

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	4
1. Листинг тестов	4
2. Листинг результатов работы тестов	5
5. Вывод	6

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,

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

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 Matrix.solveSLE(b: DoubleArray, logMiddleResults: Boolean = false):
  DoubleArray? {
2     val dim = elements.size
3     assert(elements.all { dim == it.size })
4     assert(b.size == dim)
5
6     val xs = DoubleArray(dim) { 0.0 }
7     val tmp = elements.copy().toMutableMatrix()
8     var nMutations = 0;
9
10    for (x in 0..<dim - 1) {
11        if (tmp.mutateMatrixWithVector(vector = b, start = x)) {
12            nMutations++
13        }
14        for (nextRow in x + 1..<dim) {
15            val mul = -tmp.elements[nextRow][x] / tmp.elements[x][x]
16            for (i in 0..<dim) {
17                tmp.elements[nextRow][i] += tmp.elements[x][i] * mul
18            }
19            b[nextRow] += b[x] * mul
20            tmp.elements[nextRow][x] = 0.0
21        }
22    }
23
24    var det: Double = if (nMutations % 2 == 0) {
25        1.0
26    } else {
27        -1.0
  
```

```

28     } * tmp.elements.foldIndexed(1.0) { idx, acc, d ->
29         acc * d[idx]
30     }
31     if (det.approx(0.0)) {
32         return null
33     }
34
35     for (x in xs.indices.reversed()) {
36         var tmpSum: Double = 0.0
37         for (i in x + 1..xs.lastIndex) {
38             tmpSum += tmp.elements[x][i] * xs[i]
39         }
40         xs[x] = (b[x] - tmpSum) / tmp.elements[x][x]
41     }
42     return xs
43 }
44
45 private fun MutableMatrix.mutateMatrixWithVector(vector: DoubleArray,
46     start: Int): Boolean {
47     var isMutated = false
48     var greatest: Int = start
49     for (i in start..

```

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 class Test {
2     @Test
3     fun 'input file path'() {

```

```

4      main(arrayOf("--file", "D:\\Projects\\itmo-comp-math\\lab-1\\src\\
test\\resources\\file-input-test.txt"))
5  }
6
7  @Test
8  fun 'random input'() {
9      main(arrayOf("--random", "3"))
10 }
11
12 @Test
13 fun 'docs print'(){
14     main(arrayOf("--help"))
15 }
16 }

```

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 ##### random input #####
2 #-----#
3 # You are welcome in System of Linear Equations solver. #
4 #-----#
5 Please input your matrix dimension:
6 ( 1 ? n ? 20 )
7   n=
8 Please, input your (n?n) values of matrix by rows of n elements separated
   with single space:
9 NOTE. floating point is dot ","
10
11 Thank you for your matrix.
12 Input your values of right the system right part in one row separated with
   single space:
13
14 Result of calculating vector 'x':
15 [-3.0277951234457885,
16   6.397433026161394,
17   -4.518432532993963]
18
19 Vector of deviance 'r':
20 [-0.7342930611502309,
21   0.5561207721540378,
22   -0.22871575190109894]
23
24 ##### docs print #####
25
26 DESCRIPTION
27     Utility for solving System of Liner Algebraic Equations
28
29 SYNOPSIS
30     solve [FLAG [PARAM]]
31

```

```

32 FLAG
33     -h, --help          Prints help doc of utility;
34
35     --file [path]       Receives 'path' to the input file, works in not
36                           interactive mode;
37
38     --random [n]        Receives number of matrix dimensions 'n';
39
40 ##### input file path #####
41 Result of calculating vector 'x':
42 [2.6645352591003756E-16,
43  -0.9999999999999999,
44   0.9999999999999999]
45
46 Vector of deviance 'r':
47 [0.0,
48  -1.4009999999999999,
49   0.00199999999999988916]

```

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/e1turin/itmo-comp-math/tree/dev-lab-1/lab-1>