

НИУ ИТМО

Факультет программной инженерии и компьютерной техники

## Лабораторная работа №1

по дисциплине

«Вычислительная математика».

Решение системы линейных алгебраических уравнений СЛАУ.

Вариант 8

Выполнил: студент группы Р3214  
Крюков Андрей

Преподаватель:  
Малышева Татьяна Алексеевна

Санкт-Петербург  
2021 г.









<b>Задание</b>	<b>3</b>
<b>Вычисление варианта</b>	<b>3</b>
<b>Цель работы</b>	<b>4</b>
<b>Описание метода</b>	<b>4</b>
<b>Листинг</b>	<b>5</b>
<b>Блок-схема</b>	<b>7</b>
<b>Примеры выполнения программы</b>	<b>8</b>
<b>Вывод</b>	<b>9</b>

## З а д а н и е

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

## В ы ч и с л е н и е в а р и а н т а

### Список группы 22.02.2021

№ п/п	№ таб.		Ф.И.О.
1	284726		Алексеев Михаил Николаевич
2	283809		Антонов Максим Александрович
3	286470		Бабалин Юрий Алексеевич
4	284365		Гумеров Алан Ленартович
5	284728		Дзюбак Антон Михайлович
6	285550		Ильинская Ольга Вадимовна
7	283945		Инячина Диана Александровна
8	284729		Крюков Андрей Юрьевич

Метод Гаусса с выбором главного элемента по строкам

## Цель работы

Реализовать инструмент для решения произвольных СЛАУ. Имплементировать метод Гаусса с выбором главного элемента по строкам и вычисление невязок.

## Описание метода

Идея метода Гаусса с выбором главного элемента по строкам заключается в поиске максимального числа в очередной строке и вычитании этой строки из последующих таким образом, чтобы те числа в столбце с найденным максимальным числом, которые находятся ниже него, становились равными нулю. Каждую итерацию мы меняем местами столбец с найденным максимальным числом и столбец с номером, равным номеру очередной строки. При этом мы должны записывать, какие перестановки мы совершили, чтобы в дальнейшем восстановить исходный вид матрицы и правильно сопоставить столбцы и неизвестные.

После этих преобразований, мы получаем треугольную матрицу. Здесь начинается вторая часть алгоритма, которая называется обратный ход. Мы проделываем обратные перестановки, ставя столбцы на их изначальные места. После этого, мы проходимся по строкам снизу вверх, записывая найденные неизвестные в тар и используя их для нахождения других неизвестных. Если после завершения этого цикла в тар меньше переменных, чем строк - решение не может быть найдено однозначно

Невязка - разница между левой и правой частью уравнения при подстановке полученных значений неизвестных в него.

Формула для вычисления определителя после приведения к треугольному виду

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

## Листинг

```
def calc_discrepancy(A, B, answers):
    discrepancy = []
    for i in range(len(A)):
        s = 0
        for j in range(len(A[i])):
            s += A[i][j] * answers[j]
        discrepancy.append(B[i] - s)
    return discrepancy

def find_main_elem(matrix, row):
    max_elem = -1
    max_j = -1
    for j in range(row, len(matrix[row])):
        if max_elem < abs(matrix[row][j]):
            max_elem = abs(matrix[row][j])
            max_j = j
    return max_j

def calc_determinant(A, count_swaps):
    s = 1
    for i in range(len(A)):
        s *= A[i][i]
    return s*(-1)**count_swaps

def swap(matrix, first, col):
    for i in range(len(matrix)):
        matrix[i][first], matrix[i][col] = matrix[i][col],
matrix[i][first]

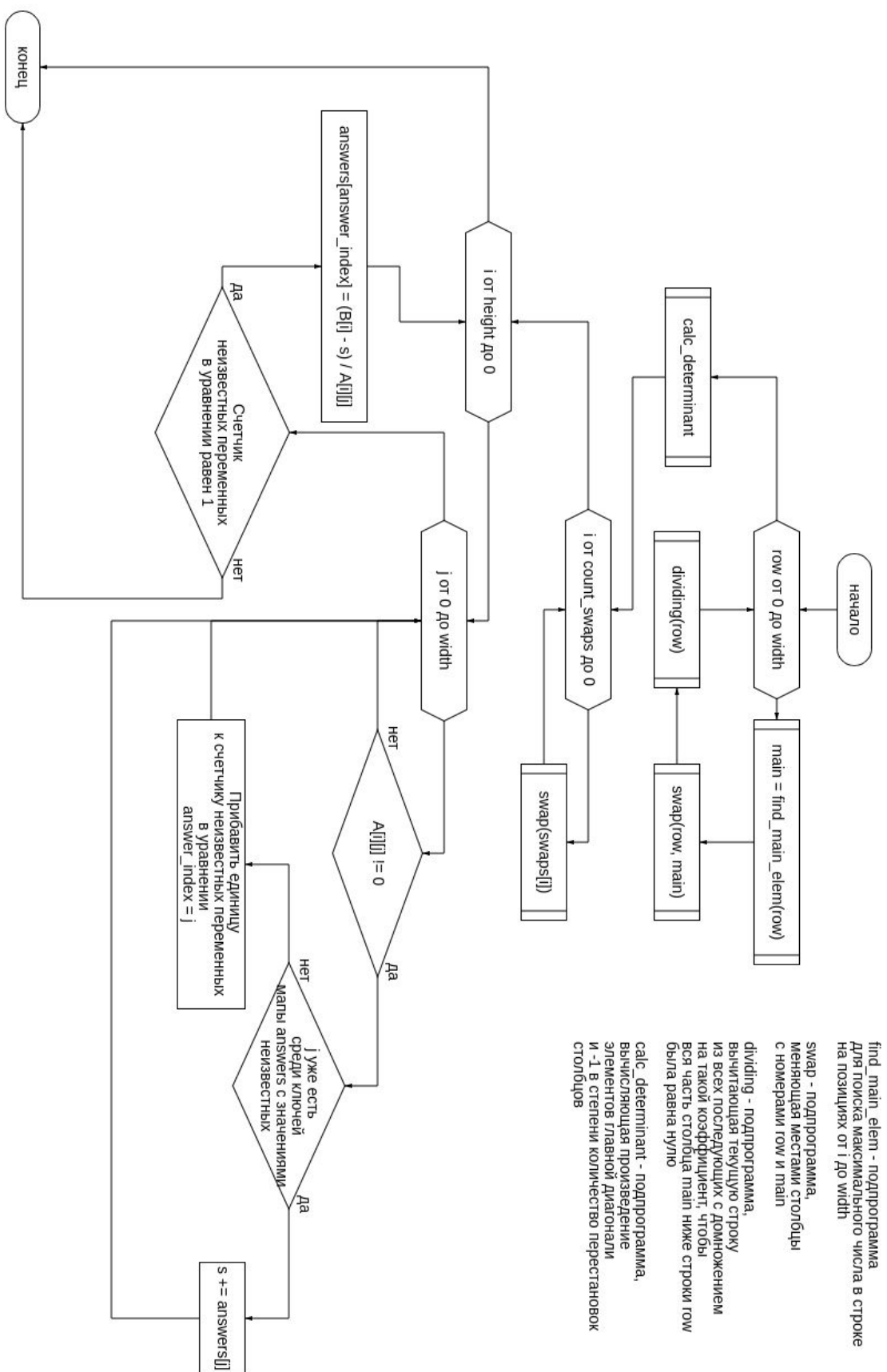
def dividing(A, B, row):
    for i in range(row, len(B)):
        if row != i:
            B[i] -= A[i][row] / A[row][row] * B[row]
    for i in range(row, len(A)):
        mm = A[i][row] / A[row][row]
        for j in range(row, len(A[i])):
            if row != i:
                A[i][j] -= mm * A[row][j]
```

```

def gauss(A, B):
    A, B = deepcopy(A), deepcopy(B)
    row = 0
    swaps = []
    while row < len(A[0]):
        main_elem = find_main_elem(A, row)
        swap(A, row, main_elem)
        swaps.append((row, main_elem))
        if A[row][row] != 0:
            dividing(A, B, row)
        row += 1
    det = calc_determinant(A, len(swaps))
    print()
    print(f"Определитель: {det}")
    print()
    print("Треугольная матрица: ")
    print_matrix(A, B)
    swaps.reverse()
    for q, w in swaps:
        swap(A, q, w)
    answers = {}
    for i in range(len(A) - 1, -1, -1):
        s = 0
        dont_know = []
        for j in range(len(A[i])):
            if abs(A[i][j]) > 10**-9:
                if j in answers:
                    s += answers[j]
                else:
                    dont_know.append(j)
        if len(dont_know) == 1:
            answers[dont_know[0]] = (B[i] - s) / A[i][dont_know[0]]
    return answers

```

## Блок-схема



# Примеры выполнения программы

```
$ python main.py input.txt
```

Треугольная матрица:

```
-----  
0 0 0 0  
0 0 0 0  
0 0 0 0  
-----
```

Cant solve. Probably, count of solutions != 1

```
$ python main.py
```

Введите количество переменных: 3

2 2 4 7

6 1 3 5

3 8 2 1

Треугольная матрица:

```
-----  
4 2 2 7  
0.0 4.5 -0.5 -0.25  
0.0 0.0 7.222222222222222 -2.388888888888889  
-----
```

Вектор неизвестных:

0.017948717948717947 -0.33076923076923076 1.8282051282051281

Вектор невязок:

0.3128205128205135 -0.2615384615384615 -0.06410256410256387

```
$ cat input.txt
```

2 2 4 7 4 5 6

6 1 3 5 2 4 5

3 8 2 1 1 1 4

2 3 6 4 5 2 1

9 9 8 4 3 6 9

2 4 8 2 7 6 5

```
$ python main.py input.txt
```

Треугольная матрица:

```
-----  
7 2 2 4 5 4 6  
0.0 4.571428571428571 -0.4285714285714286 0.1428571428571428 0.4285714285714284  
-0.8571428571428572 0.7142857142857144  
0.0 -4.440892098500626e-16 7.96875 1.34375 0.03125000000000017 0.9375000000000001 2.71875  
0.0 0.0 0.0 3.3607843137254902 -0.9450980392156859 2.6470588235294117 -3.223529411764706  
0.0 0.0 0.0 0.0 3.50291715285881 -1.9894982497082843 5.267211201866978  
0.0 2.220446049250313e-16 0.0 0.0 0.0 4.317788141239173 -1.473017988007994  
-----
```

Вектор неизвестных:

0.09580055134233845 0.3504803104257236 -1.3340436988458952 0.8039798731338461  
-0.341151056935658 1.6010547820765684

Вектор невязок:

-1.824915722729676 -1.6649689020462564 1.51295966198423 3.0489463871032925 3.857026820666783  
5.252596206202094



## Вывод

В ходе данной лабораторной работы был имплементирован метод Гаусса с выбором главного элемента по строкам, вычисление определителя и вычисление невязок в программе для решения СЛАУ, которая поддерживает ввод коэффициентов как из стандартного ввода, так и из файла.