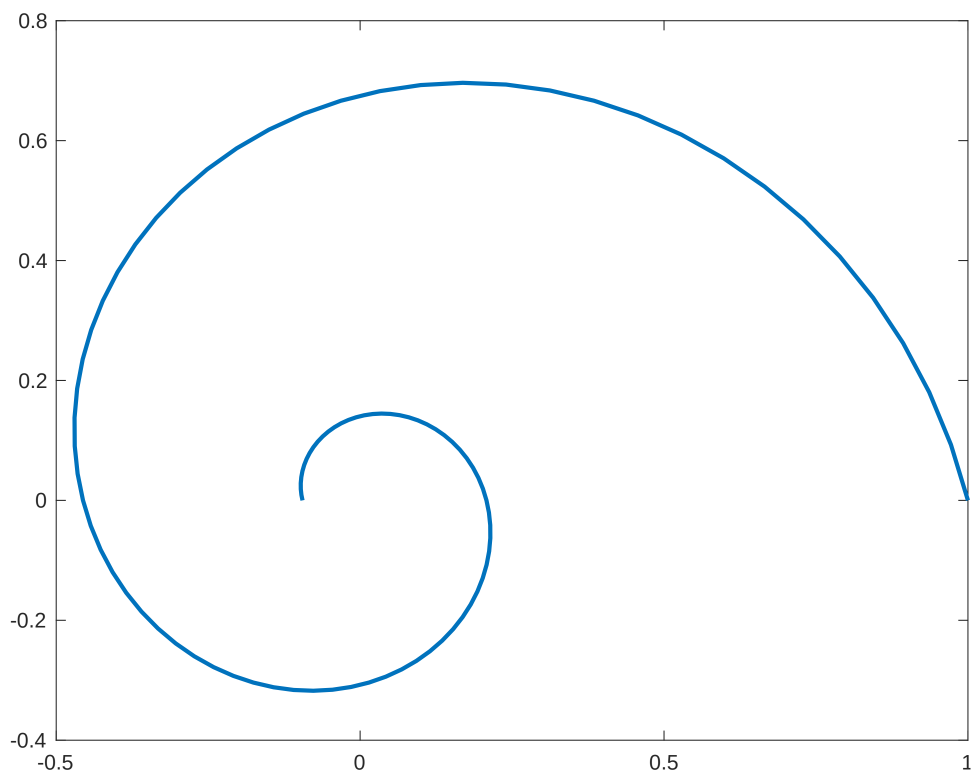
3d Plotting

Suppose we want to plot the graph of function $f(x,y) = \sin(x,y)$. The strategy that MATLAB uses for plotting the graph is very similar to the strategy for functions of one variable: it needs a bunch of data points in the domain, and then the associated values in the range. But there is a complicating issue. For 1-d plots, we have a list of domain values, e.g. `x=[1,2,3]` and a list of function values $y = [1,4,9]$ and MATLAB makes the plot by joining straight lines beteween the data points *in order*, from the first entry to the last entry. You can take advantage of this fact to make 1-d plots of curves that are not graphs of functions. For example, to graph the curve $f(t) = e^{-t/4}\cos(t)\mathbf{i} + e^{-t/4}\sin(t)\mathbf{j}$ over the parameter inverval $[0, 3\pi]$ we could use:

```
>> t = linspace(0,3*pi,100);
>> x=exp(-t/4).*cos(t); y=exp(-t/4).*sin(t);
>> plot(x,y)
```



To graph functions of two variables, MATLAB wants to arrange the data points in a 2-d grid. So if we have a function $f(x,y)$, we need a grid of x-coordinates and a grid of y-coordinates. Matlab provides the very handy `meshgrid` command to help convert lists of 1-d coordinates into grids.

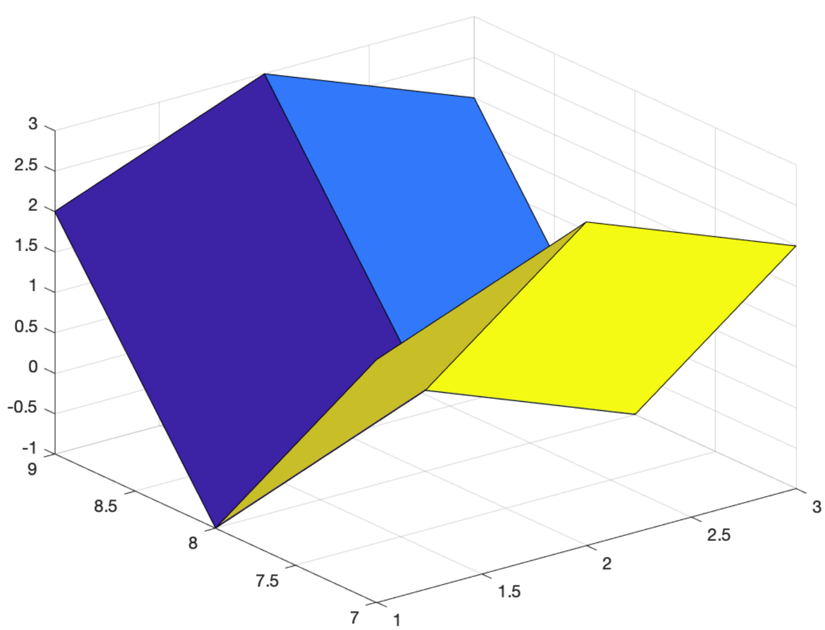
```
>> x=[1,2,3];y=[7,8,9];
>> [XX,YY]=meshgrid(x,y);
>> XX
```

```
XX =

     1     2     3
     1     2     3
     1     2     3

>> YY

YY =

     7     7     7
     8     8     8
     9     9     9
```

Notice that the XX variable has constant columns, one for each of the x values, and the YY has constant **rows**, one for each y value. There are a total of 9 points encoded by this data. Row 1 and column 2 corresponds to the input point $(2,7)$.

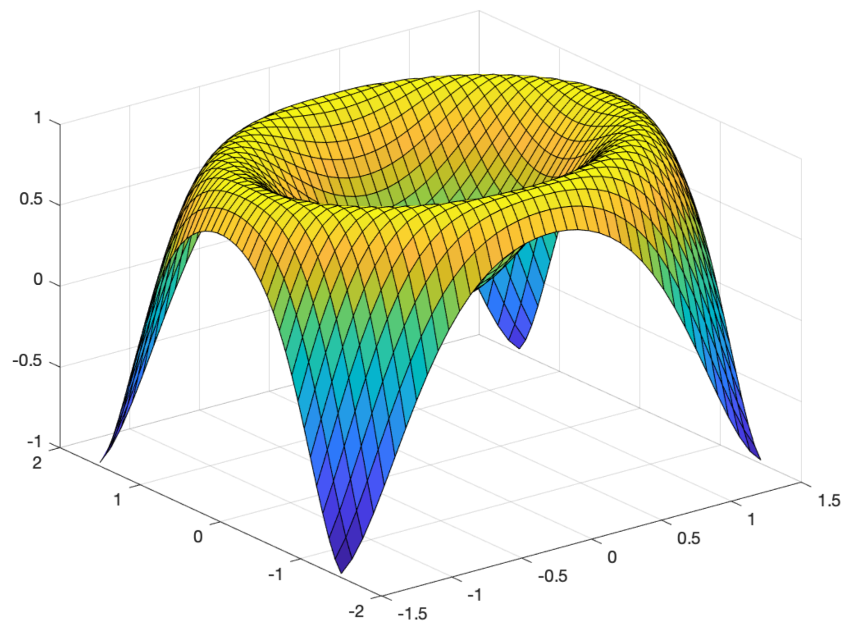
With the domain values arranged as a grid (specifically, a matrix of numbers), MATLAB knows how to connect them with little rectangles instead of little line segments. To make a 3d plot, we simply need to construct corresponding z values and then use the surf command. The following commands plot $f(x,y) = -(x-2)^2 + 3(y-8)^2$ at these 9 data points.

```
>> ZZ = -(XX-2).^2+3*(YY-8).^2;
>> surf(XX,YY,ZZ)
```



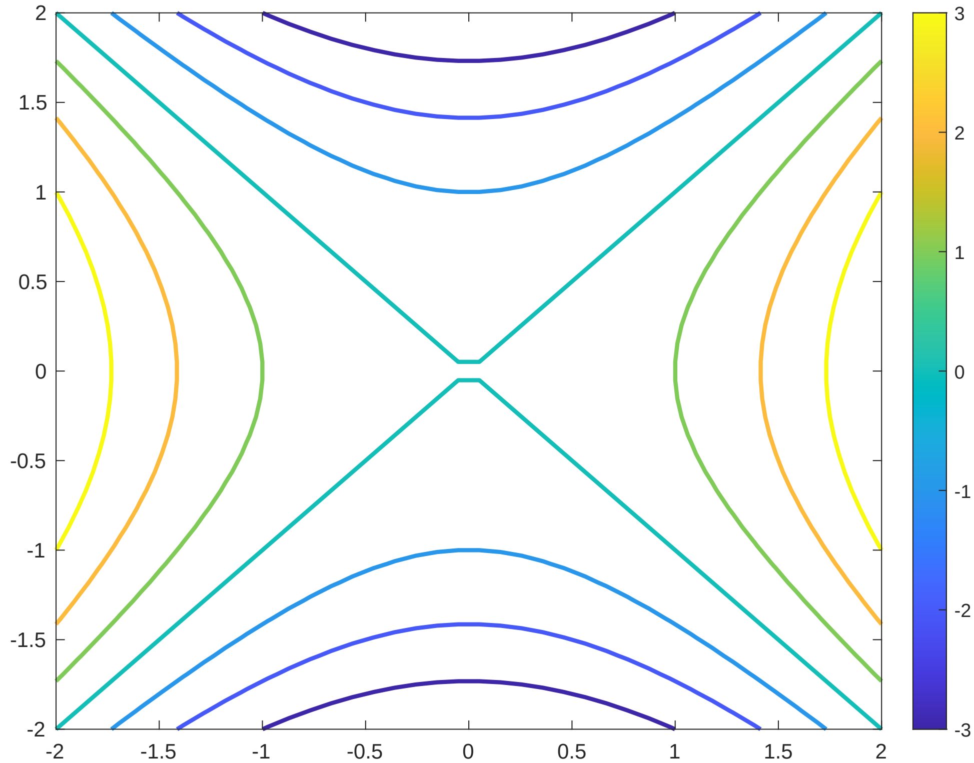
Here's a more sophisticated graph of $\sin(x^2 + y^2)$.

```
>> x=linspace(-1.5,1.5,40);y=linspace(-1.5,1.5,40);
>> [XX,YY]=meshgrid(x,y);
>> surf(XX,YY,sin(XX.^2+YY.^2))
```



We can use MATLAB to generate contour plots using the `contour` command, which uses essentially the same conventions as `surf`.

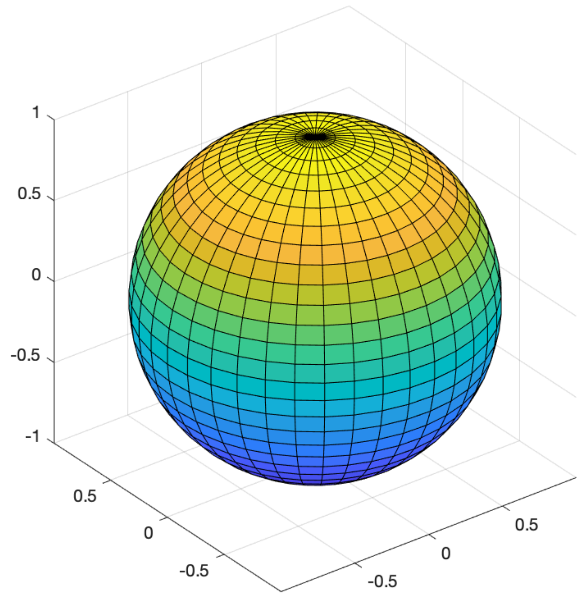
```
>> x=linspace(-2,2,40);y=linspace(-2,2,40);
>> [XX,YY]=meshgrid(x,y);
>> contour(XX,YY,XX.^2-YY.^2,'linewidth',2)
>> colorbar
```



In this graph, we added a couple of niceties: we made the contour lines a little thicker with the extra `'linewidth'` argument to `contor` and we added a color bar to the plot to help understand which contours correspond to which function values.

In the same way that we can make 2-d plots of curves that are not graphs of functions, we can make 3-d plots of surfaces that are not graphs of funtions as well. We need to parameterize the surface in terms of two variables (we'll call these u and v) and then provide x, y, and z coordinates corresponding to the (u, v) pairs. For example, the points on a sphere are given by $x = \cos(u)\cos(v)$, $y = \sin(u)\cos(v)$ and $z = \sin(v)$ where the longitude variable u satisfies $-\pi \leq u \leq \pi$ and where the latitude v satisfies $-\pi/2 \leq v$. You should check for yourself that we really do have $x^2 + y^2 + z^2 = 1$, no matter the choice of u and v.

```
>> u=linspace(-pi,pi,40);v=linspace(-pi/2,pi/2,30);
>> [UU,VV]=meshgrid(u,v);
>> XX=cos(UU).*cos(VV); YY=sin(UU).*cos(VV); ZZ=sin(VV);
>> surf(XX,YY,ZZ);
>> axis equal
```



The extra `axis equal` command ensures that the aspect ratio is the same in all directions and hence the the sphere looks round and not squashed!

As a last application, consider the hyperboloid of one sheet

$$x^2 + y^2 - 1 = z^2.$$

We can construct a graph of it as a parameterized surface using hyperbolic trignonometic functions. Recall that hyperbolic sine and hyperbolic cosine satisfy the identity

$$\cosh^2(v) - \sinh^2(v) = 1.$$

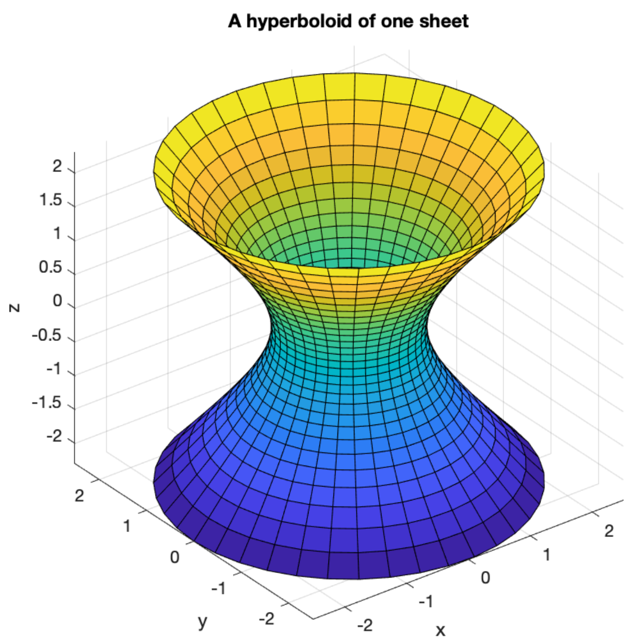
So if $x^2 + y^2 = \cosh^2(v)$, and if $z = \sinh(v)$, then $x^2 + y^2 - z^2 = 1$, exactly the condition for lying on the hyperboloid. To parameterize the curve $x^2 + y^2 = R^2$ it is convenient to use an angle: $x = R\cos(u)$ and $y = R\sin(u)$. Setting $R = \cosh(v)$ gives us the parameterization we want:

$$x = \cos(u)\cosh(v)$$
$$y = \sin(u)\cosh(v)$$
$$z = \sinh(v)$$

Neat, hey? This looks a lot like the parameterization of the sphere, except that the v variable is associated with hyperbolic trig functions. Anyway, here's the MATLAB.

```
>> u=linspace(-pi,pi,40);v=linspace(-3,3,30);
>> XX=cos(UU).*cosh(VV); YY=sin(UU).*cosh(VV); ZZ=sinh(VV);
>> surf(XX,YY,ZZ);
```

```
>> axis equal
>> xlabel('x');ylabel('y');zlabel('z')
>> title("A hyperboloid of one sheet")
```



Exercises

Exercise 1:

Graph the function $f(x, y) = xy/(x^2 + y^2)$ with $-3 \le x \le 3$ and $-3 \le y \le 3$. What's going on with this function at $(x, y) = (0, 0)$?

Exercise 2:

Graph the cone $z = \sqrt{x^2 + y^2}$.

Exercise 3:

Make a contour plot of the function $f(x, y) = \sin(x)\cos(y)$ with $-\pi \le x \le \pi$ and $-\pi \le y \le \pi$.

Exercise 4:

Make a graph of the surface from the previous exercise. Do you see how the contour plot and the graph are related to each other?

Exercise 5:

Make a graph of the ellipse in the plane $x^2 + 3y^2 = 1$. Hint: $\cos(t)^2 + 3(\sin(t)/\sqrt{3})^2 = 1$.

Exercise 6:

Use the `surf` command to construct the ellipsoid defined by $x^2 + 3y^2 + z^2 = 1$. Be sure to use the `axis equal` command so that the aspect ratio in all directions is the same. The ellipsoid should be squished in the y-direction only.