Санкт-Петербургский национальный
исследовательский университет ИТМО

Факультет программной инженерии и компьютерной техники
Направление подготовки 09.03.04 «Программная инженерия»
Дисциплина «Вычислительная математика»

Отчет
По лабораторной работе №2
«Метод Гаусса-Зейделя»

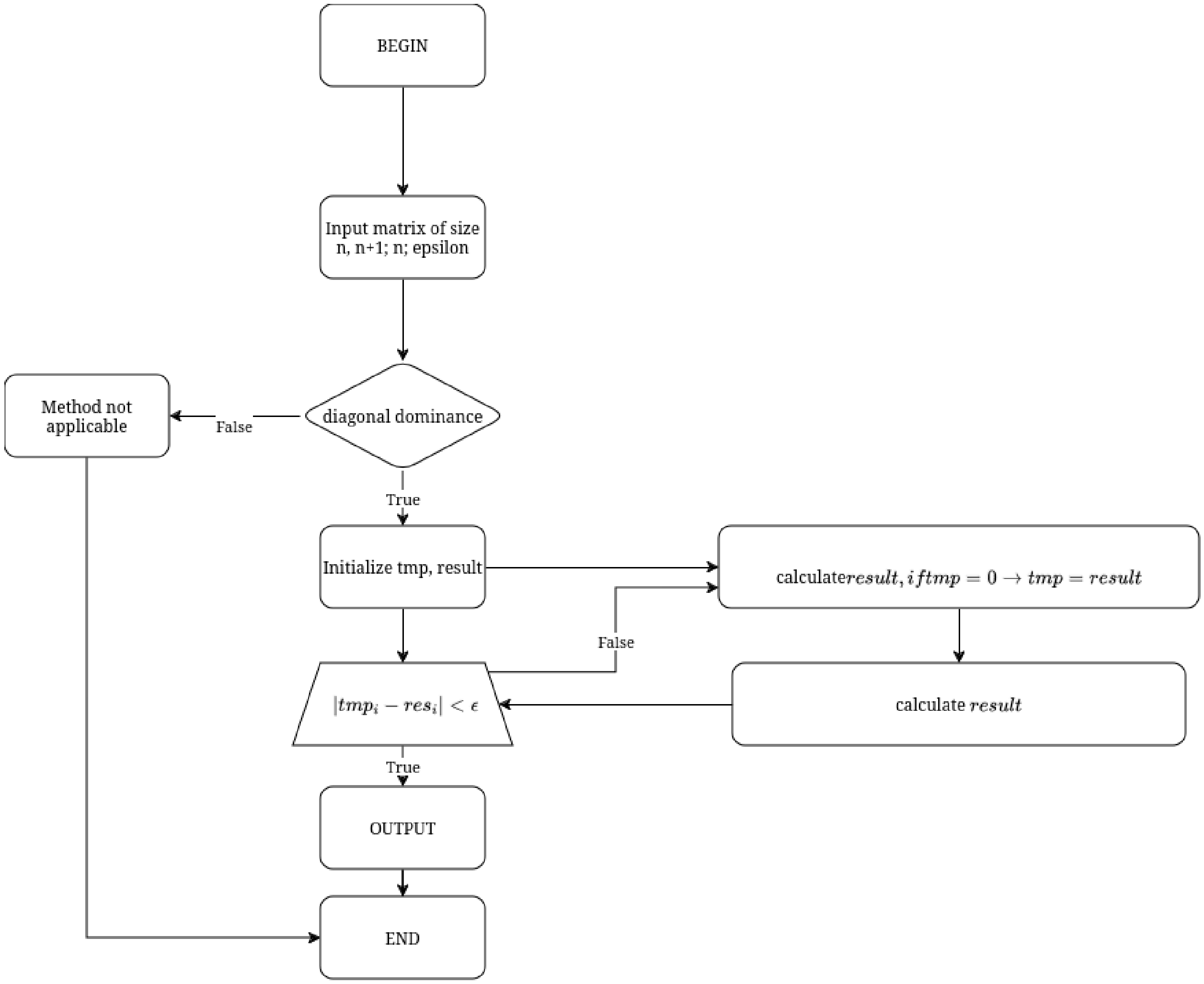
Выполнил студент:
Ведерников А.В.
P3223
Преподаватель:
Перл О.В.

Санкт-Петербург
2024

Описание численного метода:
Метод Гаусса-Зейделя - это итерационный метод решения системы линейных алгебраических уравнений. Он является модификацией метода простой итерации (метода Якоби) и имеет более быструю сходимость.

Для того чтобы метод был применим должно соблюдаться условие диагонального преобладания.

Блок-схема численного метода:



Метод реализованный на языке Python:

```
class Result:
    isMethodApplicable = True
    errorMessage = ""

    def is_applicable(n, matrix, epsilon):
        for i in range(n):
            sum_of_abs = sum(abs(matrix[i][j]) for j in range(n) if i != j)
            if abs(matrix[i][i]) <= sum_of_abs:
                Result.isMethodApplicable = False
                Result.errorMessage = "The system has no diagonal dominance for this method. Method of the Gauss-Seidel is not app
                return False
        return True


    def solveByGaussSeidel(n, matrix, epsilon):
        if not Result.is_applicable(n, matrix, epsilon):
            return []

        tmp = [0.0] * n

        while True:
            result = [0.0] * n
            for i in range(n):
                result[i] = (matrix[i][n] - sum(matrix[i][j] * tmp[j] for j in range(n) if i != j)) / matrix[i][i]
            if all(abs(result[i] - tmp[i]) < epsilon for i in range(n)):
                return result

            tmp = result
```

Тесты:

```
def test_is_applicable_and_solveByGaussSeidel():
    matrix = [[10, 1, 2, 13], [3, 8, 4, 27], [5, 6, 9, 38]]
    n = 3
    epsilon = 0.01
    result = Result.solveByGaussSeidel(n, matrix, epsilon)
    assert Result.isMethodApplicable == True
    assert abs(result[0] - 1) < epsilon
    assert abs(result[1] - 2) < epsilon
    assert abs(result[2] - 3) < epsilon

    matrix = [[-10, -1, -2, -13], [-3, -8, -4, -27], [-5, -6, -9, -38]]
    n = 3
    epsilon = 0.01
    result = Result.solveByGaussSeidel(n, matrix, epsilon)
    assert Result.isMethodApplicable == True
    assert abs(result[0] - 1) < epsilon
    assert abs(result[1] - 2) < epsilon
    assert abs(result[2] - 3) < epsilon

    matrix = [[1, 2, 3, 4], [4, 5, 6, 7], [7, 8, 9, 10]]
    n = 3
    epsilon = 0.01
    Result.solveByGaussSeidel(n, matrix, epsilon)
    assert Result.isMethodApplicable == False
    assert Result.errorMessage == "The system has no diagonal dominance for this method. Method of the Gauss-Seidel is not applicable."

    matrix = [[0, 2, 3, 4], [4, 0, 6, 7], [7, 8, 0, 10]]
    n = 3
    epsilon = 0.01
    Result.solveByGaussSeidel(n, matrix, epsilon)
    assert Result.isMethodApplicable == False
    assert Result.errorMessage == "The system has no diagonal dominance for this method. Method of the Gauss-Seidel is not applicable."

    matrix = [[1e6, 1e5, 1e4, 1e6 + 1e5 + 1e4], [1e5, 1e6, 1e5, 1e6 + 1e5 + 1e5], [1e4, 1e5, 1e6, 1e6 + 1e5 + 1e4]]
    n = 3
    epsilon = 0.01
    result = Result.solveByGaussSeidel(n, matrix, epsilon)
    assert Result.isMethodApplicable == True
    assert abs(result[0] - 1) < epsilon
    assert abs(result[1] - 1) < epsilon
    assert abs(result[2] - 1) < epsilon
```

Вывод

В ходе выполнения лабораторной работы я изучил метод Гаусса-Зейделя для решения систем линейных уравнений, и узнал о его плюсах минусах и ограничениях.

Сравнение с методом простой итерации:

- Метод Гаусса-Зейделя имеет более быструю сходимость, чем метод простой итерации, потому что он использует более новые значения переменных на каждой итерации.
- Однако, метод Гаусса-Зейделя требует большего количества вычислений на каждой итерации, чем метод простой итерации, что может увеличить время выполнения для больших систем уравнений.

Сравнение с методом сопряженных градиентов:

- Метод Гаусса-Зейделя может сходиться медленнее, чем метод сопряженных градиентов, особенно для систем уравнений с большим числом переменных.
- Однако, метод Гаусса-Зейделя обычно проще в реализации и требует меньше памяти, чем метод сопряженных градиентов.