Plotting Surfaces

Note: You may notice differences between this Maple worksheet and the equivalent Mathematica notebook. These differences were introduced to preserve the content of these modules and were necessary because of major functional differences between Maple and Mathematica.

Introduction

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

In this module, you will learn how *Maple* 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 non-trivial functions and their graphs. All the codes for creating the necessary graphs are in this *Maple* worksheets.

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

Technology Guidelines

NOTE: If you have just finished a worksheet, **restart** *Maple* before executing a new worksheet.
TO OPEN SECTIONS,
Click on the **PLUS** sign at the left hand side of the screen *or* select **Expand All Sections** from the **View** drop down menu.

TO STOP AN EXECUTION
Click on **STOP** button from the toolbar.

ORDER OF EXECUTION
Execute commands in the order given. Do not skip any *Maple* Input lines within a given worksheet

Alternatively, you can execute the entire worksheet by selecting the **Execute Worksheet** command from the **Edit** drop down menu.

SAVING WORKSHEETS.

You can save anytime to any directory you choose, and it is wise to save often.
EXPERIENCING MAJOR PROBLEMS
Save if appropriate, and then shut down *Maple* and start it up again.

Part I : Surfaces, Contours, and Level Curves

The following code gives a set of *Maple* 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.

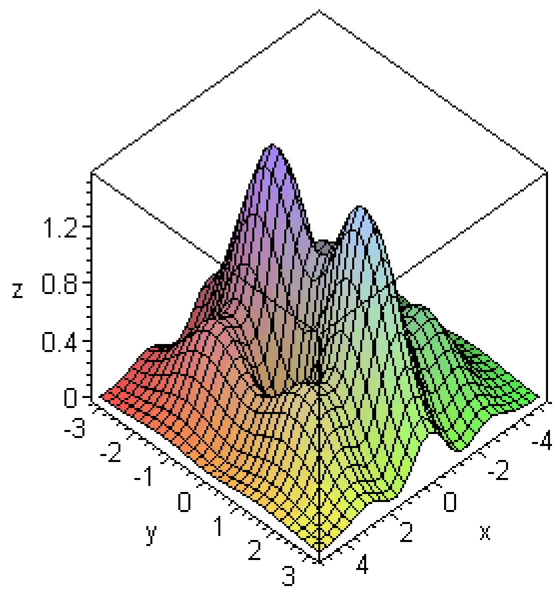
Once the graph is plotted, click anywhere on the graph and drag the mouse to show the plot from different angles. You can do this for all 3D plots.

First, we load the **plots** package.

```
> restart;
  with(plots):
```

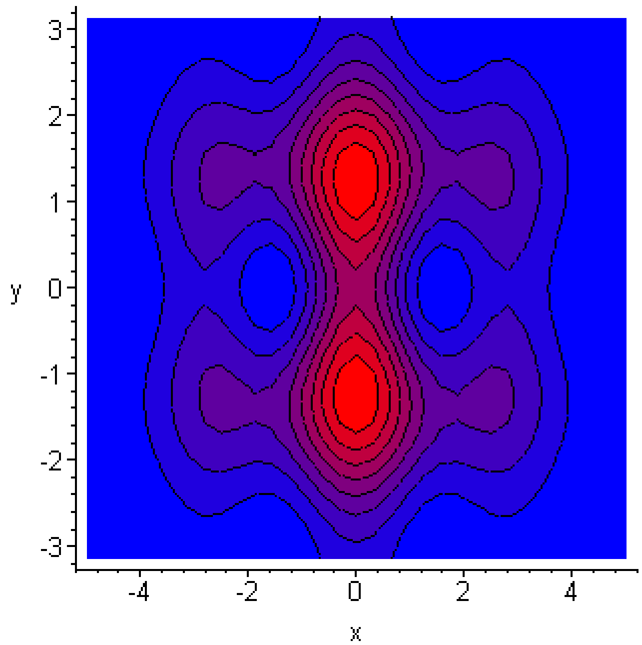
Warning, the name changecoords has been redefined

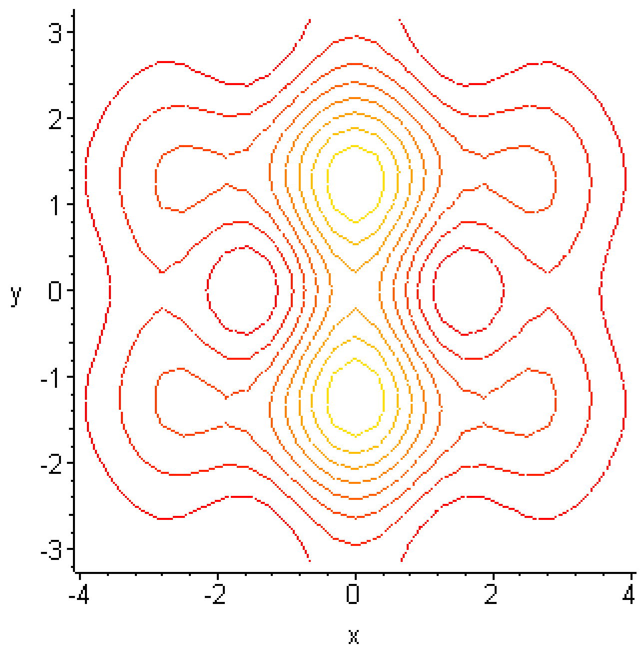
```
> f:=exp(-(x^2+y^2)/8)*((cos(x)^2+sin(y)^2)):
  xmin:=-5:
  xmax:=5:
  ymin:=-Pi:
  ymax:=Pi:
  pict:=plot3d(f(x,y), x=xmin..xmax, y=ymin..ymax, numpoints=800, labels=["x","y","z"],
  axes=BOXED):
  print(pict);
```



Now, look at the curves along which the *z*-coordinate is a constant. What do the different colors signify?

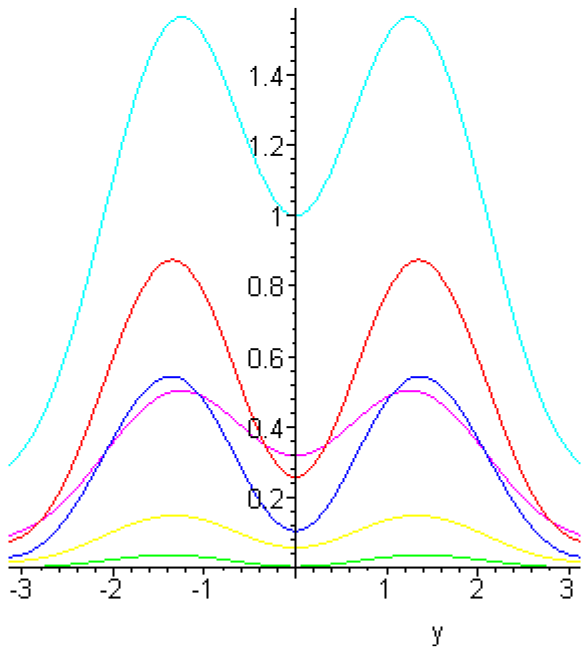
```
> contourplot(f,  x=xmin..xmax, y=ymin..ymax, numpoints=1000, axes=FRAME, labels=
  ["x","y"],  filled=true, coloring=[blue, red]);
  contourplot(f(x,y),  x=xmin..xmax, y=ymin..ymax, numpoints=1000, axes=FRAME, labels=
  ["x","y"]);
```



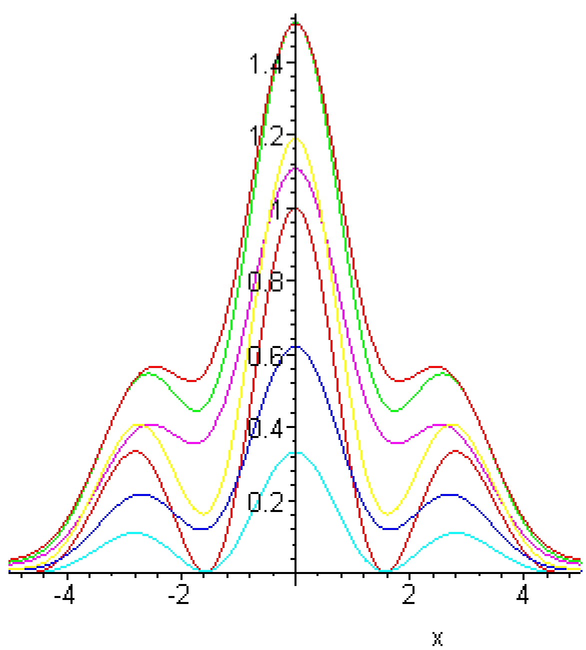


Next, look at the curves formed when you hold *x* or *y* constant.

```
> {seq(subs(x=xi,f),xi=xmin..xmax)}:
  plot(%, y=ymin..ymax);
```




```
> {seq(subs(y=yi/2,f),yi=-6..6)}:
  plot(%, x=xmin..xmax);
```



You Try It : Part I : Stump Your Classmates

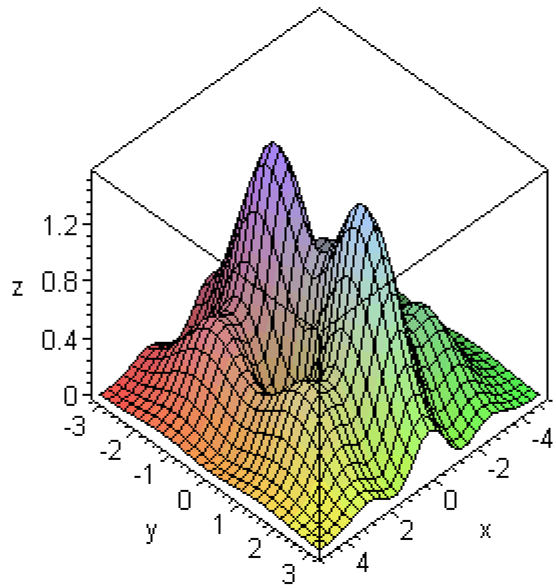
You are to submit a printout showing a function, its **3D Plot**, its **contourplot**, and its *x*-level and *y*-level plots. Print each on a different sheet. Try to select a function that will stump your classmates. Your teacher may want to include all the students' results as a set of plots to match up on the next exam, so send your worksheets to the teacher electronically.

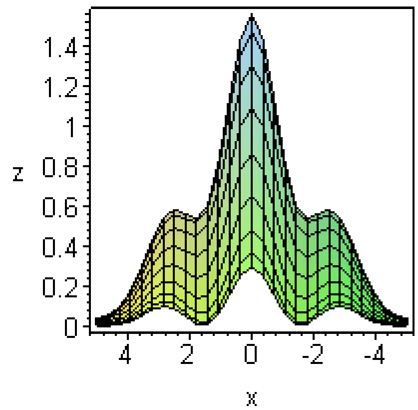
I usually have my students work in pairs to come up with a good function. Then I include their problems on the next exam and ask students match the functions-surfaces-contours-level curves, giving points to the pair who finds a function that stumps the most classmates.

Just put in your own functions and bounds. Be sure to use correct notation.

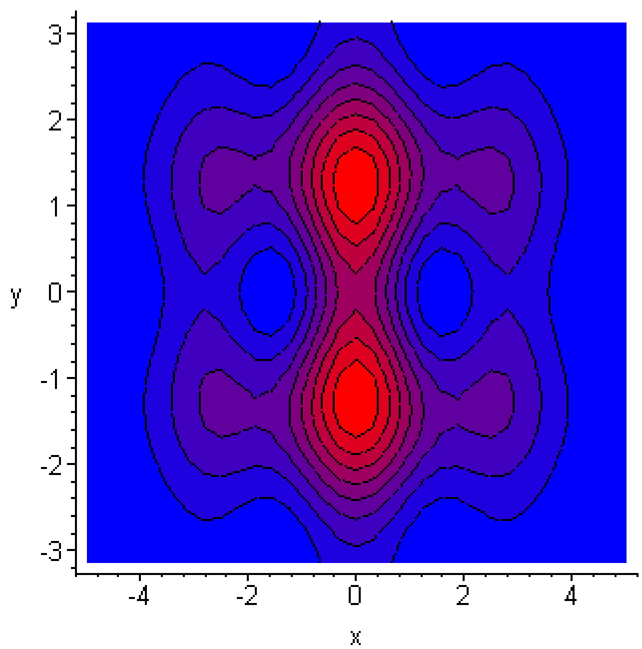
```
> f:=exp(-(x^2+y^2)/8)*(cos(x)^2+sin(y)^2);
  xmin:=-5:
  xmax:=5:
  ymin:=-Pi:
  ymax:=Pi:
  pict:=plot3d(f(x,y), x=xmin..xmax, y=ymin..ymax, numpoints=800, labels=["x","y","z"],
  axes=BOXED):
  print(pict);
  display(pict, orientation=[90,90]);
```

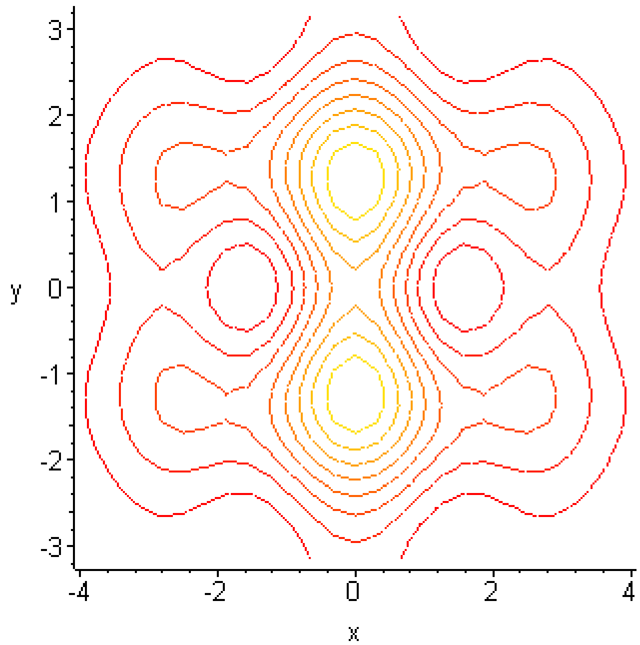
$$f := \mathbf{e}^{\left(-\frac{x^2}{8}-\frac{y^2}{8}\right)}\left(\cos(x)^2+\sin(y)^2\right)$$



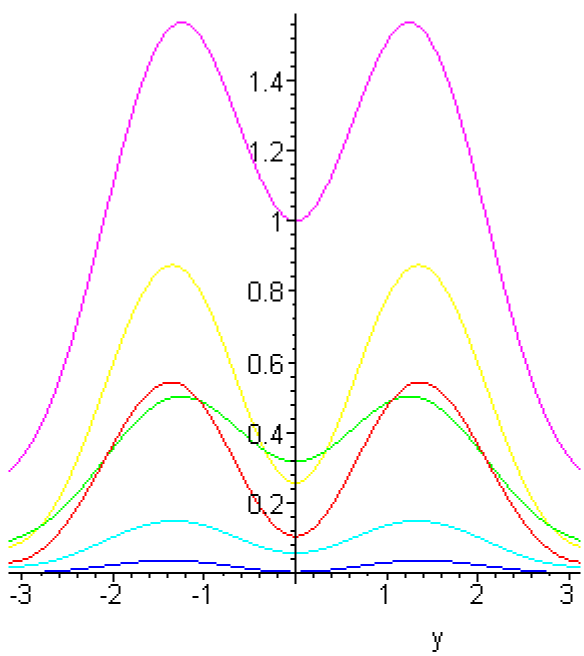



```
> contourplot(f,  x=xmin..xmax, y=ymin..ymax, numpoints=1000, axes=FRAME, labels=
  ["x","y"],  filled=true, coloring=[blue, red]);
  contourplot(f(x,y),  x=xmin..xmax, y=ymin..ymax, numpoints=1000, axes=FRAME, labels=
  ["x","y"]);
```

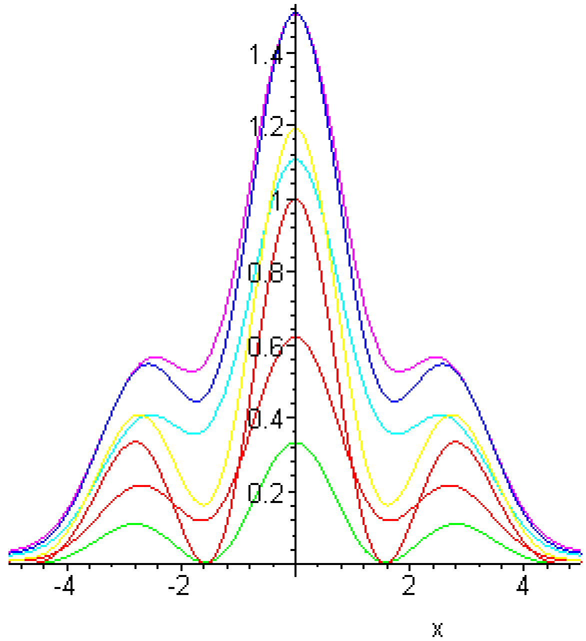




> **{seq(subs(x=xi,f),xi=xmin..xmax)}:**
> **plot(%, y=ymin..ymax);**



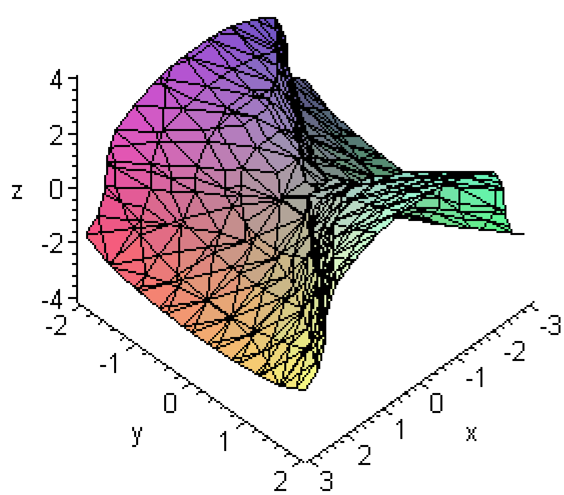
> **{seq(subs(y=yi/2,f),yi=-6..6)}:**
> **plot(%, x=xmin..xmax);**



Part II : Implicit Plots in Three Dimensions

In the last chapter, you used **implicitplot3d** and **plot3d** commands to plot cylinders and quadric surfaces. Now you understand better what contours are. Now, imagine an object in 4 dimensions. Its contours would be 3-dimensional surfaces. That is how we interpret such surfaces when plotting them in *Maple*. Recall that the default assumes that the function is equal to 0.

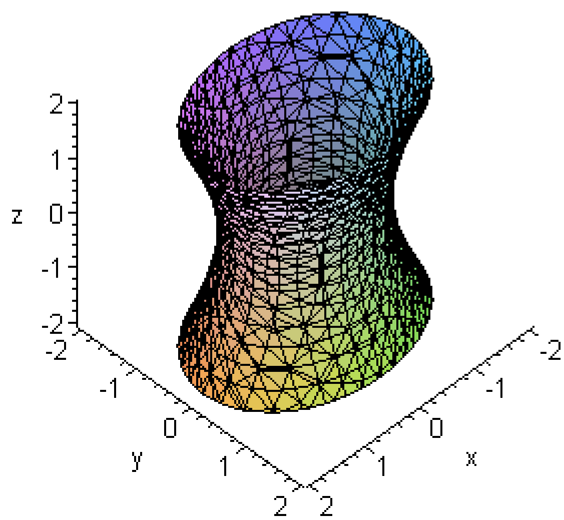
```
> f1:=z^3+2*x^2-3*y^2:
  implicitplot3d(f1, x=-3..3, y=-2..2, z=-4..4, axes=FRAME, labels=["x","y","z"]);
```



Let's see another level surfaces.

```
> f2:=9*x^2+16*y^2-4*z^2+1=16;
  implicitplot3d(f2, x=-2..2, y=-2..2, z=-2..2, axes=FRAME, labels=["x","y","z"], grid=
  [15,15,15]);
```

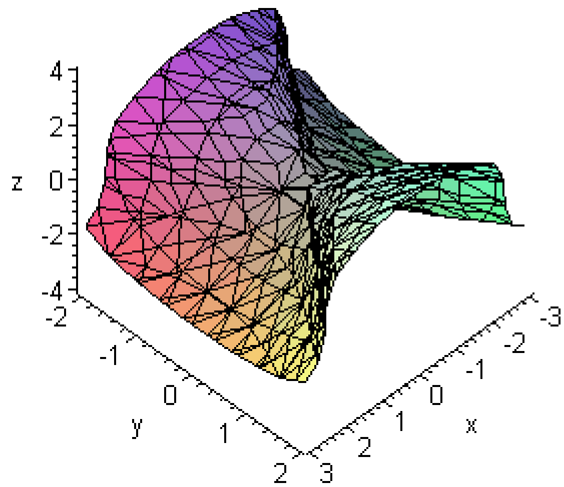
$$f2 := 9\,x^2 + 16\,y^2 - 4\,z^2 + 1 = 16$$



You Try It : Part II

Select any function of three variables that you wish. Be certain that the function is defined over the domain (*x*, *y*, and *z* bounds) you specify.

```
> f1:=z^3+2*x^2-3*y^2:
  plots[implicitplot3d](f1, x=-3..3, y=-2..2, z=-4..4, axes=FRAME, labels=["x","y","z"]);
```

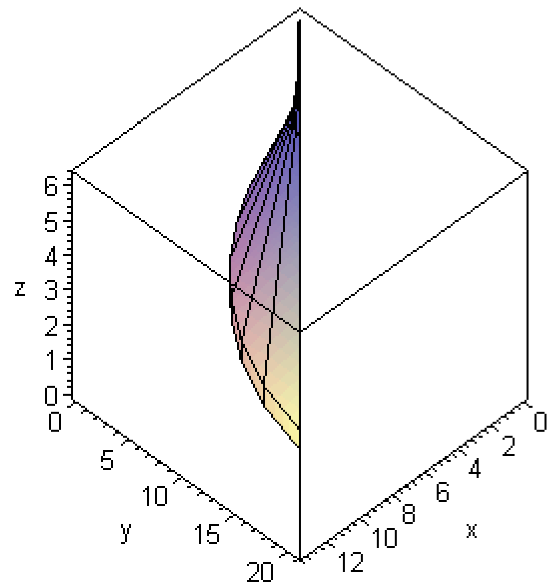


Part III : Cylindrical, Spherical, and Parametric Plots

Certain surfaces are easier than others to express in cylindrical or spherical coordinates. For cylindrical plots, it is assumed that *z* is given as a function of r and θ . For spherical plots, it is assumed that ρ is given as a function of ϕ and θ .

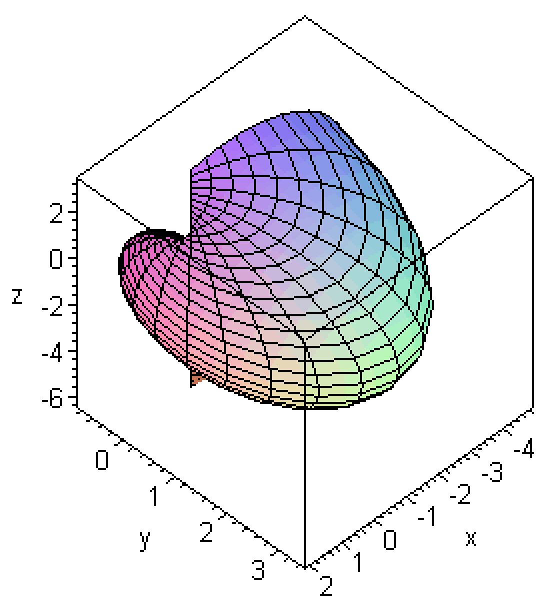
Here is the graph of $z = \frac{r^2}{1 + \cos(\theta)}$.

```
> cylinderplot(r^2/(1+cos(theta)), r=0..1, theta=0..2*Pi, labels=["x","y","z"],axes=BOXED,
  numpoints=111);
```



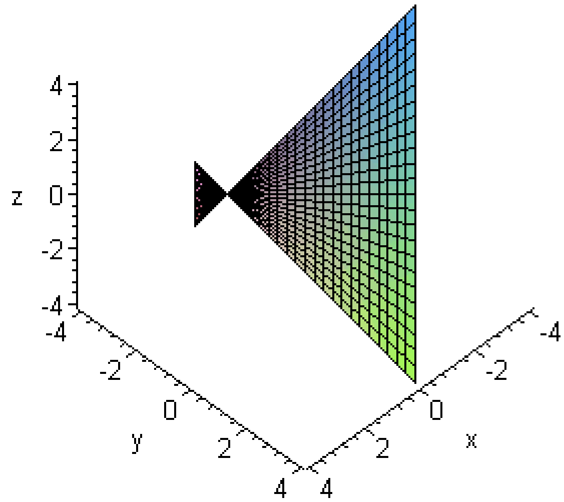
Next is a graph of $\rho = \phi - \theta$.

> **sphereplot(phi-theta, theta=0..2*Pi, phi=0..Pi, labels=["x","y","z"], axes=BOXED);**



As long as we have this package opened, let's plot a parametric surface in three dimensions. The input is of the form {*x, y,* z}. All set equal to functions of parameters *u* and *v*.

> **plot3d([u+v, u-v, u*v], u=-2..2, v=-2..2, labels=["x","y","z"], axes=FRAME);**



You Try It : Part III

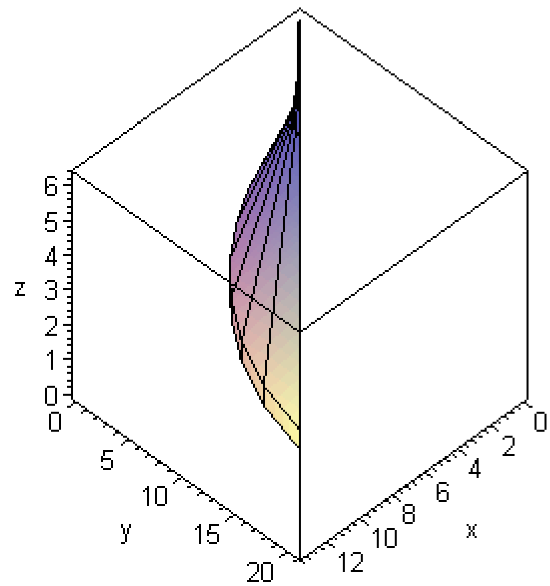
Go back to any of the plots and alter the functions as you wish. Be sure to use correct notation. Noted that many spherical plots take quite a bit of time to plot. If you find you are waiting too long, you can always click the "STOP" button on the toolbar.

Replace the function in the cylindrical plot with a function of *r* and θ .

```
> restart:
  with(plots):
```

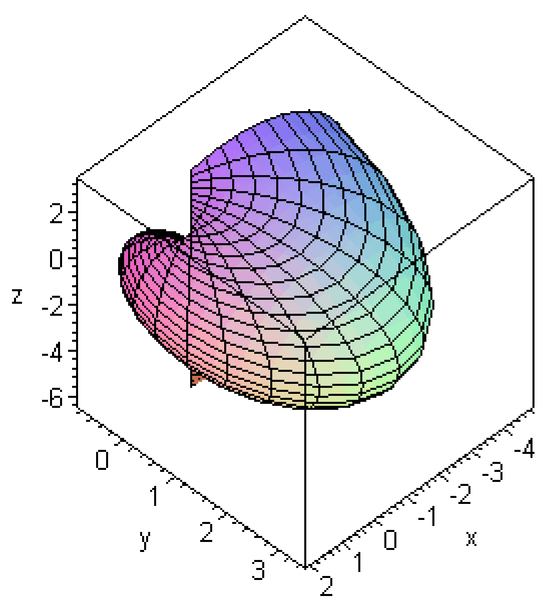
Warning, the name changecoords has been redefined

```
> cylinderplot(r^2/(1+cos(theta)), r=0..1, theta=0..2*Pi, labels=["x","y","z"],axes=BOXED,
  numpoints=111);
```



Replace function in spherical plot with a function of ϕ and θ .

```
> sphereplot(phi-theta, theta=0..2*Pi, phi=0..Pi, labels=["x","y","z"], axes=BOXED);
```



>