Introduction to *Mathematica*
Three Dimensional Graphing

Bob Bradshaw,Ohlone College

Mathematica has very good 3-dimensional graphics but, as of version 4.2, provides live, interactive rotation only through an experimental package. As a result, I recommend using *Maple* for three dimensional graphs.

Three Dimensional Curves

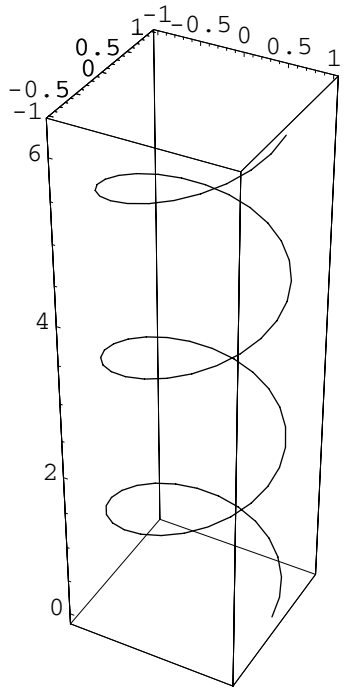
Three dimensional curves are provided through the **ParametricPlot3D** command.

In[30]:=
```
f = Cos[3 t]
g = Sin[3 t]
h = t
ParametricPlot3D[{f, g, h}, {t, 0, 2 Pi}];
```

Out[30]= `Cos[3 t]`

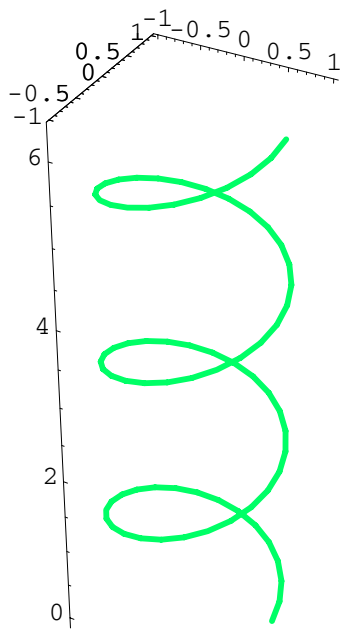
Out[31]= `Sin[3 t]`

Out[32]= `t`



To add options to the plot, place a list of options inside the first set of braces. This is different from standard graphs, in which the options go in a **PlotStyle**. However, options for the entire graph, as opposed to the curve, go at the end of the command.

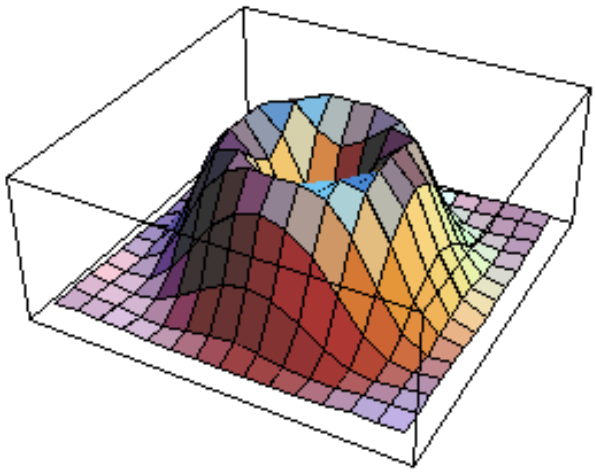
In[47]:=
```
ParametricPlot3D[{f, g, h, {Hue[0.4], Thickness[0.02]}},
  {t, 0, 2 Pi}, Axes → True, Boxed → False];
```



Surfaces

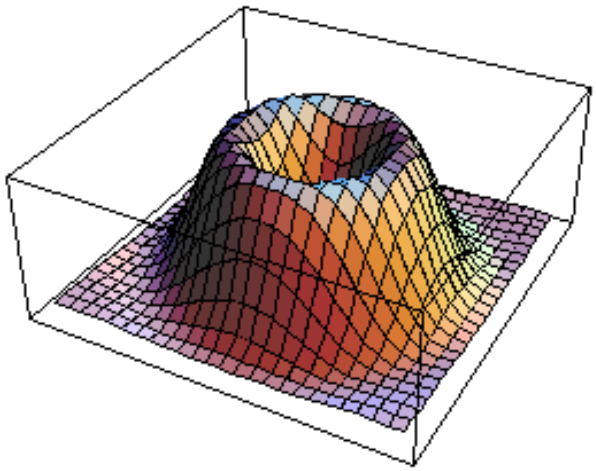
Surfaces can be plotted using the **Plot3D** command

```
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.

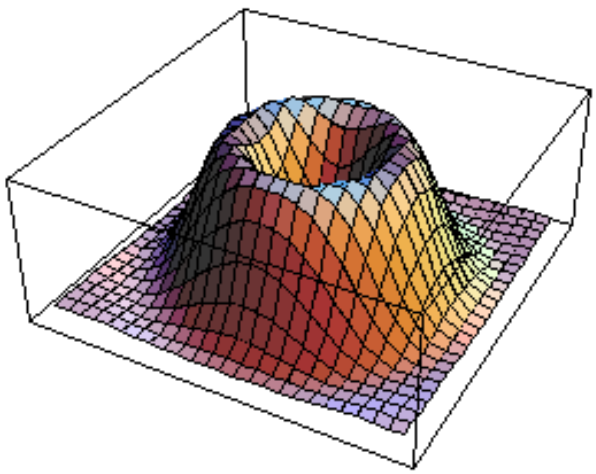
```
Plot3D[E^-(x^2 + y^2) * Sin[x^2 + y^2], {x, -2, 2}, {y, -2, 2}, PlotPoints → 25];
```



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

```
<< RealTime3D`
```

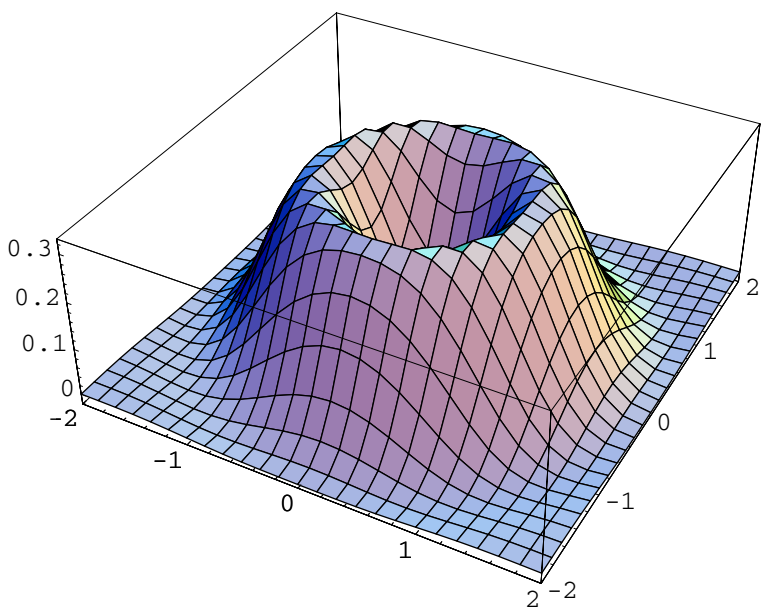
```
Plot3D[E^-(x^2 + y^2) * Sin[x^2 + y^2], {x, -2, 2}, {y, -2, 2}, PlotPoints → 25];
```



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

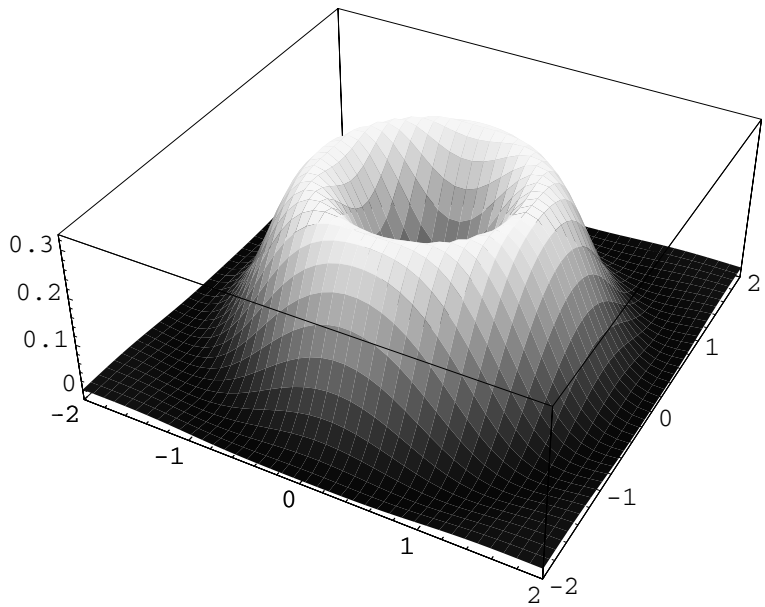
```
<< Default3D`
```

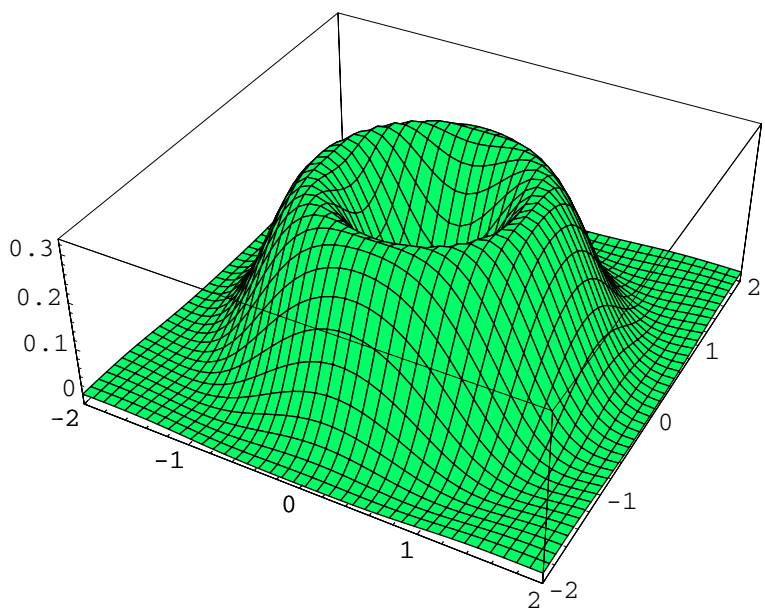
```
Plot3D[E^-(x^2 + y^2) * Sin[x^2 + y^2], {x, -2, 2}, {y, -2, 2}, PlotPoints → 25];
```



The options for a surface plot go at the end of the command, exccept when you want to specify a color for the surface.

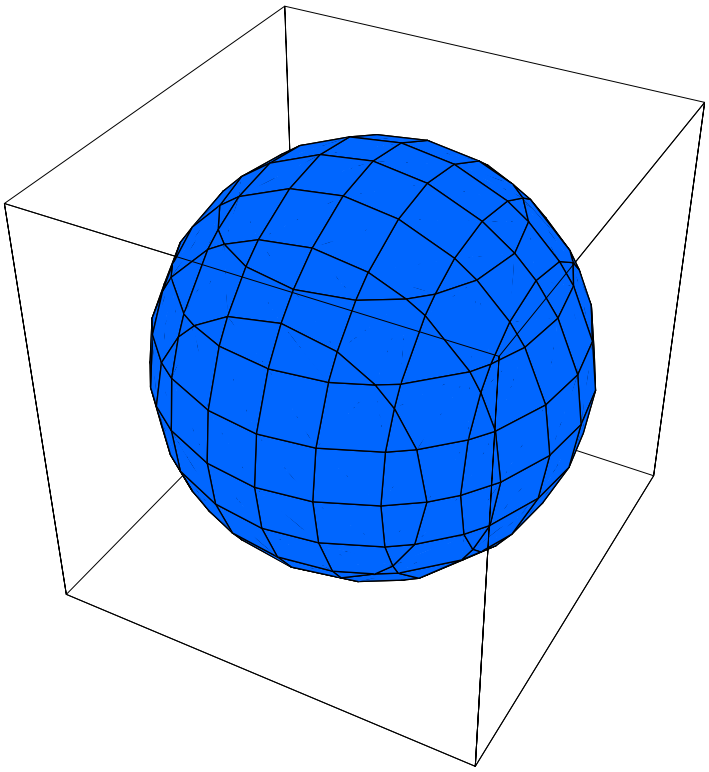
In[65]:=
```
Plot3D[E^-(x^2 + y^2) * Sin[x^2 + y^2], {x, -2, 2},
  {y, -2, 2}, PlotPoints → 40, Mesh → False, Lighting → False];
Plot3D[{E^-(x^2 + y^2) * Sin[x^2 + y^2], Hue[0.4]},
  {x, -2, 2}, {y, -2, 2}, PlotPoints → 40];
```





Mathematica can also draw implicitly defined surfaces. It requires the **ContourPlot3D** package. To graph an implicit function, enter the equation and specify the constant value at the **Contours** option. Turning the **Lighting** option off and specifying a **ContourStyle** will create a plot of a specified color.

In[6]:=
```
<< Graphics`ContourPlot3D`
```

In[17]:=

```
ContourPlot3D[x^2 + y^2 + z^2, {x, -1, 1}, {y, -1, 1}, {z, -1, 1},
  Contours → {1}, Lighting → False, ContourStyle → {Hue[0.6]}];
```