In [1]:
```
import sys
import numpy as np
import matplotlib
import matplotlib.pyplot as plt

print("Python: {}".format(sys.version))
print("NumPy: {}".format(np.__version__))
print("Matplotlib: {}".format(matplotlib.__version__))
```

```
Python: 3.7.4 (default, Aug  9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)]
NumPy: 1.18.1
Matplotlib: 3.1.1
```

In [2]:
```
# generate 2d mesh grid
nx, ny = (100,100)

x = np.linspace(0, 10, nx)
y = np.linspace(0 ,10, ny)

xv, yv = np.meshgrid(x,y)
```

In [4]:
```
xv.shape
```

Out[4]:
```
(100, 100)
```

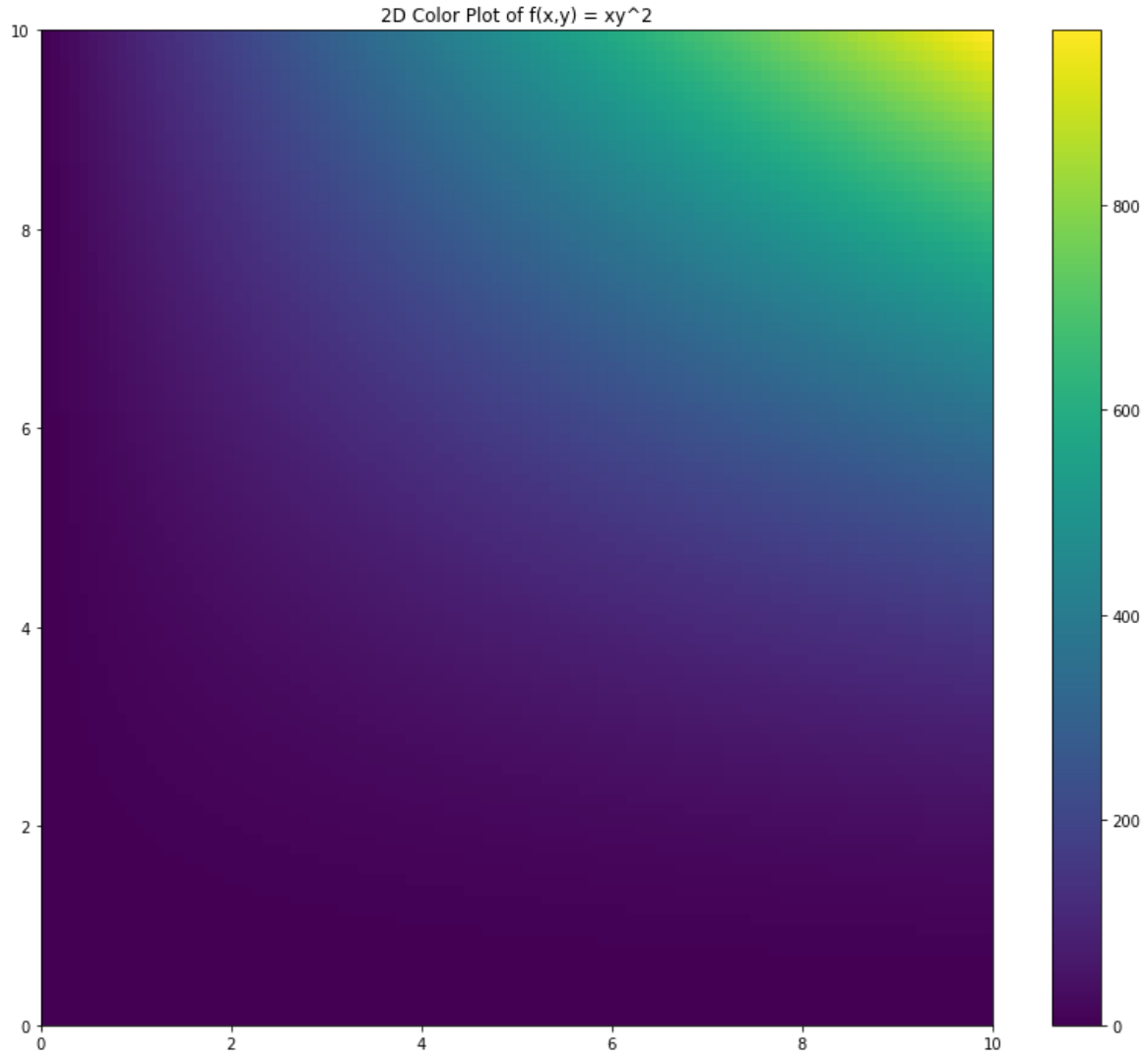
In [5]:
```
# define a function to plot
def f(x,y):
    return x* (y**2)

# calculate Z value for each x,y point
z = f(xv, yv)
```

In [6]:
```
z.shape
```

Out[6]:
```
(100, 100)
```

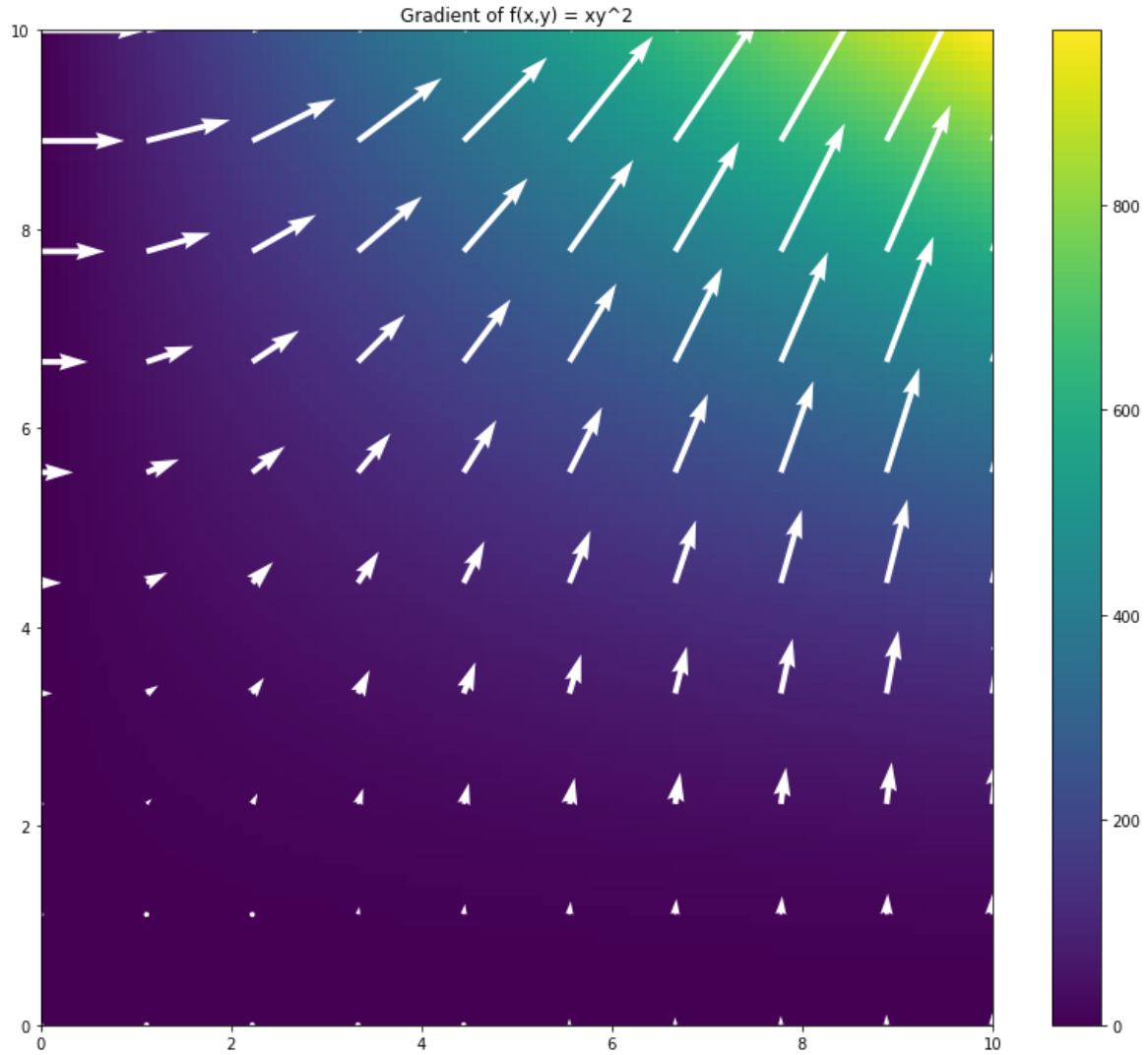
In [10]:
```
# make a color plot to display the data
plt.figure(figsize=(14,12))
plt.pcolor(xv, yv, z)
plt.title("2D Color Plot of f(x,y) = xy^2")
plt.colorbar()
plt.show()
```



In [11]:
```
# generate 2D meshgrid for the gradient
nx, ny = (10, 10)
x = np.linspace(0,10,nx)
y = np.linspace(0,10,ny)
xg,yg = np.meshgrid(x,y)

# calculate the gradient of f(x,y)
Gy, Gx = np.gradient(f(xg,yg))
```

In [12]:
```
# make a color plot to display the data
plt.figure(figsize=(14,12))
plt.pcolor(xv, yv, z)
plt.title("Gradient of f(x,y) = xy^2")
plt.colorbar()
plt.quiver(xg, yg, Gx, Gy, scale = 1000, color = 'w')
plt.show()
```

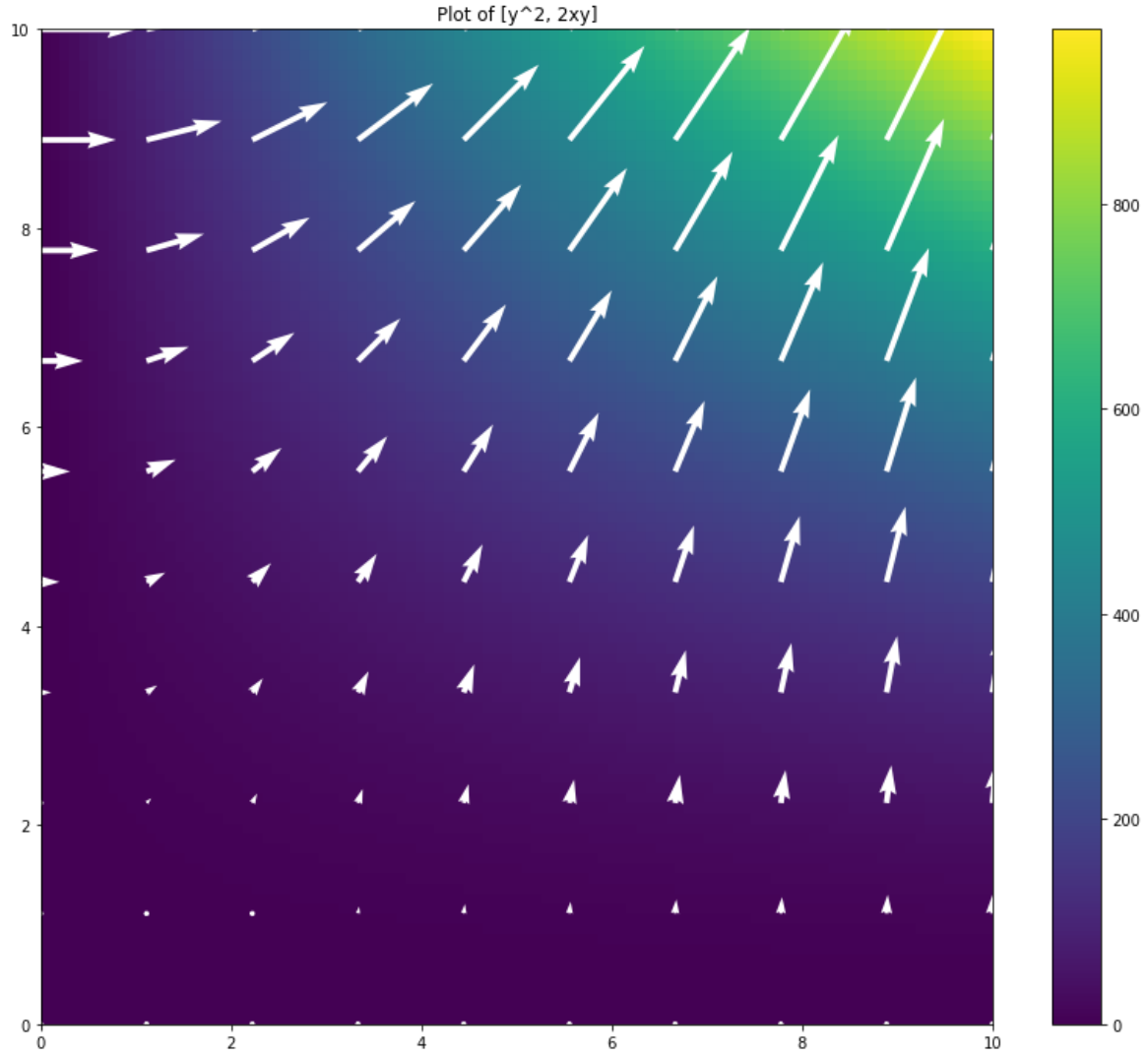


In [13]:
```
# calculate the gradient of f(x,y) = xy^2
def ddx(x,y):
    return y ** 2

def ddy(x,y):
    return 2*x*y

Gx = ddx(xg, yg)
Gy = ddy(xg, yg)
```

In [14]:
```
# make a color plot to display the data
plt.figure(figsize=(14,12))
plt.pcolor(xv, yv, z)
plt.title("Plot of [y^2, 2xy]")
plt.colorbar()
plt.quiver(xg, yg, Gx, Gy, scale = 1000, color = 'w')
plt.show()
```



In [ ]:
```

```