

PERFUM Python Camp 2023

Chapter 4 : Visualisation with Matplotlib

- Matplotlib is a library that can create a comprehensive static, animated, and interactive visualization in Python.
- It creates great vectorized image with high quality resolution that is suitable for publication.
- User can customized the image or graph and it support many files format.
- For more example and reading, user can refer to this page : https://matplotlib.org/stable/plot_types/index

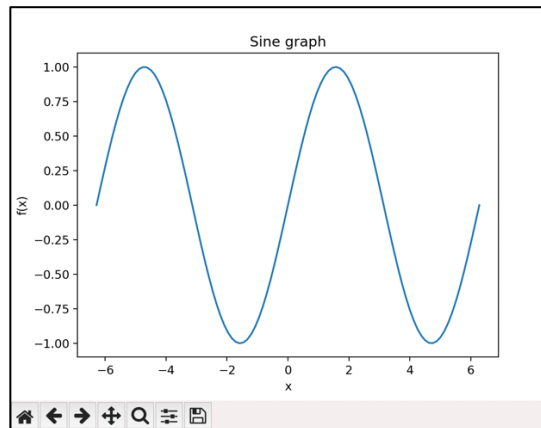
4.1 Static plot

- In this chapter, you will learn how to do simple plot by using numpy library and manipulating mathematical equation to produce beautiful plot.
- Example:

Given a function, $f(x) = \sin(x)$ plot the function within the range of $x \in [-2\pi, 2\pi]$.

```
1 #python code to plot sine plot
2
3 #import libraries
4 import matplotlib.pyplot as mpl
5 import numpy as np
6
7 #define function
8 def f(x):
9     return np.sin(x)
10
11 #choose range to plot our result, 100 point is chosen to have smooth plot
12 x = np.linspace(-2*np.pi,2*np.pi,100)
13
14 #title of the graph
15 mpl.title('Sine graph')
16
17 #x axis label
18 mpl.xlabel('x')
19
20 #y axis label
21 mpl.ylabel('f(x)')
22
23 #plot x and y
24 mpl.plot(x,f(x))
25
26 #show the graph after executing python
27 mpl.show()
```

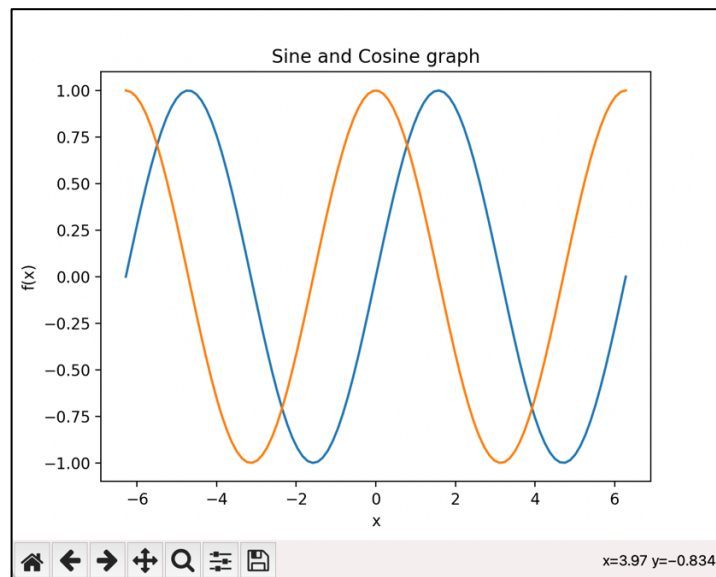
Output:



- We can combine multiple function to plot multiple figures in one graph.
- For example, we want to add $\cos(x)$ in the $\sin(x)$ graph

```
1 #python code to plot sine plot|
2 #import libraries
3 import matplotlib.pyplot as plt
4 import numpy as np
5
6 #define function for sin
7 def f(x):
8     return np.sin(x)
9
10 #define function for cos
11 def g(x):
12     return np.cos(x)
13
14 #choose range to plot our result, 100 point is chosen to have smooth plot
15 x = np.linspace(-2*np.pi,2*np.pi,100)
16
17 #title of the graph
18 plt.title('Sine and Cosine graph')
19
20 #x axis label
21 plt.xlabel('x')
22
23 #y axis label
24 plt.ylabel('f(x)')
25
26 #plot x and y
27 plt.plot(x,f(x))
28 plt.plot(x,g(x))
29
30 #show the graph after executing python
31 plt.show()
```

Output:



- You can also add label and save the files by adding several commands in the code.
- We shall learn more customized graph during the camp.

4.2 Animated plot

- We can transform our static plot into animated plot and save the plot either as gif image or movie.
- To create an animation using matplotlib, we need to import animation module from matplotlib and called specific function and writer to save our animation file.
- The basis of matplotlib animation as stated below:



- Example transforming static plot into animated plot shown below:

```
1 #python code to plot sine plot
2
3 #import libraries
4 import matplotlib.pyplot as mpl
5 from matplotlib.animation import FuncAnimation
6 import numpy as np
7
8 #create the frame for animation
9 fig, ax = mpl.subplots()
10 xdata,ydata=[],[]
11 line, = ax.plot([],[], 'bo')
12
13 #x axis label
14 mpl.xlabel('x')
15
16 #y axis label
17 mpl.ylabel('f(x)')
18
19 def init():
20     ax.set_xlim(-2*np.pi, 2*np.pi)
21     ax.set_ylim(-1,1)
22     return line,
23
24 #update the frame
25 def update(frame):
26     xdata.append(frame)
27     ydata.append(np.sin(frame))
28     line.set_data(xdata, ydata)
29     return line,
30
31 ani = FuncAnimation(fig, update, frames=np.linspace(-2*np.pi, 2*np.pi, 100),
32                     init_func=init, blit=True)
33
34 ani.save('sine.gif', writer="pillow", fps=50)
35 #show the graph after executing python
36 mpl.show()
```

You can view the gif file for this code in PPCC github page.