Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Інститут комп’ютерних наук та інформаційних технологій

Кафедра програмного забезпечення



**Звіт**

Про виконання лабораторної роботи №9

на тему:

**«Наближення функцій методом**

**найменших квадратів**»

**Лектор:**

доцент каф. ПЗ

Мельник Н.Б.

**Виконав:**

ст. гр. ПЗ-11

Солтисюк Д.А.

**Прийняла:**

доцент каф. ПЗ

Мельник Н.Б.

« \_\_ » \_\_\_\_\_\_\_\_ 2022 р.

∑ = \_\_\_\_\_ \_\_\_\_\_\_\_\_\_ .

Львів – 2022

**Тема:** Наближення функцій методом найменших квадратів

**Мета:** ознайомлення на практиці з методом найменших квадратів апроксимації (наближення) функцій.

**ТЕОРЕТИЧНІ ВІДОМОСТІ**

**Апроксимація функцій** – процес утворення поліному певного порядку на основі вхідних даних, поданих у виді таблиці певної кількості точок (*n*)*.* Утворений поліном описує криву, яка містить у собі множину вхідних даних, та дає змогу отримати її проміжні значення.

Апроксимація функцій є ефективним способом нівелювання похибок вхідних даних.

Одним із поширених методів апроксимації функцій є **метод найменших квадратів.** У якості вхідних даних отримуємо таблично задану функцію на деякому проміжку. Мета – сформувати поліном *m*-го степеня, при чому *m=0,n*.

Похибка для поліному представлена середнім квадратичним відхиленням:

Суть алгоритму – відшукати невідомі коефіцієнти поліному, при цьому, квадрат кожного відхилення повинен бути мінімальним.

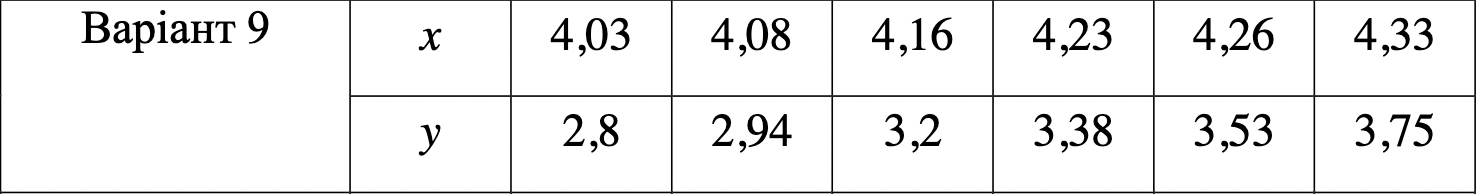
Далі – знаходимо похідні за змінними *(k = 0, m)* та прирівнюємо їх до нуля. Таким чином ми отримаємо ***нормальну систему*** методу найменших квадратів, яку можна подати як:

Отримавши СЛАР такого типу, її розв’язують довільним способом (точними або наближеними). Отримані розв’язки й будуть шуканими коефіцієнтами поліному.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

***Варіант 9***

***Завдання***

Методом найменших квадратів побудувати лінійний, квадратичний і кубічний апроксимаційні поліноми для таблично заданої функції.

ТЕКСТ ПРОГРАМИ

**main.py**

**from** common.least\_square\_approximation **import** LeastSquareApproximation

**from** common.main **import** Points

**if** \_\_name\_\_ == "\_\_main\_\_":

points: Points = ([

4.03,

4.08,

4.16,

4.23,

4.26,

4.33,

], [2.8, 2.94, 3.2, 3.38, 3.53, 3.75])

**for** degree **in** range(3):

LeastSquareApproximation(points, degree + 1).compile()

**least\_square\_approximation.py**

**import** numpy **as** np

**from** common.lu\_decomposition **import** LUDecompositionMethod

**from** common.main **import** ApproximationOrientedMethod, Points

**from** common.utils **import** print\_matrix

**class** LeastSquareApproximation(ApproximationOrientedMethod):

**def** \_\_init\_\_(self, points: Points, degree: int) -> **None**:

super().\_\_init\_\_("Least square approximation", points, degree)

**def** execute\_method(self):

x = self.points[0]

y = self.points[1]

m = self.degree + 1

n = len(x)

A = np.zeros((m, m))

B = np.zeros(m)

**for** k **in** range(m):

**for** i **in** range(n):

B[k] += y[i] \* (x[i]\*\*k)

**for** j **in** range(m):

A[k][j] += x[i]\*\*(j + k)

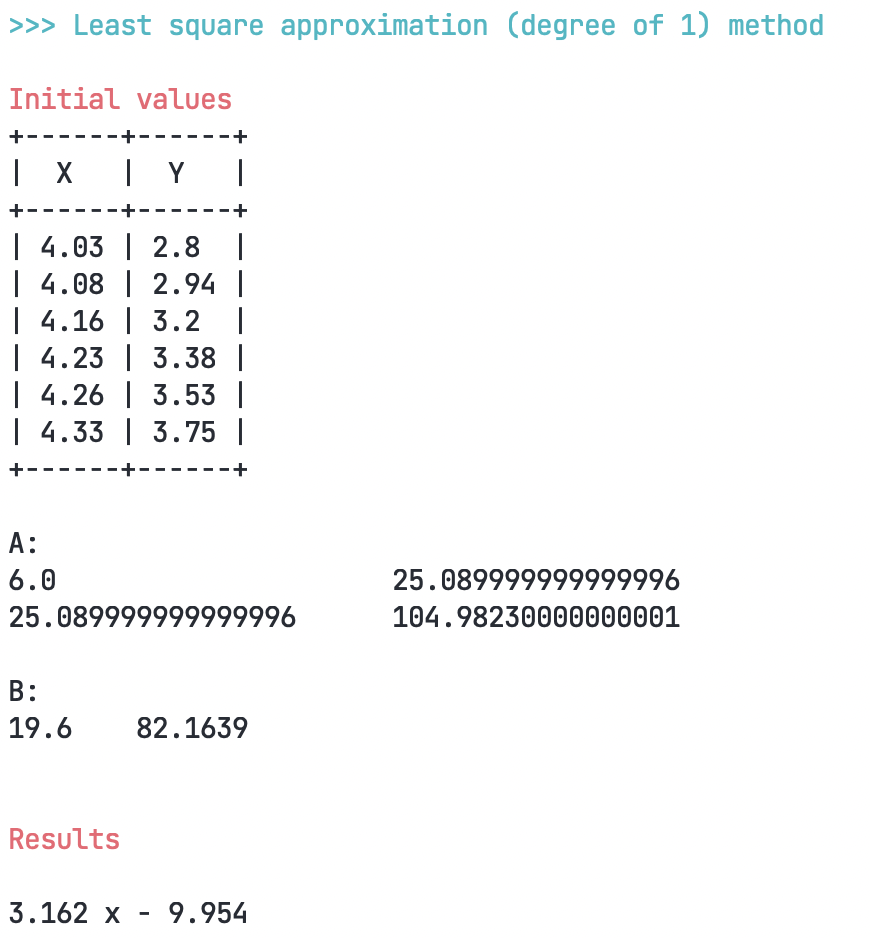
print\_matrix(A, "A")

print\_matrix([B], "B")

coeffs = LUDecompositionMethod(A, B).compile(silent=**True**)

**return** np.poly1d(coeffs[::-1])

РЕЗУЛЬТАТИ

Рис. 2 Результат   
(Лінійний поліном)

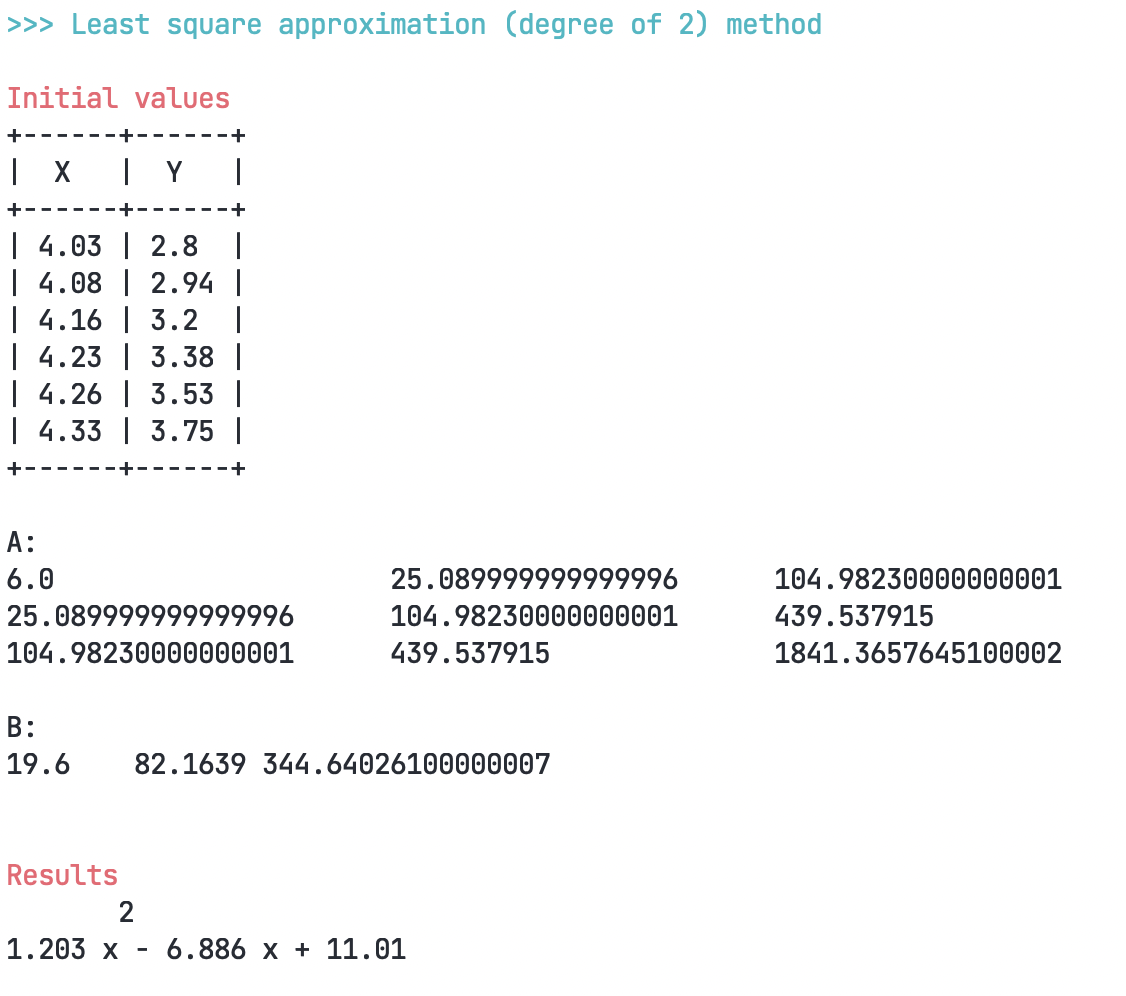
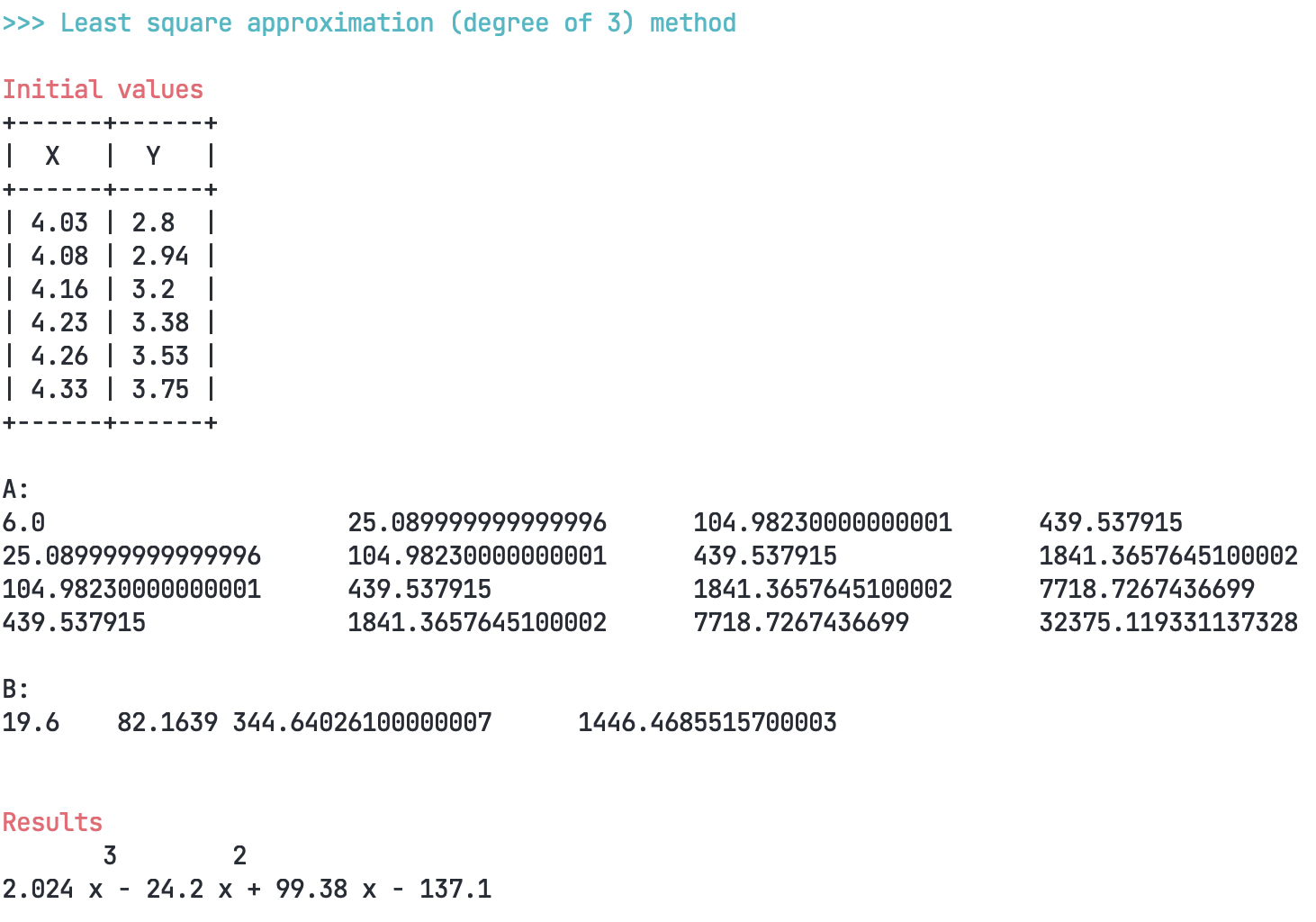


Рис. 3 Результат   
(Квадратичний поліном)

Рис. 4 Результат   
(Кубічний поліном)

ВИСНОВКИ

Виконавши лабораторну роботу №9, я отримав необхідні знання для про принципи апроксимації функцій шляхом утворення поліномів різних порядків за допомогою методу найменших квадратів. Використав отримані вміння на практиці, реалізувавши програмний алгоритм виконання цього методу.