

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

Факультет Программной Инженерии и Компьютерной Техники

**Лабораторная работа №1**  
**по дисциплине «Вычислительная математика»**  
**Решение системы линейных алгебраических уравнений СЛАУ**

Вариант: 4

Преподаватель:  
Машина Екатерина Алексеевна

Выполнил:  
Есоян Владимир Саркисович  
Группа: Р3208

Санкт-Петербург, 2025 г

## Цель работы

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

## Описание метода

Итерационные методы позволяют для системы построить последовательность векторов  $x^{(0)}$ ,  $x^{(1)}$ , ...,  $x^{(k)}$ , пределом которой должно быть точное решение  $x^{(*)}$ :  $x^{(*)} = \lim_{k \rightarrow \infty} x^{(k)}$

Построение последовательности заканчивается, как только достигается желаемая точность.

Приведем систему уравнений, выразив неизвестные  $x_1, x_2, \dots, x_n$  соответственно из первого, второго и т.д. уравнений системы.

## Листинг программы

<https://github.com/x-oc/slac-solutions>

```
import sys

from diagonal_dominance import make_matrix_diagonal_dominance
from user_interaction import get_matrix, get_accuracy,
print_matrix, print_x_values, print_errors

LIMIT = 5000
PRECISION = 6

def converges(vars_old, vars_new):
    return sum((vars_new[i] - vars_old[i]) ** 2 for i in
range(len(vars_old))) ** 0.5 < accuracy

def iteration(coefficients, vars, values):
    n = len(coefficients)
    vars_new = [0.0] * n
    for i in range(n):
        s = sum(coefficients[i][j] * vars[j] for j in range(n) if
i != j)
        vars_new[i] = (values[i] - s) / coefficients[i][i]
    return vars_new

def solve(coefs, values, precision=PRECISION):
    n = len(coefs)
    x = [0 for _ in range(n)]
    for i in range(1, LIMIT + 1):
        x_new = iteration(coefs, x, values)
        if converges(x, x_new):
            print('Решение:')
            print_x_values(x_new, precision)
            print('Итераций:', i)
            print('Погрешности:')
            print_errors(x, x_new, precision)
            break
    x = x_new
```

```

else:
    print("Видимо метод расходится...")

if __name__ == '__main__':
    accuracy = get_accuracy()
    matrix = get_matrix()
    coefs = []
    values = []
    for row in matrix:
        coefs.append(row[:-1])
        values.append(row[-1])
    if not make_matrix_diagonal_dominance(coefs, values):
        print("Матрица не может быть подготовлена для выполнения алгоритма")
        sys.exit()
    for i in range(len(matrix)):
        matrix[i] = coefs[i] + [values[i]]
    print("Преобразованная матрица:")
    print_matrix(matrix)
    solve(coefs, values)

```

## Примеры и результаты работы программы

Введите точность: 0.001

Откуда будет вводиться матрица? 1 - консоль, 2 - файл 3

Откуда будет вводиться матрица? 1 - консоль, 2 - файл 2

Введите название файла: null

Файл некорректен

Введите название файла: matrix\_3.txt

Преобразованная матрица:

|       |       |      |      |
|-------|-------|------|------|
| 3.00  | -2.00 | 1.00 | 6.00 |
| 1.00  | 3.00  | 2.00 | 2.00 |
| -1.00 | 2.00  | 4.00 | 9.00 |

Решение:

0.121874 -1.341735 2.951236

Норма матрицы (максимум сумм модулей строк): 1.0

Итераций: 16

Погрешности:

0.000238 0.000319 0.00031

**Вывод**

В результате выполнения данной лабораторной работы я познакомился с численными методами решения математических задач на примере систем алгебраических уравнений, реализовав на языке программирования Python метод простых итераций.