

Introduction

This introduction to *Mathematica* was written by Colin Gavin.

Mathematica as a Calculator

You can do almost anything in *Mathematica*, but at its most basic, *Mathematica* functions somewhat like a calculator. You enter mathematical commands; they're evaluate; and the results are displayed. Here's an example:

```
3+4*5+8/4
```

When you hit enter (or Shift+Return) you should see the following

```
25
```

To enter a command, click in an empty part of the notebook and you will see a horizontal blinking cursor. You can begin typing your command and when you have finished, press Shift+Return to evaluate it. Once you have evaluated a command, you can come back to it at any time, modify it, and then evaluate it again.

Exercise: Enter a mathematical command like the one above evaluate it.

Most mathematical expressions can be entered into *Mathematica* the way you would expect. For example + for addition, - for subtraction, * for multiplication, / for division, etc. *Mathematica* understands the standard order of operations, so $3+4/2$ is interpreted as $3+(4/2)$ instead of $(3+4)/2$. If you want to input the latter expression, use parenthesis. For example, the input

```
3+4/2
```

yields

```
5
```

but (with parenthesis) the input

```
(3+4)/2
```

yields

```
7/2
```

Exercise: Evaluate the expression $\frac{2^4+1}{3}$.

Mathematica can also evaluate familiar mathematical functions like sin, cos, tan, and take square roots, as the following example demonstrates.

```
Sin[Pi/3] + Cos[Pi] + Sqrt[2]
```

```
-1 + Sqrt[2] + Sqrt[3]/2
```

This example demonstrates several important details about functions in *Mathematica*. First, functions *always* are written with a capital letter in *Mathematica*, even if they would not be in normal mathematics. Also, *Mathematica* uses square brackets (“[“ and “]”) to indicate applying a function. Furthermore, note that trigonometric functions expect their arguments to be in radians in *Mathematica*. Finally, we see in this example that *Mathematica* knows constants like π and e , which can be entered as Pi and E, respectively.

Exercise: Evaluate the expression $\sin\left(\frac{\pi}{4} + \frac{\pi}{3}\right)$. (Note that *Mathematica* gives the result in exact form, without decimals.)

One useful tool when using *Mathematica* for calculations is the ability to store the results of steps in variables. This way, you can break up a long calculation into steps, and combine the individual results. Here’s an example:

```
width = 3*Sin[Pi/3];
height = 4*Cos[Pi/3];
area = width*height
```

```
3*sqrt(3)
```

As this example shows, you can give a name to the result of any calculation by using an “=” sign. The names of variables must not have spaces or underscores and must

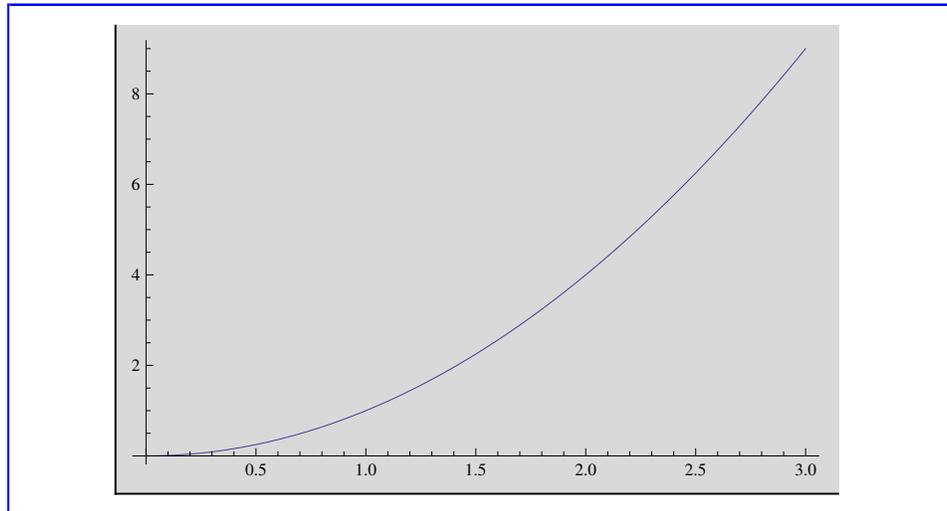
start with a lowercase letter. Also note that I have ended the first two lines of the example with semi-colons in order to prevent the results of these lines from being shown in the output.

Exercise: Compute the hypotenuse of a right triangle whose side lengths are $a = \sqrt{3 * 11 - 1}$ and $b = 6\sqrt{\cos\left(\frac{\pi}{3}\right)}$. Use variables store the side lengths in order to simplify the computation.

Plotting

Now that we have seen how to perform simple computations using *Mathematica*, we can move ahead and use *Mathematica* to graph functions. The most important function that you will learn in this section is called “Plot”. It takes a given function (or list of functions) and plots it on a given range. Here’s the most basic example of Plot:

```
Plot[x^2, {x, 0, 3}]
```

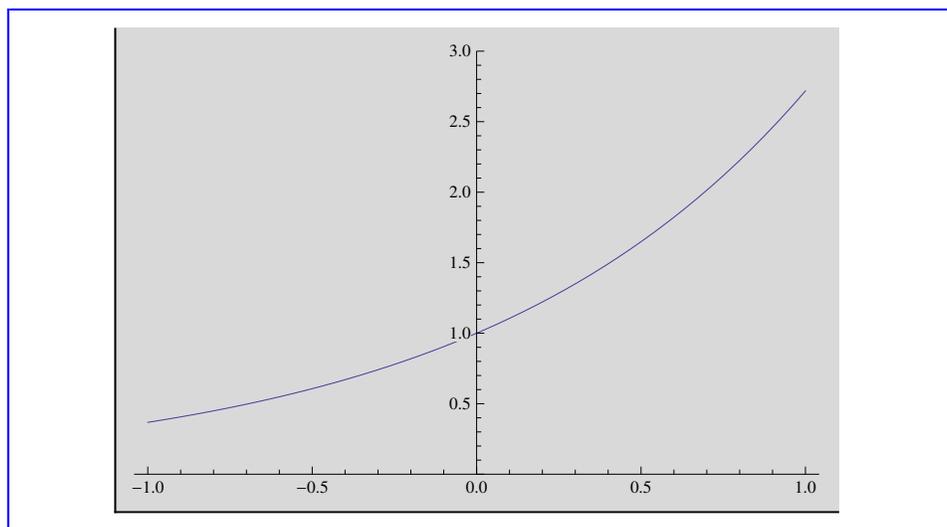


Notice that Plot is written with square braces, just like Sin and Cos. The first argument gives the function to be plotted. You write this just like any other mathematical expression in *Mathematica*, but here you can use a variable such as x to indicate the independent variable of the function that you’re plotting. The second argument is a *range*, which is written using curly brackets (“{“ and “}”). It specifies that the variable to plot is x and that the plot should start at $x = 0$ and go to $x = 3$.

Exercise: Plot the function x^3 from $x = -1$ to $x = 1$.

The function Plot also takes many options which can change how the plot looks. Here's an example of one particularly useful option called PlotRange:

```
Plot[Exp[x], {x, -1, 1}, PlotRange -> {0, 3}]
```



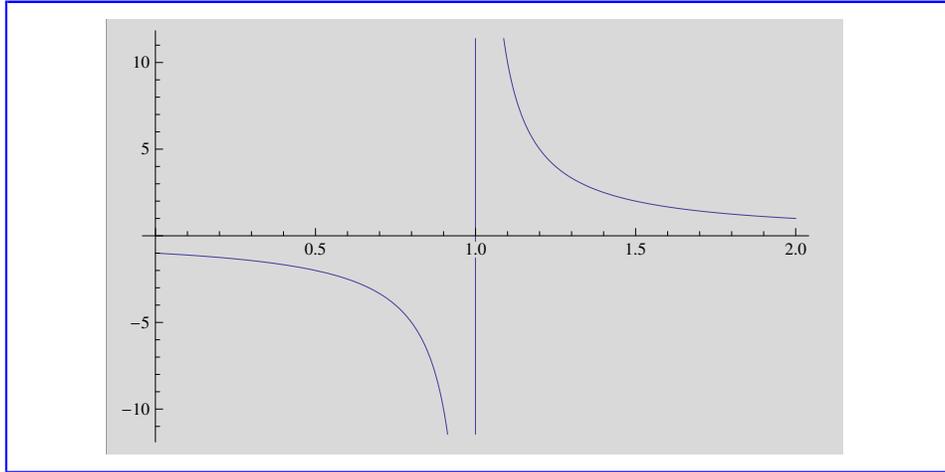
As you can see, the PlotRange option specifies the range of the y-axis to show in the plot. Here it is between $y = 0$ and $y = 3$. In general, options are added after the range and are entered in the form “OptionName \rightarrow Value”. The symbol “ \rightarrow ” is entered by typing “->” (a hyphen and then a ‘greater than’ sign), which *Mathematica* will automatically convert to a little arrow.

Because there are so many options, I will not describe them all. Instead, this is a good opportunity to learn how to use *Mathematica*'s extensive reference manual. To access the reference, click on the “Help > Documentation Center” menu. This will open a window where you can search for any of the functions built into *Mathematica*. Typing “Plot” into the search bar should bring you to the reference page for the Plot function. This page gives a great deal of information about the Plot function, including many examples. In particular, there is a section called “Options” which gives details and examples on all the options that Plot accepts.

Exercise: Using the Documentation Center's page on “Plot” as a reference, plot the function $\sin(x)$ on the domain $0 \leq x \leq 2\pi$ in red. *Hint:* read about the PlotStyle option.

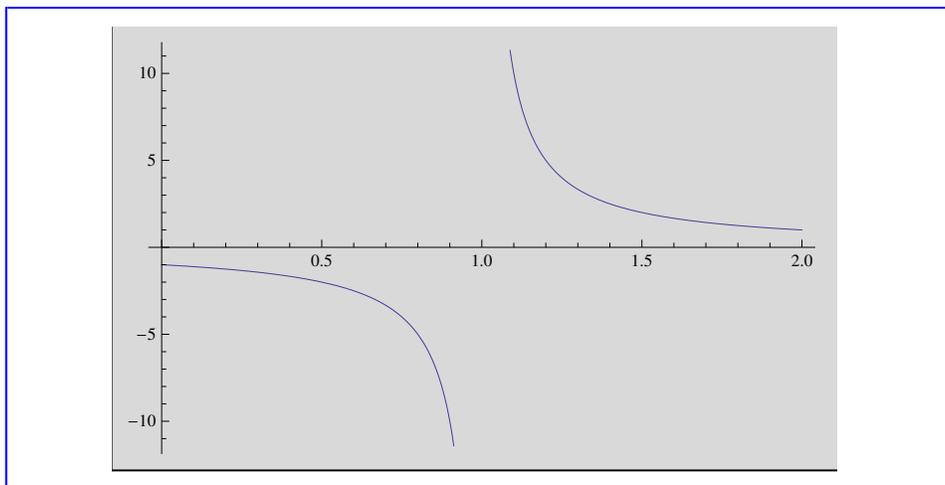
One very useful option that I will discuss is “Exclusions”. First, here’s an example of a case when “Exclusions” is necessary:

```
Plot[1/(x-1), {x, 0, 2}]
```



The vertical line visible in this plot is not actually part of the graph and is present because the value of $\frac{1}{x-1}$ jumps at the point $x = 1$. To remove the vertical line, we can use “Exclusions” to ensure that Plot does not consider the point $x = 1$ to be part of the domain of the function:

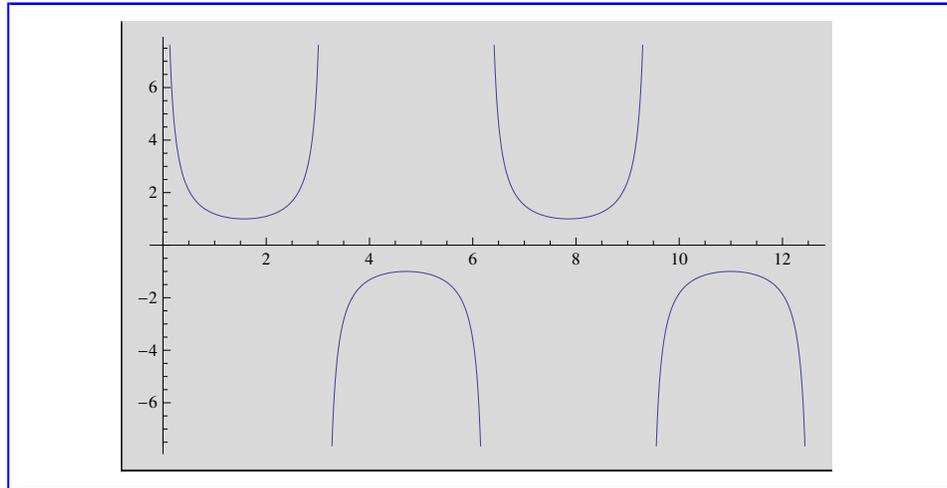
```
Plot[1/(x - 1), {x, 0, 2}, Exclusions -> 1]
```



We can also specify a list of points to exclude from the domain. Here, I plot the function $\frac{1}{\sin(x)}$ which jumps at multiples of π . Thus, I need to specify that any

multiple of π from 0 to 4π should be excluded from the domain. To do this, I give a list of exclusions.

```
Plot[1/Sin[x], {x, 0, 4Pi},
      Exclusions -> {0, Pi, 2Pi, 3Pi, 4Pi}]
```



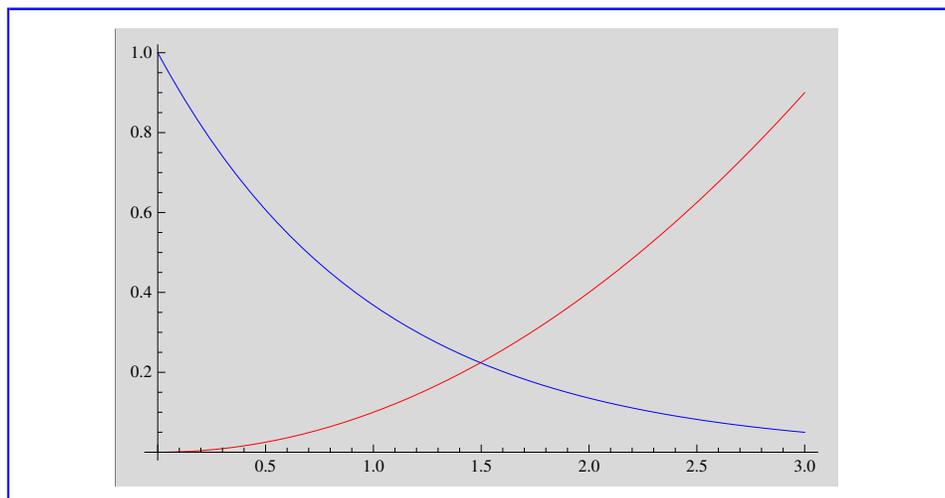
Notice that to specify a list of values, I use curly braces and commas.

Exercise: Modify the code below to exclude points from the domain where the function's value jumps.

```
Plot[1/((x-2)(x-3)(x-4)) , {x, 0, 5}]
```

Using our new knowledge of how to specify lists, we can plot multiple functions at once.

```
Plot[{x^2/10, Exp[-x]}, {x, 0, 3},
      PlotStyle -> {Red, Blue}]
```



As you can see, this is as simple as giving a list of functions to be plotted instead of a single function. Notice that the independent variable must be the same in each of the functions. As you can see in this example, we can also specify different options for each of the functions being plotted by giving a list of values for the option. Here, in the `PlotStyle` option `Red` refers to the plot of $\frac{x^2}{10}$ while `Blue` refers to the plot of e^{-x} .

Exercise: Plot the functions $\cos(x)$ and $\tan(x)$ from $x = 0$ to $x = 4\pi$ on the same graph. Be sure to remove any jumps that occur.

Functions

We can define mathematical functions like $f(x) = 2x$ and $g(x) = \sin(3x)$ easily in *Mathematica*. This frees us from constantly typing the same expression, and makes substituting in values very easy. In this section, we'll learn several ways to define functions and then return to plotting to show how they can be used. Here's a simple example to start with:

```
f[x_]:=2x + 1;
f[3]
```

The output will be

```
7
```

This defines a new function called `f` which works like the mathematical function $f(x) = 2x + 1$. As we can see, evaluating `f[3]` gives 7 as we expect. There are

some important things to notice about how we define functions. First, the arguments on the left hand side must be followed by underscores (“_”). Here, this indicates that “x” should be treated as a placeholder variable. Also, the colon before the equals sign is required to indicate that we are defining a function.

Exercise: Define the function $g(x) = \sin(3x)$ and evaluate it at $x = \frac{\pi}{6}$.

We can also define functions that take two or more arguments in a similar way. Here’s an example:

```
h[x_, y_] := Sqrt[x^2 + y^2];
h[3, 4]
```

Exercise: Define the function $b(u, v) = \sin(u) \cos(v)$ and evaluate it at $u = \frac{\pi}{3}$, $v = \frac{\pi}{4}$.

Another common task in calculus and other areas of math is to use piecewise functions, where different formulas are used depending on the arguments. For example, we can define a function $w(x)$ that is equal to x^2 when $0 \leq x \leq 2$ and is equal to $2x$ when $x > 2$. Here’s how to enter such a function in *Mathematica*:

```
w[x_] := Piecewise[{{x^2, 0 ? x ? 2}, {2x, x > 2}}];
{w[3/2], w[3]}
```

The output will be

```
{9/4, 6}
```

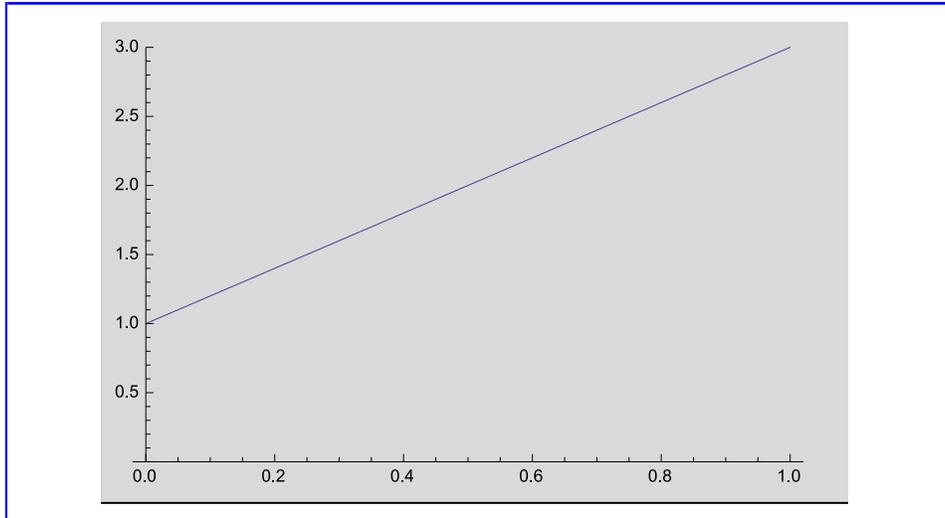
This uses the built-in function `Piecewise`, which takes a list of pairs {output, condition} and returns the output for the condition that is true. So, in our example, when we evaluate `w[3/2]`, the condition $0 \leq 3/2 \leq 2$ is true so `Piecewise` returns $(3/2)^2$ which is equal to $9/4$.

Exercise: Define your own function $q(x)$ which is equal to $x^3 - 1$ when $x \leq 0$ and is equal to $x - 1$ when $x > 0$. Evaluate $q(-2)$.

Now that we’ve defined some functions, let’s plot them. This is just like plotting a built-in function like `Sin` or `Sqrt`.

```
Plot[f[x], {x, 0, 1}, PlotRange -> {0, 3}]
```

The result is

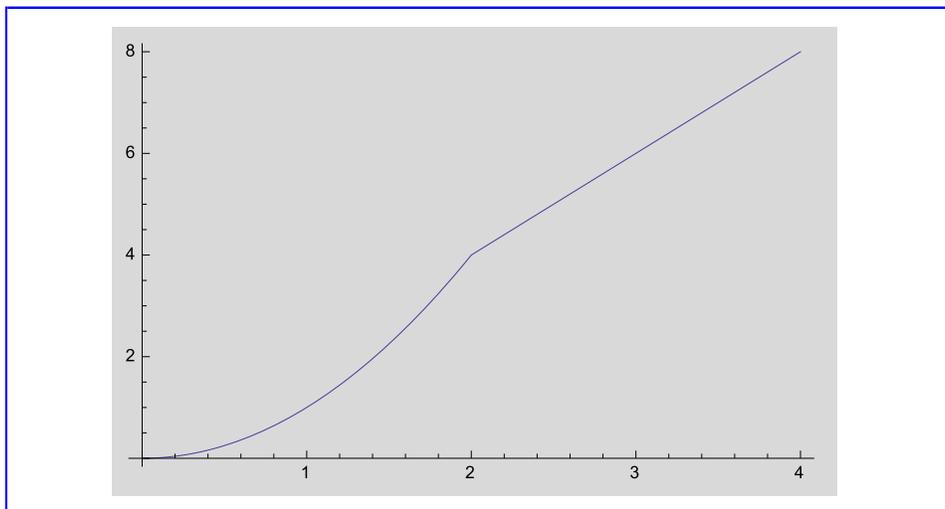


Exercise: Plot the function $g(x)$ that you defined above from $x = 0$ to $x = 2\pi$.

We can also plot piecewise functions in the exact same way:

```
Plot[w[x], {x, 0, 4}]
```

The result is



Exercise: Plot the function $q(x)$ that you defined above on the range $x = -1$ to $x = 1$.

Algebra and Symbolic Manipulation

You might have noticed that in the previous part when we graphed functions we gave them arguments that weren't actually numbers. Instead, we wrote things like $w[x]$ where x is what is called a symbol. This is like a variable in normal mathematics: something which stands for an undetermined value. *Mathematica* is very good at manipulating such symbolic expressions. Suppose we wanted to make the substitution $x = \frac{1}{2}u$ in our function $f(x)$ that we defined above. *Mathematica* can do this and simplify the answer for us:

```
Simplify[f[1/2 u]]
```

The output is

```
1 + u
```

As we expected we get the answer $1 + u$ because $f\left(\frac{1}{2}u\right) = 2\left(\frac{1}{2}u\right) + 1 = u + 1$. The important thing to notice is that we never gave u a value in this computation. We used the function `Simplify` here to make sure that *Mathematica* simplified the resulting expression as much as it could.

Exercise: Make the substitution $x = u^2$ in the function $q(x)$ that you defined above. Be sure to simplify. Is this the result you were expecting?

Another common algebraic task that *Mathematica* excels at is solving equations. Here's an example:

```
Solve[f[x] == 0, x]
```

The result is

```
( x -> -1/2 )
```

This finds the value of x such that $f(x) = 0$. As we expect, there is one answer $x = -\frac{1}{2}$. The function `Solve` that we used here takes two arguments. The first is an equation, or a list of equations, to solve. The second is a variable, or a list of variables, that will be solved for.

We didn't necessarily have to use a function in the arguments to `Solve`. Instead, we can write out an equation directly.

```
quadraticSolutions=Solve[x^2-5x+6==0, x]
```

The result is

$$\left(\begin{array}{l} x \rightarrow 2 \\ x \rightarrow 3 \end{array} \right)$$

Because this equation was a quadratic, it gave us two solutions: $x = 3$ and $x = 2$.

Exercise: Find the roots of the polynomial $-12 + 20x - 7x^2 - 2x^3 + x^4$.

You might be wondering what we can do with the answers that Solve gives us. For instance, can I substitute the results into another equation or use them as arguments to a function? Yes, but we have to use what's called replacement. The answers that solve gives are in the form of lists of rules, which specify that a variable should become a given value. To apply such rules we use the `/.` command as follows:

```
x^2/.quadraticSolutions
```

The result of which is

```
{4,9}
```

What happened here? When I solve the quadratic equation above, I stored its answer in the variable `quadraticSolutions`. There are two rules solution, either x becomes 2 or x becomes 3. *Mathematica* went through consecutively and set the x in x^2 to 2, giving 4, and then to 3, giving 9. The result was a list of possible values that x^2 could become.

We can do the same thing with functions instead of directly entered expressions.

```
f[x]/.Solve[x^2-1==0,x]
```

The result is

```
{-1,3}
```

Here, there were two solutions for x : either $x = -1$ or $x = 1$. These were then given to f as arguments, giving $f(-1) = -1$ and $f(1) = 3$.

Exercise: Use *Mathematica* to solve the following problem: An orange weighs $\frac{2}{5}$ of a pound and costs \$0.76. If I buy 6 pounds of orange, how much will I spend?