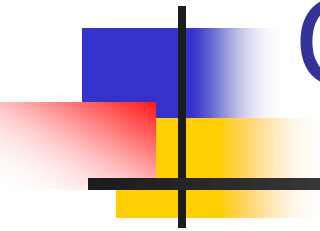
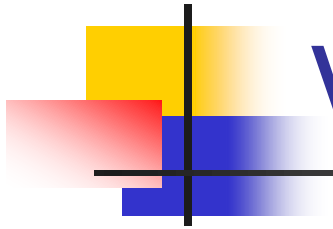


Computer Methods (MAE 3403)

Visualization



Visualization

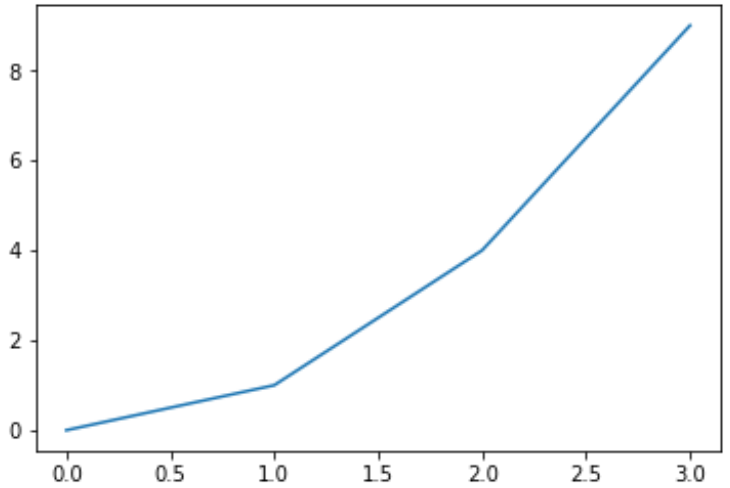
- A necessary component to develop, validate code and work with data

- module (similar to MATLAB plotting): matplotlib.pyplot
  - import matplotlib.pyplot as plt
  - Need to install separately

2D plot

- Simplest plot: plot(x,y)

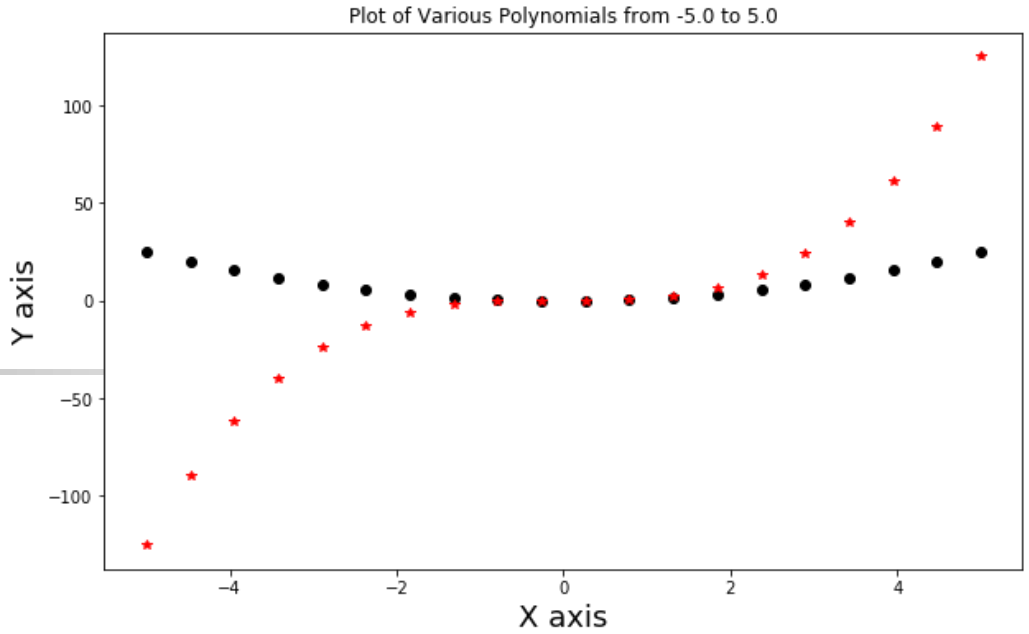
```
x = [0, 1, 2, 3]
y = [0, 1, 4, 9]
plt.plot(x, y)
plt.show()
```



- Change marker or line or color: plot(x,y,'ro')
  - color: b, g, r, c, m, y, k, w
  - marker: ., o, x, +, *, s, d
  - line style: -, -., :, -,

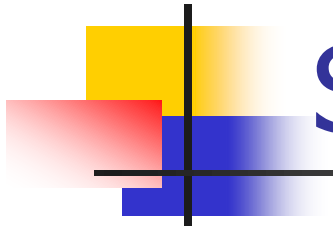
```
x = np.linspace(-5,5,20)
plt.plot(x, x**2, 'ko')
plt.plot(x, x**3, 'r*')
plt.show()
```

Labels



- Always give a title and axis labels
  - title('Trajectory of the rocket')
  - xlabel('time (sec)')
    ylabel('position (m)')
- Change the size of a figure
  - plt.figure(figsize = (10,6))

```
plt.figure(figsize = (10,6))
x = np.linspace(-5,5,20)
plt.plot(x, x**2, 'ko')
plt.plot(x, x**3, 'r*')
plt.title(f'Plot of Various Polynomials from
{x[0]} to {x[-1]}')
plt.xlabel('X axis', fontsize = 18)
plt.ylabel('Y axis', fontsize = 18)
plt.show()
```



Styles

- Almost any part of the figure can be customized, e.g., change font size, color, etc.
- Use xlim, ylim to change the limits of the axis. Use grid to turn on the grid of the figure.
  - plt.xlim(-6, 6), plt.grid()
- Predefined plotting style

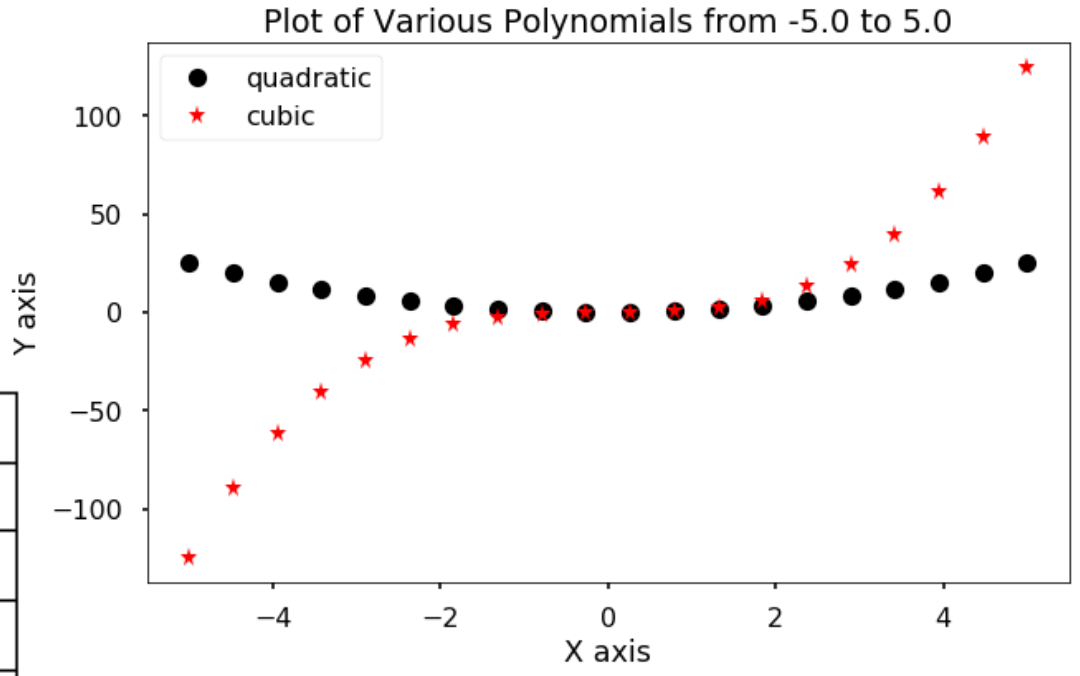
```
print(plt.style.available)
plt.style.use('seaborn-v0_8')
```

Legend

- Add label in the plot function. Legend function specifies location of the legend (loc = 0 – 10)

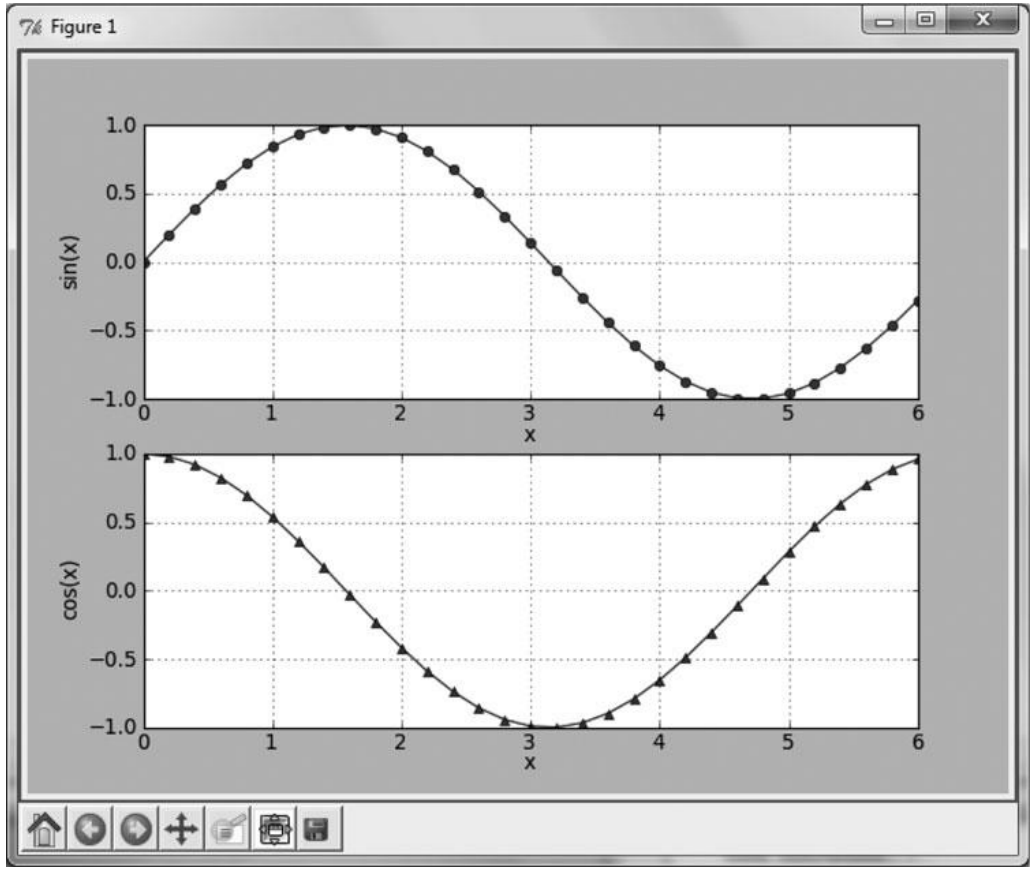
```
plt.figure(figsize = (10,6))
x = np.linspace(-5,5,20)
plt.plot(x, x**2, 'ko', label = 'quadratic')
plt.plot(x, x**3, 'r*', label = 'cubic')
plt.title(f'Plot of Various Polynomials from
{x[0]} to {x[-1]}')
plt.xlabel('X axis')
plt.ylabel('Y axis')
plt.legend(loc = 2)
plt.show()
```

0	"Best" location
1	Upper right
2	Upper left
3	Lower left
4	Lower right

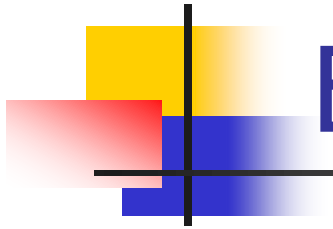


subplot

- subplot(rows, cols, plot number) creates a subplot window, dividing the figure into a row X col grid
- subplot(2, 1, 1): divide the figure into a 2 X 1 grid, then plot on the first figure.



Existing plotting functions



- scatter(x,y): same as plot(x,y,'ro')
- bar(x,y): plot bars centered at x with height y
- errorbar: plot x vs. y with error bars
- polar: plot in polar coordinates
- loglog, semilogx, and semiology plot the data in x and y with the x and y axis on a log scale, the x axis on a log scale and the y axis on a log scale.
- hist: histogram of a dataset
- boxplot: statistical summary of a dataset
- pie: Pie chart

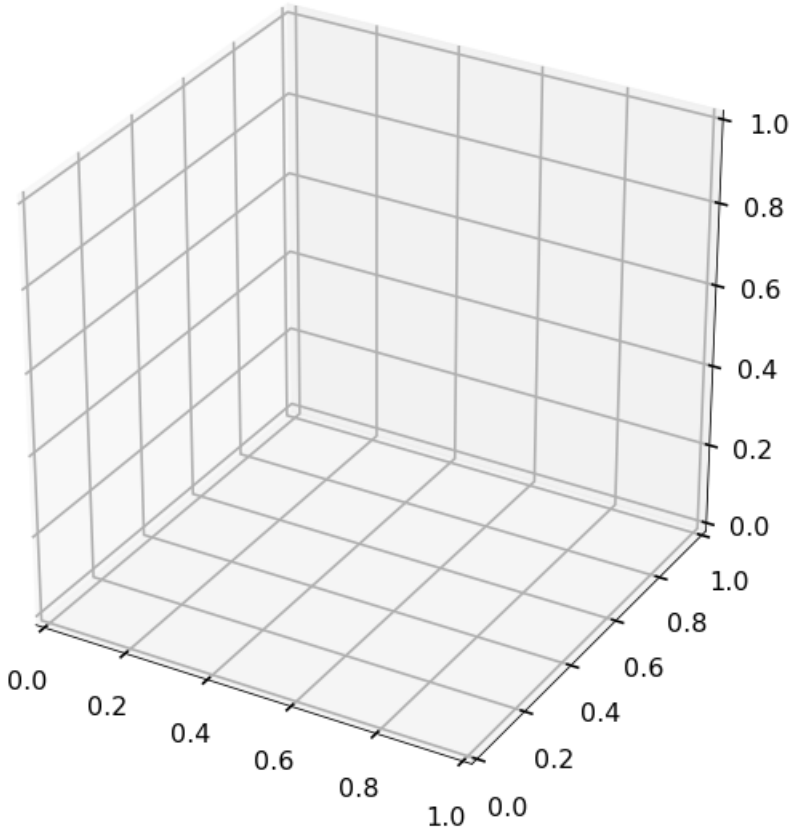
Misc

- plt.tight_layout(): no overlapping between subfigures.
- plt.savefig('image.pdf', format ='pdf')

3D plot

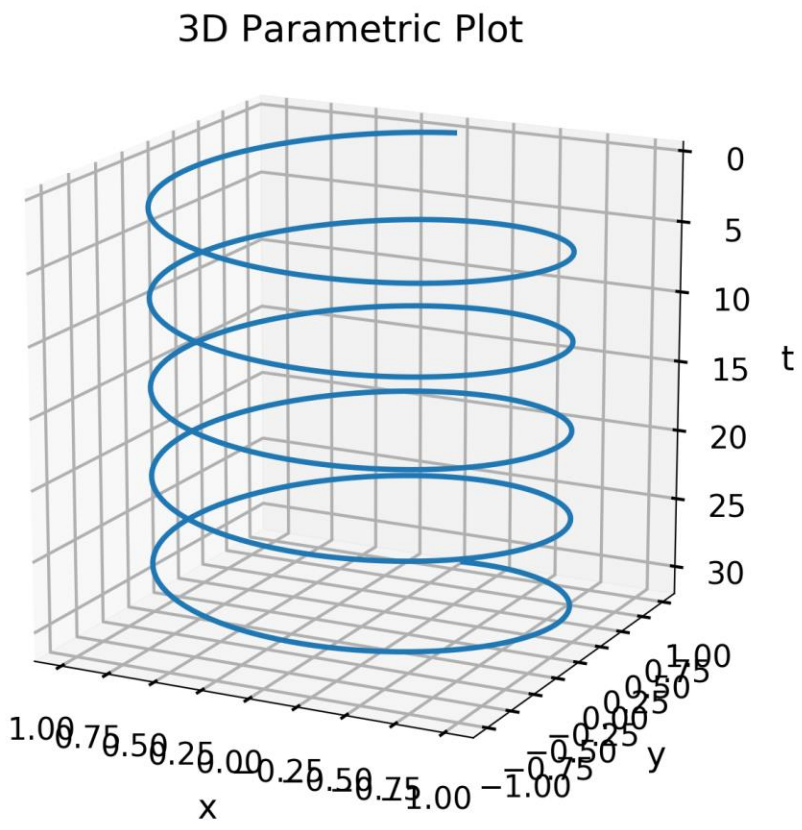
```
from mpl_toolkits import mplot3d
import matplotlib.pyplot as plt


fig = plt.figure(figsize = (10,10))
# create 3D axis object
ax = plt.axes(projection='3d')
plt.show()
```



Example

```
fig = plt.figure(figsize = (8,8))
ax = plt.axes(projection='3d')
ax.grid()
t = np.arange(0, 10*np.pi, np.pi/50)
x = np.sin(t)
y = np.cos(t)
ax.plot3D(x, y, t)
ax.set_title('3D Parametric Plot')
# Set axes label
ax.set_xlabel('x', labelpad=20)
ax.set_ylabel('y', labelpad=20)
ax.set_zlabel('t', labelpad=20)
plt.show()
```



Surface plot

- Create a mesh given lists/arrays of x and y: listing all possible combinations of x, y
  - X, Y = np.meshgrid(x, y)

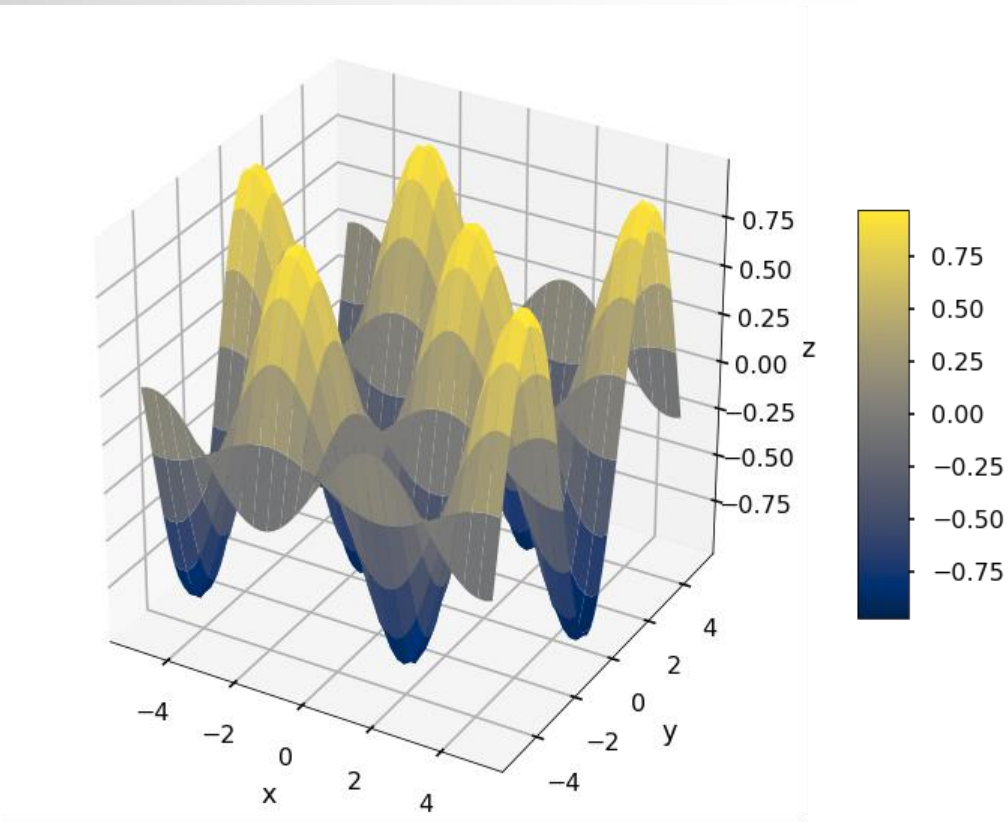
```
x = [1, 2, 3, 4]
y = [3, 4, 5]
X, Y = np.meshgrid(x, y)
print(X)
print(Y)
```

```
[[1 2 3 4]        [[3 3 3 3]
 [1 2 3 4]         [4 4 4 4]
 [1 2 3 4]]        [5 5 5 5]]
```

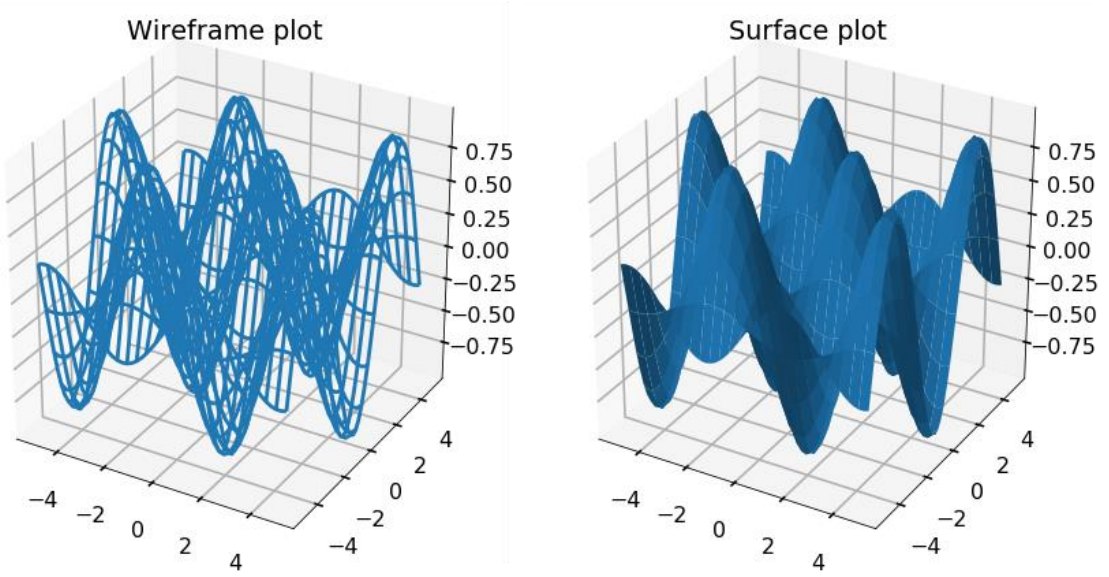
- Once the meshgrid is created, evaluate the function f at each grid point
  - Z = f(X,Y)
  - Plot using ax.plot_surface

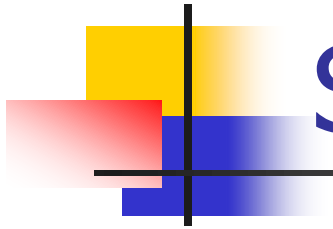
Example: f(x,y) = sin(x)cos(y)

```
fig = plt.figure(figsize = (12,10))
ax = plt.axes(projection='3d')
x = np.arange(-5, 5.1, 0.2)
y = np.arange(-5, 5.1, 0.2)
X, Y = np.meshgrid(x, y)
Z = np.sin(X)*np.cos(Y)
surf = ax.plot_surface(X, Y, Z, cmap = plt.cm.cividis)
# Set axes label
ax.set_xlabel('x', labelpad=20)
ax.set_ylabel('y', labelpad=20)
ax.set_zlabel('z', labelpad=20)
fig.colorbar(surf, shrink=0.5, aspect=8)
plt.show()
```



```
fig = plt.figure(figsize=(12,6))
ax = fig.add_subplot(1, 2, 1, projection='3d')
ax.plot_wireframe(X,Y,Z)
ax.set_title('Wireframe plot’)
ax = fig.add_subplot(1, 2, 2, projection='3d')
ax.plot_surface(X,Y,Z)
ax.set_title('Surface plot’)
plt.tight_layout()
plt.show()
```





Some illustrations

- Refer to uploaded python files and Pyplot Plotting – Basics.pdf