МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технологический университет «СТАНКИН»
(ФГБОУ ВО «МГТУ «СТАНКИН»)

Институт информационных систем и технологий

КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ

Вычислительная математика

Отчет по лабораторной работе

«Обработка экспериментальных данных методом наименьших квадратов»

Вариант 12

Выполнил студент гр. ИДБ-21-06 Кильдишов А.А.

Проверил преподаватель Красикова Е.М.

Москва 2023г.

Цель работы: изучить метод наименьших квадратов и применить его на практике для получения коэффициентов линейной и квадратичной функциональных зависимостей.

Краткие теоретические сведения

Метод наименьших квадратов (МНК) — математический метод, применяемый для решения различных задач, основанный на минимизации суммы квадратов отклонений некоторых функций от искомых переменных.

Для линейной функциональной зависимости получим:

$$y_i = a x_i + b + \delta_i \Rightarrow \delta_i = y_i - (a x_i + b)$$

$$F(a,b) = \sum_{i=1}^{n} \delta_i^2 = \sum_{i=1}^{n} (y_i - (a x_i + b))^2 \to min \;\; F(a,b)$$

$$\begin{cases} \frac{\partial F(a,b)}{\partial a} = 0 \\ \frac{\partial F(a,b)}{\partial b} = 0 \end{cases} \qquad \begin{cases} -2\sum_{i=1}^{n} (y_i - (a x_i + b))x_i = 0 \;\; | : (-2) \\ -2\sum_{i=1}^{n} (y_i - (a x_i + b)) = 0 \quad | : (-2) \end{cases}$$

Система линейных алгебраических уравнений для линейной апромаксирующей функции:

$$\begin{cases} a\sum_{i=1}^{n} x_i + bn = \sum_{i=1}^{n} y_i \\ a\sum_{i=1}^{n} x_i^2 + b\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} x_i y_i \end{cases}$$

Для квадратичной зависимости получим:

$$y_i = a_0 x_i^2 + a_1 x_i + a_2 + \delta_i \Rightarrow \delta_i = y_i - (a_0 x_i^2 + a_1 x_i + a_2)$$

$$F(a_0, a_1, a_2) = \sum_{i=1}^{n} \delta_i^2 = \sum_{i=1}^{n} y_i - (a_0 x_i^2 + a_1 x_i + a_2) \to min \;\; F(a_0, a_1, a_2)$$

$$\begin{cases} \frac{\partial F}{\partial a_0} = 0 \\ \frac{\partial F}{\partial a_1} = 0 \\ \frac{\partial F}{\partial a_2} = 0 \end{cases} \qquad \begin{cases} -2\sum_{i=1}^{n} (y_i - (a_0 x_i^2 + a_1 x_i + a_2))x_i^2 = 0 | : (-2) \\ -2\sum_{i=1}^{n} (y_i - (a_0 x_i^2 + a_1 x_i + a_2))x_i = 0 \;\; | : (-2) \\ -2\sum_{i=1}^{n} (y_i - (a_0 x_i^2 + a_1 x_i + a_2)) = 0 \quad | : (-2) \end{cases}$$

Система линейных алгебраических уравнений для квадратичной аппроксимирующей функции:

$$\begin{cases} a_0\sum_{i=1}^{n} x_i^2 + a_1\sum_{i=1}^{n} x_i + a_2 n = \sum_{i=1}^{n} y_i \\ a_0\sum_{i=1}^{n} x_i^3 + a_1\sum_{i=1}^{n} x_i^2 + a_2\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i x_i \\ a_0\sum_{i=1}^{n} x_i^4 + a_1\sum_{i=1}^{n} x_i^3 + a_2\sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} y_i x_i^2 \end{cases}$$

Блок-схема

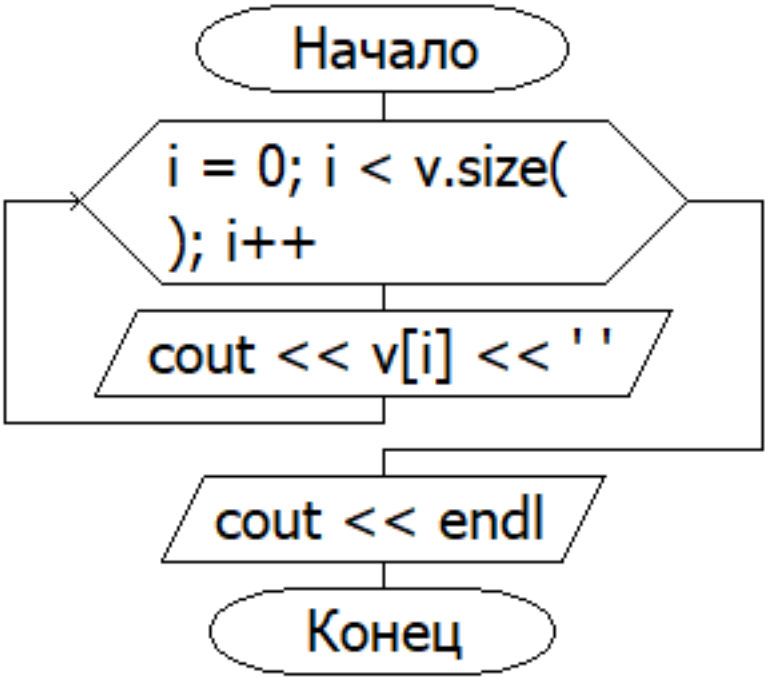

Рис.1. Print - блок-схема

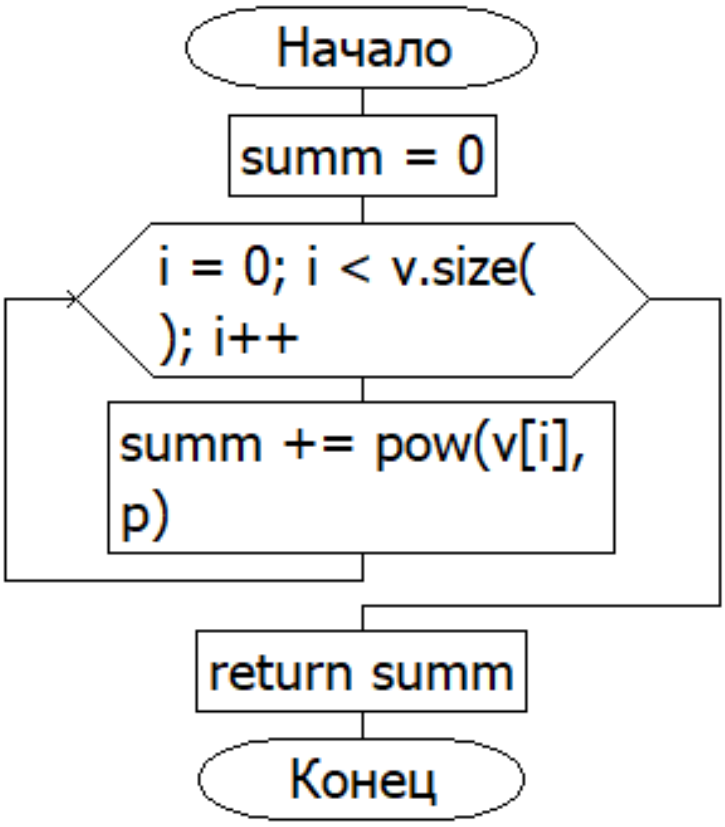

Рис.2. Summ - блок-схема

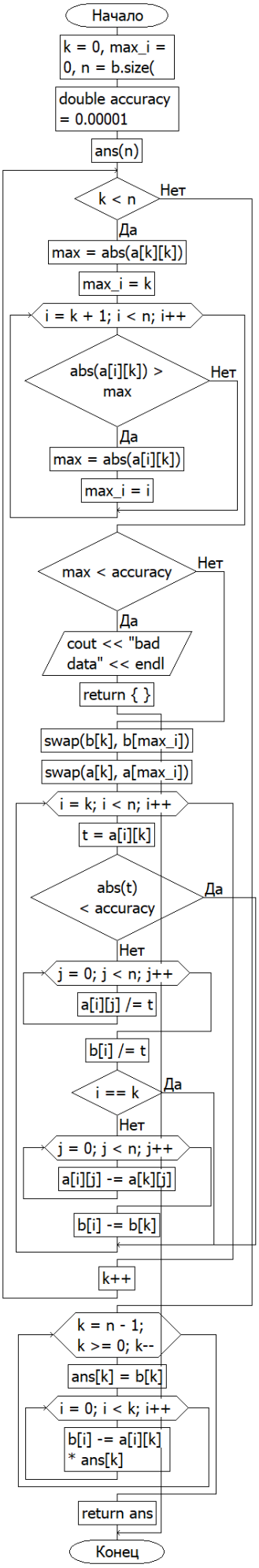

Рис.3. Guss - блок-схема

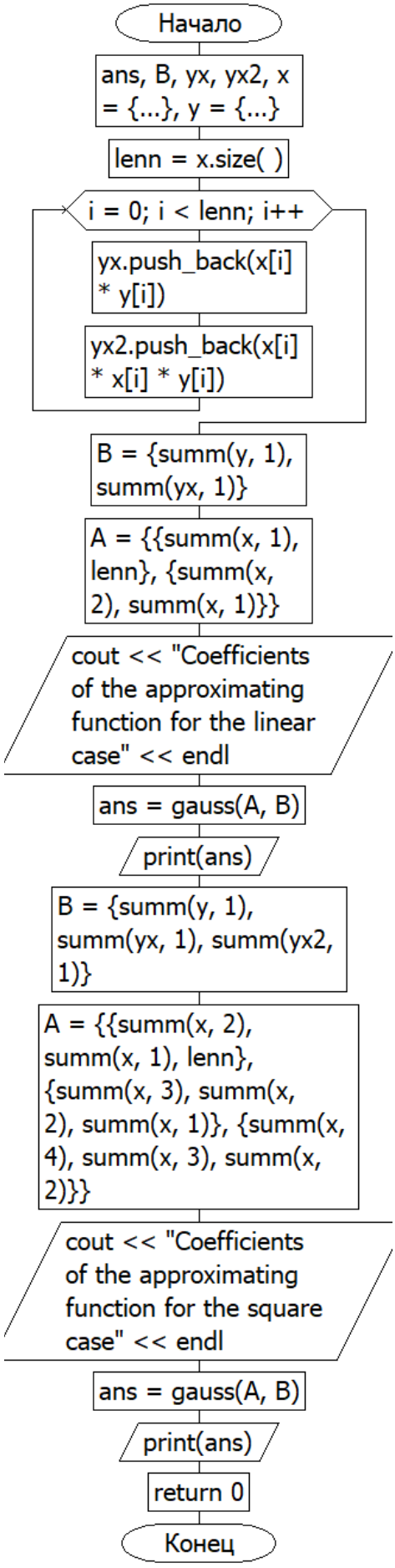

Рис.4.Main - блок-схема

Код программы

```
#include <iostream>
#include <cmath>
#include <vector>
#include <algorithm>

using namespace std;

double summ(vector<double> v, int p) {
    double summ = 0;
    for (int i = 0; i < v.size(); i++)
        summ += pow(v[i], p);
    return summ;
}

void print(vector<double> v) {
    for (int i = 0; i < v.size(); i++)
        cout << v[i] << ' ';
    cout << endl;
}

vector<double> gauss(vector< vector<double> > a, vector<double> b)
{
    double max, t;
    int k = 0, max_i = 0, n = b.size();
    const double accuracy = 0.00001;
    vector<double> ans(n);
    while (k < n)
    {
        max = abs(a[k][k]);
        max_i = k;
        for (int i = k + 1; i < n; i++)
        {
            if (abs(a[i][k]) > max)
            {
                max = abs(a[i][k]);
                max_i = i;
            }
        }
        if (max < accuracy)
        {
            cout << "bad data" << endl;
            return {};
```

```
        }
        swap(b[k], b[max_i]);
        swap(a[k], a[max_i]);
        for (int i = k; i < n; i++)
        {
            t = a[i][k];
            if (abs(t) < accuracy)
                continue;
            for (int j = 0; j < n; j++)
                a[i][j] /= t;
            b[i] /= t;
            if (i == k)
                continue;
            for (int j = 0; j < n; j++)
                a[i][j] -= a[k][j];
            b[i] -= b[k];
        }
        k++;
    }
    for (k = n - 1; k >= 0; k--)
    {
        ans[k] = b[k];
        for (int i = 0; i < k; i++)
            b[i] -= a[i][k] * ans[k];
    }
    return ans;
}

int main()
{
    vector< vector<double> > A;
    vector<double> ans, B, yx, yx2,
      x = { 0.168,
            0.115,
            0.928,
            0.962,
            0.129,
            0.762,
            0.646,
            0.055,
            0.186,
            0.563
            },
        y = { 5.524,
            5.605,
            3.264,
```

```
        3.072,
        5.497,
        3.579,
        3.645,
        5.667,
        5.131,
        4.127
        };
    double lenn = x.size();
    for (int i = 0; i < lenn; i++)
    {
       yx.push_back(x[i] * y[i]);
       yx2.push_back(x[i] * x[i] * y[i]);
    }
    B = { summ(y,1), summ(yx, 1) };
    A = { {summ(x,1), lenn},
          {summ(x,2),summ(x,1)} };
    cout << "Coefficients of the approximating function for the linear case" << endl;
    ans = gauss(A, B);
    print(ans);
    B = { summ(y,1), summ(yx, 1), summ(yx2, 1) };
    A = { {summ(x,2), summ(x,1), lenn},
          {summ(x,3), summ(x,2),summ(x,1)},
          {summ(x,4), summ(x,3),summ(x,2)} };

    cout << "Coefficients of the approximating function for the square case" << endl;
    ans = gauss(A, B);
    print(ans);

    return 0;
}
```

```
Coefficients of the approximating function for the linear case
-2.95409 5.84458
Coefficients of the approximating function for the square case
1.22149 -4.16722 6.00164
```

Рис.5. Результат работы программы

Графики

Y - экспериментальные данные Yл - значения по линейной аппроксимирующей функции Yк - значения по квадратной аппроксимирующей функции										
X	0,168	0,115	0,928	0,962	0,129	0,762	0,646	0,055	0,186	0,563
Y	5,524	5,605	3,264	3,072	5,497	3,579	3,645	5,667	5,131	4,127
Yл	5,348	5,505	3,103	3,003	5,464	3,594	3,936	5,682	5,295	4,181
Yк	5,336	5,539	3,186	3,123	5,484	3,535	3,819	5,776	5,269	4,043

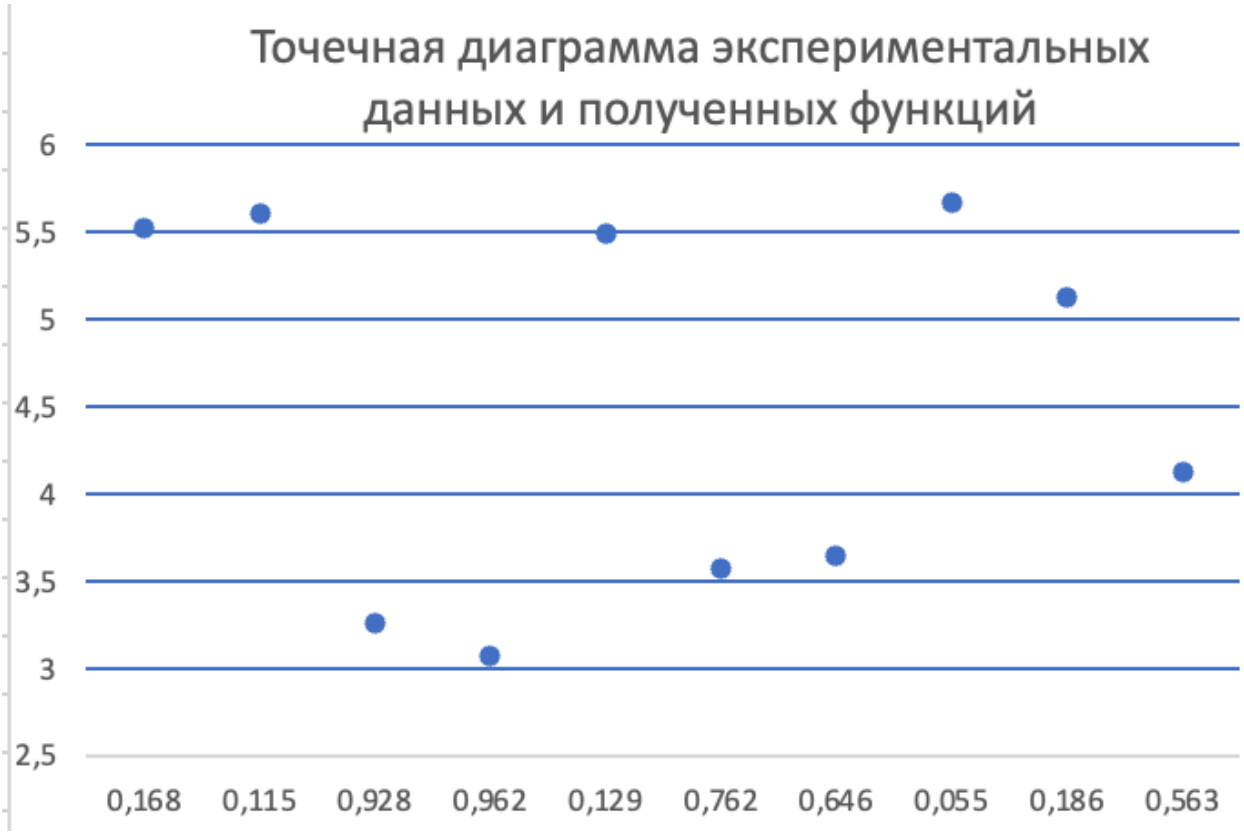

Рис.6.Точеченая диаграмма экспериментальных данных полученных в функции

δ_i линейного случая	0,176	0,100	0,161	0,069	0,033	-0,015	-0,291	-0,015	-0,164	-0,054
δ_i квадратного случая	0,188	0,066	0,078	-0,051	0,013	0,044	-0,174	-0,109	-0,138	0,084

Общая квадратичная погрешность линейного случая: 0,188
Общая квадратичная погрешность квадратного случая: 0,119

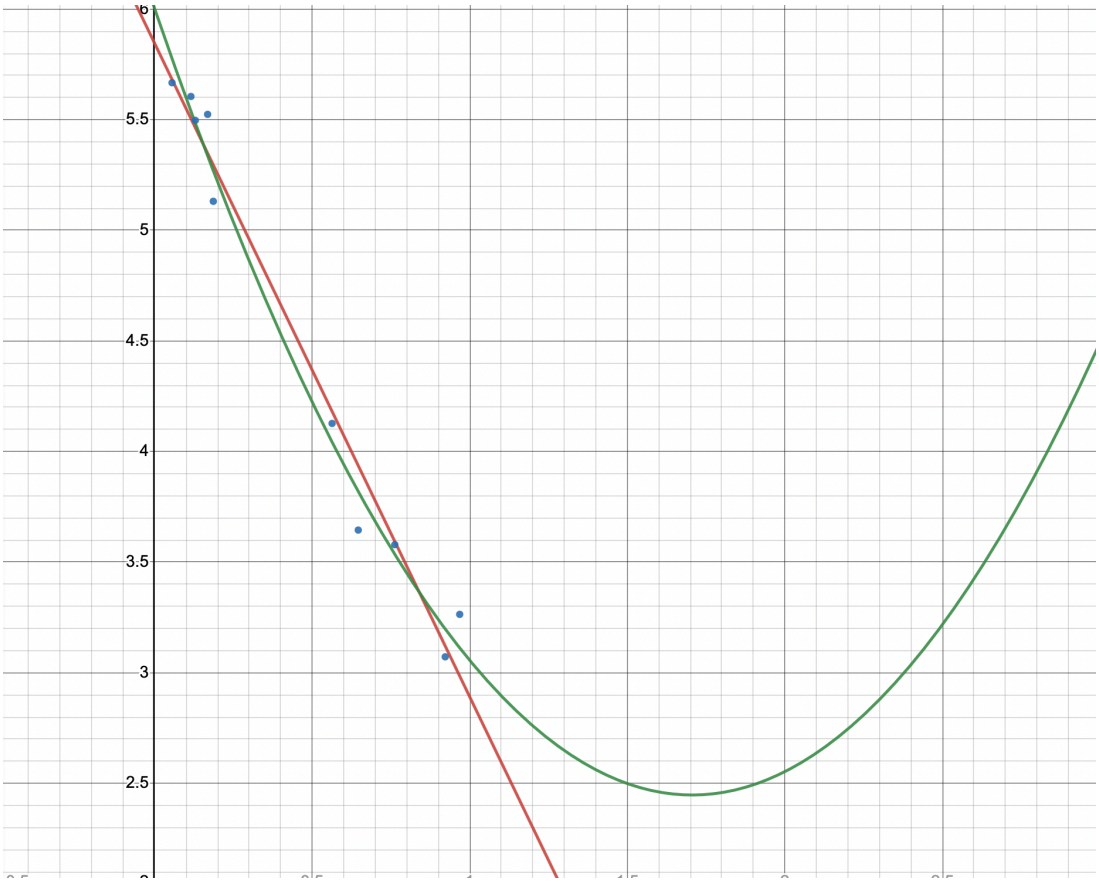

Рис.7. Точечная диаграмма экспериментальных данных и графики полученных функций

Вывод

Был изучен метод наименьших квадратов (МНК). Данный метод был применён на практике для получения коэффициентов линейной и квадратичной функциональной зависимостей. В моем случае лучшей оказался метод квадратной аппроксимации(погрешность на 0.069 меньше)