Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра програмного забезпечення

ЗВІТ

Про виконання лабораторної роботи № 6

«РОЗВ'ЯЗУВАННЯ ПЕРЕВИЗНАЧЕНИХ СИСТЕМ ЛІНІЙНИХ АЛГЕБРАЇЧНИХ РІВНЯНЬ»

з дисципліни «Чисельні методи»

Лектор:

доцент кафедри ПЗ

Мельник Н.Б.

Виконав:

студ. групи ПЗ-15

Бабіля О.О.

Прийняв:

асистент кафедри ПЗ

Гарматій Г.Ю.

«___» ______ 2022 р.

$\sum$ = ____ ___________

Львів – 2022

Мета: ознайомлення на практиці з методами розв’язування перевизначених систем лінійних алгебраїчних рівнянь

Теоретичні відомості:

Метод найменших квадратів для розв'язування перевизначених СЛАР

Розглянемо систему лінійних алгебраїчних рівнянь, у якій кількість рівнянь є більшою за кількість невідомих

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1m}x_m = b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2m}x_m = b_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nm}x_m = b_n, \end{cases} \qquad (\ \ 1)$$

де $n > m$.

У загальному випадку система рівнянь (1) є несумісною. Якщо із даної системи вибрати m рівнянь та розв’язати їх, то отриманий розв’язок не буде задовольняти всі рівняння системи (1). Тому поступимо інакше: знайдемо розв’язок системи $x_1, x_2, ..., x_m$ наближено, але щоб він задовольняв усі рівняння системи (1), а саме

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1m}x_m - b_1 = \varepsilon_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2m}x_m - b_2 = \varepsilon_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nm}x_m - b_n = \varepsilon_n. \end{cases} \qquad (\ \ 2)$$

Розв’язок системи (2) будемо знаходити з використанням умови мінімізації суми квадратів відхилень, тобто з умов мінімізації функції

$$S(x_1, x_2, ..., x_m) = \sum_{j=1}^{m} \varepsilon_i^2 \qquad (\ \ 3)$$

і вимагатимемо, щоб виконувалась умова

$$\sum_{i=1}^{m} \varepsilon_i^2 \to \min . \qquad (\ \ 4)$$

Проведемо деякі перетворення над системою (2), використовуючи умову (4). Розглянемо функцію

$$S(x_1, x_2, \ldots, x_m) = \sum_{i=1}^{n} \left(\sum_{j=1}^{m} a_{ij} x_j - b_j \right)^2 . \qquad (\ \ 5)$$

Необхідною умовою мінімуму функції від багатьох змінних є рівність нулеві її частинних похідних. Використаємо цей факт і продиференціюємо функцію (5) за змінними x_i $(i = \overline{1, m})$. У результаті отримаємо

$$\frac{\partial S}{\partial x_k} = 2 \sum_{i=1}^{n} a_{ik} \left(\sum_{j=1}^{m} a_{ij} x_j - b_j \right), \qquad k = \overline{1, m} . \qquad (\ \ 6)$$

Прирівнявши вирази(6) до нуля, отримаємо нормальну систему рівнянь

$$\sum_{i=1}^{n} a_{ik} \left(\sum_{j=1}^{m} a_{ij} x_j - b_j \right) = 0, \qquad k = \overline{1,m}, \tag{7}$$

в якій кількість рівнянь системи дорівнює кількості невідомих. Нормальні системи лінійних алгебраїчних рівнянь характеризуються тим, що матриці їх коефіцієнтів завжди є симетричними, а діагональні елементи - додатніми.

Систему (7) розв'язують довільними прямими або ітераційними методами. Якщо матриця коефіцієнтів системи рівнянь (7) є додатньо визначеною (визначник матриці є більшим за нуль), то рекомендують для її розв’язування використовувати метод квадратного кореня.

Запишемо систему лінійних алгебраїчних рівнянь (1) у матричному вигляді

$$AX = B, \tag{8}$$

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & & & \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}, \quad X = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}, \quad B = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{pmatrix},$$

де A - матриця коефіцієнтів системи розмірності $m \times n$, X -- матриця-стовпець невідомих розмірності $m \times 1$, B - матриця-стовпець вільних членів системи розмірності $m \times 1$.

Матричне рівняння (8) помножимо на транспоновану матрицю A^T до матриці A. У результаті отримаємо матричне рівняння

$$NX = C, \tag{9}$$

де N – матриця коефіцієнтів нормальної системи

$$N = A^T A,$$

C-стовпець вільних членів

$$C = A^T B.$$

Розв’язавши нормальну систему лінійних алгебраїчних рівнянь, отримаємо її точний розв’язок (якщо використано прямі методи) або наближений розв’язок (якщо використано ітераційні методи). Отриманий розв’язок буде наближеним для СЛАР (1).

Індивідуальне завдання

Розв’язати перевизначену систему лінійних алгебраїчних рівнянь методом найменших квадратів. Отриману відповідну нормальну систему розв’язати методом квадратного кореня.

1. $$\begin{cases} 3x_1 + x_2 - x_3 = 2 \\ -5x_1 + x_2 + 3x_3 = -10 \\ 2x_1 + x_3 = 1 \\ x_1 - 5x_2 + 3x_3 = 7 \\ 2x_1 - 4x_2 - x_3 = 11 \end{cases}$$

Хід роботи:

$$\begin{cases} 3x_1 + x_2 - x_3 = 2 \\ -5x_1 + x_2 + 3x_3 = -10 \\ 2x_1 + x_3 = 1 \\ x_1 - 5x_2 + 3x_3 = 7 \\ 2x_1 - 4x_2 - x_3 = 11 \end{cases}$$

$$AX = B$$

$$A = \begin{vmatrix} 3 & 1 & -1 \\ -5 & 1 & 3 \\ 2 & 0 & 1 \\ 1 & -5 & 3 \\ 2 & -4 & -1 \end{vmatrix} \quad B = \begin{vmatrix} 2 \\ -10 \\ 1 \\ 7 \\ 11 \end{vmatrix} \quad X = \begin{vmatrix} x_1 \\ x_2 \\ x_3 \end{vmatrix}$$

$$N = A^T \cdot A =$$

$$= \begin{vmatrix} 3 & -5 & 2 & 1 & 2 \\ 1 & 1 & 0 & -5 & -4 \\ -1 & 3 & 1 & 3 & -1 \end{vmatrix} \cdot \begin{vmatrix} 3 & 1 & -1 \\ -5 & 1 & 3 \\ 2 & 0 & 1 \\ 1 & -5 & 3 \\ 2 & -4 & -1 \end{vmatrix} = \begin{pmatrix} 43 & -15 & -15 \\ -15 & 43 & -9 \\ -15 & -9 & 21 \end{pmatrix}$$

$$C = A^T \cdot B = \begin{vmatrix} 3 & -5 & 2 & 1 & 2 \\ 1 & 1 & 0 & -5 & -4 \\ -1 & 3 & 1 & 3 & -1 \end{vmatrix} \cdot \begin{vmatrix} 2 \\ -10 \\ 1 \\ 7 \\ 11 \end{vmatrix} = \begin{vmatrix} 87 \\ -87 \\ 21 \end{vmatrix}$$

$$NX = C =$$

$$= \begin{vmatrix} 43 & -15 & -15 \\ -15 & 43 & -9 \\ -15 & -9 & 21 \end{vmatrix} \cdot \begin{vmatrix} x_1 \\ x_2 \\ x_3 \end{vmatrix} = \begin{vmatrix} 87 \\ -87 \\ 21 \end{vmatrix}; \quad \det N = 16896 > 0$$

Застосуєм метод квадратного кореня:

$$N = L^T \cdot L$$

$$l_{11} = \sqrt{n_{11}} = 6,55; \quad l_{21} = \frac{n_{21}}{l_{11}} = \frac{-15}{6,55} = -2,28; \quad l_{31} = \frac{n_{31}}{l_{11}} = -2,28$$

$$l_{22} = \sqrt{n_{22} - l_{21}^2} = \sqrt{43 - (-2,28)^2} = 6,14; \quad l_{32} = \frac{n_{32} - l_{31} l_{21}}{l_{22}} = -2,31$$

$$l_{33} = \sqrt{n_{33} - l_{31}^2 - l_{32}^2} = 3.22$$

$$N = \begin{pmatrix} 6,55 & 0 & 0 \\ -2,28 & 6,14 & 0 \\ -2,28 & -2,31 & 3,22 \end{pmatrix} \cdot \begin{pmatrix} 6,55 & -2,28 & -2,28 \\ 0 & 6,14 & -2,31 \\ 0 & 0 & 3,22 \end{pmatrix}$$

$$LY = C =$$

$$= \begin{pmatrix} 6,55 & 0 & 0 \\ -2,28 & 6,14 & 0 \\ -2,28 & -2,31 & 3,22 \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 13,2 \\ -9,2 \\ -3,7 \end{pmatrix}$$

$$L^T X = Y =$$

$$= \begin{pmatrix} 6,55 & -2,28 & -2,28 \\ 0 & 6,14 & -2,31 \\ 0 & 0 & 3,22 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 13,2 \\ -9,2 \\ -3,7 \end{pmatrix} = \begin{pmatrix} 0,94 \\ -1,93 \\ -1,15 \end{pmatrix}$$

$x_1 = 0,94$, $x_2 = -1,93$, $x_3 = -1,15$.

Отже, розв’язком даної СЛАР є:

X_1=0.94,

X_2=-1.93,

X_3=-1.15.

Код програми:

Source.c

```c
#define CRT_SECURE_NO_WARNINGS
#include <math.h>
#include <iostream>
#define I 5
#define J 3
#define R 5
void Print(double** A, double* B, int n1, int n2)
{
    for (int i = 0; i < n1; i++)
    {
        for (int j = 0; j < n2; j++)
            printf("%+.3lf ", A[i][j]);

        printf("= %+.3lf\n", B[i]);
    }
    printf("\n");
}
void Show(double** A, int n1, int n2)
{
```

```
    for (int i = 0; i < n1; i++)
    {
        for (int j = 0; j < n2; j++)
            printf("%+.3lf ", A[i][j]);

        printf("\n");
    }
    printf("\n");
}
double** Multiplication(double A[3][3])
{
    int n = 3;
    int m = 3;
    double** N = new double* [n];
    for (int i = 0; i < n; i++)
        N[i] = new double[m];
    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            N[i][j] = 0;
    for (int i = 0; i < 3; i++)
        for (int j = 0; j < 3; j++)
            for (int r = 0; r < R; r++)
                N[i][j] += A[r][i] * A[r][j];
    return N;
}
double* Multiplication(double A[3][3], double B[3])
{
    int n = 3;
    double* C = new double[n];
    for (int i = 0; i < n; i++)
        C[i] = 0;

    for (int i = 0; i < 3; i++)
        for (int j = 0; j < R; j++)
            C[i] += A[j][i] * B[j];

    return C;
}
double** L(double** N)
{
    int n = 3;
    int m = 3;
    double** L = new double* [n];
    for (int i = 0; i < n; i++)
        L[i] = new double[m];

    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            L[i][j] = 0;

    for (int i = 0; i < 3; i++)
        for (int j = 0; j <= i; j++)
        {
            if (i == j)
            {
                L[i][j] = N[i][j];
                for (int r = 0; r < i; r++)
                    L[i][j] -= pow(L[i][r], 2);

                L[i][j] = sqrt(L[i][j]);
            }
```

```
            else if (j == 0)L[i][j] = N[i][j] / L[0][0];

            else
            {
                L[i][j] = N[i][j];
                for (int r = 0; r < j; r++)
                    L[i][j] -= L[i][r] * L[j][r];

                L[i][j] = L[i][j] / L[j][j];
            }
        }
    return L;
}
double* Y(double** L, double* C)
{
    int n = 3;
    double* Y = new double[n];
    for (int i = 0; i < n; i++)
        Y[i] = 0;

    for (int i = 0; i < 3; i++)
    {
        Y[i] = C[i];
        for (int m = 0; m <= i - 1; m++)
            Y[i] = Y[i] - L[i][m] * Y[m];

        Y[i] = Y[i] / L[i][i];
    }
    return Y;
}
void Transpon(double** A)
{
    int n = 3;
    int m = 3;
    double** Temp = new double* [n];
    for (int i = 0; i < n; i++)
        Temp[i] = new double[m];

    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            Temp[i][j] = 0;

    for (int i = 0; i < 3; i++)
        for (int j = 0; j < 3; j++)
            Temp[i][j] = A[j][i];

    for (int i = 0; i < 3; i++)
        for (int j = 0; j < 3; j++)
            A[i][j] = Temp[i][j];

    delete[]Temp;
}
double* X(double** L, double* Y)
{
    int n = 3;
    double* X = new double[n];
    for (int i = 0; i < n; i++)
        X[i] = 0;

    for (int i = 2; i >= 0; i--)
    {
```

```
        X[i] = Y[i];
        for (int m = i + 1; m < 3; m++)
            X[i] -= L[i][m] * X[m];

        X[i] = X[i] / L[i][i];
    }
    return X;
}
int main() {
    double A[I][J] = {
            {3, 1, -1   },
            {-5, 1,3},
            {2, 0, 1},
            {1, -5, 3},
            {2, -4, -1} };
    double B[I] = { 2,-10,1,7,11 };
    double** N = Multiplication(A);
    double* C = Multiplication(A, B);
    printf("C matrix:\n");
    for (int i = 0; i < R; i++)
        printf("%lf\n", C[i]);

    printf("\nN matrix:\n");
    Show(N, 3, 3);
    double** _L = L(N);
    printf("L matrix:\n");
    Show(_L, 3, 3);
    double* _Y = Y(_L, C);
    printf("Y matrix:\n");
    for (int i = 0; i < R; i++)
        printf("%lf\n", _Y[i]);

    printf("\nLX=Y:\n");
    Print(_L, _Y, 3, 3);
    Transpon(_L);

    double* _X = X(_L, _Y);
    for (int i = 0; i < 3; i++)
        printf("X%i = %f\n", i + 1, _X[i]);

    return 0;
}
```

Вигляд виконаної програми

```
C matrix:
87.000000
-87.000000
-21.000000


N matrix:
+43.000 -15.000 -15.000
-15.000 +43.000 -9.000
-15.000 -9.000 +21.000

L matrix:
+6.557 +0.000 +0.000
-2.287 +6.146 +0.000
-2.287 -2.316 +3.226

Y matrix:
13.267376
-9.218283
-3.720337



LX=Y:
X1 = 0.946023
X2 = -1.934659
X3 = -1.153409

C:\Users\home\source\repos\ЧМ 06\Debug\ЧМ 06.exe (процесс 22232) завершил работу с кодом 0.
```

Висновки:

На даній лабораторній роботі я ознайомився на практиці з методом найменших квадратів для розв'язування перевизначених СЛАР. Розв’язав СЛАР,
згідно до індивідуального завдання:

1. $$\begin{cases} 3x_1 + x_2 - x_3 = 2 \\ -5x_1 + x_2 + 3x_3 = -10 \\ 2x_1 + x_3 = 1 \\ x_1 - 5x_2 + 3x_3 = 7 \\ 2x_1 - 4x_2 - x_3 = 11 \end{cases}$$

Розв’язками стали

X_1=0.94,

X_2=-1.93,

X_3=-1.15.

Середнє значення похиюки = 0.03