

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

```
import pandas as pd
```

Line Plot

Here Im writing a function that mimics $f(x_1, x_2)$

```
def func(x1,x2):
    f = []
    for i in range(len(x1)):
        f.append((x2[i]-x1[i])**4 + 8*x1[i]*x2[i] -x1[i]+x2[i]+3)
    return np.array(f)
```

```
# lets generate range of values for f(x)
```

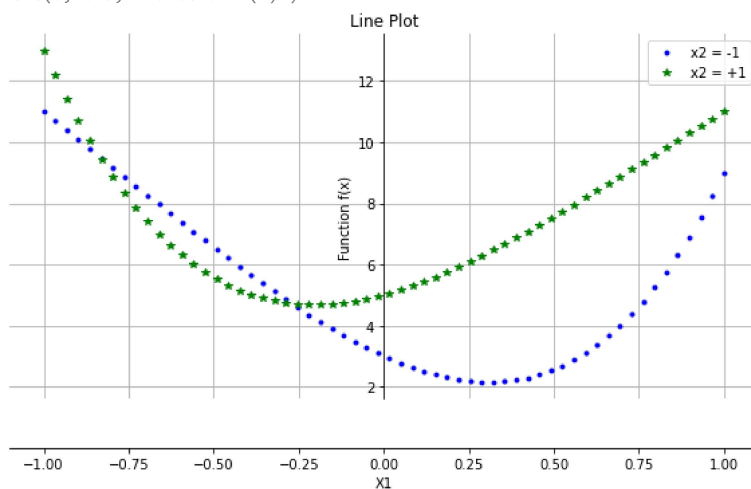
```
ran = np.linspace(-1,1,60)
```

```
fig = plt.figure(figsize = (10,5))
ax = fig.add_subplot(1, 1, 1)
ax.spines['left'].set_position('center')
ax.spines['bottom'].set_position('zero')
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.xaxis.set_ticks_position('bottom')
ax.yaxis.set_ticks_position('left')
```

```
plt.plot(ran,func(ran,[-1]*len(ran)), 'b.')
plt.plot(ran,func(ran,[1]*len(ran)), 'g*')
plt.legend(['x2 = -1', 'x2 = +1'])
```

```
plt.grid()
plt.title('Line Plot')
plt.xlabel('X1')
plt.ylabel('Function f(x)')
```

```
Text(0, 0.5, 'Function f(x)')
```



```
#### The block of code below is to generate the ***surface plot*** for function f(x)
```

```
from mpl_toolkits import mplot3d
```

```
fig, ax = plt.subplots(subplot_kw={"projection": "3d"},figsize= (10,7))
```

```
x1 = np.array(np.linspace(-1,1,60))
x2 = np.array(np.linspace(-1,1,60))
```

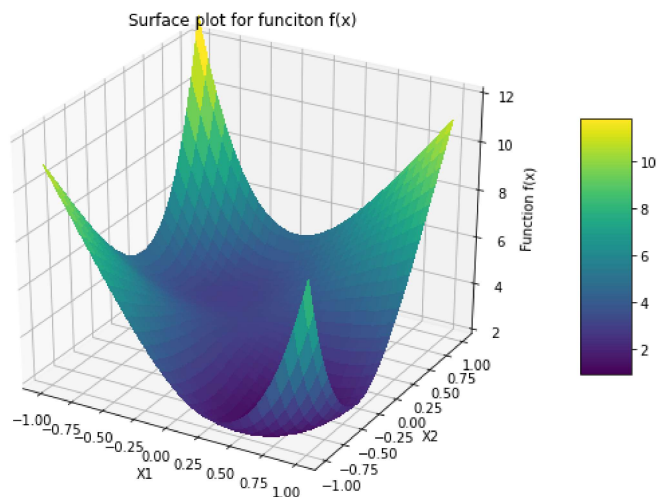
```
x1,x2 = np.meshgrid(x1,x2)
```

```
surf = ax.plot_surface(np.array(x1),np.array(x2),np.array(func(x1,x2)), cmap='viridis',
                      linewidth=0, antialiased=False)
```

```
# Customize the z axis.
ax.set_zlim(2, 12)
```

```
#from matplotlib.ticker import LinearLocator
#ax.zaxis.set_major_locator(LinearLocator(10))
# A StrMethodFormatter is used automatically
#ax.zaxis.set_major_formatter('{x:.02f}')

# Add a color bar which maps values to colors.
fig.colorbar(surf, shrink=0.5, aspect=5)
plt.xlabel('X1')
plt.ylabel('X2')
ax.set_zlabel('Function f(x)')
plt.title('Surface plot for funciton f(x)')
plt.show()
```



```
fig, ax = plt.subplots(1,1,figsize= (10,7))

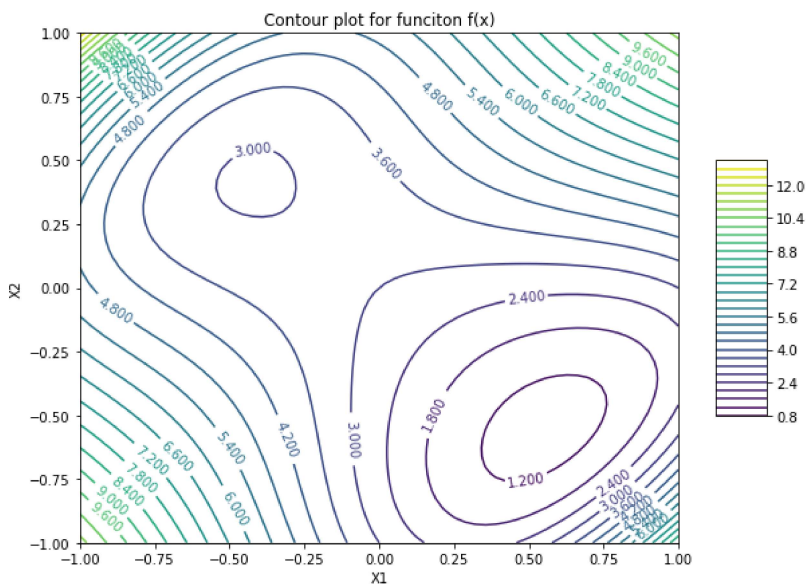
x1 = np.array(np.linspace(-1,1,60))
x2 = np.array(np.linspace(-1,1,60))

x1,x2 = np.meshgrid(x1,x2)

contour = ax.contour(np.array(x1),np.array(x2),np.array(func(x1,x2)),levels = 20)
ax.clabel(contour, inline=True, fontsize=10)

# Add a color bar which maps values to colors.
fig.colorbar(surf, shrink=0.5, aspect=5)
plt.xlabel('X1')
plt.ylabel('X2')

plt.title('Contour plot for funciton f(x)')
plt.show()
```



Colab paid products - [Cancel contracts here](#)

✓ 1s completed at 6:39 PM

