**Міністерство Освіти І НАУКИ України**

**Національний університет "Львівська політехніка"**

Інститут **ІКНІ**

Кафедра **ПЗ**



### ЗВІТ

До лабораторної роботи №9

**На тему:** *“Наближення функцій методом найменших квадратів”*

**З дисципліни:** *“Чисельні методи”*

**Лектор:**

доцент каф. ПЗ

Мельник Н.Б.

**Виконав:**

ст. гр. ПЗ-16

Шеремета А.І.

**Прийняла:**

асистент кафедри ПЗ

Бутрак І. О.

« \_\_\_\_ » \_\_\_\_\_\_\_\_ 2022 р.

∑= \_\_\_\_

Львів – 2022

**Тема роботи:** Наближення функцій методом найменших квадратів

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

**Індивідуальне завдання**

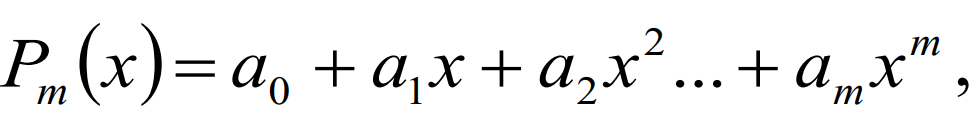
Методом найменших квадратів побудувати лінійний, квадратичний і кубічний апроксимаційні поліноми для таблично заданої функції.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Варіант 10 | x | 0,05 | 0,10 | 0,17 | 0,25 | 0,30 | 0,36 |
| y | 0,54 | 0,51 | 0,47 | 0,45 | 0,42 | 0,38 |

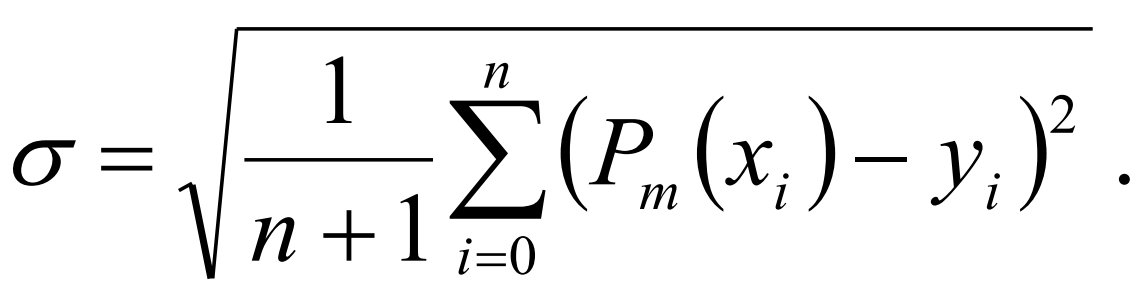
**Теоретичні відомості**

Розглянемо функцію *y* = *f* (*x*), задану таблицею своїх значень *yi* = *f* (*xi* ), *i* = 0, *n*.

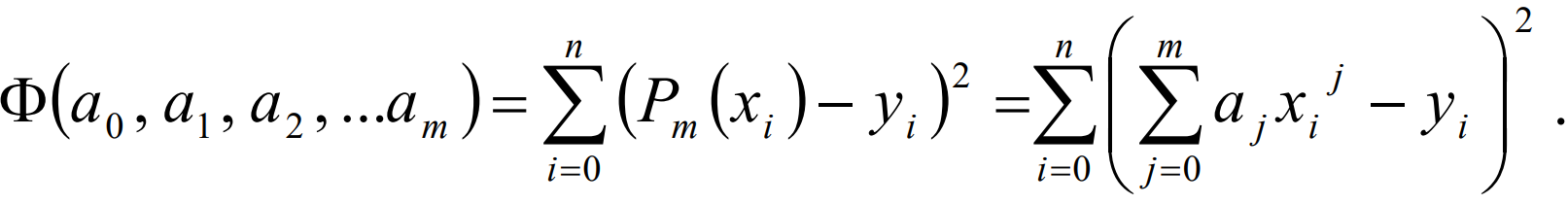
Потрібно знайти поліном фіксованого *m*-го степеня (*m* = 0, *n*)



для якого похибкою апроксимації є середнє квадратичне відхилення



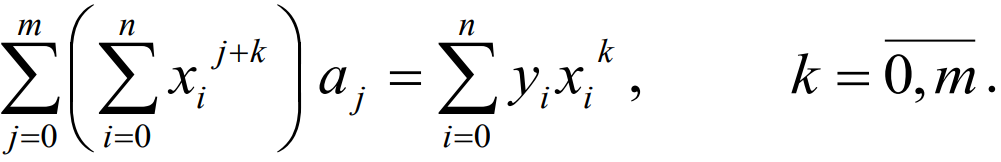
Оскільки поліном (10.1) містить невизначені коефіцієнти *ai* (*i* = 0, *m*) , то необхідно їх підібрати таким чином, щоб мінімізувати функцію



У цьому і полягає суть використання методу найменших квадратів для апроксимації функцій.

Використовуючи необхідну умову екстремуму ( *k* = 0, *m* ) функції

від багатьох змінниї Ф(*a*0 , *a*1 , *a*2 , ...*am* ), отримуємо так звану нормальну систему методу найменших квадратів дял визначення коефіцієнтів *ai* (*i* = 0,*m*) апроксимаційного полінома



Отримана система - це система лінійних алгебраїчних рівнянь відносно невідомих *a*0 , *a*1 , *a*2 , ...*am* . Можна показати, що визначник цієї системи відмінний від нуля, тобто її розв’язок існує і єдиний. Однак для високих степенів m система є погано обумовленою. Тому метод найменших квадратів застосовують для знаходження поліномів невисоких степенів m ≤ 5. Розв’язок нормальної системи шукають, використовуючи прямі або наближені методи розв’язування систем лінійних алгебраїчних рівнянь.

**Код програми**

let xI = [0.05, 0.10,   0.17,   0.25,   0.30,   0.36];

let yI = [0.54, 0.51,   0.47,   0.45,   0.42,   0.38];

let a = [];

let b = [];

let m = 2;

function findA() {

    for (let k = 0; k != m; ++k) {

        let Sum = 0;

        let stringA = [];

        for (let j = 0; j != m; ++j) {

            for (let i = 0; i != xI.length; ++i) {

              Sum += Math.pow(xI[i],k+j)

            }

            stringA.push(Sum);

            Sum = 0;

        }

        a.push(stringA)

     }

     for (let i = 0; i != m; ++i) {

        let sum = 0

        for (let j = 0; j != xI.length; ++j) {

            sum += yI[j] \* Math.pow(xI[j], i);

        }

        b.push(sum)

        sum = 0;

    }

}

function ludcmp(A, update) {

    var d = true

    var n = A.length

    var idx = new Array(n)

    var vv = new Array(n)

    for (var i=0; i<n; i++) {

        var max = 0

        for (var j=0; j<n; j++) {

            var temp = Math.abs(A[i][j])

            if (temp > max) max = temp

        }

        if (max == 0) return

        vv[i] = 1 / max

    }

    if (!update)

        var Acpy = new Array(n)

        for (var i=0; i<n; i++) {

            var Ai = A[i]

            Acpyi = new Array(Ai.length)

            for (j=0; j<Ai.length; j+=1) Acpyi[j] = Ai[j]

            Acpy[i] = Acpyi

        }

        A = Acpy

    }

    var tiny = 1e-20

    for (var i=0; ; i++) {

        for (var j=0; j<i; j++) {

            var sum = A[j][i]

            for (var k=0; k<j; k++) sum -= A[j][k] \* A[k][i];

            A[j][i] = sum

        }

        var jmax = 0

        var max = 0;

        for (var j=i; j<n; j++) {

            var sum = A[j][i]

            for (var k=0; k<i; k++) sum -= A[j][k] \* A[k][i];

            A[j][i] = sum

            var temp = vv[j] \* Math.abs(sum)

            if (temp >= max) {

                max = temp

                jmax = j

            }

        }

        if (i <= jmax) {

            for (var j=0; j<n; j++) {

                var temp = A[jmax][j]

                A[jmax][j] = A[i][j]

                A[i][j] = temp

            }

            d = !d;

            vv[jmax] = vv[i]

        }

        idx[i] = jmax;

        if (i == n-1) break;

        var temp = A[i][i]

        if (temp == 0) A[i][i] = temp = tiny

        temp = 1 / temp

        for (var j=i+1; j<n; j++) A[j][i] \*= temp

    }

    return {A:A, idx:idx, d:d}

}

     var A = lu.A

    var idx = lu.idx

    var n = idx.length

     if (!update) {

        var bcpy = new Array(n)

        for (var i=0; i<b.length; i+=1) bcpy[i] = b[i]

        b = bcpy

    }

     for (var ii=-1, i=0; i<n; i++) {

        var ix = idx[i]

        var sum = b[ix]

        b[ix] = b[i]

        if (ii > -1)

            for (var j=ii; j<i; j++) sum -= A[i][j] \* b[j]

        else if (sum)

            ii = i

        b[i] = sum

    }

    for (var i=n-1; i>=0; i--) {

        var sum = b[i]

        for (var j=i+1; j<n; j++) sum -= A[i][j] \* b[j]

        b[i] = sum / A[i][i]

    }

    return b

}

function lusolve(A, b, update) {

    var lu = ludcmp(A, update)

    if (lu === undefined) return

    return lubksb(lu, b, update)

}

findA(m);

console.log(a,b);

let x = lusolve(a,b);

console.log(`${x[1].toFixed(3)}x + ${x[0].toFixed(3)} = y`)

a = [];

b=[];

m++

findA(m);

console.log(a,b);

x = lusolve(a,b);

console.log(`${x[2].toFixed(3)}x^2+ ${x[1].toFixed(3)}x + ${x[0].toFixed(3)} = y`)

a = [];

b=[];

m++

findA(m);

console.log(a,b);

x = lusolve(a,b);

console.log(`${x[3].toFixed(3)}x^3+ ${x[2].toFixed(3)}x^2+ ${x[1].toFixed(3)}x + ${x[0].toFixed(3)} = y`)

a = [];

b=[];

**Протокол роботи**

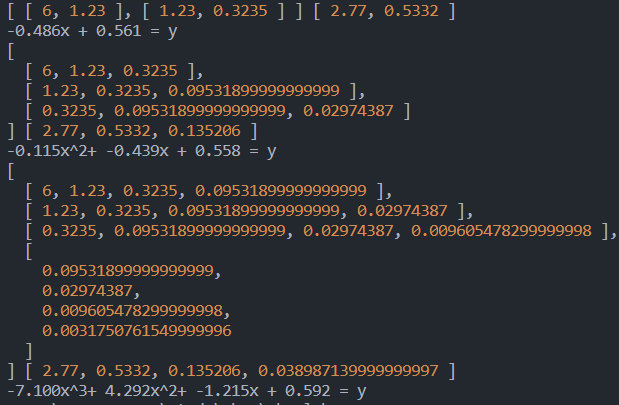


Рис.1. Результат виконання програми

**Висновки**

На даній лабораторній роботі я ознайомився на практиці з методом найменших квадратів апроксимації функцій та склав програму для знаходження апроксимаційних поліномів.