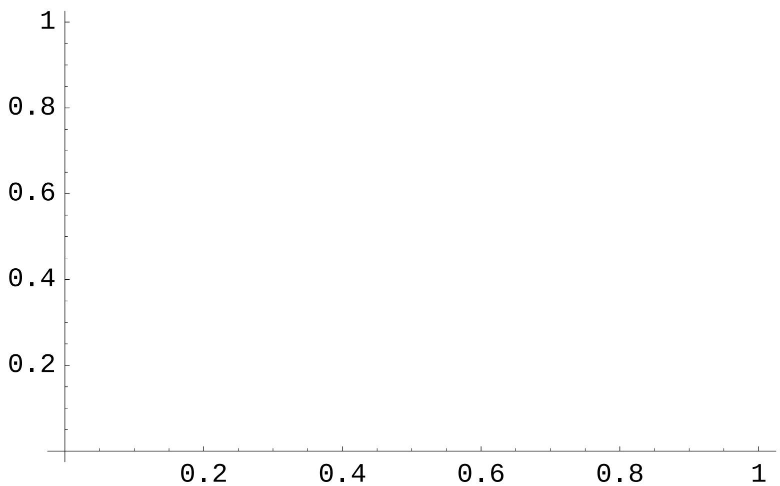
```
In[102]:= x y
Out[102]= 6
```

Failing to Capitalize

When the following input cell is executed, error messages are obtained.

In[103]:= **Plot[sin[x], {x, 0, 2 π}]**

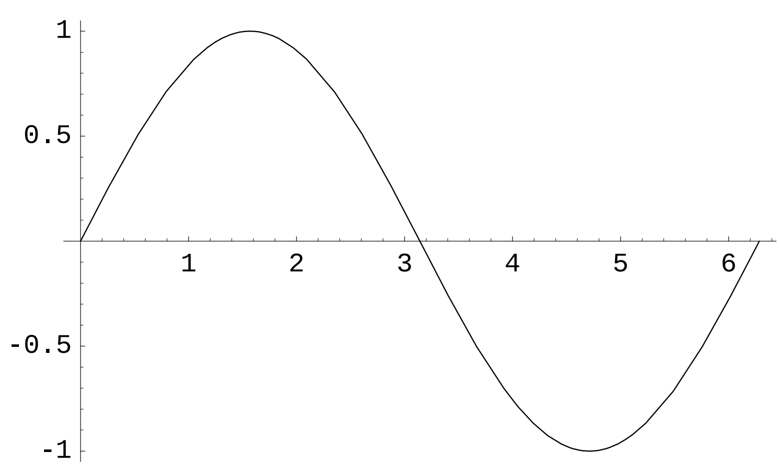
```
General::spell1 : Possible spelling error: new
   symbol name "sin" is similar to existing symbol "Sin".

Plot::plnr :
 sin[x] is not a machine-size real number at x = 2.617993877991494`*^-7.

Plot::plnr :
 sin[x] is not a machine-size real number at x = 0.25488992540742256`.

Plot::plnr :
 sin[x] is not a machine-size real number at x = 0.5328694051959509`.

General::stop :
 Further output of Plot::plnr will be suppressed during this calculation.
```



Out[103]= - Graphics -

As indicated by the first part of the error message above, the sine function was not properly not capitalized. All *Mathematica* functions and commands begin with a capital letter. Here is the correct input and output:

In[104]:= **Plot[Sin[x], {x, 0, 2 π}]**



Out[104]= - Graphics -

Using Reserved *Mathematica* Words

The following assignment statement attempts to assign the value of 5 to the letter N.

```
In[105]:= N = 5

          Set::wrsym : Symbol N is Protected.

Out[105]= 5
```

An error occurred because N is a reserved *Mathematica* command which finds the numerical approximation of a given number. *Since all reserved Mathematica words begin with a capital letter, always begin any of your own user-defined names with a lower case letter.*

Misinterpreting a Warning Messages as an Error Message

In the following, the name roots is assigned to the list {1, 2, 3}. Upon executing this assignment statement, a blue warning message appears. This is not an error message, but rather it is a warning message just stating that the name roots is similar to the name of a reserved *Mathematica* command. In this case, you can ignore the message and continue on with your work. You can also be certain that you did not use a reserved *Mathematica* word since you used the reserved word *roots* starting with a lower case letter r.

```
In[106]:= roots = {1, 2, 3}

          General::spell1 : Possible spelling error: new
             symbol name "roots" is similar to existing symbol "Roots".

Out[106]= {1, 2, 3}
```

Using an Add-on Command or Constant Before Loading it into Memory

An add-on package, called PhysicalConstants, contains a constant called SpeedOfLight. If we attempt to access this constant before loading the package into memory, nothing happens.

```
In[107]:= SpeedOfLight

Out[107]= SpeedOfLight
```

Now if we load the package, found in the file Miscellaneous, a warning message is displayed and if we again attempt to access the constant, nothing happens.

```
In[108]:= << Miscellaneous`PhysicalConstants`

          SpeedOfLight::shdw : Symbol SpeedOfLight appears in
             multiple contexts {Miscellaneous`PhysicalConstants`, Global`};
             definitions in context Miscellaneous`PhysicalConstants` may
             shadow or be shadowed by other definitions.

In[109]:= SpeedOfLight

Out[109]= SpeedOfLight
```

In order to access the speed of light constant from the package, first use the Remove command as shown in the following input cell.

In[110]:= **Remove[SpeedOfLight]**

In[111]:= **SpeedOfLight**

Out[111]= $\frac{299792458\ \text{Meter}}{\text{Second}}$

■ Where Do You Go From Here?

This short introduction to *Mathematica* in now concluded. There are three other files which your instructor may ask you to study as you begin using *Mathematica* in your course work:

Derivatives.nb: This notebook contains information on using *Mathematica* to investigate ideas found in chapters 2 - 4 of the text concerning differentiation and its applications.

Integrals.nb: This file shows you how to use *Mathematica* to perform integration and to study concepts and ideas found in chapters 5 - 8.

Multivariable.nb: A *Mathematica* notebook dealing with topics covered in the remaining part of the text.

In addition to using these notebook files, the Help menu is an excellent resource for finding information on computing with *Mathematica*. The entire *Mathematica* Book is online, so any information you need regarding *Mathematica* is available under **Help** in the Help menu.