

# Plotting Surfaces

---

## Introduction

**OBJECTIVE:** To learn how to use *Mathematica* to plot surfaces, contours, and level curves.

In this module, you will learn how *Mathematica* can help plot functions of two variables. For the standard form,  $z = f(x,y)$ , you will examine contours ( $z$ -level curves) and  $x$ -level and  $y$ -level curves. Can you come up with a function that will stump your classmates when they attempt to match your functions with surface plots, contours, and level curves? This is a challenge for you to explore nontrivial functions and their graphs. All the codes for creating the necessary graphs are in this *Mathematica* notebook.

When functions of two variables are expressed in cylindrical or spherical coordinates or are expressed parametrically, *Mathematica* can help you plot these surfaces.

## ■ Technology Guidelines

**NOTE:** If you have just finished a module, restart *Mathematica* or close the *Kernel* before executing a new module.

**TO OPEN CELLS,** put your cursor on the right cell bracket and double click.

**TO STOP AN EXECUTION**

Select the *Kernel* pull-down menu and click on *Abort Evaluation*.

**ORDER OF EXECUTION**

Execute cells in the order given. Do not skip any Input cells within a given notebook.

**SAVING NOTEBOOKS**

You can save anytime to any directory you choose, and it is wise to save often.

However, before you do your final save, it is a good idea to delete all your output by selecting the

*Delete All Output* selection under the *Kernel* pull-down menu.

**EXPERIENCING MAJOR PROBLEMS**

Save if appropriate, then shut down *Mathematica* and start it up again.

---

## Part I: Surfaces, Contours, and Level Curves

The following code gives a set of *Mathematica* commands that can be used in defining and

plotting surfaces in three-dimensional space. You can get more information on these commands and their options in the Help menu. You can insert several new functions of your own, each of a different type, to develop a visualization of different surfaces. Try some new options for the plots as well.

In[1]:=

```
Off[General::spell]

Off[General::spell1]

Clear[x, y, z, f]

f[x_, y_] := Exp[-(x^2 + y^2) / 8] ((Cos[x])^2 + (Sin

xmin = -5;

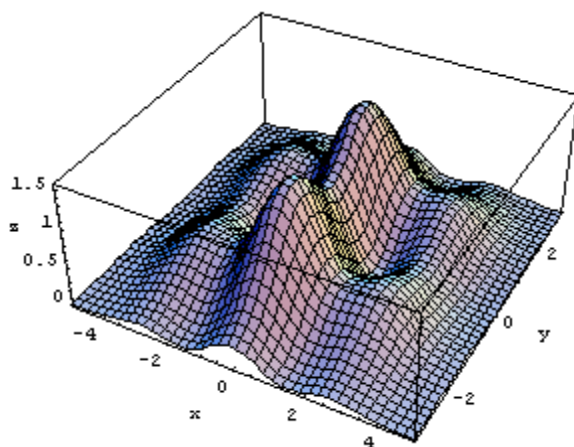
xmax = +5;

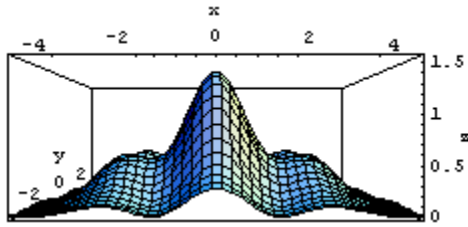
ymin = -N[π];

ymax = +N[π];

pict = Plot3D[f[x, y], {x, xmin, xmax}, {y, ymi
  PlotPoints -> 40, AxesLabel -> {x, y, z}];

Show[pict, ViewPoint -> {0, -2, 0}];
```





Determine which axis is which, and experiment with different viewpoints for the above plot. Click on [About Mathematica](#) if you need help with this.

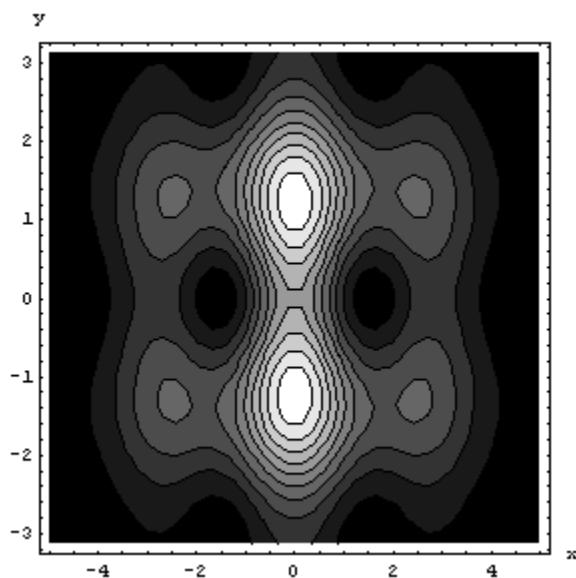
"> [About Mathematica](#)

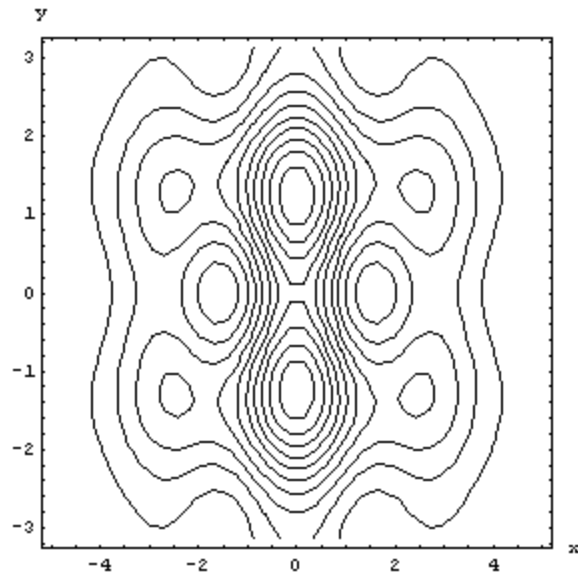
Now, look at the curves along which the  $z$ -coordinate is a constant. What is the difference between the two plots below, and what does light to dark signify in the first plot?

In[11]:=

```
ContourPlot[f[x, y], {x, xmin, xmax}, {y, ymin, ymax},
  PlotPoints -> 40, Axes -> True, AxesLabel -> {x,
```

```
y}];
ContourPlot[f[x, y], {x, xmin, xmax}, {y, ymin, ymax},
  ContourShading -> False, PlotPoints -> 40, Axes
  AxesLabel -> {x, y}];
```





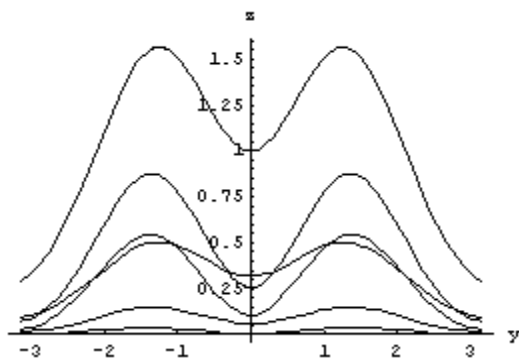
Next, look at the curves formed when you hold  $x$  or  $y$  constant.

In[13]:=

```
Clear[x, y]
```

```
xlevel = Table[f[x, y], {x, xmin, xmax}];
```

```
Plot[Evaluate[xlevel], {y, ymin, ymax}, AxesL
```

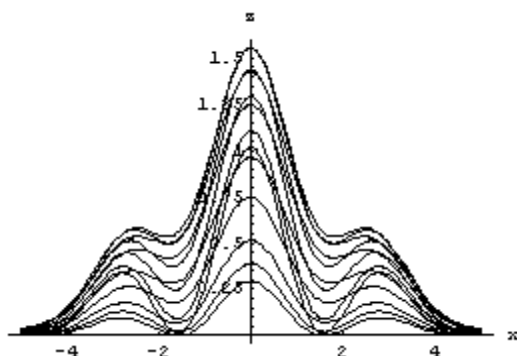


In[16]:=

```
Clear[x, y]
```

```
ylevel = Table[f[x, y], {y, ymin, ymax, .5}];
```

```
Plot[Evaluate[ylevel], {x, xmin, xmax}, AxesL
```



## You Try It: Part I: Stump Your Classmates

You are to submit a printout showing a function, its surface plot, its contour plot, and its  $x$ -level and  $y$ -level plots. Print each on a different sheet. Try to select a function that will stump your classmates. Your teacher may want to include all the students' results as a set of plots to match up on the next exam, so send your notebooks to the teacher electronically.

I usually have my students work in pairs to come up with a good function. Then I include their problems on the next exam and ask students to match the functions-surfaces-contours-level curves, giving points to the pair who find a function that stumps the most classmates.

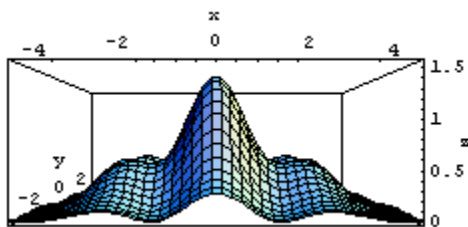
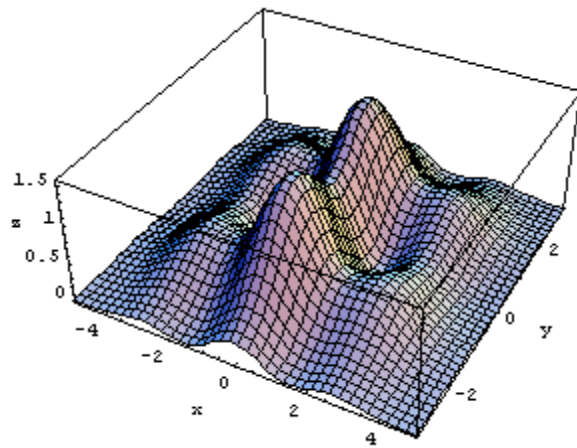
Just put in your own functions and bounds for the items in red. Be sure to use correct notation. You can select any viewpoint you wish for the **Show** command.

In[19]:=

```
Clear[x, y, z, f]

f[x_, y_] := Exp[-(x^2 + y^2) / 8] ((Cos[x])^2 + (Sin

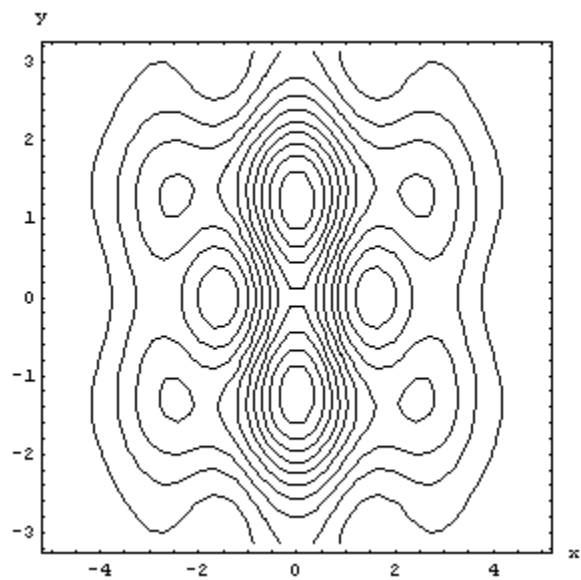
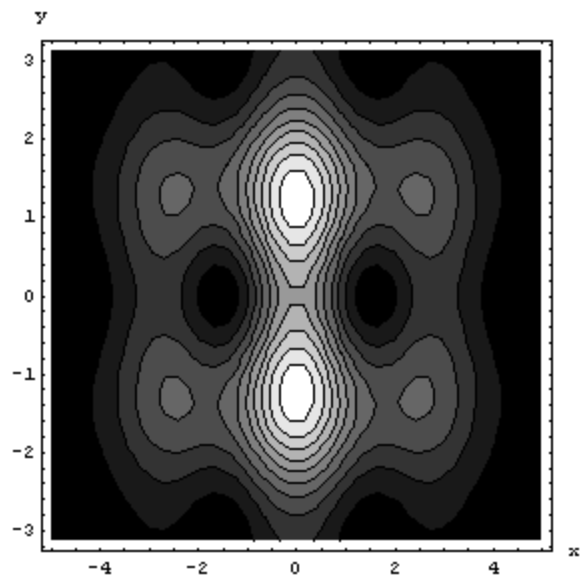
xmin = -5;
xmax = +5;
ymin = -N[ $\pi$ ];
ymax = +N[ $\pi$ ];
pict = Plot3D[f[x, y], {x, xmin, xmax}, {y, ymi
  PlotPoints -> 40, AxesLabel -> {x, y, z}];
Show[pict, ViewPoint -> {0, -2, 0}];
```



In[22]:=

```
ContourPlot[f[x, y], {x, xmin, xmax}, {y, ymin, ymax},
  PlotPoints -> 40, Axes -> True, AxesLabel -> {x,
```

```
ContourPlot[f[x, y], {x, xmin, xmax}, {y, ymin, ymax},
  ContourShading -> False, PlotPoints -> 40, Axes
  AxesLabel -> {x, y}];
```

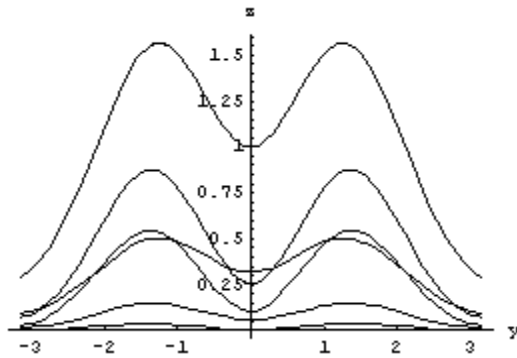


In[24]:=

```
Clear[x, y, z]
```

```
xlevel = Table[f[x, y], {x, xmin, xmax, (xmax -
```

```
Plot[Evaluate[xlevel], {y, ymin, ymax}, AxesL
```

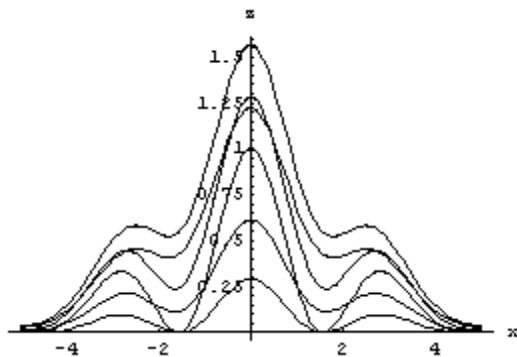


In[27]:=

```
Clear[x, y, z]
```

```
ylevel = Table[f[x, y], {y, ymin, ymax, (ymax -
```

```
Plot[Evaluate[ylevel], {x, xmin, xmax}, AxesL
```



## Part II: Implicit Plots in Three Dimensions

In the last chapter, you used a **ContourPlot3D** command to plot cylinders and quadric surfaces. Now you understand better what contours are. Imagine an object in 4 dimensions. Its contours would be 3-dimensional surfaces. That is how we interpret such surfaces when plotting them in *Mathematica*. We first need to read in the **ContourPlot3D** package. Recall that the default assumes that the function is equal to 0.

In[30]:=

```
<< Graphics`ContourPlot3D`
```

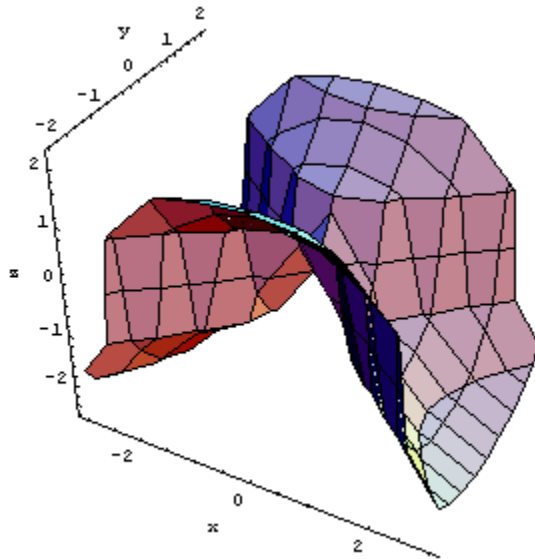
In[31]:=



```
Clear[x, y, z]
```

```
function = z3 + 2 x2 - 3 y2;
```

```
ContourPlot3D[function,
  {x, -3, 3}, {y, -2, 2}, {z, -4, 4},
  Axes -> True, AxesLabel -> {x, y, z}, Boxed ->
```

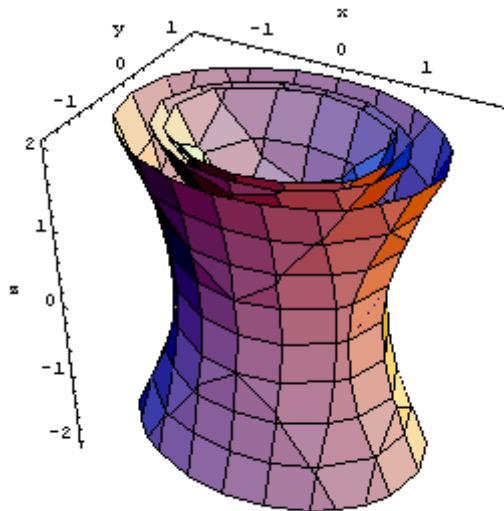


Let's see what it looks like when we draw three level surfaces.

```
In[34]:=
```

```
function = 9 x2 + 16 y2 - 4 z2 + 1;
```

```
ContourPlot3D[ function,
  {x, -2, 2}, {y, -2, 2}, {z, -2, 2}, Contours -
  Axes -> True, AxesLabel -> {x, y, z}, Boxed ->
```




---

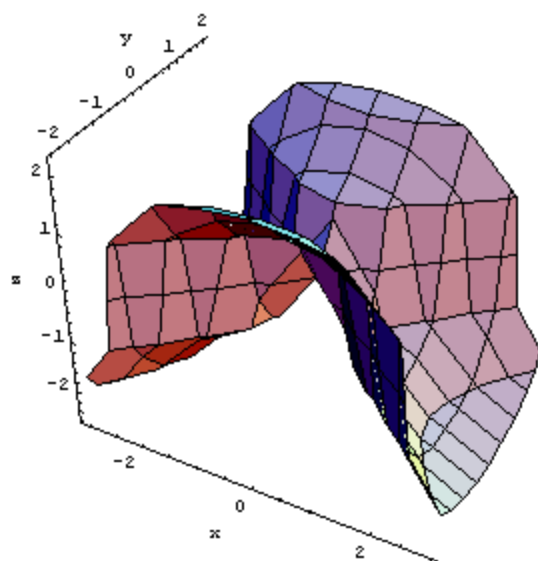
## You Try It: Part II

Be sure that you have already read in the package in Part II. If you have, there is no need to read it in again. Now, select a function of three variables; be certain that the function is defined over the domain (x, y, and z bounds) you specify.

In[36]:=

```
Clear[x, y, z]

function =  $z^3 + 2x^2 - 3y^2$ ;
ContourPlot3D[function,
  {x, -3, 3}, {y, -2, 2}, {z, -4, 4},
  Axes -> True, AxesLabel -> {x, y, z}, Boxed ->
```



## Part III: Cylindrical, Spherical, and Parametric Plots

Certain surfaces are easier than others to express in cylindrical or spherical coordinates. To plot such surfaces in *Mathematica*, we must first read in a plotting package. As you can tell from its name, this package plots parametrically in 3 dimensions also. For cylindrical plots, it is assumed that  $z$  is given as a function of  $r$  and  $\theta$ . For spherical plots, it is assumed that  $\rho$  is given as a function of  $\phi$  and  $\theta$ .

In[38]:=

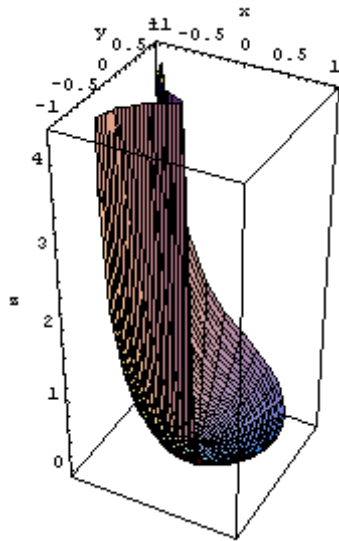
```
<< Graphics`ParametricPlot3D`
```

Here is the graph of  $z = r^2 / (1 + \cos[\theta])$

In[39]:=

```
Clear[r, θ]
```

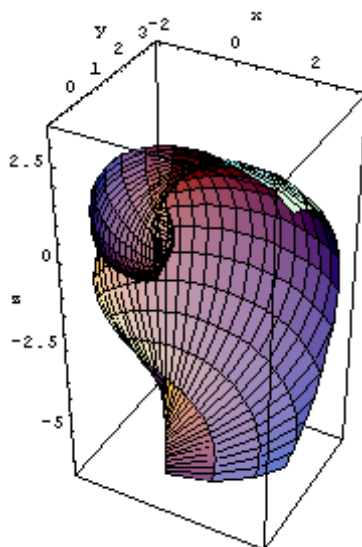
```
CylindricalPlot3D[r^2 / (1 + Cos[θ]),  
  {r, 0, 1}, {θ, 0, 2 Pi}, AxesLabel -> {}]
```



Next is a graph of  $\rho = \phi - \theta$ .

In[41]:=

```
SphericalPlot3D[ $\phi - \theta$ ,
  { $\theta$ , 0, 2 Pi}, { $\phi$ , 0, Pi}, AxesLabel -> {x, y, z}]
```

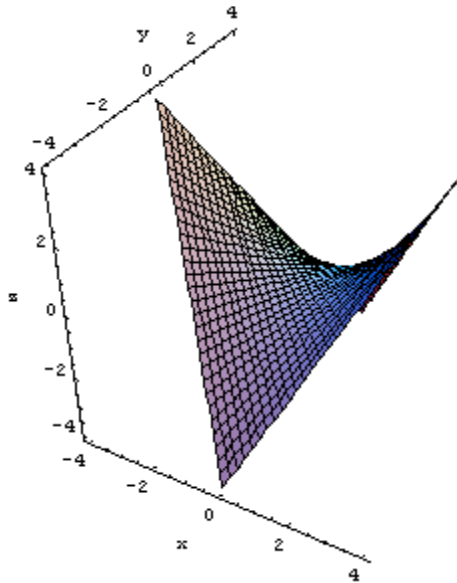


As long as we have this package opened, let's plot a parametric surface in three dimensions. The input is an ordered triple of the form  $\{x,y,z\}$ . These variables are set equal to functions of parameters  $u$  and  $v$ .

In[42]:=

```
Clear[u, v]
```

```
ParametricPlot3D[{u + v, u - v, u v}, {u, -2, 2},  
  AxesLabel -> {x, y, z}, Boxed -> False];
```




---

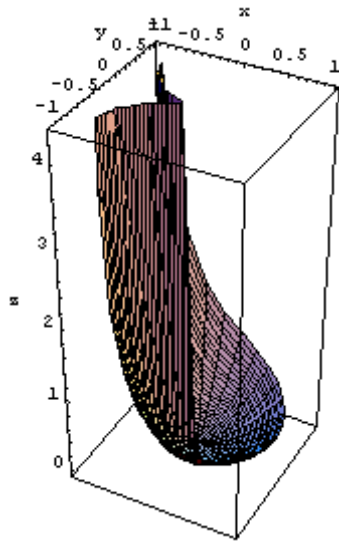
## You Try It: Part III

Go back to any of the plots and alter the functions as you wish. Be sure to use correct notation. Note that many spherical plots take quite a bit of time to plot. If you find you are waiting too long, you can always pull down the Kernel and select Abort Evaluation.

Replace the red in the cylindrical plot with a function of  $r$  and  $\theta$ .

```
In[44]:=
```

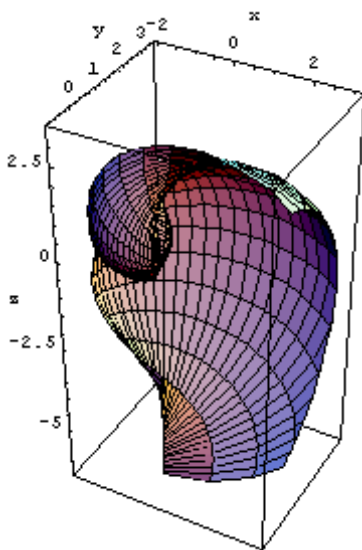
```
CylindricalPlot3D[ $r^2 / (1 + \cos[\theta])$ ,  
  {r, 0, 1}, { $\theta$ , 0, 2 Pi}, AxesLabel -> {r,  $\theta$ , z}];
```



Replace the red in the spherical plot with a function of  $\phi$  and  $\theta$ .

In[45]:=

```
SphericalPlot3D[ $\phi - \theta$ ,
  { $\theta$ , 0, 2 Pi}, { $\phi$ , 0, Pi}, AxesLabel -> {x, y, z}]
```




---

## About *Mathematica*

You can change the viewpoint manually in the last command, or you can highlight the **ViewPoint** selector command and then select the **3D ViewPoint Selector** button under the

Input heading at the top. Move the box around to select a new viewpoint; then paste that viewpoint into your code.

[Go back.](#)

---

Created by [Mathematica](#) (April 2, 2005)

