

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ
НАПРАВЛЕНИЕ СИСТЕМНОГО И ПРИКЛАДНОГО ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

курса «Вычислительная математика»

**по теме: «Решение системы линейных алгебраических уравнений
СЛАУ»**

Вариант № 2 (№ 22 в списке группы)

Выполнил студент:

Тюрин Иван Николаевич

группа: Р32131

Преподаватель:

Малышева Т. А.,

Бострикова Д. К.

Санкт-Петербург, 2023 г.

Содержание

Лабораторная работа № 1. Решение системы линейных алгебраических уравнений СЛАУ	2
1. Задание варианта № 2 (№ 22 в списке группы)	2
2. Цель работы	2
3. Описание метода, расчетные формулы	2
1. Листинг программы	3
4. Примеры и результаты работы программы	5
1. Листинг тестов	5
2. Листинг результатов работы тестов	6
5. Вывод	7

Лабораторная работа № 1

Решение системы линейных алгебраических уравнений СЛАУ

1. Задание варианта № 2 (№ 22 в списке группы)

, , ,
Метод Гаусса с выбором главного элемента по столбцам
, , ,

2. Цель работы

Изучить способы численных методов решения системы линейных алгебраических уравнений и реализовать один из них.

3. Описание метода, расчетные формулы

Метод Гаусса с выбором главного элемента по столбцам — прямой метод решения СЛАУ. Метод состоит из двух этапов:

1. подготовка матрицы («прямой ход»),
2. вычисление вектора переменных («обратный ход»).

Во время первого этапа происходит приведение данной квадратной матрицы к треугольному виду с помощью последовательного исключения переменных из всех нижележащих строчек. Математически шаг с исключением переменной x_i из уравнений начиная с $i + 1$ можно описать, как

$$a'_{kj} = a_{kj} - \frac{a_{ki}}{a_{ii}} a_{kj}, \quad \forall i, j \in [1..n], k \in [(i + 1)..n]. \quad (1.1)$$

При этом модификация *выбора главного элемента по столбцам* заключается в том, что перед началом исключения мы, если это нужно, переставляем две строки в матрице так, чтобы главный элемент в текущей строке был как можно большим по абсолютному значению, т.е. $a_{ii} \geq a_{ji} \forall i \in [1..n], j \in [(i+1)..n]$. Это действие позволяет повысить точность вычислений на компьютере, т.к. таким образом мы стараемся избавиться от погрешности от деления на близкое к нулю число. В добавок к этому, посчитав количество перестановок строчек, мы можем установить знак определителя нашей изначальной матрицы (его значение, с точностью до знака, мы можем вычислить перемножив элементы на диагонали полученной треугольной матрицы), а именно

$$\det A = (-1)^k \prod_{i=1}^n a_{ii}, \quad (1.2)$$

где k — число перестановок строк (или столбцов) матрицы при ее приведении к треугольному виду в соответствии с модификацией.

3.1. Листинг программы

Основную часть программной реализации на языке программирования Kotlin можно посмотреть в листинге 1.1. Весь код представлен в личном репозитории [1].

```

1 fun Double.approx(double: Double): Boolean {
2     return (this - double).absoluteValue < 1e-6
3 }
4
5 /**
6  * extension function on matrix to solve system of linear equations
7  * described by this and vector 'b'.
8  */
9 @OptIn(ExperimentalStdlibApi::class)
10 fun Matrix.solveSLE(b: DoubleArray, logMiddleResults: Boolean = false):
11     DoubleArray? {
12     val dim = elements.size // matrix must be square unless we can slice
13     square matrix from it
14     assert(elements.all { dim == it.size }) { "Matrix dimensions mismatch
15     (not square matrix)"}
16     assert(b.size == dim) { "Matrix and vector dimensions mismatch (must
17     have same number of rows)" }
18
19     val xs = DoubleArray(dim) { 0.0 }
20     val tmp = elements.copy().toMutableMatrix()
21
22     var nRowSwaps = 0;
23
24     // Forward
25     for (x in 0..<dim - 1) {
26         if (tmp.mutateMatrixWithVector(vector = b, start = x)) {
27             nRowSwaps++
28         }
29     }
30 }

```

```

24
25     for (nextRow in x + 1..

```

```

82         if (elements[i][start].absoluteValue > elements[greatest][start].
absoluteValue) {
83             greatest = i
84             mutated = true
85         }
86     }
87     if (mutated) {
88         elements[start] = elements[greatest].also { elements[greatest] =
elements[start] }
89         vector[start] = vector[greatest].also { vector[greatest] = vector[
start] }
90     }
91     return mutated
92 }

```

Листинг 1.1: Реализация на языке программирования Kotlin основной логики решения СЛАУ

4. Примеры и результаты работы программы

В утилите реализована возможность ввода данных через файл специального формата данных, заполнение матрицы случайными числами и режим интерактивного ввода.

4.1. Листинг тестов

Для проверки работоспособности программы были написаны тесты, их содержимое представлено в листинге 1.2.

```

1
2 class Test {
3     @Test
4     fun 'input data from file'() {
5         println("##### input data from file #####")
6         main(arrayOf("--file", "D:\\Projects\\itmo-comp-math\\lab-1\\src\\
test\\resources\\file-input-test.txt"))
7     }
8
9     @Test
10    fun 'input data by random dim=3'() {
11        println("##### input data by random #####")
12        main(arrayOf("--random", "3"))
13    }
14
15    @Test
16    fun 'print documentation'(){
17        println("##### print documentation #####")
18        main(arrayOf("--help"))
19    }
20 }

```

Листинг 1.2: Реализация на языке программирования Kotlin тестов утилиты для решения СЛАУ

4.2. Листинг результатов работы тестов

Соответственно вывод результатов тестов представлен в листинге 1.3. Текстовый файл использовавшийся при тестировании так же можно найти среди ресурсов для тестов в личном репозитории [1].

```
1 ##### input data from file #####
2
3 Triangle matrix:
4 [[10.0, -7.0, 0.0],
5  [0.0, 2.5, 5.0],
6  [0.0, 0.0, 6.002]]
7
8 Modified b:
9 [7.0,
10  2.5,
11  6.001999999999999]
12
13 Det(Matrix)=-150.04999999999998
14
15 Result of calculating vector 'x':
16 [2.6645352591003756E-16,
17  -0.9999999999999997,
18  0.9999999999999999]
19
20 Vector of deviance 'r':
21 [0.0,
22  -1.4009999999999999,
23  0.0019999999999988916]
24 ##### print documentation #####
25
26 DESCRIPTION
27     Utility for solving System of Liner Algebraic Equations
28
29     NOTE: floating point is comma (",")
30
31 SYNOPSIS
32     solve [FLAG [PARAM]]
33
34 FLAG
35     -h, --help          Prints help doc of utility;
36
37     --file [path]       Receives 'path' to the input file, works in not
38     interactive mode;
39
40     --random [n]         Receives number of matrix dimensions 'n';
41
42 ##### input data by random #####
43 #-----#
44 # Your are welcome in System of Linear Equations solver. #
45 #-----#
46 Please input your matrix dimension:
47 ( 0 < n < 21 )
48 n= 3
49 Please, input your (n x n) values of matrix by rows of n elements
50 separated with single space:
51
52 Thank you for your matrix:
53 [[0.47605838251574817, 0.8113645483712928, 0.26142577878135476],
54  [0.19243939663868048, 0.30967754101767, 0.12289604698407952],
```

```

53 [0.6040888727423883, 0.3408073537056102, 0.6314133409219941]]
54
55 Next step is input values of your system's right-hand part in one row
    separated with single space:
56
57 Thank you for your vector:
58 [0.14767079427067065,
59  0.43526862948970035,
60  0.4361731660679654]
61
62 Triangle matrix:
63 [[0.6040888727423883, 0.3408073537056102, 0.6314133409219941],
64  [0.0, 0.5427878457115936, -0.23616592889834392],
65  [0.0, 0.0, 0.009254467679650977]]
66
67 Modified b:
68 [0.4361731660679654,
69  -0.19605990712835591,
70  0.36896326648887656]
71
72 Det(Matrix)=0.003034466822004519
73
74 Result of calculating vector 'x':
75 [-50.532697350685176,
76  16.985568739373555,
77  39.868664439788894]
78
79 Vector of deviance 'r':
80 [0.28850237179728977,
81  -0.6313285366180561,
82  -0.06720989957909007]

```

Листинг 1.3: Вывод с тандартный поток тестов утилиты для решения СЛАУ

5. Вывод

В ходе выполнения данной лабораторной работы углубили понимание работы методов решения СЛАУ, реализовали на языке Kotlin требуемую утилиту для их решения.

Литература

- [1] Ссылка на личный репозиторий GitHub: <https://github.com/elturin/itmo-comp-math/tree/main/lab-1>