Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Отчет по лабораторной работе №1

«Решение системы линейных алгебраических уравнений СЛАУ»

По дисциплине «Вычислительная математика»

Вариант 8

Выполнила: Иванова Мария Максимовна

Группа: P3208

Преподаватель: Машина Екатерина Алексеевна

Санкт-Петербург

~ 2024 ~

Цель работы: изучение прямых и итерационных методов решения систем линейных алгебраических уравнений СЛАУ.

Описание используемого метода:

$a11x1 + a12x2 + \ldots + a1n\, xn = b1,$

$a21x1 + a22x2 + \ldots + a2n\, xn = b2,$

……………………………………..

$an1x1 + an2x2 + \ldots + ann\, xn = bn.$

Прямой ход:

Шаг 1 (считаем $a11 \neq 0$): Исключим $x1$ из второго уравнения: умножим первое уравнение на ($-a21/a11$) и прибавим ко второму. Исключим $x1$ из третьего уравнения: умножим первое уравнение на ($-a31/a11$) и прибавим к третьему… Исключим $x1$ из последнего уравнения: умножим первое уравнение на ($-an1/a11$) и прибавим к последнему. Получим равносильную систему уравнений.

Шаг 2: Исключим $x2$ из третьего уравнения: умножим второе уравнение на ($-a32'\, a22'$) и прибавим к третьему (и т.д. для следующих уравнений) Исключим $x2$ из последнего уравнения: умножим второе уравнение на ($-an2'/a22'$) и прибавим к последнему.

Продолжим до тех пор, пока матрица системы (3) не примет треугольный вид :

$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}\, xn = b1,$

$a_{22}{}^{(1)}\, x_2 + a_{23}{}^{(1)}\, x_3 + \cdots + a_{2n}{}^{(1)}\, x_n = b2,$

$a_{33}{}^{(2)}\, x_3 + \ldots + a_{3n}{}^{(2)}\, x_n = b3^{(2)}$

$a_{nn}{}^{(n-1)}\, x_n = bn^{(n-1)}$

Матрица системы имеет треугольный вид $\rightarrow$ конец прямого хода.

Требование: если в процессе исключения неизвестных, коэффициенты: $\boldsymbol{a11}$, $\boldsymbol{a22}^1$, $\boldsymbol{a33}^2$…. = 0 , тогда необходимо соответственным образом переставить уравнения системы. Перестановка уравнений должна быть предусмотрена в вычислительном алгоритме при его реализации на компьютере.

Обратный ход:

$xn = bn\ (n-1)\ ann\ (n-`1)$

………………………..

$x2 = 1/a22\ 1\ (b2\ 1 - a23\ 1\ x3 - \cdots - a2n\ 1\ xn)$

$x1 = 1/a11\ (b1 - a12x2 - a13x3 - \cdots - a1nxn)$

Трудоемкость метода. Для реализации метода Гаусса требуется примерно 2/3 n^3 операций для прямого хода и n^2 операций для обратного хода. Общее количество операций: 2/3 $n^3 + n^2$.

Листинг программы:

```
public class GaussMethod {
    private final int n;
    private double[][] matrix;


    public GaussMethod(int n, double[][] matrix) {
        this.n = n;
        this.matrix = matrix;
    }

    public void getDiscrepancy(double[][] matrix, double[] x, int n)
    {
        double[] discrepancy = new double[n];

        for(int i = 0; i < n; i++){
            double r = matrix[i][n];
            for(int j = 0; j < n; j++){
                r -= matrix[i][j]*x[j];
            }

            discrepancy[i] = r;
        }

        for (double i : discrepancy) System.out.print(i+" ");

    }


    public void getDeterminant(double[][] matrix) {
        double determinant = matrix[0][0];
        for (int i = 1; i < n; i++) {
            determinant *= matrix[i][i];
            if (determinant == 0) {
                System.out.println("Определитель равен нулю. Система имеет
бесконечное количество решений.");
                break;
            }
        }
        System.out.println("Определитель = " + determinant);
    }

    public void calculate() {
        double [][] matrix2 = new double[n][n+1];
        matrix2 = matrix;
        double[] x = new double[n];
        for (int i = 0; i < n; i++) {
            // Приведение к треугольному виду
            for (int j = i + 1; j < n; j++) {
                double factor = matrix[j][i] / matrix[i][i];
                for (int k = i; k < n + 1; k++) {
                    matrix[j][k] -= factor * matrix[i][k];
                }
            }
        }

        System.out.println("Треугольная матрица:");
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n+1; j++) {
                System.out.printf("%.2f", matrix[i][j]);
                System.out.print(" ");
            }
```

```
            System.out.println();
        }

        getDeterminant(matrix);

        // Решение системы обратным ходом

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

        // Вывод результата
        System.out.println("Вектор неизвестных:");
        for (int i = 0; i < n; i++) {
            System.out.print("x" + (i + 1) + " = ");
            System.out.printf("%.2f", x[i]);
            System.out.println();
        }

        System.out.println("Вектор невязки:");
        getDiscrepancy(matrix2, x, n);

    }
}
```

Результат работы программы:

Выберете: 1 - чтение с консоли, 2 - чтение из файла, 3 - генерация матрицы со случайными коэффициентами

2

Размер матрицы: 3

Исходная матрица:

10,000 -7,000 0,000 7,000

-3,000 2,000 6,000 4,000

5,000 -1,000 5,000 6,000

Треугольная матрица:

10,00 -7,00 0,00 7,00

0,00 -0,10 6,00 6,10

0,00 0,00 155,00 155,00

Определитель = -155.00000000000003

Вектор неизвестных:

x1 = -0,00

x2 = -1,00

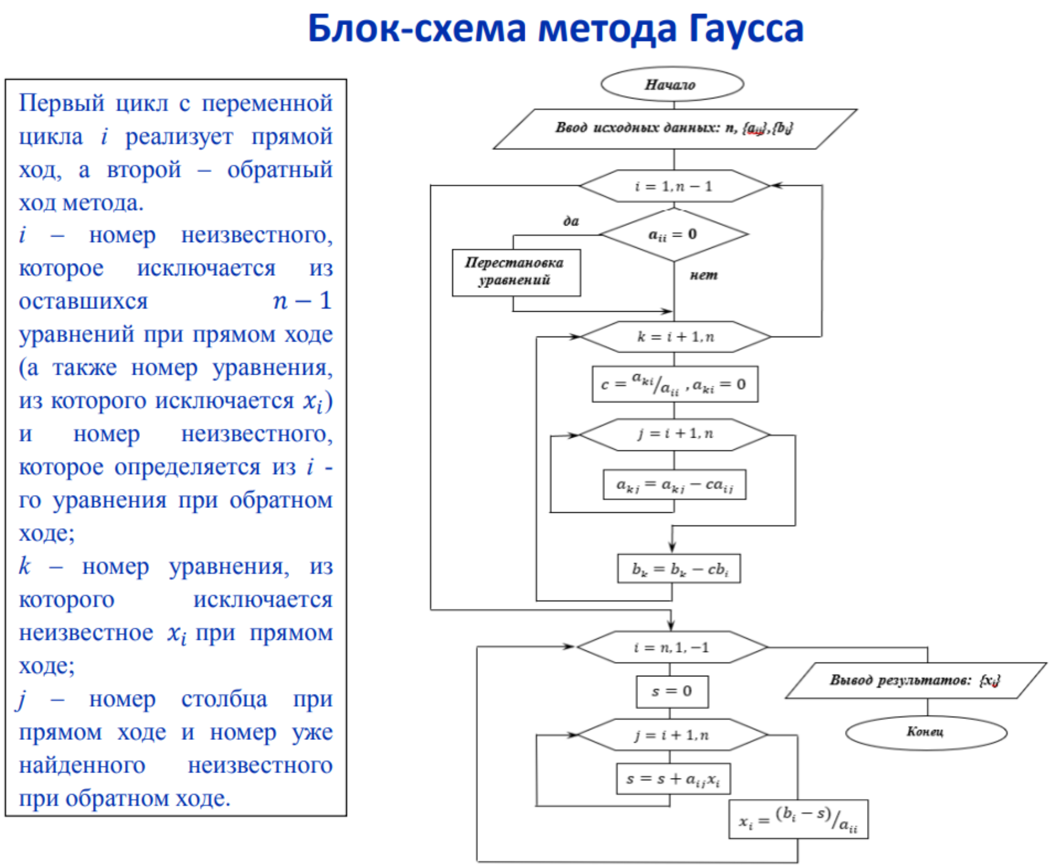
x3 = 1,00

Вектор невязки:

0.0 0.0 0.0

Process finished with exit code 0

Блок-схема метода:



Вывод: в ходе работы мне удалось изучить прямые и итерационные методы решения СЛАУ, а также написать программу, реализующую метод Гаусса.