# Matplotlib Tutorial: 5. 3D plotting

The 3D plotting toolkit introduced in matplotlib version 1.0 can lead to some very nice plots. We'll explore a few of the options here: for more examples, the matplotlib tutorial is a great resource.

Again we'll use inline plotting (though keep in mind that it can be useful to skip the "inline" backend to allow interactive manipulation of the 3D plots).

In [1]:
```
%matplotlib inline

from __future__ import print_function, division
import matplotlib.pyplot as plt
import numpy as np
```

In [2]:
```
# This is the 3D plotting toolkit
from mpl_toolkits import mplot3d
```

Just as before when we created a 2D axes and called several plotting methods on it, here we'll create some 3D axes objects and call 3D plotting routines. Below are several examples.

## 3D scatter Plot

The 3D scatter plot takes all the same keyword parameters as the 2D scatter plot, so its use should be familiar. To create a 3D axes, we need to pass the argument `projection='3d'`.

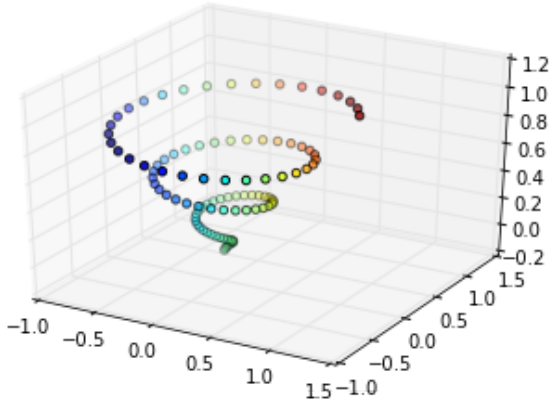
In [3]:
```
fig = plt.figure()
ax = plt.axes(projection='3d')

z = np.linspace(0, 1, 100)
x = z * np.sin(20 * z)
y = z * np.cos(20 * z)

c = x + y

ax.scatter(x, y, z, c=c)
```

Out[3]:
```
<mpl_toolkits.mplot3d.art3d.Path3DCollection at 0x106dc1c10>
```

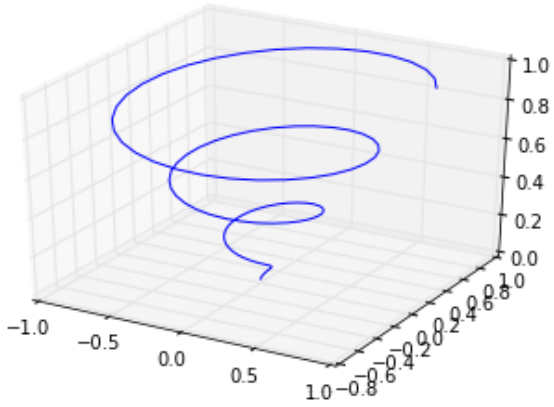



## 3D Line Plot

Like the 2D and 3D `scatter` command, the 2D and 3D `plot` command have the same argument structure. Thus plotting a 3D line plot is straightforward:

In [4]:
```
fig = plt.figure()
ax = plt.axes(projection='3d')

ax.plot(x, y, z, '-b');
```



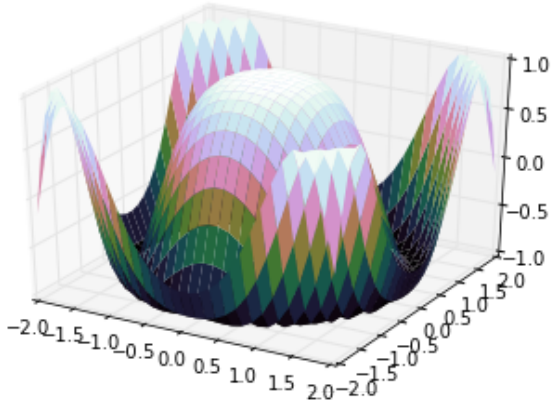

## Surface Plot

Surface plots are connected plots of x, y, and z coordinates. They can be used to make some pretty interesting shapes:

In [5]:
```
x = np.outer(np.linspace(-2, 2, 30), np.ones(30))
y = x.copy().T
z = np.cos(x ** 2 + y ** 2)

fig = plt.figure()
ax = plt.axes(projection='3d')

ax.plot_surface(x, y, z, cmap='cubehelix', rstride=1, cstride=1, linewidt
h=0);
```




## Wire-frame plot

Wire-frame plots draw lines between nearby points. They can be displayed using the `plot_wireframe` method. Here we'll plot a parametrized sphere:

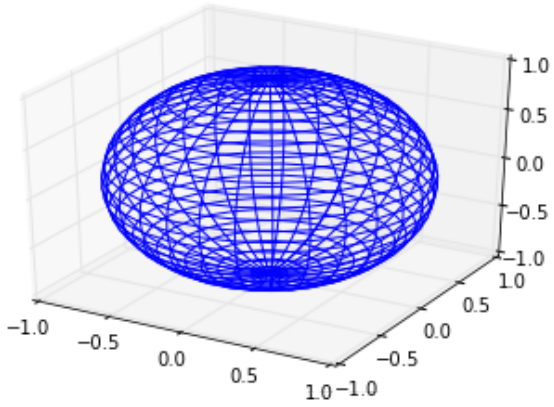
In [6]:
```
u = np.linspace(0, np.pi, 30)
```

```
v = np.linspace(0, 2 * np.pi, 30)

x = np.outer(np.sin(u), np.sin(v))
y = np.outer(np.sin(u), np.cos(v))
z = np.outer(np.cos(u), np.ones_like(v))


fig = plt.figure()
ax = plt.axes(projection='3d')

ax.plot_wireframe(x, y, z);
```




## More 3D Plot Examples

There are many more 3D plotting examples to explore; see the matplotlib documentation for more information.