**Code**

import autograd.numpy as np

from autograd import grad

##### PART A #####

def func(x): # y = 2 + e^(-bx)sin(x)

    return 2 + np.exp(-b \* x) \* np.sin(x)

def slope(x1, x2, y1, y2):

    return (y2 - y1) / (x2 - x1)

b=0.05

x = np.linspace(0, 30, num=100)

slope\_of\_loch = grad(func)

print("Slope at 0.1 for b=0.05:\n%s" % slope\_of\_loch(0.1))

print("Slope at 0.01 for b=0.05:\n%s\n" % slope\_of\_loch(0.01))

b=0.1

x = np.linspace(0, 30, num=100)

slope\_of\_loch = grad(func)

print("Slope at 0.1 for b=0.1:\n%s" % slope\_of\_loch(0.1))

print("Slope at 0.01 for b=0.1:\n%s\n" % slope\_of\_loch(0.01))

##### PART B #####

x = [0,1,2,3,4,5]

y = [10,9,6,4,6,10]

x1 = 1

x2 = 2

print('Slope between x=1 and x=2:\n%s' % slope(x1,x2,y[x1],y[x2]))

x4 = 4

x5 = 5

print('Slope between x=4 and x=5:\n%s' % slope(x1,x2,y[x4],y[x5]))

**Results**

Slope at 0.1 for b=0.05:

0.9850747865301321

Slope at 0.01 for b=0.05:

0.9989504086560138

Slope at 0.1 for b=0.1:

0.9752197026578588

Slope at 0.01 for b=0.1:

0.9979515663747239

Slope between x=1 and x=2:

-3.0

Slope between x=4 and x=5:

4.0