Physics 326G
Winter 2008

Class 8

We have learned how to do two-dimensional (*x*-*y*) plots, but Matlab also has a number of built-in functions to do three-dimensional graphics. These are useful when you are looking at functions of two variables, or for interpretation of data in the form of images. Some of these functions are similar to the functions we have used for 2D plotting, but some are quite different.

There is a 3D version of the PLOT command, called PLOT3. This command will plot data on three dimensional (*x*-*y*-*z*) axes. You can specify the line or symbol type and control other properties of the plot in the same way as you did with PLOT. The syntax is

```
plot3(x,y,z,'line specifiers','PropertyName',property value)
```

The vectors *x*, *y*, and *z* are the coordinates of your data points. All three vectors must have the same number of elements. The line specifiers and property values are the same as in two dimensions.

Suppose we had a particle moving through space so that its *x*, *y*, and *z* coordinates for a range of time *t* were given by the following:

```
t = 0:.1:50;
x = t.*sin(t);
y = t.*cos(t);
z = 2*t;
```

We can plot these data with the command

```
plot3(x,y,z)
```

The function SCATTER3 plots a symbol at each data point, and has the amusing feature that the size and color of the symbol can vary for each point (there is also a function SCATTER for 2D graphing). Try

```
scatter3(x,y,z,(z/4).^2+1,y)
```

which gives a plot in which the size of the circles gets bigger as *z* gets bigger, and the color depends on the *y* coordinate.

Matlab plots the 3D graph as if it was viewed from a certain direction. You can change the viewing direction in two ways. First, there is the command VIEW. In its simplest form, the input arguments to VIEW are the altitude and azimuth angles that define the direction of the viewer.

```
view(0,90)
```

gives you a top view.

```
view(0,0)
```
or
```
view(90,0)
```

give you a side views. The default settings are -37.5 degrees and +30 degrees respectively.

The second way to change the view is interactively in the plot window. On the tools menu is a tool called *rotate 3d*, or there is a button on the plot window tool bar for the same tool. Click it, then you can drag the mouse across the plot to chage the viewpoint. Try it.

There are many other types of 3D plots useful for plotting functions of the type $z = f(x,y)$, where z is evaluated on a grid of x,y points. In this case, z defines a surface in the three dimensional space (x,y,z). Let's create some data. We can make a grid with MESHGRID as before.

```
[x,y] = meshgrid(-10:10,-10:10);
```

will create a grid with x and y both running from -10 to 10. Then we can calculate z at all of these grid points:

```
z = cos(x/2).*exp(-((x/5).^2+(y/5).^2));
```

Now let's plot the surface defined by z with the function SURF.

```
surf(x,y,z)
```

There are, as usual, many properties of surface graphs that you can control by setting property values. For example, you can put symbols at each data point like this:

```
surf(x,y,z,'Marker','square');
```

You can change the appearance of the graph by changing the way Matlab colors the pieces of the surface. The command to do this is SHADING and the choices are

```
shading flat
shading faceted (which is the default)
shading interp
```

Try them all. You can also change the colormap used by Matlab to determine how to color the plot. Try

```
colormap spring
```

and

```
colormap gray
```

There are several other colormaps to choose from; you can see a list of them by typing

```
help graph3D
```

You can make a similar plot but without filling in the ‘patches’ between the grid point at all by using the command MESH. Try it.

There are several variations on MESH and SURF that look a little different. You should look them up with HELP and try them so you know what they do. They are

```
meshz
surfl
waterfall
```

More useful, in my opinion anyway, are the following, which draw contour plots of the function as well as the mesh or surface plot.

```
surfc
meshc
```

You can get the contour plots by themselves using

```
contour(x,y,z,n)
```

where *n* is an optional argument specifying the number of contour lines to draw. You can get a 3D contour plot with

```
contour3(x,y,z,n)
```

Make a new set of data like this:

```
[x,y] = meshgrid(-100:100,-100:100);
z = exp(-((x/30).^2+(y/30).^2)).*(y.^2+x.^2);
```

and try out the various ways of making 3D plots. You will probably find that some plots don’t look very good. In fact it is often very hard to present 3D data in a way that is easy to interpret, especially when you have a lot of data. It takes a lot of practice and skill to use 3D plots in a way that makes them helpful rather than confusing. Use them with discretion.

Sometimes you may want to plot a vector field, such as an electric field or a velocity field, and want to show the direction of the field as well as its magnitude. Matlab has a function called QUIVER that allows you to do this. Consider two 1 microCoulomb electric charges, one at $(x,y) = (1,0)$ and the other at $(x,y) = (-1,0)$. The field around a point charge is

$$\vec{E} = \frac{kq}{r^2}\hat{r}$$

where *r* is the distance from the charge and $\hat{r}$ is a unit vector in the radial direction. The MAT file *charge.mat* contains data for the grid coordinates (in matrices called *x* and *y*) as well as the *x* and *y* components of the total electric field due to the two charges in the region $-0.5 < x < 0.5$ and $-0.5 < y < 0.5$, in between the two charges. Load this file, and use QUIVER to make a plot of the electric field vector.

QUIVER3 allows you to do the same thing in 3 dimensions.

Here are some practice problems to work through using 3D graphics.

1. A projectile is launched due north at an angle of 60 degrees to the horizontal with an initial velocity of 250 m/s. There is a strong wind blowing to the east, which causes the projectile to acquire a constant eastward component of it velocity of v_w m/s. Plot (on the same graph) the trajectory of the particle in 3 dimensions for different values of v_w. Use a different colored line for each value of v_w.
2. The amplitude of the current in a series LRC circuit is given by $I = \dfrac{V}{\sqrt{R^2 + (\omega L - 1/\omega C)^2}}$ where $L = 0.240$ H is the inductance, $C = 15 \times 10^{-6}$ F is the capacitance, *R* is the resistance, $V = 24$ V is the amplitude of the applied voltage, and ω is the frequency of the applied AC voltage. Calculate *I* for *R* from 10 to 40 Ohms and ω from 300 to 700 s^{-1}. Make a 3D plot of the results. Also make a contour plot of your results. Estimate the resonant frequency of the circuit. Does your estimate agree with the theoretical prediction $\omega_0 = \sqrt{1/LC}$?

3. The ideal gas law again! We know that $P = nRT/V$ where *n* is the number of moles of gas and *R* is the gas constant, 8.31 J/mol K. Taking *T* from 200 to 500 K and V from 0.0005 to 0.002 m^3, calculate the pressure for one mole of gas. Make a 3D plot of your results.