

# **Национальный исследовательский университет информационных технологий, механики и оптики**

---

**Факультет Программной Инженерии и Компьютерной Техники**

---

## **Лабораторная работа 1. «Решение системы линейных алгебраических уравнений СЛАУ»**

---

**Вариант № 21**

**«Вычислительная математика»**

Работу выполнила:

Студентка группы Р3212

Сенина Мария Михайловна

Преподаватель:

Малышева Татьяна Алексеевна

Санкт-Петербург

2022

## Цель работы

---

Вычисление решения СЛАУ методом Гаусса.

## Задание

---

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 входные/выходные данные передаются в качестве параметров.
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).

## Описание метода, расчетные формулы

---

**Определитель** матрицы  $n, m$  можно почсчитать по формуле

$\sum_{k=1}^n (-1)^{k+1} a_{k1} M_{k1}$ , где  $M_{k1}$  – определитель минора элемента  $a_{k1}$ . Т.е. С помощью рекурсивной программы можно вычислить определитель всей матрицы раскладывая матрицу и её миноры по первой строке.

**Решение СЛАУ** можно вычислить методом Гаусса:

Сначала прямым ходом по порядку от первой строки расширенной матрицы до последней, для каждой  $i$ ой строки необходимо:

1. Нормировать строку на элемент  $a_{i,i}$
2. Для остальных строк с индексом  $j$  больше  $i$  нужно вычесть строку  $j$  умноженную на строку с номером  $i$

Таким образом мы приведём матрицу к треугольному виду.

Далее необходимо обратным ходом привести её к диагональному виду. В цикле по всем строкам для каждой строки с номером  $i > 1$ :

1. Для каждой строки с номерами  $j$  меньше  $i$  нужно вычесть эту строку домноженную на строку с номером  $i$ .

## Листинг программы

---

[Ссылка репозиторий с кодом всей программы](#)

Метод вычисления определителя:

```
def count_matrix_det(matrix):  
  
    det = 0  
  
    for i in range(len(matrix[0])):  
  
        if(len(matrix) > 1):  
  
            minor = [line[:i] + line[(i + 1):] for line in matrix[1:]]  
  
            det += (-1)**(i)*matrix[0][i]*count_matrix_det(minor)  
  
        elif (len(matrix) == 1):  
  
            return matrix[0][0]  
  
    else:
```

```
print("Error: can't count det of matrix with dim less then 1.")

return det
```

Метод решающий СЛАУ методом Гаусса:

```
def count_result(matrix):

    n = len(matrix)

    matrix = triangulize_matrix(matrix)

    for i in range(1, n):

        for j_before_i in range(0, i):

            i_head = matrix[j_before_i][i]

            for k in range(i, n + 1):

                matrix[j_before_i][k] = matrix[j_before_i][k] - matrix[i][k]*i_head

    return [round(matrix[i][n], 3) for i in range(0, len(matrix))]
```

Метод триангулирующий матрицу:

```
def triangulize_matrix(matrix):

    n = len(matrix)

    for i in range(0, n):
```

```
if(matrix[i][i] == 0):

    matrix = swap_lines(matrix, i)

    i_i = matrix[i][i]

    for j_in_i_line in range(i, n + 1):

        matrix[i][j_in_i_line] = matrix[i][j_in_i_line]/i_i

    for j_after_i in range(i + 1, n):

        j_line_head = matrix[j_after_i][i]

        for k in range(i, n + 1):

            matrix[j_after_i][k] = matrix[j_after_i][k] - j_line_head*matrix[i][k]

    return matrix
```

Метод вычисления вектора невязок:

```
def count_residual_vector(matrix, solution):

    return [sum([solution[i]*line[i] for i in range(len(line) - 1)]) - line[len
```

## Примеры и результаты работы программы

---

### Пример 1

Ввод:

y

input.txt

Содержимое файла input.txt:

1 2 4 5 6

3 5 6 7 8

1 2 6 3 7

9 8 6 5 3

Вывод:

Matrix det:-140.0

Triangle matrix:

\_\_\_\_\_triangle matrix\_\_\_\_\_

[1.0, 2.0, 4.0, 5.0, 6.0]

[0.0, 1.0, 6.0, 8.0, 10.0]

[0.0, 0.0, 1.0, -1.0, 0.5]

[0.0, 0.0, 0.0, 1.0, 0.486]

-----

Matrix solution:[-0.771, 0.2, 0.986, 0.486]

Residual vector:[0.000429, -0.0, 0.000286, 0.000286]

## Пример 2

Ввод:

n

0 2 4 5 6

3 5 6 0 8

1 0 6 0 7

9 8 6 5 3 EOF

Вывод:

Matrix det:-940.0

Triangle matrix:

-----triangle matrix-----

[1.0, 1.667, 2.0, 0.0, 2.667]

[0.0, 1.0, 2.0, 2.5, 3.0]

[0.0, 0.0, 1.0, 0.568, 1.273]

[0.0, 0.0, 0.0, 1.0, -0.119]

-----

Matrix solution:  $[-1.043, 0.617, 1.34, -0.119]$

Residual vector:  $[-2.1e-05, -0.000447, -0.000426, 0.000149]$

## Выводы

---

В этой лабораторной работе я научилась решать СЛАУ методом Гаусса, написала ПО на языке python для этого. Разобралась в том, от чего зависят погрешности вычисления и узнала о новых методах подсчёта решения СЛАУ.