

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
    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
- 3 - Сгенерировать случайную матрицу

Ваш выбор (1, 2 или 3): 3

Введите размер случайной матрицы (от 1 до 20): 3

Сгенерированная матрица:

20.96 -9.70 7.24

-8.76 19.20 -8.47

-3.07 5.02 10.24

Сгенерированный вектор свободных членов: 8.60 -2.60 7.76

Введите точность (например, 1e-6): ds

Ошибка ввода. Введите число.

Введите точность (например, 1e-6): 1e-10

Решение: [0.31575680979492093, 0.3162558903992179,
0.6975948614184524]

Количество итераций: 39

Норма матрицы: 37.90878291397181

Вектор погрешностей: [8.623868286150582e-11,
9.046507987164887e-11, 3.8020919745918036e-11]

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.