LAB # 8

**SCIPY AND MATPLOTLIB**

**OBJECTIVE :** Scientific computing and technical computing with plotting using SciPy and matplotlib library in python

**Task 1:**

**Code:**

import matplotlib.pyplot as Plt

import numpy as np

x = np.arange(10)

Plt.plot(x,x,)

Plt.plot(x, 2 \* x)

Plt.plot(x, 3 \* x)

Plt.plot(x, 4 \* x)

Plt.gca().get\_lines()[0].set\_color("orange")

Plt.gca().get\_lines()[1].set\_color("red")

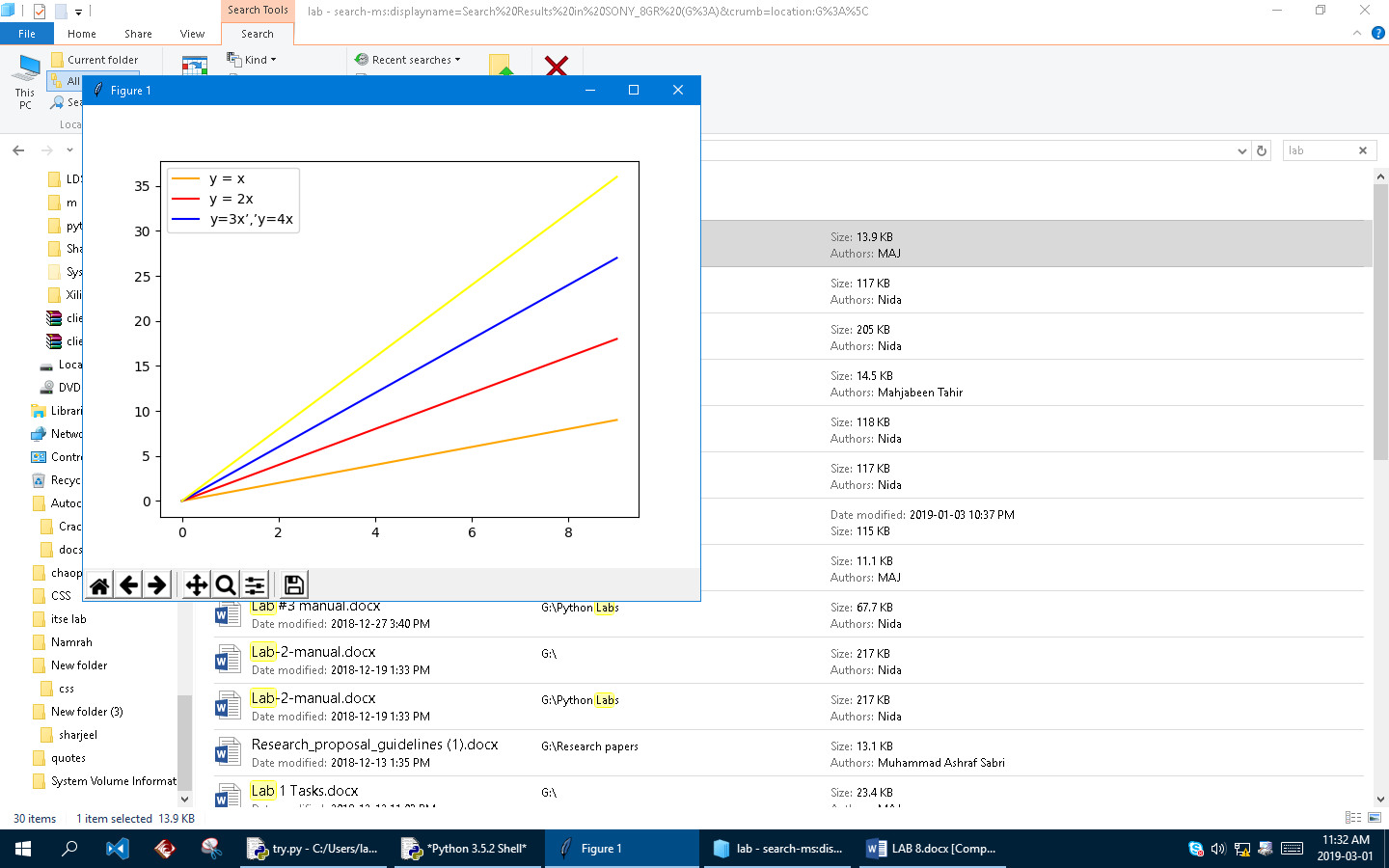
Plt.gca().get\_lines()[2].set\_color("blue")

Plt.gca().get\_lines()[3].set\_color("yellow")

Plt.gca().legend(('y = x', 'y = 2x', 'y=3x’,’y=4x'))

Plt.show()

**Output:**



**TASK 2**

**CODE:**

import matplotlib.pyplot as plt

import numpy as np

x = np.linspace(-10, 10, 1000)

y = x\*\*2 + 2\*x + 2

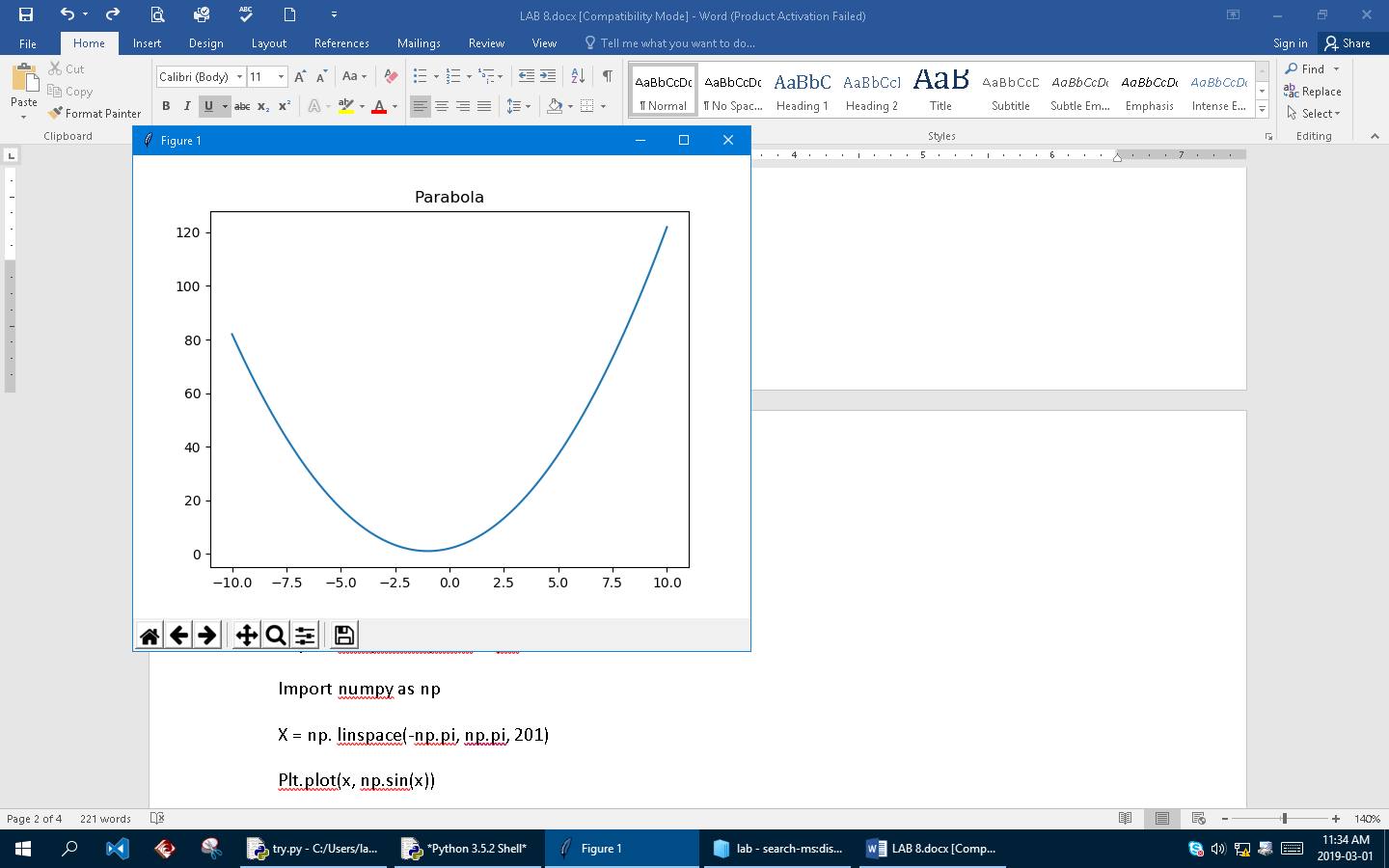
fig, ax = plt.subplots()

ax.plot(x,y)

plt.title('Parabola')

plt.show()

OUTPUT



**TASK 3**

**CODE**

import matplotlib.pylab as Plt

import numpy as np

x = np. linspace(-np.pi, np.pi, 201)

Plt.plot(x, np.sin(x))

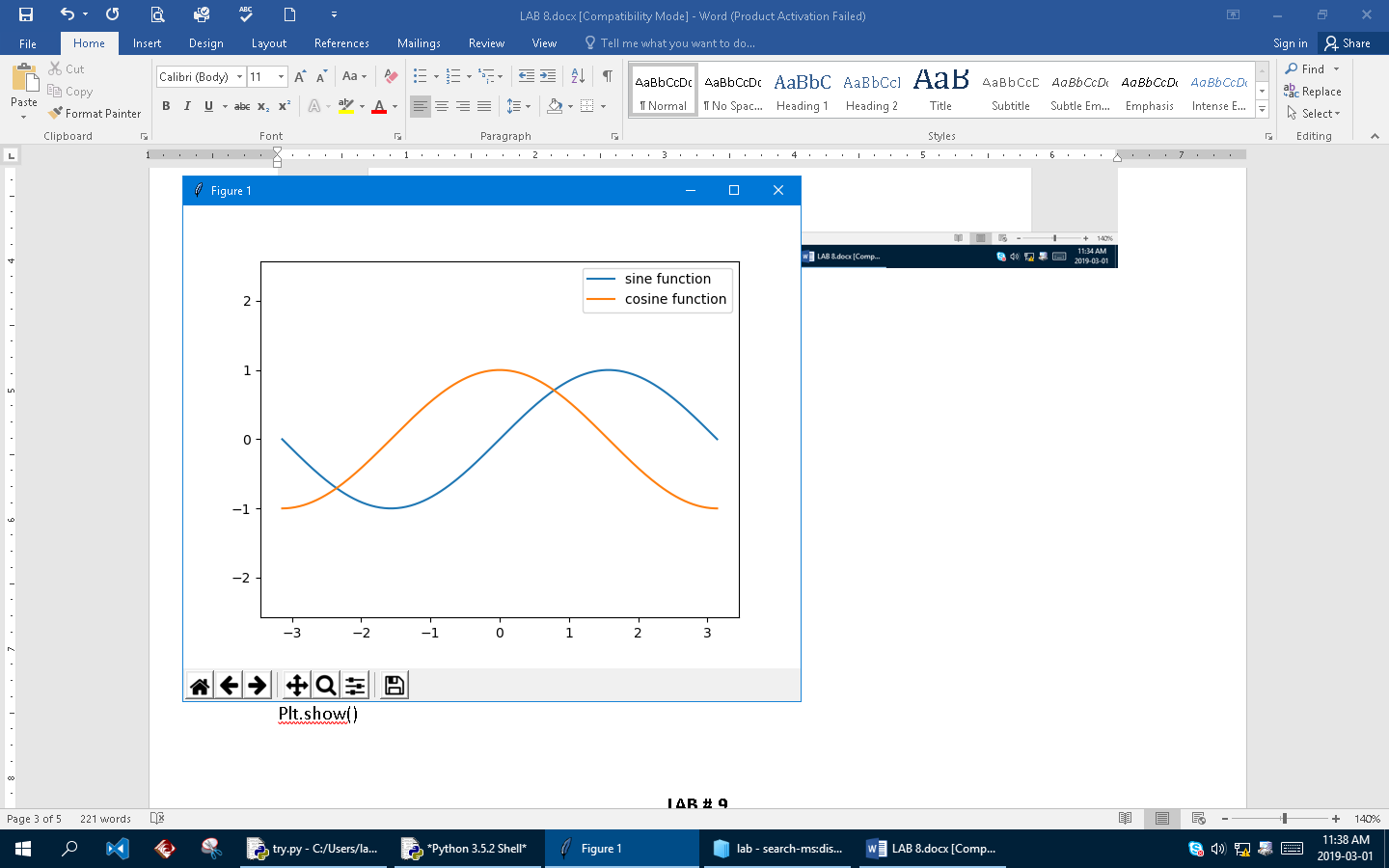
Plt.plot(x, np.cos(x))

Plt.axis('equal')

Plt.gca().legend(('sine function','cosine function'))

Plt.show()

**OUTPUT**



**LAB # 9**

**OBJECTIVE**

Familiarization with fuzzy logic system using Scikit fuzzy

**TASK 1**

1. Implementation of Union and intersection in fuzzy using Built-in function in skfuzzy

**Code**

import numpy as np

import skfuzzy as fuzz

uni\_x = np.array([1,2,4,5,6])

uni\_y = np.array([1,2,4,5,6])

mfx = np.array([0.3,0.1,0.7,0.9],float)

mfy = np.array([0.6,0.2,1,0.3],float)

print(fuzz.fuzzy\_or(uni\_x,mfx,uni\_y,mfy))

print(fuzz.fuzzy\_and(uni\_x,mfx,uni\_y,mfy))

**OUTPUT**

(array([1, 2, 4, 5, 6]), array([0.6, 0.2, 1. , 0.9]))

(array([1, 2, 4, 5, 6]), array([0.3, 0.1, 0.7, 0.3]))

1. Implementation of cartesian addition of fuzzy membership vector using algrebraic method using cartadd(x,y)

**CODE**

import numpy as np

import skfuzzy as fuzz

uni\_x = np.array([1,2,4,5,6])

uni\_y = np.array([1,2,4,5,6])

mfx = np.array([0.3,0.1,0.7,0.9],float)

mfy = np.array([0.6,0.2,1,0.3],float)

print(fuzz.cartadd(mfx,mfy))

**Output:**

[[0.9 0.5 1.3 0.6]

[0.7 0.3 1.1 0.4]

[1.3 0.9 1.7 1. ]

[1.5 1.1 1.9 1.2]]

1. Implementation of cartesian product of two fuzzy membership vector using algrebraic method using cartprod(x,y)

**CODE**

import numpy as np

import skfuzzy as fuzz

uni\_x = np.array([1,2,4,5,6])

uni\_y = np.array([1,2,4,5,6])

mfx = np.array([0.3,0.1,0.7,0.9],float)

mfy = np.array([0.6,0.2,1,0.3],float)

print(fuzz.cartprod(mfx,mfy))

**Output:**

[[0.3 0.2 0.3 0.3]

[0.1 0.1 0.1 0.1]

[0.6 0.2 0.7 0.3]

[0.6 0.2 0.9 0.3]]