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

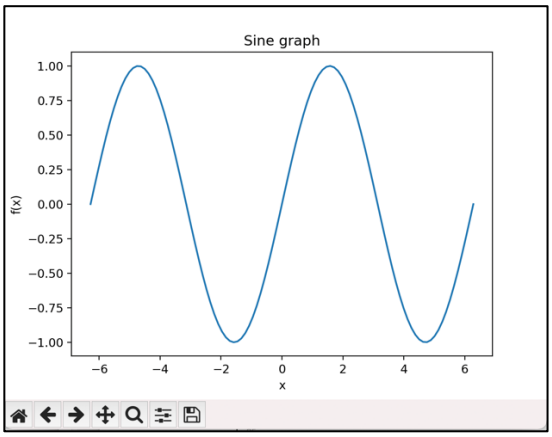
4.1 Static plot

- 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 $x \in [-2\pi, 2\pi]$.

```python
#python code to plot sine plot

#import libraries
import matplotlib.pyplot as mpl
import numpy as np

#define function
def f(x):
   return np.sin(x)

#choose range to plot our result, 100 point is chosen to have smooth plot
x = np.linspace(-2*np.pi,2*np.pi,100)

#title of the graph
mpl.title('Sine graph')

#x axis label
mpl.xlabel('x')

#y axis label
mpl.ylabel('f(x)')

#plot x and y
mpl.plot(x,f(x))

#show the graph after executing python
mpl.show()
```

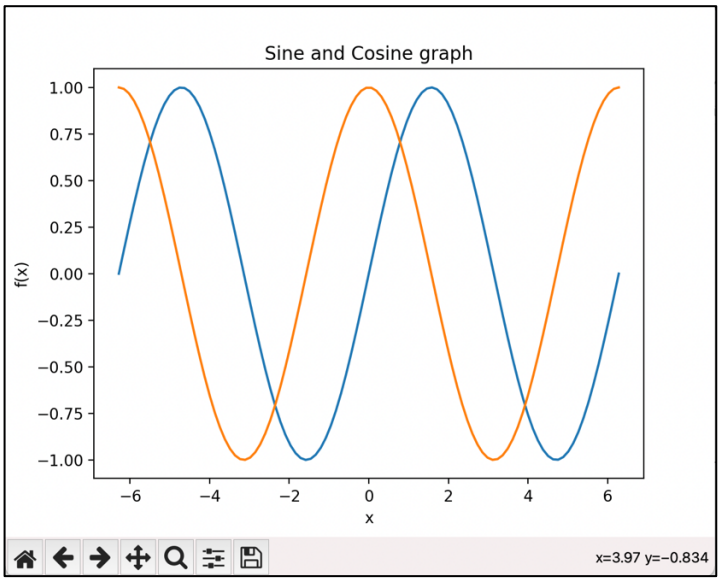
  Output:



- 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

```python
#python code to plot sine plot
#import libraries
import matplotlib.pyplot as mpl
import numpy as np

#define function for sin
def f(x):
   return np.sin(x)

#define function for cos
def g(x):
   return np.cos(x)

#choose range to plot our result, 100 point is chosen to have smooth plot
x = np.linspace(-2*np.pi,2*np.pi,100)

#title of the graph
mpl.title('Sine and Cosine graph')

#x axis label
mpl.xlabel('x')

#y axis label
mpl.ylabel('f(x)')

#plot x and y
mpl.plot(x,f(x))
mpl.plot(x,g(x))

#show the graph after executing python
mpl.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:

```python
#python code to plot sine plot

#import libraries
import matplotlib.pyplot as mpl
from matplotlib.animation import FuncAnimation
import numpy as np

#create the frame for animation
fig, ax = mpl.subplots()
xdata,ydata=[],[]
line, = ax.plot([],[],'bo')

#x axis label
mpl.xlabel('x')

#y axis label
mpl.ylabel('f(x)')

def init():
   ax.set_xlim(-2*np.pi,2*np.pi)
   ax.set_ylim(-1,1)
   return line,

#update the frame
def update(frame):
    xdata.append(frame)
    ydata.append(np.sin(frame))
    line.set_data(xdata, ydata)
    return line,

ani = FuncAnimation(fig, update, frames=np.linspace(-2*np.pi, 2*np.pi, 100),
                    init_func=init, blit=True)

ani.save('sine.gif', writer="pillow", fps=50)
#show the graph after executing python
mpl.show()
```

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