

Maple Tutorial #2: Functions, Plots, and Equations*

Scott R. Fulton and Christos A. Xenophontos
Department of Mathematics and Computer Science
Clarkson University

Maple V Release 5
Fall 1999

Introduction

After working through the first tutorial, you should be able to use Maple as a calculator and work with simple algebra. You should know how to enter commands to Maple (and edit them if you make mistakes), and be able to get help when needed.

This barely scratches the surface of what Maple can do. In this second tutorial, we'll start to dig a bit deeper. The goal is to learn how to:

- Define and use functions
- Plot curves
- Solve equations

Along the way, you'll encounter Maple constructions known as sets and sequences. As before, you'll get the most out of this tutorial if you read the text carefully and *try* the commands as shown. After finishing this tutorial, you should have enough background to be able to explore Maple in whatever direction your interests and needs might dictate.

Functions

In mathematics, we say one quantity y is a function of another quantity x if each value of x has a unique value of y associated with it. Typically we write $y = f(x)$, where f is

*Technical Report 99-04, Department of Mathematics and Computer Science, Clarkson University, Potsdam, NY 13699-5815.

the name of the function. Often (but not always!) there is a formula which expresses this relationship between input and output.

Maple understands functions in exactly the same way. For example, to define the “square” function (which takes an input number x , multiplies it by itself, and returns x^2) you type:

```
> f := x -> x^2;
```

$$f := x \rightarrow x^2$$

Note carefully what’s going on here. On the left side of the assignment operator `:=` is the name `f` which you are giving to the function. The right side is the function itself, expressed as a mapping from input `x` to output `x^2` using the arrow operator `->` (minus sign followed by greater than sign with no space between).

You can plug values into a function easily: for example,

```
> f(1); f(-19); f(33.27);
```

$$1$$

$$361$$

$$1106.8929$$

If you want to see what the output of f is in general (that is, the “rule” which defines the function), type:

```
> f(x);
```

$$x^2$$

However, just typing `f`; simply tells you that `f` is itself (try it).

Note carefully: *there is a difference between an expression and a function. An expression* consists of constants and/or variables joined by operators (such as `+`, `-`, `*`, etc.):

```
> expr := 3*x^2 - 4*x - 5;
```

$$expr := 3x^2 - 4x - 5$$

whereas a *function* involves the idea of mapping input to output:

```
> g := x -> 3*x^2 - 4*x - 5;
```

$$g := x \rightarrow 3x^2 - 4x - 5$$

While an expression may be a bit easier to type, it’s often more convenient to use the corresponding function. For example, both of the above give the same result when evaluated at $x = 3.5$:

```
> subs( x=3.5, expr ); g(3.5);
```

$$17.75$$

17.75

but it's probably easier just to plug the number into the function directly. Be warned: forgetting the distinction between an expression and a function can lead to confusion!

Plots

While a function may be defined by a formula, often one of the best ways to understand the function is to plot it on a graph. Maple has many commands for plotting; here we'll just look at the simplest. The Maple command `plot` requires two input arguments: something to plot (usually an expression or a function), and the domain of the independent variable. For example, to plot the "square" function you defined above, type:

```
> plot( f(x), x=-5..5 );
```

Note that the domain is specified by giving the name of the independent variable and the minimum and maximum values to use (separated by two dots).

Shortly after the command is executed, the plot should appear and underneath it a new prompt. By clicking with the mouse on the plot, the plot menus appear (in place of the file menus). The plot menus give you some control over the appearance of the plot. One useful option is selecting "constrained" from the Projection menu; this forces both axes to use the same scaling (try it). You may want to experiment with the options in the other menus. When you're done with a plot, simply place the cursor on the next input (prompt) line.

When working with plots, don't leave all of the thinking to Maple. For example, suppose you want to understand the function

```
> h := t -> 4*t^3 - 3*t;
```

$$h := t \rightarrow 4t^3 - 3t$$

A first step would be to plot it. Try the following (this also specifies a title, which must be enclosed with either double or *backward* quotes—you may have to hunt to find the key marked ' on the keyboard), while the double quotes (") are usually on the left of the enter key :

```
> plot( h(t), t=-10..10, title="Domain Too Large" );
```

This looks a lot like the graph of $y = x^3$, so let's zoom in (by changing the domain) and look just near the origin:

```
> plot( h(t), t=-0.1..0.1, title="Domain Too Small" );
```

Now it looks like a straight line—what's going on? As the title says, now the domain is too small to see much of the function. When we finally find an appropriate domain, we get a better picture of what the function is like:

```
> plot( h(t), t=-2..2, title="Domain Just Right" );
```

Restricting the domain further would leave off too much information. However, we can still get a better picture of the behavior near the origin by specifying the range as another optional

parameter (rather than letting Maple choose it internally). The range must immediately follow the domain:

```
> plot( h(t), t=-2..2, y=-4..4, title="Restricted Range" );
```

The moral of this example is to be careful with plots: experiment until you are sure you understand what you're looking at. There are other options which allow you to further improve upon the plot (or just jazz it up); to explore these, try starting with the help screen for `plot[options]`. Maple also has many other plot commands; for information on these, consult the help browser.

Another common task is plotting more than one curve on the same graph. For example, consider the two functions f and g which you defined above: to see again what they were, you can type

```
> f(x); g(x);
```

$$x^2$$

$$3x^2 - 4x - 5$$

Is there a solution to the equation $f(x) = g(x)$? One way to see is to simply plot both:

```
> plot( {f(x),g(x)}, x=0..5 );
```

Note that the two expressions to be plotted are enclosed in curly braces `{ }`; this is an example of what Maple calls a set. Your plot should show one point where the two curves intersect. This gives you an (approximate) solution x of this equation. Are there any other solutions?

Equations

You've just seen a way to use Maple to obtain graphical information about solutions of equations. In addition, Maple can deal with equations analytically (symbolically) and numerically. To express an equation in Maple, simply type it in: for example,

```
> x^2 + 2*x - 3 = 0;
```

$$x^2 + 2x - 3 = 0$$

Note that an equation in Maple consists of two expressions (here, $x^2 + 2x - 3$ and 0) joined by an equals sign. Since you're going to work with this equation for a while, it helps to give it a label (recall the role of `%`):

```
> eqn1 := %;
```

$$eqn1 := x^2 + 2x - 3 = 0$$

While the command `factor` will work on this equation as it did for expressions (try it!), you can also solve the equation by simply saying

```
> solve(eqn1,x);
-3,1
```

You can verify that these solutions actually work by substituting them into the original equation:

```
> subs(x=1,eqn1); subs(x=-3,eqn1);
0 = 0
0 = 0
```

Sometimes it's handy to refer to just one side of an equation. For example, to change the right-hand side of the above equation to 2, we can construct a new equation using the command `lhs` to select just the left-hand side, and set it equal to 2:

```
> lhs(eqn1) = 2;
x2 + 2x - 3 = 2
```

(there is an analogous command `rhs` for the right-hand side). To solve this equation, type

```
> solve(%,x);
-1 + √6, -1 - √6
```

Note that it is possible to “nest” commands: the previous example could be expressed as

```
> solve( lhs(eqn1)=2, x );
-1 + √6, -1 - √6
```

Some people find this way simpler—do you?

You may have noticed that when Maple finds more than one solution of an equation, it returns them in a sequence. For example, define the equation

```
> eqn2 := x3 + 2*x2 - x - 2 = 0;
eqn2 := x3 + 2x2 - x - 2 = 0
```

and assign the output of the `solve` command to `S`:

```
> S := solve( eqn2, x );
S := 1, -2, -1
```

Now, if you try to check the answers, you find:

```
> subs( x=S, eqn2 );
Error, invalid terms in product
```

What does this cryptic error message mean? Take another look at the solution:

```
> S;
1, -2, -1
```

Note that **S** is no longer a label for a single constant, but rather is a sequence of values. Here (unlike in a set), the order of the elements of the sequence matters. This allows us to refer to each individually: the first is element 1, the second is element 2, etc. Use square brackets to select these elements:

```
> S[1]; S[3];
1
-1
```

To check that the second solution actually solves the equation, you can plug it in as follows:

```
> subs( x=S[2], eqn2 );
0 = 0
```

As you might expect, Maple can solve quadratic equations exactly; it can also do so with cubics and quartics (fourth-degree equations). However, for higher-degree polynomial equations and other types, **solve** sometimes can't find a solution analytically. For example,

```
> eqn3 := x = cos(4*x);
eqn3 := x = cos(4 x)
```

is a transcendental equation (note: the input to trigonometric functions is in radians—not degrees—in Maple, just as in mathematics). If we ask Maple to solve this analytically, it fails:

```
> solve(eqn3,x);

$$\frac{1}{4}\text{RootOf}(-Z - 4\cos(-Z))$$

```

This output from Maple (loosely) means it could not find analytic solution. In such cases, you may be able to find a solution numerically (if one exists). Maple uses the command **fsolve** for this (the **f** means “solve in floating-point arithmetic”—rather than exactly):

```
> fsolve(eqn3,x);
.3130883085
```

While `fsolve` generally works for polynomial equations (for which it returns all real roots), for other equations it will find at most one root. In such cases, it's usually a good idea to plot both sides of the equation to see graphically where the root(s) must lie. You can then use this information to tell `fsolve` where to look for roots (see the help screen for `fsolve` for more details).

Equations need not be of the form `expression=constant`. Recall the equation you solved graphically above:

```
> eqn4 := f(x) = g(x);
      eqn4 := x^2 = 3x^2 - 4x - 5
```

Maple can solve this exactly (so could you):

```
> solve(eqn4, x);
      1 + 1/2 sqrt(14), 1 - 1/2 sqrt(14)
```

To convert this answer to floating point (decimal) form, use:

```
> evalf(%);
      2.870828694, -.870828694
```

It's also possible to solve sets of simultaneous equations. For example, try defining the two equations

```
> linear1 := 5*x - 7*y = 37; linear2 := 3*x - 11*y = 17;
      linear1 := 5x - 7y = 37
      linear2 := 3x - 11y = 17
```

To solve them, type

```
> soln := solve( {linear1, linear2}, {x, y} );
      soln := { x = 144/17, y = 13/17 }
```

Note that `solve` is now working on a set of equations (note the curly braces `{ }`); you've asked it to solve for a set of variables (which it did), and assigned to that set the name `soln`. You could verify that this solution satisfies the equations by substituting it into the equations:

```
> subs(soln, linear1); subs(soln, linear2);
      37 = 37
      17 = 17
```

How can you use Maple to find the solution of this system graphically?

Review

You have now seen many of the basic features and commands of Maple. You should be able to enter and work with functions, plot curves, and solve equations. You should also understand the difference between an *expression* and a *function* in Maple. The commands introduced in this tutorial are summarized in the table on the next page.

More Maple Commands

Command	Description
<code>f := x -> expr;</code>	define <code>f</code> as the function mapping <code>x</code> to <code>expr</code>
<code>f(c);</code>	evaluate the function <code>f</code> at the point <code>c</code>
<code>plot(expr, x=a..b);</code>	plot <code>expr</code> as a function of <code>x</code> from <code>x=a</code> to <code>x=b</code>
<code>plot({expr1,expr2}, x=a..b);</code>	plot two curves specified by <code>expr1</code> and <code>expr2</code>
<code>eqn := expr1 = expr2;</code>	defines an equation and labels it <code>eqn</code>
<code>lhs(eqn);</code>	returns the left-hand side of equation <code>eqn</code>
<code>rhs(eqn);</code>	returns the right-hand side of equation <code>eqn</code>
<code>solve(eqn,x);</code>	solves equation <code>eqn</code> for <code>x</code> (analytically)
<code>fsolve(eqn,x);</code>	solves equation <code>eqn</code> for <code>x</code> (numerically)
<code>S[n];</code>	returns the <code>n</code> th element of a sequence <code>S</code>

Remember:

- Always end each command with a semicolon;
- Use capital letters only where needed—Maple is Case Sensitive.