Introduction to *Mathematica*: Overview

Bob Bradshaw, Ohlone College

Help

There is an extensive online help feature. Help with individual commands can be accessed using ***?command name*** and the options for a command can be accessed using **Options[*command name*]**. The results will come up within your document.

```
? Plot
```

```
Options[Plot]
```

Restart

It sometimes becomes necessary to clear the entire *Mathematica* memory. To do this, type **Quit**

Syntax

Mathematica uses an upper case letter for the start of built-in functions and square brackets for the arguments of functions.

```
Sin[Pi / 4]
```

Parentheses are used only for a grouping symbol. Curly braces are used to indicate intervals, as in the domain of plot statement.

```
Plot[x^2, {x, -2, 2}]
```

No punctuation is required at the end of a statement. Using a semicolon will suppress the output of a command.

```
3 + 4
```

```
4 + 5;
```

Multiplication does not require a multiplication sign but an asterisk can be used. Spaces will also work for multiplication. Lack of a space indicates a new variable. Multiplication by a number does not require a space.

```
x = 3;
y = 4;
x y
xy
10 x
```

Functions

Functions are declared using the notation **f[x_]:=*expression***

```
f[x_] := 2 x - 7
f[10]
```

Substitutions

Substitution is most readily done using the **/.** operation.

```
x^2 /. x → 4
```

What went wrong? We used $x = 3$ in an earlier statement. We should first use `Clear`

```
Clear[x]
x^2 /. x → 4
```

Numerical Values

To get a numerical value, use **N[*expression*]**.

```
N[Pi]
N[Pi, 50]
```

Tables

A table of values can be created for functions.

```
mytable1 = Table[x^2, {x, 1, 5}]
```

This can also be made into a standard math table and include headings.

```
mytable2 =
 TableForm[Table[{x, x^2}, {x, 1, 5}], TableHeadings -> {None, {x, x^2}}]
```

Graphics

Simple Plots

The standard `Plot` command has the form **Plot[*expression*,{x,*start*,*end*},PlotStyle->{*settings*}]**

```
Plot[x^2, {x, -2, 2}, PlotStyle → {Hue[0.5]}];
```

The thickness of a curve is set as a percentage of the width of the domain.

```
Plot[x^2, {x, -2, 2}, PlotStyle → {Hue[0.5], Thickness[0.05]}];
```

You can also set the thickness of a curve in terms of points (12 points per inch.)

```
Plot[x^2, {x, -2, 2}, PlotStyle → {Hue[0.5], AbsoluteThickness[5]}];
```

Colors

Colors can be declared either by the **Hue[#]** or **RGB[#,#,#]** commands. Alternatively, loading the following package allows you to use the name of the color.

```
<< Graphics`Colors`
```

```
AllColors
```

To see what these colors are, check here.

Combining Plots

By naming several plots, they can be combined using the **Show [ ]** command.

```
p1 = Plot[x^2, {x, -2, 2}, Background → Raspberry,
   PlotStyle → {DarkTurquoise, Thickness[0.02]}];
p2 = Plot[x^2 + 2, {x, -2, 2}, Background → Raspberry,
   PlotStyle → {Black, Thickness[0.02]}];
p3 = Plot[x^2 + 4, {x, -2, 2}, Background → Raspberry,
   PlotStyle → {MintCream, Thickness[0.02]}];
Show[
  p1,
  p2,
  p3];
```

Notice that ending the **Plot** command with a semicolon only supressed the "Graphics" output, not the graph itself.

To supress the graph, use **DisplayFunction->Identity.**

To resume displaying a graph, use **DisplayFunction->$DisplayFunction**

```
p1 = Plot[x^2, {x, -2, 2},
  Background → Mint,
  PlotStyle → {Tomato, Thickness[0.02]},
  DisplayFunction -> Identity];
p2 = Plot[x^2 + 2, {x, -2, 2},
  Background → Mint,
  PlotStyle → {Black, Thickness[0.02]},
  DisplayFunction -> Identity];
p3 = Plot[x^2 + 4, {x, -2, 2},
   Background → Mint,
   PlotStyle → {MintCream, Thickness[0.02]}];
Show[p1, p2, p3,
  DisplayFunction → $DisplayFunction,
  Frame → True,
  FrameStyle → {{Blue, Thickness[0.017]},
    {Red, Thickness[0.012]},
    {Blue, Thickness[0.012]},
    {Green, Thickness[0.012]}}];
```

Surfaces

Surfaces can be plotted using the **Plot3D** command

```
Plot3D[E^-(x^2 + y^2) * Sin[x^2 + y^2], {x, -2, 2}, {y, -2, 2}];
```

Mathematica uses fewer points than *Maple* for the default resolution. Here is the same picture at the higher resolution.

```
Plot3D[E^-(x^2 + y^2) * Sin[x^2 + y^2], {x, -2, 2}, {y, -2, 2}, PlotPoints → 25];
```

To get a mouse driven rotation, load the **RealTime3D** package.

```
<< RealTime3D`
```

```
Plot3D[E^-(x^2 + y^2) * Sin[x^2 + y^2], {x, -2, 2}, {y, -2, 2}, PlotPoints → 25];
```

To return to standard graphics, use **Default3D**

```
<< Default3D`
```

```
Plot3D[E^-(x^2 + y^2) * Sin[x^2 + y^2], {x, -2, 2}, {y, -2, 2}, PlotPoints → 25];
```

Parametric Functions

Parametric curves and surfaces can be drawn through three different commands.

```
ParametricPlot[{Cos[3 t] * Cos[t], Cos[3 t] * Sin[t]},
  {t, 0, 2 Pi}, PlotStyle → {Blue, Thickness[0.02]}];
```

There is some inconsistent notation here.

```
ParametricPlot3D[
  {Cos[t] * Cos[t], Cos[t] * Sin[t], Sin[5 t], {Blue, Thickness[0.02]}},
  {t, 0, 2 Pi}, BoxRatios → {1, 1, 1}];
```

Animations

There are multiple methods of creating animations. I like to use the Movie Plot commands as they are consistent for different types of curves. All of them require loading the package `Graphics`Animation`

```
<< Graphics`Animation`
```

For two dimensional functions,use the **MoviePlot** command.

```
MoviePlot[Sin[x * t], {x, -10, 10}, {t, 0, 4}]
```

To animate a surface, use the **MoviePlot3D** command

```
Clear[x, y, t];


MoviePlot3D[
 E^(-t * (x^2 + y^2)) * Sin[x^2 + y^2],
 {x, -2, 2}, {y, -2, 2}, {t, 1, 3},
 PlotRange → {-0.3, 0.3},
 PlotPoints → 25]
```

To rotate a surface, first create it using the appropriate command and then use **SpinShow** to rotate it.

```
g = ParametricPlot3D[{Sin[u] Cos[v], Sin[u] Sin[v], Cos[v]},
   {u, 0, 2 Pi}, {v, 0, 2 Pi}];

SpinShow[g, Frames → 30,
  SpinRange → {0 Degree, 360 Degree}, SpinTilt → {0, 45 Degree}];
```

Calculus

Surface of Revolution

There is a special command to generate a surface of revolution. First load the package `Graphics`SurfaceOfRevolution`

```
<< Graphics`SurfaceOfRevolution`
```

The **SurfaceOfRevolution** command revolves a function $z = f(x)$ around an axis between the origin and a point. The default axis is to use the z axis. If you want to revolve around the x axis, use **RevolutionAxis->{1,0,0}**.

```
SurfaceOfRevolution[x^2, {x, -1, 1}, RevolutionAxis → {1, 0, 0},
  BoxRatios → {2, 1, 1}, Axes → False, Boxed → False];
```

Derivatives

Derivatives of functions can be found and evaluated using "prime" notation.

```
f[x_] := Cos[3 x]
f'[x]
f''[x] /. x → 1
```

To evaluate the derivative at a particular number, you could use /. notation (see above) but standard math notation is easier.

```
f'[Pi / 12]
```

To find the derivative of an expression, Use the **D** command

```
D[x^2 + 6 x - 2, x]
```

Integration

Integration can be done in either exact or numerical form.

```
Integrate[Sqrt[Tan[x]], {x, 0, Pi / 4}]
NIntegrate[Sqrt[Tan[x]], {x, 0, Pi / 4}]
```

Series

Infinite series expansions are found using the **Series** command.

```
Series[Log[x], {x, 2, 5}]
```

To remove the "Big O" notation at the end, use the **Normal** command either at the beginning or end of the **Series** command.

```
Normal[Series[Log[x], {x, 2, 5}]]
Series[Log[x], {x, 2, 5}] // Normal
```

Sum of a Series

To get the sum of a series, use the **Sum** command

```
Sum[1 / x, {x, 1, 100}]
```

Other Stuff

As far as I know, the following are unique to *Mathematica*

Special Characters

The special characters used in math are available either through a palette on the **File** menu or through commands. For example, ESC e ESC will create the "is an element of" symbol. ϵ

Palettes

You can create a palette by going to **Input->CreateTable/Natrix/Palette** menu. In each cell, just type the form of the desired command and save the results. Let me know if you want details. For an example, look at my palette.

Presentation

Look at the **Format->StyleSheet** menu.

Sound

Mathematica can produce the sound of an equation. I find this useful one day a year, if I happen to teach differential equations.

```
Play[Sin[4 Pi t] * Sin[1200 Pi t], {t, 11, 14}];
```