11. Matplotlib (Abhijit Kar Gupta)

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

```
In [2]: #help(plt.plot)
```

Simple Commands:

1. Plot x and y using blue circle markers: **plt.plot(x,y, 'bo')** .
2. Plot with red plusses: **plt.plot(x,y, 'r+')** .
3. Plot y using x from 0 to n (default).

1. Name and Size a figure (dpi - dots per inch).
2. Save a figure.

```
In [3]: plt.figure('Demo figure', figsize=(8,4), dpi=100)
        x1 = np.linspace(0,1.2,50)
        plt.subplot(1,2,1)
        plt.plot(range(10))
        plt.grid()

        plt.subplot(1,2,2)
        for i in range(5):
            plt.plot(x1,x1**i)
        plt.grid()
        plt.savefig('fig1.png')
        plt.savefig('fig1.pdf')
```
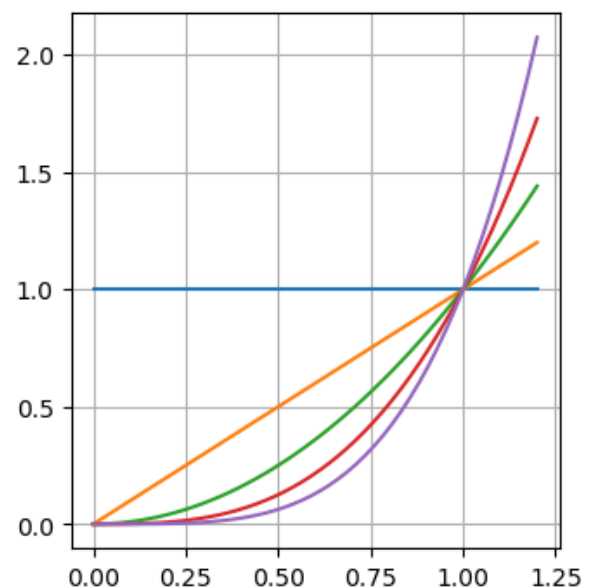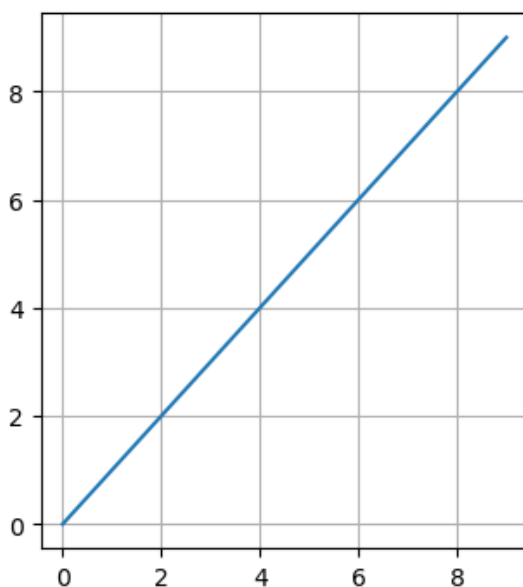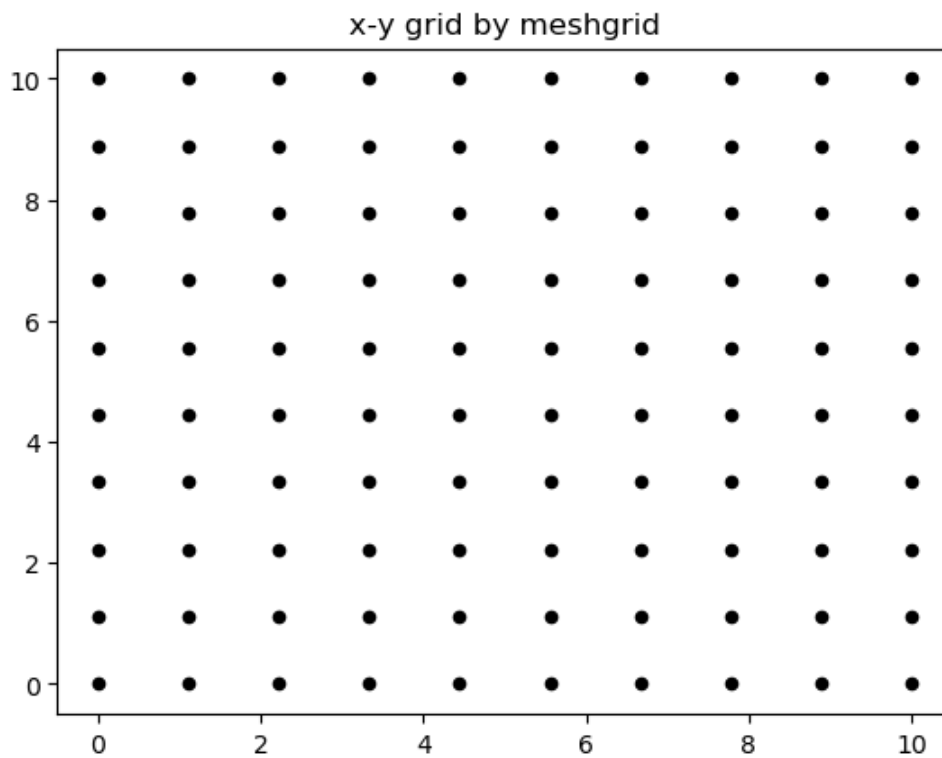


```
In [ ]:
```

```
In [ ]:
```

In [ ]:

Contour Plot

In [4]:
```python
x = np.linspace(0,10,10)
y = np.linspace(0,10,10)
xm, ym = np.meshgrid(x,y)

plt.plot(xm, ym, '.', ms=9, color='k')
plt.title('x-y grid by meshgrid')
plt.show()
```
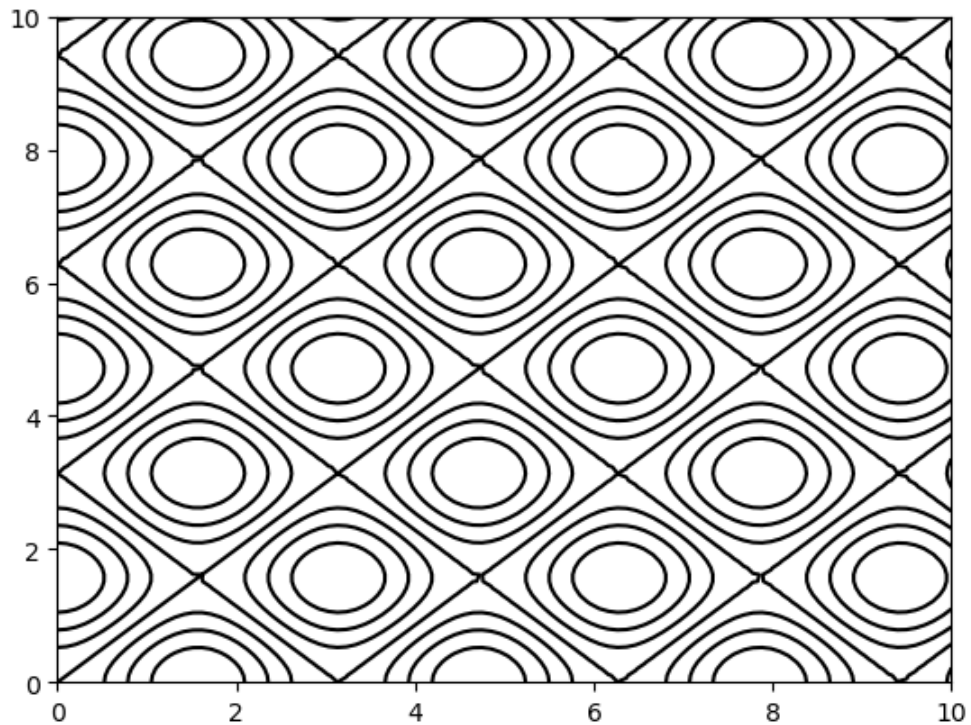

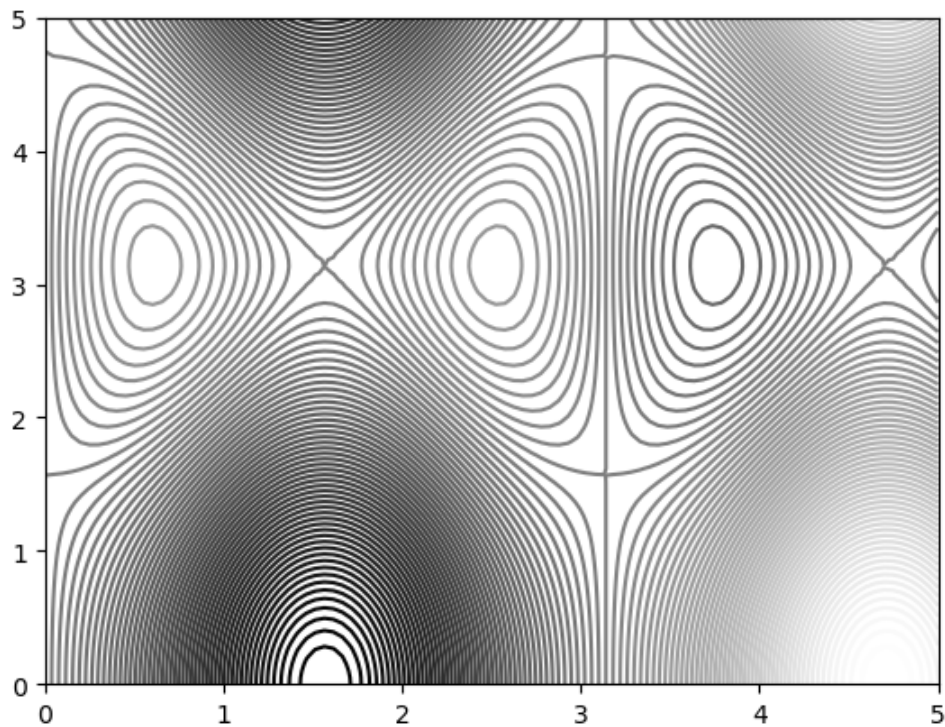x-y grid by meshgrid

In countour plots we plot $z(x, y)$.

In [5]:
```python
x = np.linspace(0,10,100)
y = np.linspace(0,10,100)
xm, ym = np.meshgrid(x,y)
Z = np.sin(xm)**2 + np.cos(ym)**2

plt.contour(xm, ym, Z, colors='k')
```

Out[5]: <matplotlib.contour.QuadContourSet at 0x1bbca77b730>

```
In [6]: def f1(x,y):
            return np.sin(x)**3 + np.cos(y)*np.sin(x)

        x = np.linspace(0,5,100)
        y = np.linspace(0,5,100)
        xm, ym = np.meshgrid(x, y)
        z1 = f1(xm, ym)

        plt.contour(xm, ym, z1, 100, cmap='binary')
        plt.show()
```
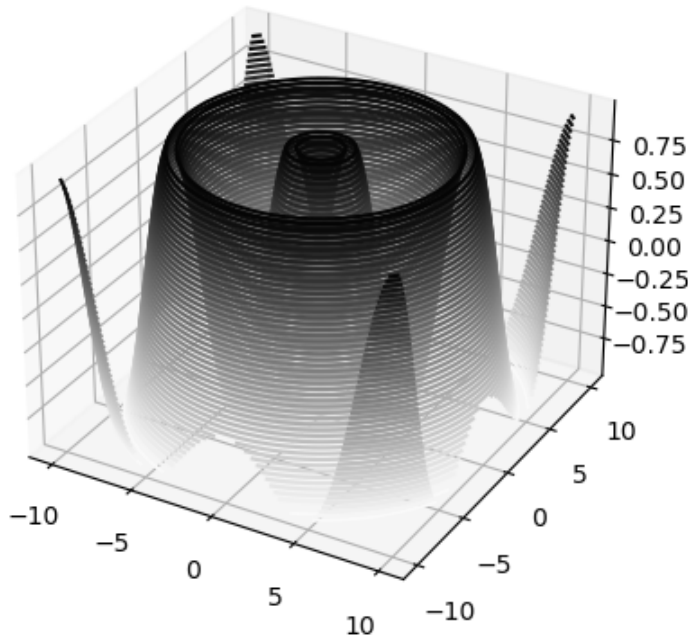

Here, the number 100 is to control the density of contours.

11.2 3D Plot

```
In [7]: from mpl_toolkits import mplot3d
```

```
In [8]: x = np.linspace(-10,10,100)
        y = np.linspace(-10,10,100)
        xm, ym = np.meshgrid(x, y)
        Z = np.sin(np.sqrt(xm**2+ym**2))

        #fig = plt.figure()
        ax = plt.axes(projection='3d')
        ax.contour3D(xm, ym, Z, 50, cmap='binary')
        plt.show()
```


We can produce color contours and 3D color images through **cmap** (color map) parameter. See *matplotlib surface plot documentation section (matplotlib website)* for details.

```
x = np.linspace(0,10,50)
y = np.linspace(0,10,50)
z = np.linspace(0,10,50)
xm, ym, zm = np.meshgrid(x, y, z)
f1 = xm**2 + ym**2 + zm**2
fig = plt.figure()
ax = plt.axes(projection='3d')
ax.plot(xm, ym, f1, 50, cmap='binary')
plt.show()
```

In [ ]: