Отчёт по летней практике

Кошелев Александр

3 июня 2023 г.

1 Задание

Обработка результатов эксперимента.
Метод наименьших квадратов.

В результате эксперимента была определена некоторая табличная зависимость.

С помощью метода наименьших квадратов определить линию регрессии, рассчитать коэффициент корреляции, подобрать функциональную зависимость заданного вида, вычислить коэффициент регрессии. Построить график экспериментальной зависимости, линию регрессии и график подобранной зависимости. Определить суммарную квадратичную ошибку и среднюю ошибку для линии регрессии и подобранной функциональной зависимости. Написать программу на языке С(С++) для решения задачи. При необходимости напишите функцию решения системы линейных алгебраических уравнений.

Вариант № 10. $Z=At^4+Bt^3+Ct^2+Dt+K$

t	0.66	0.9	1.17	1.47	1.7	1.74	2.08	2.63	3.12
Z	38.9	68.8	64.4	66.5	64.95	59.36	82.6	90.63	113.5

Код:

```
#include <iostream>
#include <cmath>
#include <fstream>

using namespace std;

// Function to solve a system of linear equations using Gaussian elimination method
void solveLinearSystem(double A[][5], double b[], double x[], int n) {
    for (int i = 0; i < n; i++) {
        int maxRow = i;
        for (int j = i + 1; j < n; j++) {
            if (abs(A[j][i]) > abs(A[maxRow][i]))
                maxRow = j;
        }
        swap(A[i], A[maxRow]);
        swap(b[i], b[maxRow]);

        for (int j = i + 1; j < n; j++) {
            double ratio = A[j][i] / A[i][i];
            for (int k = i; k < n; k++) {
                A[j][k] -= ratio * A[i][k];
            }
            b[j] -= ratio * b[i];
```

```
        }
    }

    for (int i = n - 1; i >= 0; i--) {
        double sum = 0;
        for (int j = i + 1; j < n; j++) {
            sum += A[i][j] * x[j];
        }
        x[i] = (b[i] - sum) / A[i][i];
    }
}

double calculateSumSquaredError(double t[], double Z[], double A_val, double B_val, double
    double sumSquaredError = 0;
    for (int i = 0; i < n; i++) {
        double predictedValue = A_val * pow(t[i], 4) + B_val * pow(t[i], 3) + C_val * pow(
        double error = Z[i] - predictedValue;
        sumSquaredError += pow(error, 2);
    }
    return sumSquaredError;
}

double calculateMeanError(double t[], double Z[], double A_val, double B_val, double C_val
    double meanError = 0;
    for (int i = 0; i < n; i++) {
        double predictedValue = A_val * pow(t[i], 4) + B_val * pow(t[i], 3) + C_val * pow(
        double error = Z[i] - predictedValue;
        meanError += error;
    }
    meanError /= n;
    return meanError;
}

double calculateCorrelationCoefficient(double t[], double Z[], double A_val, double B_val,
    double sumZ = 0, sumZSquared = 0, sumPredicted = 0, sumPredictedSquared = 0, sumZPredi
    for (int i = 0; i < n; i++) {
        double predictedValue = A_val * pow(t[i], 4) + B_val * pow(t[i], 3) + C_val * pow(
        sumZ += Z[i];
        sumZSquared += pow(Z[i], 2);
        sumPredicted += predictedValue;
        sumPredictedSquared += pow(predictedValue, 2);
        sumZPredicted += Z[i] * predictedValue;
    }
    double numerator = n * sumZPredicted - sumZ * sumPredicted;
    double denominator = sqrt((n * sumZSquared - pow(sumZ, 2)) * (n * sumPre-dictedSquared
    double correlationCoefficient = numerator / denominator;
    return correlationCoefficient;
}

double calculateRegressionCoefficient(double t[], double Z[], double A_val, dou-ble B_val,
    double sumZ = 0, sumZSquared = 0, sumPredicted = 0, sumPredictedSquared = 0, sumZPredi
    for (int i = 0; i < n; i++) {
        double predictedValue = A_val * pow(t[i], 4) + B_val * pow(t[i], 3) + C_val * pow(
        sumZ += Z[i];
        sumZSquared += pow(Z[i], 2);
        sumPredicted += predictedValue;
        sumPredictedSquared += pow(predictedValue, 2);
        sumZPredicted += Z[i] * predictedValue;
```

```
    }
    double regressionCoefficient = sumZPredicted / sqrt(sumZSquared * sumPre-dictedSquared
    return regressionCoefficient;
}

int main() {
    // Input data
    double t[] = { 0.66, 0.9, 1.17, 1.47, 1.7, 1.74, 2.08, 2.63, 3.12 };
    double Z[] = { 38.9, 68.8, 64.4, 66.5, 64.95, 59.36, 82.6, 90.63, 113.5 };
    int n = sizeof(t) / sizeof(t[0]);
    // Creating the matrix and vector for linear system
    double A[5][5] = { {0, 0, 0, 0, 0}, {0, 0, 0, 0, 0}, {0, 0, 0, 0, 0}, {0, 0, 0, 0, 0},
    double b[5] = { 0, 0, 0, 0, 0 };
    double x[5] = { 0, 0, 0, 0, 0 };

    // Filling the matrix and vector
    for (int i = 0; i < n; i++) {
        double t2 = t[i] * t[i];
        double t3 = t2 * t[i];
        double t4 = t3 * t[i];
        A[0][0] += t4;
        A[0][1] += t3;
        A[0][2] += t2;
        A[0][3] += t[i];
        A[0][4] += 1;
        b[0] += Z[i];
        A[1][1] += t2;
        A[1][2] += t[i];
        A[1][3] += 1;
        b[1] += Z[i];
        A[2][2] += 1;
        A[2][3] += t[i];
        A[2][4] += 1;
        b[2] += Z[i];
        A[3][3] += t[i];
        A[3][4] += 1;
        b[3] += Z[i];
        A[4][4] += 1;
        b[4] += Z[i];
    }

    // Solving the linear system
    solveLinearSystem(A, b, x, 5);

    double A_val = x[0];
    double B_val = x[1];
    double C_val = x[2];
    double D_val = x[3];
    double K_val = x[4];

    // Calculating the errors and coefficients
    double sumSquaredError = calculateSumSquaredError(t, Z, A_val, B_val, C_val, D_val, K_
    double meanError = calculateMeanError(t, Z, A_val, B_val, C_val, D_val, K_val, n);
    double correlationCoefficient = calculateCorrelationCoefficient(t, Z, A_val, B_val, C_
    double regressionCoefficient = calculateRegressionCoefficient(t, Z, A_val, B_val, C_va

    // Outputting the results
    cout << "A: " << A_val << endl;
```

```
    cout << "B: " << B_val << endl;
    cout << "C: " << C_val << endl;
    cout << "D: " << D_val << endl;
    cout << "Sum of Squared Error: " << sumSquaredError << endl;
    cout << "Mean Error: " << meanError << endl;
    cout << "Correlation Coefficient: " << correlationCoefficient << endl;
    cout << "Regression Coefficient: " << regressionCoefficient << endl;

    // Creating a data file for plots
    ofstream dataFile("data.txt");
    for (int i = 0; i < n; i++) {
        dataFile << t[i] << " " << Z[i] << endl;
    }
    dataFile.close();

    // Creating a gnuplot script file
    ofstream scriptFile("script.plt");
    scriptFile << "set title 'Experimental data'\n";
    scriptFile << "set xlabel 't'\n";
    scriptFile << "set ylabel 'Z'\n";
    scriptFile << "plot 'data.txt' with linespoints title 'Experimental data', \\" << endl
    scriptFile << "     " << A_val << "*x**4 + " << B_val << "*x**3 + " << C_val << "*x**2
    scriptFile << "     " << A_val << "*x**2 + " << B_val << "*x + " << C_val << " title '
    scriptFile << "pause -1 'Press any key to exit'\n";
    scriptFile.close();


    // Running gnuplot to display the plots
    system("gnuplot script.plt");

    return 0;
}
```

Скриншоты работы программы:

```
A: -8.09989
B: 20.5123
C: 0
D: 0
Sum of Squared Error: 43260.6
Mean Error: 6.94753e-14
Correlation Coefficient: -0.73466
Regression Coefficient: 0.658847
Нажмите любую клавишу для выхода
```