

**Федеральное государственное автономное образовательное учреждение высшего
образования «Национальный исследовательский университет ИТМО»**

Факультет программной инженерии и компьютерной техники

Лабораторная работа №1

По Вычислительной математике

Вариант №11

Выполнил:

Ступин Тимур Русланович

Группа № Р3208

Поток № 1.3

Преподаватель:

Машина Екатерина Алексеевна

Санкт-Петербург 2025

Содержание

Цель	3
Описание метода	3
Листинг программы	4
Пример работы программы	5
Вывод	5

Цель

Изучить прямые и численные методы решения систем линейных уравнений и реализовать один из них средствами программирования.

Описание метода

Метод Гаусса основан на приведении матрицы системы к треугольному виду так, чтобы ниже ее главной диагонали находились только нулевые элементы. Выделяют прямой и обратных ход в методе Гаусса.

Прямой ход состоит в последовательном исключении неизвестных из уравнений системы. Сначала с помощью первого уравнения исключается x_1 из всех последующих уравнений системы. Затем с помощью второго уравнения исключается x_2 из третьего и всех последующих уравнений и т.д. Этот процесс продолжается до тех пор, пока в левой части последнего (n-го) уравнения не останется лишь один член с неизвестным x_n , т. е. матрица системы будет приведена к треугольному виду.

Обратный ход состоит в последовательном вычислении искомых неизвестных: решая последнее уравнение, находим неизвестное x_n . Далее, из предыдущего уравнения вычисляем x_{n-1} и т. д. Последним найдем x_1 из первого уравнения.

Листинг программы

```
def gauss_direct_route(a, b):
    n = len(a)
    for i in range(n - 1):
        if a[i][i] == 0:
            l = i
            for m in range(i+1, n):
                if a[m][i] != 0:
                    l = m
                    break
            if l != i:
                for j in range(i, n):
                    a[i][j], a[l][j] = a[l][j], a[i][j]
                b[i], b[l] = b[l], b[i]
            else:
                continue
        for k in range(i + 1, n):
            c = a[k][i] / a[i][i]
            a[k][i] = 0
            for j in range(i + 1, n):
                a[k][j] = a[k][j] - a[i][j] * c
            b[k] = b[k] - b[i] * c

def gauss_reverse_route(a, b):
    n = len(a)
    x = [0 for _ in range(n)]
    for i in range(n - 1, -1, -1):
        s = 0
        for j in range(i + 1, n):
            s = s + a[i][j] * x[j]
        if a[i][i] == 0:
            if b[i] != 0:
                exit_with_error("СЛАУ не имеет решений!")
            else:
                exit_with_error("СЛАУ имеет бесконечно много решений!")
        x[i] = (b[i] - s) / a[i][i]
    return x
```

Полный код: <https://github.com/timur1516/computational-math/blob/main/lab1/main.py>

Пример работы программы

```
Решение СЛАУ методом Гаусса
Выберите способ ввода:
1 - ввод в консоли
2 - ввод из файла
1
Выбран ввод в консоли
Введите n: 3
Введите матрицу построчно (и А и В):
4 5 2 1
3 -3 2 2
2 -3 1 3
Считанная система:
4.0x1 + 5.0x2 + 2.0x3 = 1.0
3.0x1 + -3.0x2 + 2.0x3 = 2.0
2.0x1 + -3.0x2 + 1.0x3 = 3.0
Вычисленный определитель матрицы: 11.00
Определитель матрицы через numpy: 11.00
Треугольная матрица:
4.00    5.00    2.00    1.00
0.00   -6.75    0.50    1.25
0.00    0.00   -0.41    1.48
Решение системы методом Гаусса:
x1 = 2.64
x2 = -0.45
x3 = -3.64

Решение системы через numpy:
x1 = 2.64
x2 = -0.45
x3 = -3.64

Невязка:
r1 = -0.00
r2 = -0.00
r3 = 0.00
```

Вывод

В ходе выполнения данной работы я познакомился с прямыми и численными методами решения систем линейных уравнений, реализовал метод Гаусса на языке python и проверил корректность его работы используя невязки а также сравнение с результатами полученными с помощью библиотеки numpy.