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

Корольчук В.М.

**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 13***

***Завдання***

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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| x | 0.15 | 0.20 | 0.25 | 0.30 | 0.40 | 0.50 |
| y | 4.01 | 3.78 | 3.52 | 3.42 | 3.19 | 2.95 |

ТЕКСТ ПРОГРАМИ

#include <iostream>

#include <iomanip>

#include <string>

#include <cmath>

#define SIZE 6

using namespace std;

void printMatrix(double\*\* Matrix, int m, string title)

{

cout << setprecision(2) << fixed;

cout << "\n" << title << "\n";

for (int i = 0; i < m + 1; i++)

{

cout << "|" << " ";

for (int j = 0; j < m + 2; j++)

{

cout << setw(6) << Matrix[i][j] << " ";

}

cout << " " << "|" << "\n";

}

}

void printPolynome(double\* matrix, int size)

{

cout << "\n";

cout << "Formed polynom: ";

for (int i = 0; i < size - 1; i++)

{

cout << fabs(matrix[i]) << "x^";

cout << i;

if (i > 0 && i != size - 2)

{

cout << " + ";

}

else if (i != size - 2)

{

cout << " - ";

}

}

}

void DoGaussMethod(double\*\* AMatrix, double\* BColumn, int m)

{

m = m + 1;

double\* X = new double[m];

double\* B = new double[m];

double\*\* Matrix = new double\* [m];

if (AMatrix[0][0] <= 0.0000001)

{

cout << "\n[ INVALID SLAE ]" << endl;

return;

}

for (int i = 0; i < m; i++)

{

Matrix[i] = new double[m];

for (int j = 0; j < m; j++)

{

Matrix[i][j] = AMatrix[i][j];

}

B[i] = BColumn[i];

X[i] = { 0.0 };

}

for (int i = 0; i < m; i++)

{

double MaxValue = 0.0;

int MaxValueInd = -1;

for (int j = i; j < m; j++)

{

if (fabs(Matrix[j][i]) >= MaxValue)

{

MaxValue = fabs(Matrix[j][i]);

MaxValueInd = j;

}

}

swap(Matrix[i], Matrix[MaxValueInd]);

swap(B[i], B[MaxValueInd]);

for (int p = i; p < m; p++)

{

if (p != i)

{

double Mult = -(Matrix[p][i] / Matrix[i][i]);

for (int j = i; j < m; j++)

{

if (j == i)

{

Matrix[p][j] = 0.0;

}

else

{

Matrix[p][j] += (Matrix[i][j] \* Mult);

}

}

B[p] += (B[i] \* Mult);

}

}

}

for (int i = m - 1; i >= 0; i--)

{

for (int j = i + 1; j < m; j++)

{

X[i] += X[j] \* Matrix[i][j];

}

X[i] = (B[i] - X[i]) / Matrix[i][i];

}

printPolynome(X, m + 1);

}

void Method(double x[SIZE], double y[SIZE], int m)

{

double\*\* NormMatrix = new double\*[m + 1];

for (int i = 0; i < m + 1; i++)

{

NormMatrix[i] = new double[m + 2];

}

for (int i = 0; i < m + 1; i++)

{

for (int j = 0; j < m + 2; j++)

{

NormMatrix[i][j] = 0.0;

}

}

int n = SIZE;

for (int k = 0; k < m + 1; k++)

{

for (int j = 0; j < m + 1; j++)

{

for (int i = 0; i < n; i++)

{

NormMatrix[k][j] += pow(x[i], j + k);

}

}

for (int i = 0; i < n; i++)

{

NormMatrix[k][m + 1] += (y[i] \* pow(x[i], k));

}

}

double\*\* A = new double\* [m + 1];

double\* B = new double[m + 2];

for (int i = 0; i < m + 1; i++)

{

A[i] = new double[m + 2];

}

printMatrix(NormMatrix, m, "Normal matrix: \n");

for (int i = 0; i < m + 1; i++)

{

for (int j = 0; j < m + 1; j++)

{

A[i][j] = NormMatrix[i][j];

}

B[i] = NormMatrix[i][m + 1];

}

DoGaussMethod(A, B, m);

for (int i = 0; i < m + 1; i++)

{

delete NormMatrix[i];

}

delete[] NormMatrix;

for (int i = 0; i < m + 1; i++)

{

delete A[i];

}

delete[] A;

delete[] B;

}

int main()

{

double X[SIZE] = { 0.15, 0.20, 0.25, 0.30, 0.40, 0.50 };

double Y[SIZE] = { 4.01, 3.78, 3.52, 3.42, 3.19, 2.95 };

cout << "\n[ INPUT DATA ]\n" << endl;

cout << setprecision(2) << fixed;

char corner\_top\_left = (char)218,

corner\_top\_right = (char)191,

corner\_bottom\_left = (char)192,

corner\_bottom\_right = (char)217,

middle\_to\_right = (char)195,

middle\_to\_left = (char)180;

cout << corner\_top\_left << "----------------------------------------" << corner\_top\_right<< endl;

cout << "| X | ";

for (int i = 0; i < SIZE; i++)

{

cout << X[i] << " |";

}

cout << "\n" << middle\_to\_right << "----------------------------------------" << middle\_to\_left;

cout << "\n| Y | ";

for (int i = 0; i < SIZE; i++)

{

cout << Y[i] << " |";

}

cout << "\n" << corner\_bottom\_left << "----------------------------------------" << corner\_bottom\_right << endl;

cout << "\n[ LINEAR POLYNOM ]\n";

Method(X, Y, 1);

cout << "\n\n\n\n[ QUADRATIC POLYNOM ]\n";

Method(X, Y, 2);

cout << "\n\n\n\n[ CUBIC POLYNOM ]\n";

Method(X, Y, 3);

return getchar();

}

РЕЗУЛЬТАТИ

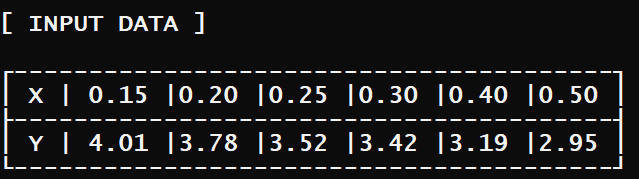


Рис. 1 – Виведення вхідних даних

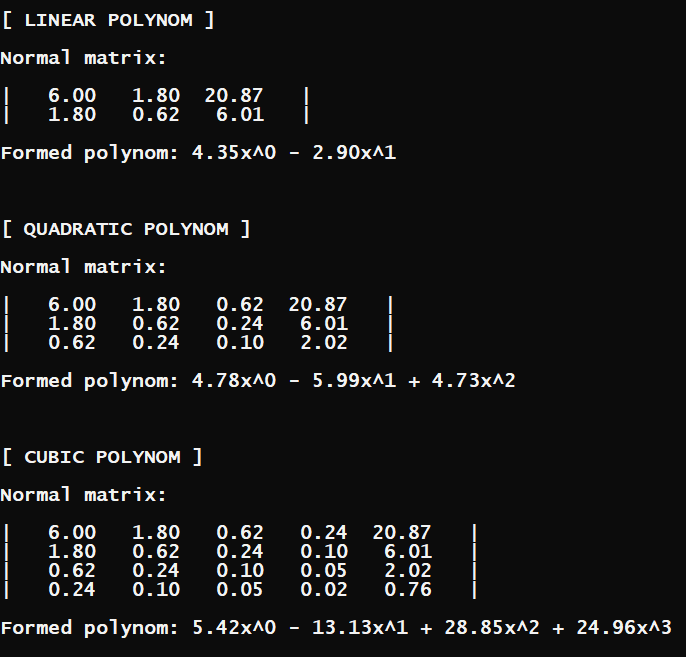


Рис. 2 Результат   
(Лінійний поліном)

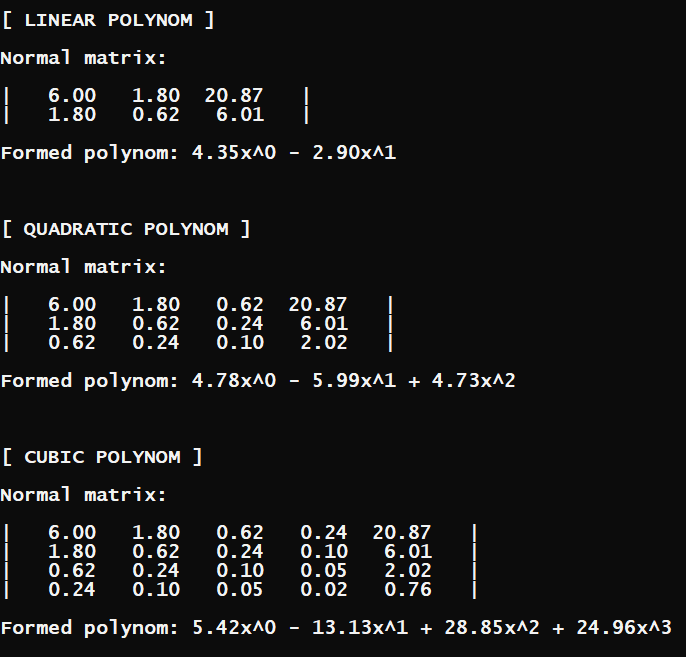


Рис. 3 Результат   
(Квадратичний поліном)

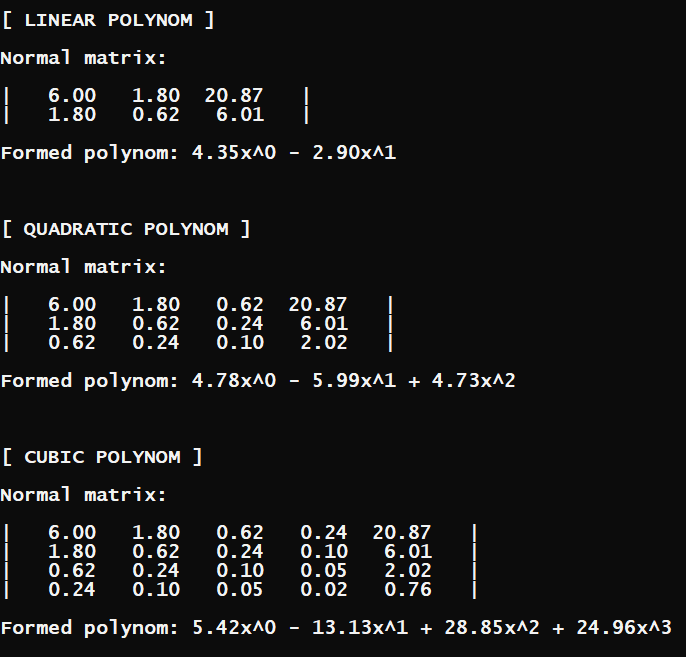


Рис. 4 Результат   
(Кубічний поліном)

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.