Лабораторна робота №7

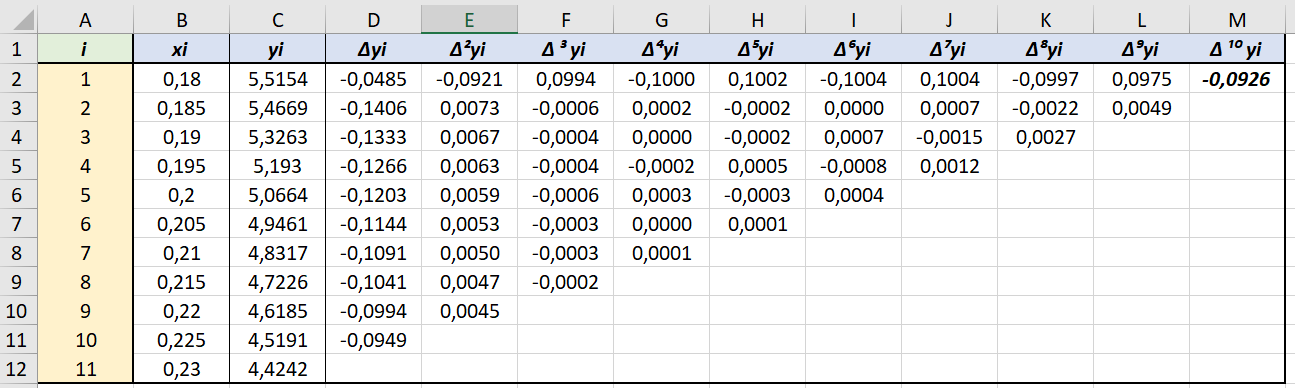
*Інтерполяційний багаточлен Ньютона*

Виконала Гальчинська Софія, студентка ФІТ 2-8. Варіант 4.

[*https://github.com/GalchynskaSofiia/Numerical\_Methods\_Of\_Programming*](https://github.com/GalchynskaSofiia/Numerical_Methods_Of_Programming)

**Завдання 1:** наближено відбудувати функцію *y = f(x)*, що задана таблицею, у довільній точці х за допомогою інтерполяційних багаточленів Ньютона. За наявним набором значень побудувати графік інтерполяційної функції.

*Складемо для заданої функції таблицю кінцевих різниць:*

**

**Інтерполяційні формули Ньютона:**

import numpy as np

from math import factorial

import matplotlib.pyplot as plt

x = np.array ([0.180, 0.185, 0.190, 0.195, 0.200, 0.205, 0.210, 0.215, 0.220, 0.225, 0.230])

y = np.array([5.5154, 5.4669, 5.3263, 5.1930, 5.0664, 4.9461, 4.8317, 4.7226, 4.6185, 4.5191, 4.4242])

# перша інтерполяційна формула

def first\_interpolation(x, y, x0):

n = len(x)

f = np.zeros((n, n))

f[:, 0] = y

for j in range(1, n):

for i in range(n - j):

f[i, j] = (f[i+1, j-1] - f[i, j-1]) / (x[i+j] - x[i])

ans = 0

for j in range(n):

prod = f[0, j]

for i in range(j):

prod \*= (x0 - x[i])

ans += prod

return ans

# друга інтерполяційна формула

def second\_interpolation(x, y, x0):

n = len(x)

f = np.zeros((n, n))

f[:, 0] = y

for j in range(1, n):

for i in range(n - j):

f[i, j] = (f[i+1, j-1] - f[i, j-1]) / (x[i+j] - x[i])

ans = f[0, 0]

for j in range(1, n):

prod = f[0, j]

for i in range(j):

prod \*= (x0 - x[i])

ans += prod

return ans

x1 = 0.184

x2 = 0.189

x3 = 0.193

x4 = 0.221

x5 = 0.227

x6 = 0.216

y1 = first\_interpolation(x, y, x1)

y2 = first\_interpolation(x, y, x2)

y3 = first\_interpolation(x, y, x3)

y4 = second\_interpolation(x, y, x4)

y5 = second\_interpolation(x, y, x5)

y6 = second\_interpolation(x, y, x6)

print(f"f({x1}) = {y1}")

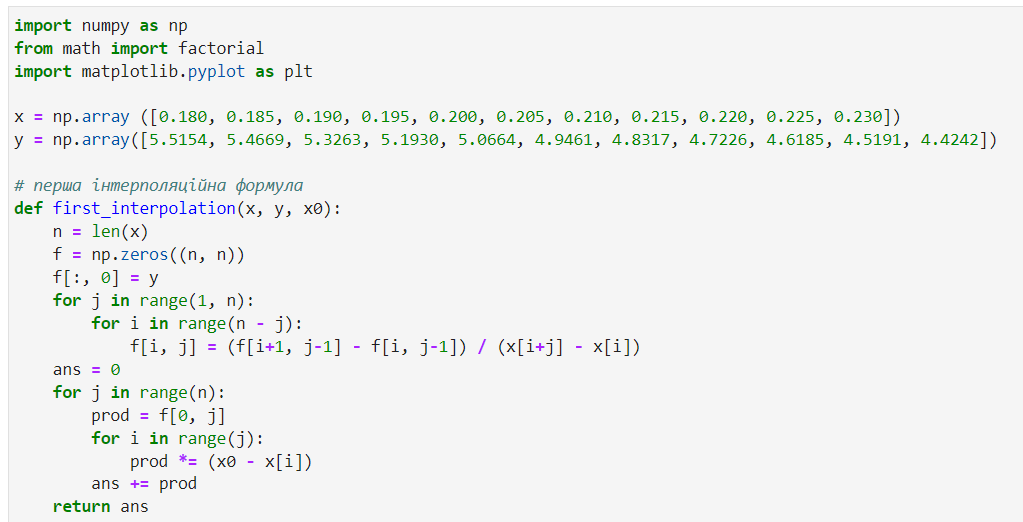
print(f"f({x2}) = {y2}")

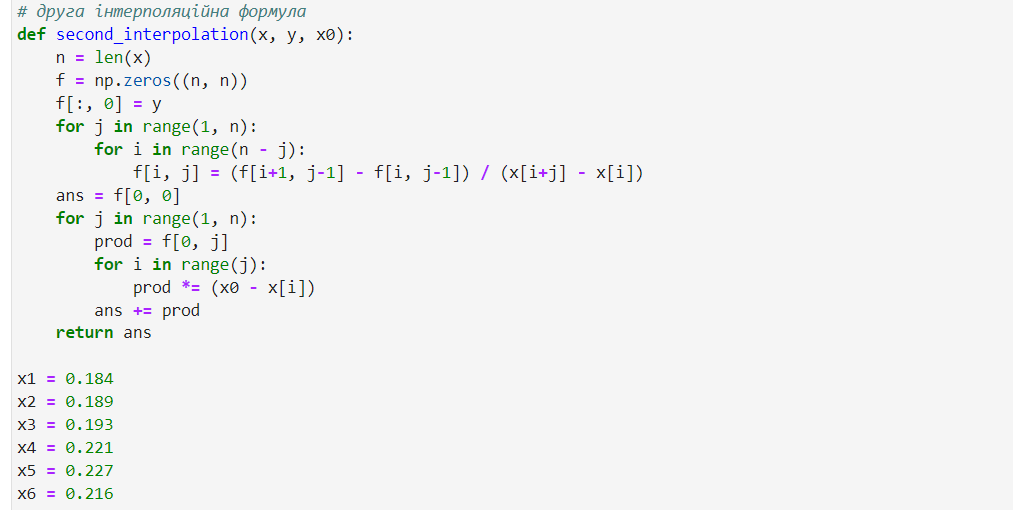
print(f"f({x3}) = {y3}")

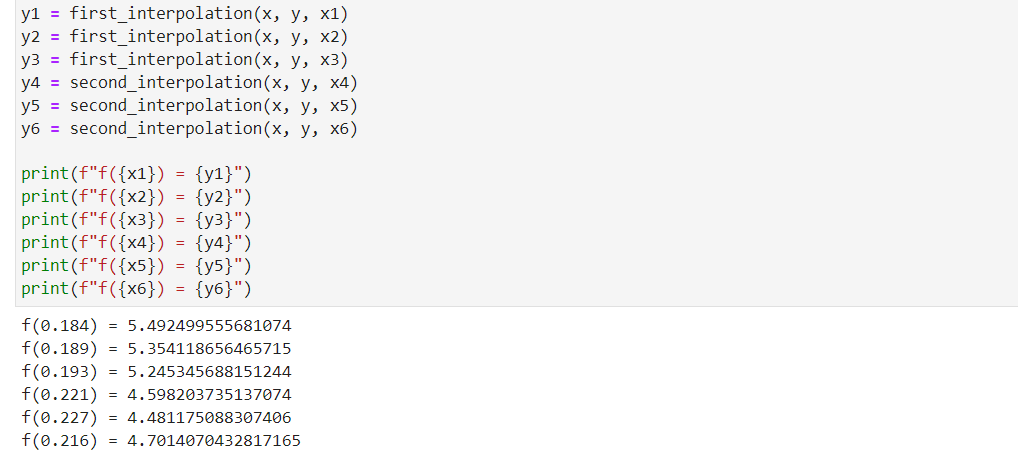
print(f"f({x4}) = {y4}")

print(f"f({x5}) = {y5}")

print(f"f({x6}) = {y6}")







**Графік інтерполяційної функції:**

xx = np.linspace(np.min(x), np.max(x), 100)

yy = np.zeros\_like(xx)

for i in range(len(xx)):

yy[i] = second\_interpolation(x, y, xx[i])

plt.plot(x, y, 'o', label='Дані точки')

plt.plot(xx, yy, label="багаточлен Ньютона")

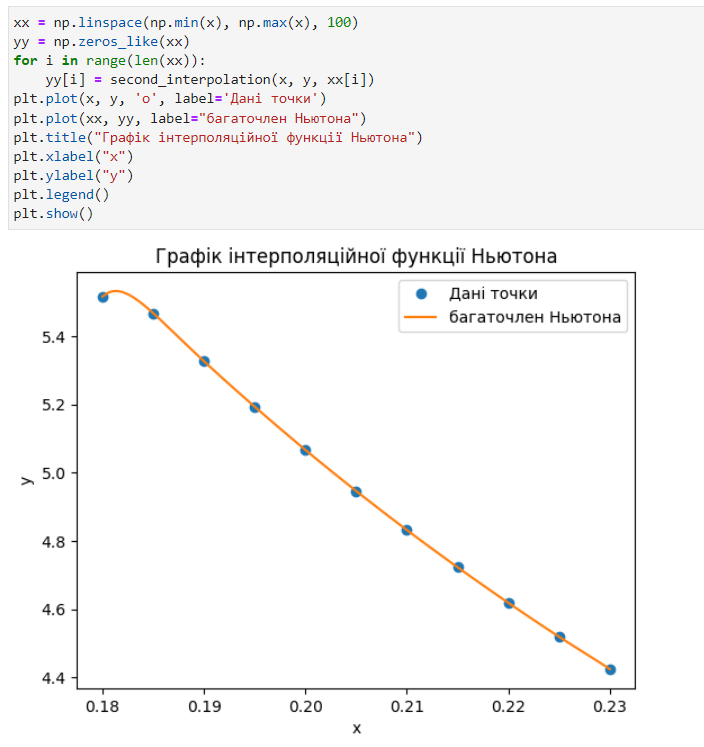
plt.title("Графік інтерполяційної функції Ньютона")

plt.xlabel("x")

plt.ylabel("y")

plt.legend()

plt.show()



**Висновки:** Отже, під час виконання практичної роботи ознайомились з інтерполяційним багаточленом Ньютона та навчились наближено будувати функцію, що задана таблицею. Побудова була проведена за допомогою програмування, зокрема використання Python для обчислення багаточлена Ньютона та його значень у заданих точках. Також проведено обчислення кінцевих різниць yi. Також було побудовано графік полінома Ньютона з відображенням на графіку точок (*x, f(x)*).