

Университет ИТМО

Вычислительная математика
Лабораторная работа №1
«Решение системы линейных алгебраических уравнений СЛАУ»

Работу выполнил:
Бавыкин Роман
Группа: Р3210
Вариант 2

Санкт-Петербург
2022 г.

Цель работы:

Разработать программу для решения СЛАУ методом Гаусса с выбором главного элемента по столбцам.

Задание лабораторной работы:

- № варианта определяется как номер в списке группы согласно ИСУ.
- В программе численный метод должен быть реализован в виде отдельной подпрограммы или класса, в который входные/выходные данные передаются в качестве параметров.
- Размерность матрицы $n \leq 20$ (задаётся из файла или с клавиатуры - по выбору конечного пользователя).
- Должна быть реализована возможность ввода коэффициентов матрицы, как с клавиатуры, так и из файла (по выбору конечного пользователя).

Для прямых методов должно быть реализовано:

- Вычисление определителя
- Вывод треугольной матрицы (включая преобразованный столбец В)
- Вывод вектора неизвестных: x_1, x_2, \dots, x_n
- Вывод вектора невязок: r_1, r, \dots, r_n

Описание метода, расчётные формулы:

Схема с выбором главного элемента является одной из модификаций метода Гаусса.

Среди ведущих элементов могут оказаться очень маленькие по абсолютной величине. При делении на такие ведущие элементы получается большая погрешность округления.

Идеей метода Гаусса с выбором главного элемента является такая перестановка уравнений, чтобы на k -ом шаге исключения ведущим элементом a_{ii} оказывался наибольший по модулю элемент k -го столбца.

Т.е. на очередном шаге k в уравнениях, начиная от k до последнего ($i=k, k+1, \dots, n$) в столбце k выбирают максимальный по модулю элемент и строки i и k меняются местами. Это выбор главного элемента по столбцу.

Для исключения x_i из уравнения из уравнений от $i+1$ до n -ого (k), умножим i -ое уравнение на

$\left(-\frac{a_{ki}}{a_{ii}} \right)$ и прибавим к k -ому уравнению.

$$x_n = \frac{b_n^{(n-1)}}{a_{nn}^{(n-1)}}$$

$$\dots x_2 = \frac{1}{a_{22}^{(1)}} (b_2^{(1)} - a_{23}^{(1)} x_3 - \dots - a_{2n}^{(1)} x_n)$$

$$x_1 = \frac{1}{a_{11}} (b_1 - a_{12} x_2 - a_{13} x_3 - \dots - a_{1n} x_n)$$

Листинг программы:

```
1 using comp_math_1.exceptions;
2 using comp_math_1.io;
3 using comp_math_1.matrix;
4 using comp_math_1.util;
5
6 namespace comp_math_1.lineq;
7
8 public static class GaussianMethod
9 {
10
11     public static Pair<DoubleMatrix> SolveEquation(DoubleMatrix a, DoubleMatrix b)
12     {
13         double det = a.GetDeterminant();
14         Console.WriteLine("Determinant = " + det);
15         if (det == 0)
16         {
17             throw new ZeroDeterminantException();
18         }
19
20         DoubleMatrix a1 = (DoubleMatrix) a.Clone();
21         DoubleMatrix b1 = (DoubleMatrix) b.Clone();
22
23         for (int i = 0; i < a.getN(); i++)
24         {
25             ChoiceMainElem(a, b, i);
26             SubAllRows(a, b, i);
27         }
28         ConsoleOutput.WriteEquation(a, b);
29         DoubleMatrix x = new DoubleMatrix(a.getN(), 1);
30         for (int i = a.getN() - 1; i >= 0; i--)
31         {
32             double s = 0;
33             for (int j = i + 1; j < a.getM(); j++)
34             {
35                 s += a.GetElement(i, j) * x.GetElement(j, 0);
36             }
37             x.SetElement((b.GetElement(i, 0) - s) / a.GetElement(i, i), i, 0);
38         }
39
40         DoubleMatrix r = new DoubleMatrix(a.getN(), 1);
41         for (int i = 0; i < a1.getN(); i++)
42         {
43             double s = 0;
44             for (int j = 0; j < a1.getM(); j++)
45             {
46                 s += a1.GetElement(i, j) * x.GetElement(j, 0);
47             }
48             r.SetElement(s - b1.GetElement(i, 0), i, 0);
49         }
50         return new Pair<DoubleMatrix>(x, r);
51     }
52
53     private static void ChoiceMainElem(DoubleMatrix a, DoubleMatrix b, int i)
54     {
```

```

55     double max = Math.Abs(a.GetElement(i, i));
56     int maxIndex = i;
57     for (int j = i + 1; j < a.getN(); j++)
58     {
59         if (Math.Abs(a.GetElement(j, i)) > max)
60         {
61             max = Math.Abs(a.GetElement(j, i));
62             maxIndex = j;
63         }
64     }
65     a.SwapRows(i, maxIndex);
66     b.SwapRows(i, maxIndex);
67 }
68
69 private static void SubRow(DoubleMatrix a, DoubleMatrix b, int i1, int i2)
70 {
71     double c = a.GetElement(i2, i1) / a.GetElement(i1, i1);
72     a.SetElement(0, i2, i1);
73     for (int j = i1 + 1; j < a.getM(); j++)
74     {
75         a.SetElement(a.GetElement(i2, j) - c * a.GetElement(i1, j), i2, j);
76     }
77     b.SetElement(b.GetElement(i2, 0) - c * b.GetElement(i1, 0), i2, 0);
78 }
79
80 private static void SubAllRows(DoubleMatrix a, DoubleMatrix b, int i)
81 {
82     for (int j = i + 1; j < a.getN(); j++)
83     {
84         SubRow(a, b, i, j);
85     }
86 }
87 }

```

Примеры и результаты работы программы:

1)
Data from file (F) or from keyboard (K)?
f
Input file path: /home/robq/repositories/comp_math_1/test.txt
Determinant = -150.04999999999998
10 -7 0 | 7
0 2.5 5 | 2.5
0 0 6.002 | 6.001999999999999
Vector of unknowns (X):
2.6645352591003756E-16
-0.9999999999999997
0.9999999999999999
Vector of incompatibilities (R):
0
-8.881784197001252E-16
0
Матрица в файле:
3
10 -7 0 7
-3 2.099 6 3.901
5 -1 5 6

2)
Data from file (F) or from keyboard (K)?
k
n = 2
1 1 1
1 1 1
Determinant = 0
Determinant is zero => no resolves or infinity resolves

Выводы:

Во время выполнения лабораторной работы я узнал про разные численные методы решения СЛАУ, и разработал программу, реализующую метод Гаусса с выбором главного элемента по столбцу.