

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

#Question 2 Finite difference method

x = [0, 0.375, .75, 1.125, 1.5, 1.875, 2.25, 2.625, 3]
y = [1, -.2571, -0.9484, -1.9689, -3.2262, -4.6414, -6.1503, -7.7051, -9.275]

#numerical derivative definitions
def diff_forwards( x, y, i):
    return (y[i + 1] - y[i])/(x[i + 1] - x[i])

def diff_backwards( x, y, i):
    return (y[i] - y[i-1])/(x[i] - x[i-1])

def diff_center(x, y, i):
    return (y[i+1] - y[i-1])/(x[i+1] - x[i-1])

#second order derivatives
def diff2_forward(x, y, i):
    return (y[i+2] - 2 * y[i + 1] + y[i])/((x[i + 1] - x[i])**2)

def diff2_backward(x, y, i):
    return (y[i] - 2 * y[i - 1] + y[i - 2])/((x[i] - x[i-1])**2)

def diff2_center(x, y, i):
    return (y[i+1] - 2 * y[i] + y[i-1])/((x[i + 1] - x[i])**2)

#forward diff
slopes = [diff_forwards(x,y,0)]
#central diff
for i in range(1 , len(x) - 1):
    slopes.append(diff_center(x,y,i))
#backward diff
slopes.append(diff_backwards(x,y,-1))

#finding derivative of slope:
#forward difference for first value
dTheta = [diff_forwards(x,slopes,0)]
for i in range(1 , len(slopes) - 1):
    dTheta.append( diff_center(x, slopes, i))
#backwards difference for last value
dTheta.append( diff_backwards(x, slopes, -1))

#finding second y derivative:
dY2 = [diff2_forward(x,y,0)]
for i in range(1, len(y) - 1):
    dY2.append( diff2_center(x,y,i))
dY2.append(diff2_backward(x,y,-1))

#Finding bending moment
E = 200 #[GPa]
I = 0.0003 #[m^4]
My = np.multiply( dY2, E * I)
Mt = np.multiply( dTheta, E * I)

data = {
    ("X values") : x,
    ("Y values") : y,
    ("Slope") : slopes,
    ("dTheta") : dTheta,
    ("dY2") : dY2
}

df = pd.DataFrame(data)
df.index = range(1, len(x)+1)

print(df)
'''
plt.figure()
plt.plot(x, y)
plt.title("Profile of rod")
'''
plt.figure()

```

```
plt.plot(x, Mt, label='M theta')
plt.plot(x, My, label='M Y')
plt.title("Second Derivatives Comparison")
plt.xlabel("X [m]")
plt.ylabel("Moment")
plt.legend()
plt.show()
```

```
#The second derivative of the y values matches the expected curve for more values than the derivative of the slope. Th
# derivative of the y values is a more accurate method
```