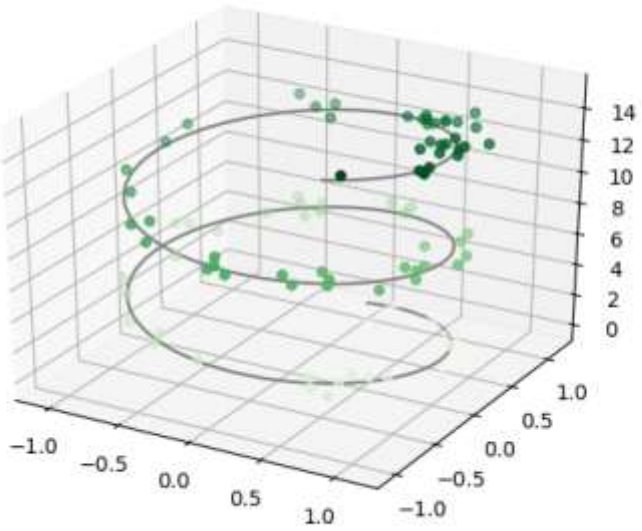
# 3D Graphs with Matplotlib

AMAN KHARWAL / MAY 2, 2020 / MACHINE LEARNING

## Three-Dimensional points and lines

The most basic three-dimensional plot is a line or scatter plot created from sets of (x,y,z) triples. In analogy with more common two-dimensional plots, we can create these using the ax.plot3D and ax.scatterd3D functions. The call signature of these is nearly identical to that of their two-dimensional counterparts. Here we will plot a trigonometric spiral, along with some points drawn randomly near the line:

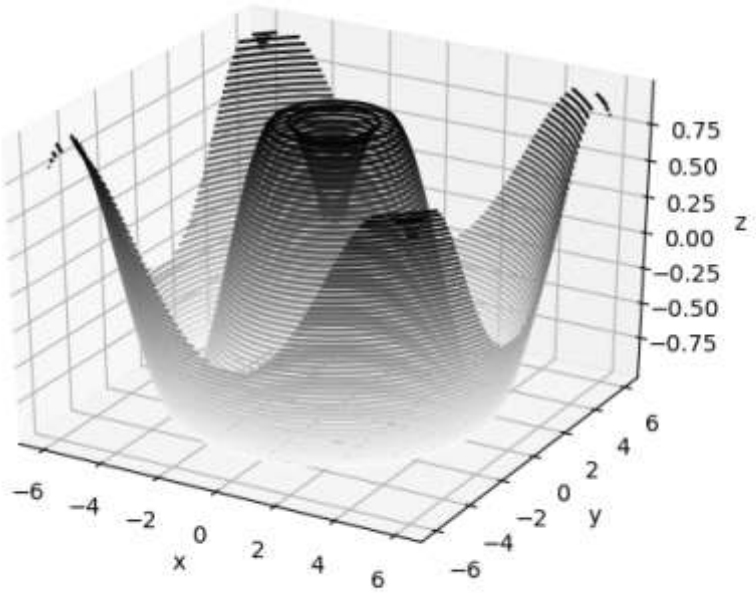
```
import numpy as np
import matplotlib.colors as col
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
#Data for a three dimensional line
z = np.linspace(0, 15, 1000)
x = np.sin(z)
y = np.cos(z)
ax.plot3D(x, y, z, 'grey')
#Data for three dimensional scattered points
z = 15 * np.random.random(100)
x = np.sin(z) + 0.1 * np.random.randn(100)
y = np.cos(z) + 0.1 * np.random.randn(100)
ax.scatter3D(x, y, z, c=z, cmap='Greens')
plt.show()
```



# Three-Dimensional contour plots

Like two-dimensional ax.contour plots, ax.contour3D requires all the input data to be in the form of two-dimensional regular grids, with the z data evaluated at each point. Here we will show a three-dimensional contour diagram of a three-dimensional sinusoidal function:

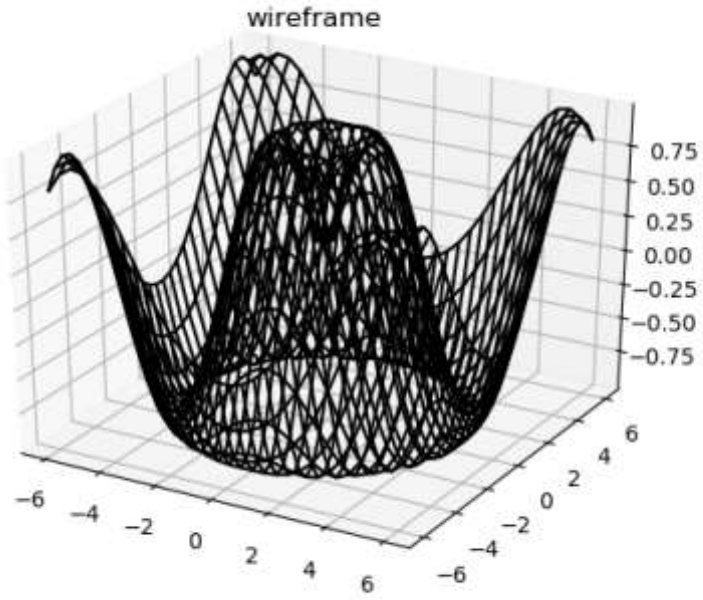
```
def f(x, y):
    return np.sin(np.sqrt(x ** 2 + y ** 2))
x = np.linspace(-6, 6, 30)
y = np.linspace(-6, 6, 30)
x, y = np.meshgrid(x, y)
z = f(x, y)
fig = plt.figure()
ax = plt.axes(projection='3d')
ax.contour3D(x,y,z,50, cmap='binary')
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('z')
plt.show()
```




# Wireframes and Surface Plots

Two other types of three-dimensional plots that work on gridded data are wireframes and surface plots. These take a grid of values and project it onto the specified three-dimensional surface, and can make the resulting three-dimensional forms quite easy to visualize. Here's an example using a wireframe:

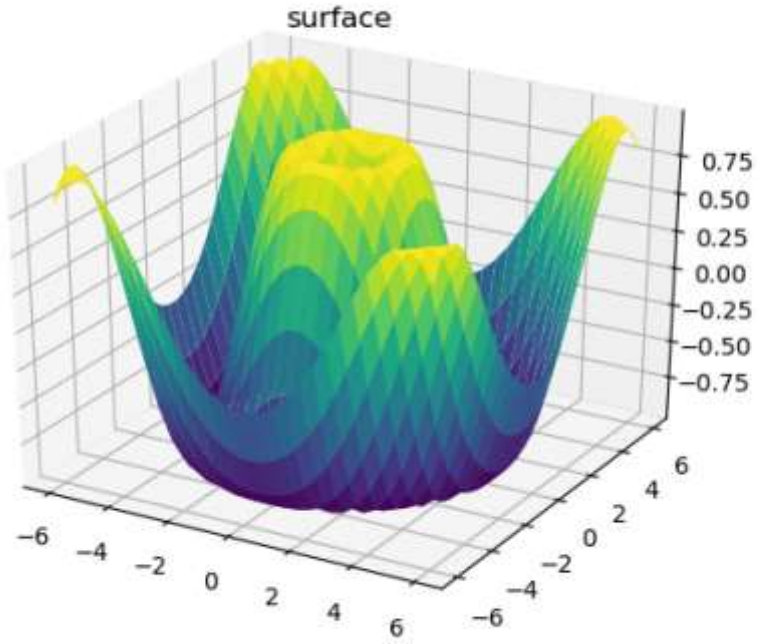
```
fig = plt.figure()
ax = plt.axes(projection='3d')
ax.plot_wireframe(x,y,z, color='black')
ax.set_title('wireframe')
plt.show()
```




A surface plot is like a wireframe plot, but each face of the wireframe is a filled polygon. Adding a colormap to the filled polygons can aid perception of the topology of the surface being visualized:

```
ax = plt.axes(projection='3d')
ax.plot_surface(x, y, z, rstride=1,
                cstride=1, cmap='viridis',
```

```
                edgecolor='none')
ax.set_title('surface')
plt.show()
```




**Aman Kharwal**

Data Strategist at Statso. My aim is to decode data science for the real world in the most simple words.

ARTICLES: 1614

PREVIOUS POST

Python GUI App for – Student Details

NEXT POST

Who Uses Python Today?

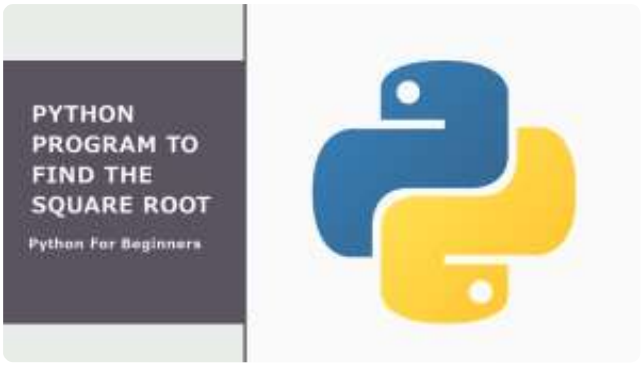
# Recommended For You


## Break and Continue in Python

March 25, 2022

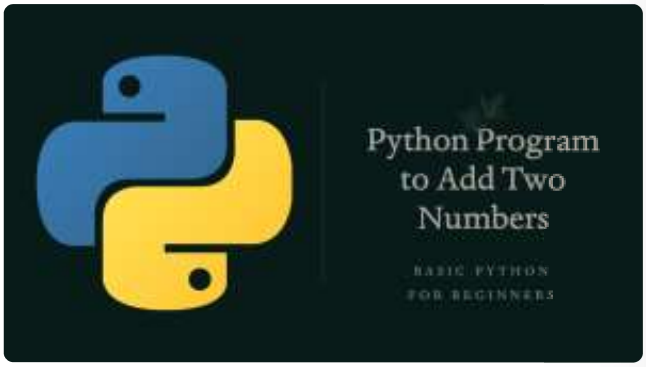

## Area of Triangle with Python

October 11, 2020




## Square Root with Python

October 10, 2020




## Python Program to Add Two Numbers

October 8, 2020

## Leave a Reply

FACEBOOK INSTAGRAM MEDIUM LINKEDIN