Plotting with *Mathematica*

Graphing a function of one variable

Start by typing in and executing the following:

```
Plot[Sin[x], {x, 0, 2 Pi}]
```

The required arguments for **Plot** are an expression (with just one variable) and a list giving the variable and its range. You can also include options. Try the next input. Note: To get the arrow →, type a dash followed by the right angle bracket (on the period key) and then a space. This will get turned into the arrow when you enter the next character.

```
Plot[Sin[x], {x, 0, 2 Pi}, PlotStyle → {Dashed, Thick, Red}]
```

Here's another option to try:

```
Plot[Sin[x], {x, 0, 2 Pi}, AxesLabel → {x, y}]
```

Now try plotting another function:

```
Plot[Sin[x] / x, {x, -20, 20}]
```

The window chosen by *Mathematica* cuts off part of this graph. To choose a different window, you can use the **PlotRange** option. Below are three ways to use PlotRange. Run each one and study its effect.

```
Plot[Sin[x] / x, {x, -20, 20}, PlotRange → All]
```

```
Plot[Sin[x] / x, {x, -20, 20}, PlotRange → {-1, 2}]
```

```
Plot[Sin[x] / x, {x, -20, 20}, PlotRange → {{-10, 10}, {-1, 2}}]
```

Here's another example with other options:

```
Plot[Sin[x], {x, -2 Pi, 2 Pi}, Axes → False, Frame → True, FrameLabel → {x, y}]
```

You can use the **Ticks** option to specify your own tick marks on the axes:

```
Plot[Sin[x], {x, -2 Pi, 2 Pi}, Ticks → {{-2 Pi, -Pi, 0, Pi, 2 Pi}, {-1, 0, 1}}]
```

You can plot more than one function by giving a list:

```
Plot[{Sin[x], Cos[x]}, {x, -2 Pi, 2 Pi}]
```

Visualizing functions of two variables

Try a basic contour plot:

```
ContourPlot[x * y, {x, -2, 2}, {y, -2, 2}]
```

If you put the cursor over a level curve, you will get the value for that curve. (This may not be the default option on the Windows version.) Try these options:

```
ContourPlot[x * y, {x, -2, 2}, {y, -2, 2}, ContourShading → False, ContourLabels → True]
```

For a graph of the function, try

```
Plot3D[x * y, {x, -2, 2}, {y, -2, 2}]
```

Note that you can use the cursor to rotate a 3D plot.

Below are two options to adjust the view. The **RegionFunction** option gives a restriction on the domain that is plotted. The **MeshFunctions** option controls which mesh curves are drawn. The syntax for this option is not very intuitive. In this case, #1 refers to the first variable (x here), #2 refers to the second variable (y here), and #3 refers to the output.

```
Plot3D[x y, {x, -2, 2}, {y, -2, 2},
 RegionFunction → Function[{x, y}, x^2 + y^2 < 4],
 MeshFunctions → {#3 &}]
```

Plotting other surfaces

To plot the graph of an equation involving three variables, you can use the **ContourPlot3D** command. Here's an example going back to the quadric surfaces from the early part of the course. Note the use of **==** in the equation.

```
ContourPlot3D[x^2 + y^2 - z^2 == 1, {x, -6, 6}, {y, -6, 6}, {z, -5, 5}]
```

If the mesh curves are distracting, you can remove them using **Mesh → None**. You can also replace **None** by a number to control how many mesh curves are drawn.

```
ContourPlot3D[x^2 + y^2 - z^2 == 1, {x, -6, 6}, {y, -6, 6}, {z, -5, 5}, Mesh → None]
```