

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

Факультет Программной Инженерии и Компьютерной Техники

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

Вариант: 9

Преподаватель:
Наумова Надежда Александровна

Выполнил:
Нестеров Владислав Алексеевич
Группа: Р3210

Санкт-Петербург, 2025 г

Цель работы

Ознакомиться с численными методами решения систем линейных алгебраических уравнений и разработать программу, реализующую один из таких методов.

Описание метода

Метод простых итераций используется для решения системы уравнений $Ax = b$, где A — квадратная матрица, x — вектор неизвестных, b — вектор свободных членов.

Система представляется в виде:

$$x^{k+1} = Bx^k + g$$

где B и g вычисляются из преобразованной системы $Ax = b$, обеспечивая сходимость метода.

Итерационный процесс продолжается до выполнения условия:

$$\|x^{k+1} - x^k\| < \varepsilon$$

где ε — заданная точность

Листинг программы

https://github.com/urasha/Slae_Iteration_Method — полный код проекта

solver.py:

```
def simple_iteration_method(matrix, vector, epsilon=1e-6, max_iterations=1000):
    size = len(matrix)
    current_solution = [0.0] * size

    iteration_matrix = []
    constant_terms = []

    for row_index in range(size):
        diagonal_element = matrix[row_index][row_index]
        transformed_row = [(-matrix[row_index][col_index] / diagonal_element) if row_index != col_index
else 0.0
        for col_index in range(size)]
        iteration_matrix.append(transformed_row)
        constant_terms.append(vector[row_index] / diagonal_element)

    iteration_count = 0

    error_vector = []
    while iteration_count < max_iterations:
        previous_solution = current_solution[:]
        new_solution = [
            sum(iteration_matrix[row_index][col_index] * previous_solution[col_index] for col_index in
range(size)) +
            constant_terms[row_index]
            for row_index in range(size)
        ]

        error_vector = [abs(new_solution[i] - previous_solution[i]) for i in range(size)]

        if max(error_vector) < epsilon:
            current_solution = new_solution
            break

        current_solution = new_solution
        iteration_count += 1

    matrix_norm = max(sum(abs(value) for value in row) for row in matrix)

    return current_solution, iteration_count, matrix_norm, error_vector
```

Примеры работы программы:

Выберите способ ввода данных:

1 - Ввести вручную

2 - Считать из файла

Ваш выбор (1 или 2): чыф

Ошибка ввода. Введите целое число.

Ваш выбор (1 или 2): 1

Введите размер матрицы (целое число от 1 до 20): 2

Введите 2 коэффициентов строки 1 через пробел: 2 2

Введите 2 коэффициентов строки 2 через пробел: 1 1

Введите свободные члены через пробел: 3 3

Решение: [-750.0, 750.0]

Количество итераций: 1000

Норма матрицы: 4.0

Вектор погрешностей: [3.0, 1.5]

Вывод:

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