**Лабораторна робота №4**

**Виконав студент 211 групи  
Вайнагій Данило Вікторович**

**Варіант 2**

**Тема: Наближення функцій**

**Завдання:** **Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис**

**Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис**

Код:

import numpy as np

import matplotlib.pyplot as plt

def phi(x\_array: np.ndarray, k: int):

    return x\_array \*\* k

def get\_coef(x: np.ndarray, y: np.ndarray, n: int): #Отримати коефіцієнти інтерполяційного полінома n-го степеня

    matrix = np.zeros((n, n))

    b = np.zeros(n)

    for i in range(n):

        for j in range(n):

            matrix[i, j] = np.sum(phi(x, i) \* phi(x, j))

        b[i] = np.sum(y \* phi(x, i))

    return np.linalg.solve(matrix, b)[::-1]

def max\_error(f\_res: np.ndarray, phi\_res: np.ndarray): #Максимальна похибка

    return np.max(np.abs(f\_res - phi\_res))

def mean\_error(f\_res: np.ndarray, phi\_res: np.ndarray): #Середня похибка

    return np.mean(np.abs(f\_res - phi\_res))

def mean\_squared\_error(f\_res: np.ndarray, phi\_res: np.ndarray): #Середньоквадратична похибка

    return np.sqrt(np.mean((f\_res - phi\_res) \*\* 2))

X = np.arange(2006, 2016, dtype = np.float64)

Y = np.array([54512, 79955, 101659, 103396, 117343, 126236, 116454, 135065, 142079, 126308], dtype = np.float64)

new\_x = np.linspace(X[0], X[-1], 100)

coef = get\_coef(X, Y, 15)

f = np.poly1d(coef) #повертає поліноміальну функцію для заданих коефіціентів

print('Максимальна похибка: ', max\_error( Y, f(X) ) )

print('Середня похибка', mean\_error( Y, f(X) ) )

print('Середньоквадратична похибка', mean\_squared\_error( Y, f(X) ) )

plt.figure( figsize = (8, 6) )

plt.plot(new\_x, f(new\_x), color='black', label='Середньоквадратичне наближення')

plt.plot(X, Y, '--go',  color='Blue', label='Початкова функція')

plt.legend()

plt.show()

Результати:



