2D Plotting with matplotlib

Written by Dave Pawlowski,

Indroduction

There are several ways to make a 2D plot using matplotlib. We are only going to discuss one of these here: the contour plot. Making a contour plot is not much different than plotting 1D data. There are a couple differences. The first one is that contour plots need 3 variables instead of just 2. It is best to have all three variables be of the same dimension. To accomplish this, I use Numpy's meshgrid function. Finally, instead of calling the plot function, we use pyplot's contour function. Let's do an example:

```
import numpy as np
from matplotlib import pyplot as p

#Establish a grid.
x = np.arange(-.051,.05,.002)
y = np.arange(-.051,.05,.002)

#contour wants the x and y values to be defined at every point
#on the grid.  I.e. x and y should really be 2D variables.
X, Y = np.meshgrid(x,y)

#We need some data to plot
r1 = np.sqrt((X-0.03011)**2+(Y-0)**2)
r2 = np.sqrt((X+0.03011)**2+(Y-0)**2)
epsilon_0 = 8.85e-12
Z = 1e6/(4*np.pi*epsilon_0*r1)+1e6/(4*np.pi*epsilon_0*r2)

#plot it
p.contour(X,Y,Z,100)
p.title('Potential due to two point charges')
p.xlabel('x (cm)')
p.ylabel('y (cm)')
p.savefig('plot.ps')
```

Note that the 4th argument in the contour call, '100', is optional. That simply specifies the number of contour levels, or lines, to display. When used, pyplot figures out the spacing between levels automatically. The results are shown in Figure 1.

Alternatively, one can use a filled contour plot simply by using contourf instead of contour. See Figure 2.

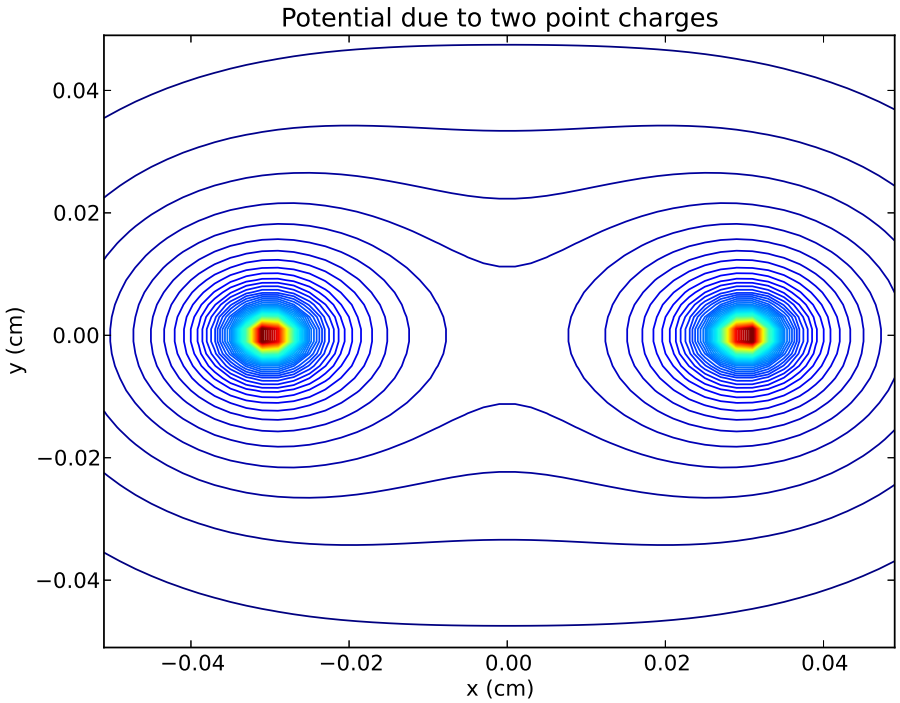

Figure 1: Potential due to point charges

```
.....
#plot it
p.contourf(X,Y,Z,100)
p.title('Potential due to two positive point charges')
...
```

In both cases, you'll notice that something is missing. We have no idea what the colors represent. We need a colorbar! To do this, we need to change our call to contourf slightly.

```
.....
#plot it
cont = p.contourf(X,Y,Z,100)
cb = p.colorbar(cont)
cb.set_label('Potential (V)')
...
```

Here we have saved the result of the contour to a variable so that we can let the colorbar method operate on it. Python does all the work behind the scenes to produce a professional plot. We simply need to add a label. See Figure 3.

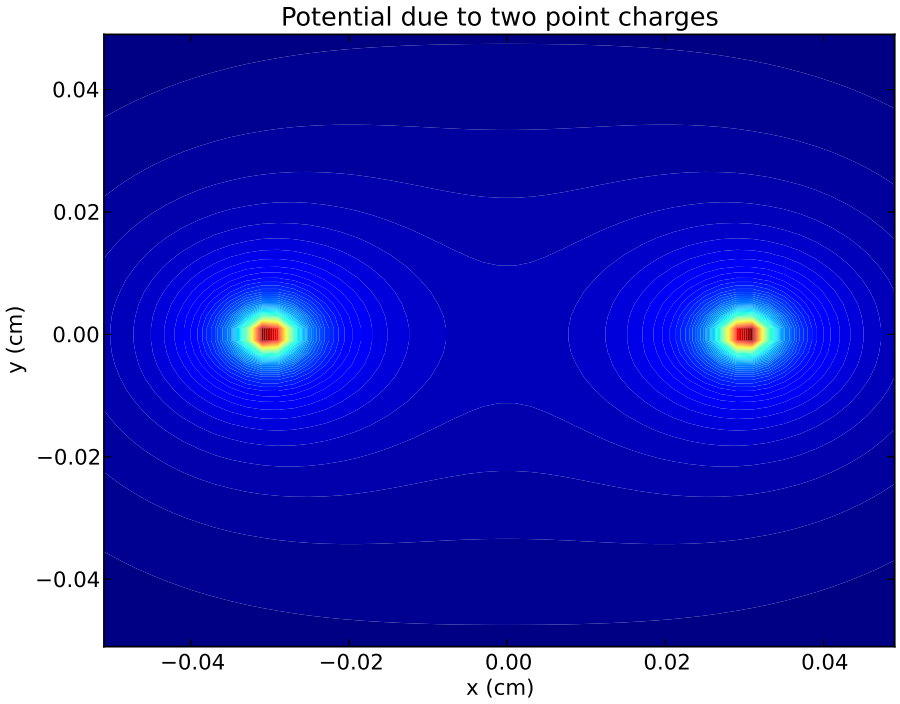

Figure 2: Potential due to positive point charges, filled

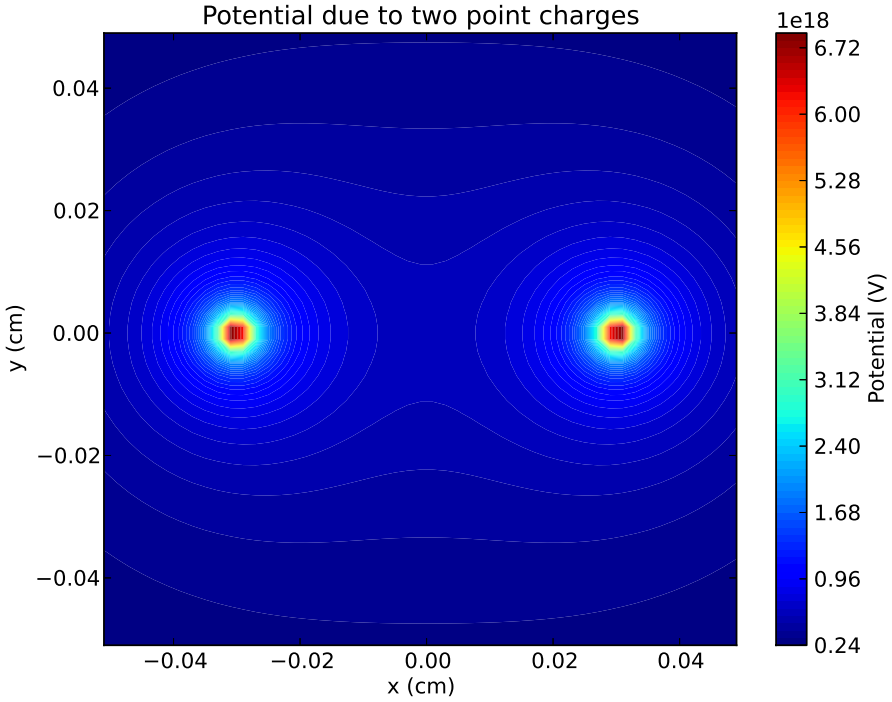

Figure 3: Potential due to positive point charges- with CB