НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
Факультет Программной инженерии и компьютерных технологий
Направление: Нейротехнологии и программная инженерия

Дисциплина: Вычислительная математика

Лабораторная работа № 2

“Метод Гаусса-Зейделя”

Выполнил студент
Рязанов Демид Витальевич
Группа P3221
Преподаватель: Перл Ольга Вячеславовна

г. Санкт-Петербург
2024

Содержание

Описание метода.....3
Блок-схема.....4
Исходный код.....6
Примеры работы.....7
Вывод.....9

Описание метода

Метод Гаусса-Зейделя – итерационный алгоритм для поиска решений СЛАУ. На каждой итерации метода получается более точное приближение решения, и при достижении определенной точности алгоритм завершает свою работу. В отличии от метода простых итераций для вычисления новых значений приближения используются значения с текущей итерации.

Алгоритм:

1) Сначала проверяем условие применимости метода ($|a_{ii}| \geq \Sigma_{i \neq k} |a_{ik}| (i, k = 1,2, \dots, n)$) функция **isDDM** (DDM – diagonal dominance matrix)

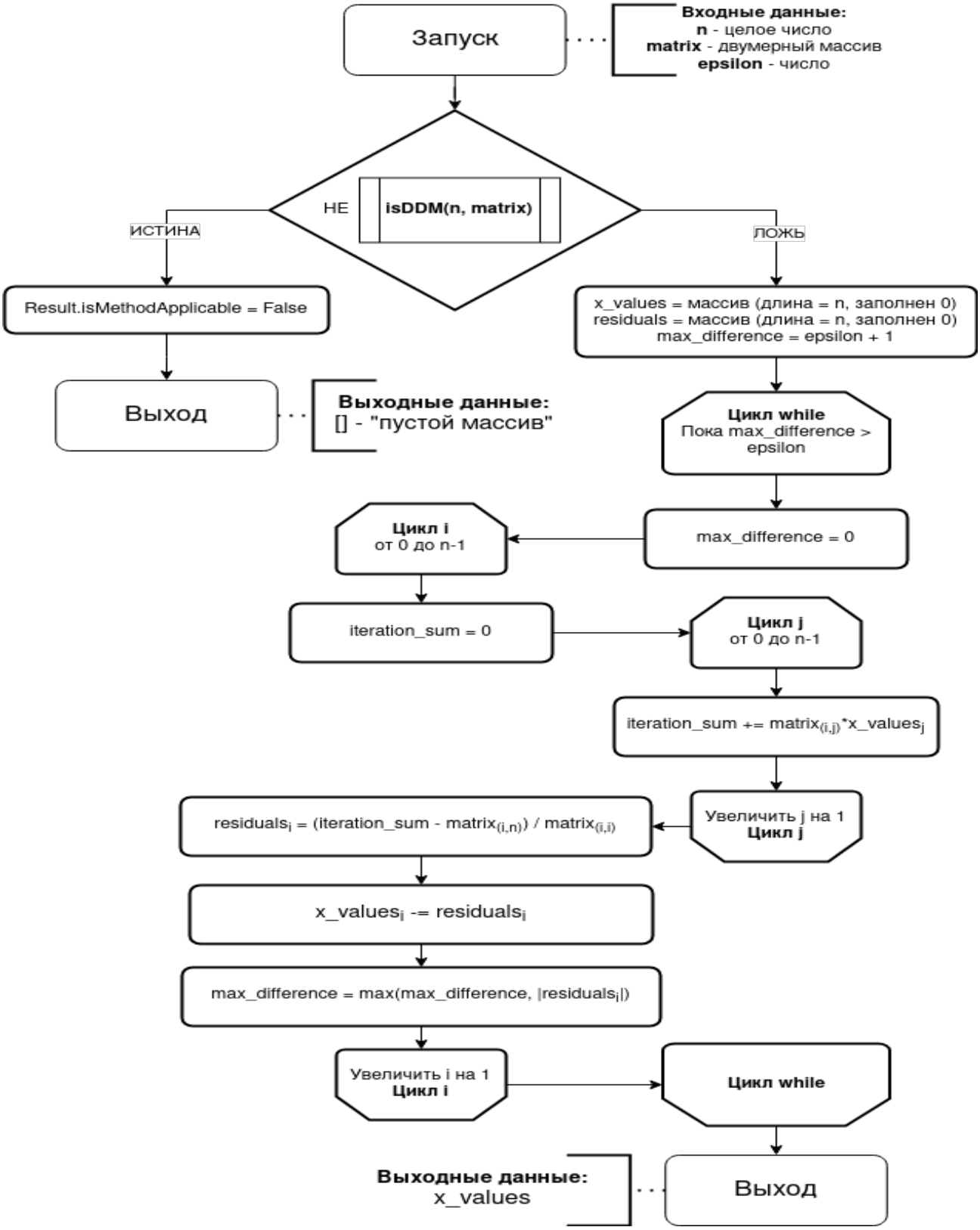
2) Определяем начальное приближение $\vec{x}^{(0)}$ (его можно брать произвольно, в нашем случае берем нули) и **max_difference = epsilon + 1**, чтобы начать итерационный процесс

3) Для x_i высчитываем **residuals** разницу между $x_i^{(k)}$ и $x_i^{(k+1)}$: $residuals_i = \frac{\Sigma_{j=0}^{n-1} (x_j * matrix_{i,j}) - matrix_{i,n}}{matrix_{i,i}}$, где **matrix** – расширенная матрица СЛАУ.

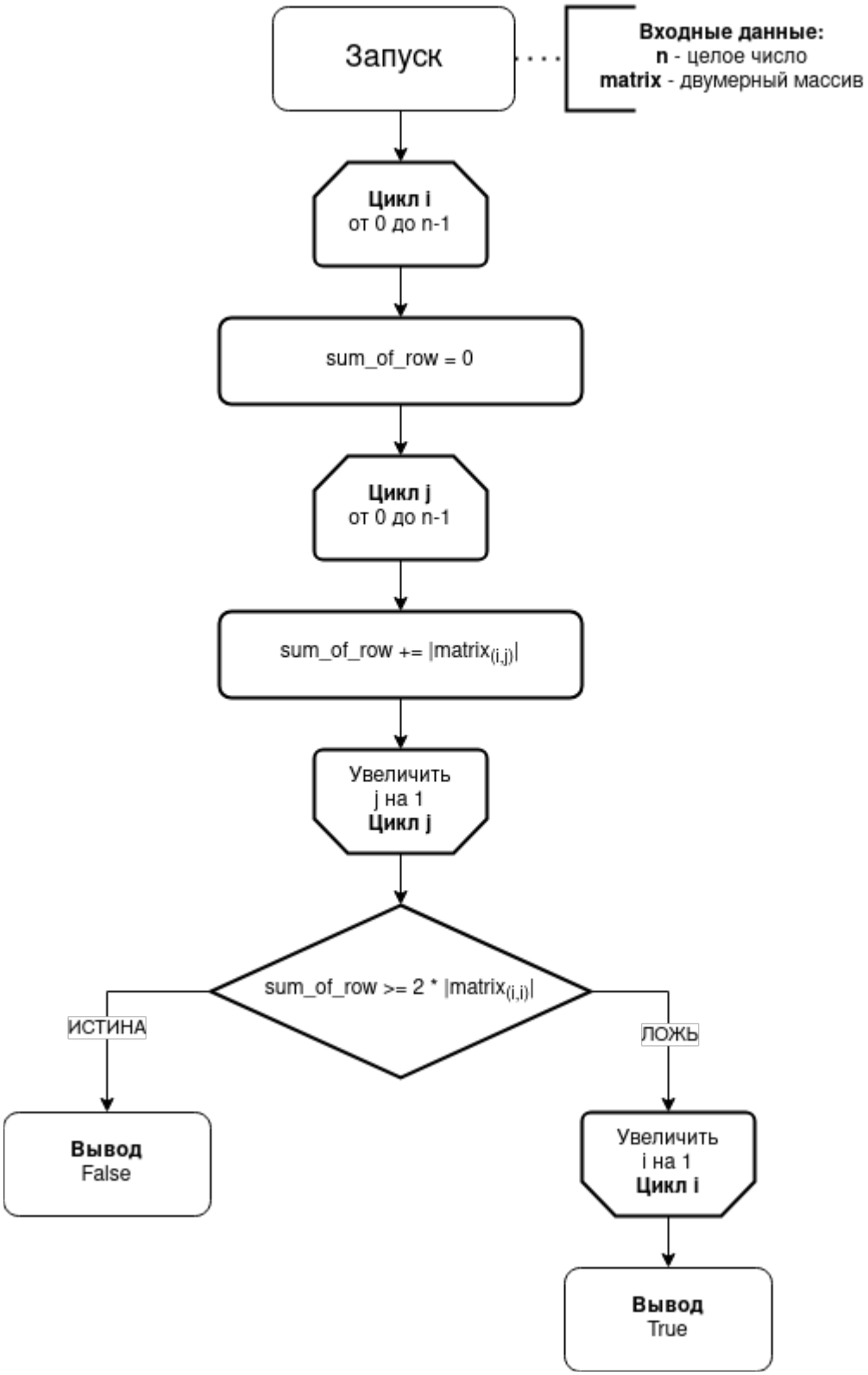
С помощью **residuals** сразу меняем значение x_i и обновляем **max_difference**. Выполняем так для каждого x_i , $i = 0, .., n-1$

4) Выполняем пункт 3, пока **max_difference** > **epslion**, ответом будет массив значений **x**

Блок-схема


Запуск
Входные данные:
n - целое число
matrix - двумерный массив
epsilon - число
НЕ
isDDM(n, matrix)
ИСТИНА
ЛОЖЬ
Result.isMethodApplicable = False
Выход
Выходные данные:
[] - "пустой массив"
x_values = массив (длина = n, заполнен 0)
residuals = массив (длина = n, заполнен 0)
max_difference = epsilon + 1
Цикл while
Пока max_difference > epsilon
max_difference = 0
Цикл i
от 0 до n-1
iteration_sum = 0
Цикл j
от 0 до n-1
iteration_sum += matrix(i,j)*x_valuesj
Увеличить j на 1
Цикл j
residualsi = (iteration_sum - matrix(i,n)) / matrix(i,i)
x_valuesi -= residualsi
max_difference = max(max_difference, |residualsi|)
Увеличить i на 1
Цикл i
Цикл while
Выходные данные:
x_values
Выход

Функция **isDDM**



Исходный код

```
class Result:
    isMethodApplicable = True
    errorMessage = "The system has no diagonal dominance for this method. Method of the Gauss-Seidel is not applicable."

    def solveByGaussSeidel(n, matrix, epsilon):
        try:
            if not Result.isDDM(n, matrix):
                Result.isMethodApplicable = False
                return []
            x_values = [0 for _ in range(n)]
            residuals = [0 for _ in range(n)]
            max_difference = epsilon + 1
            while max_difference > epsilon:
                max_difference = 0
                for i in range(n):
                    iteration_sum = 0
                    for j in range(n):
                        iteration_sum += matrix[i][j] * x_values[j]
                    residuals[i] = (iteration_sum - matrix[i][n]) / matrix[i][i]
                    x_values[i] -= residuals[i]
                    max_difference = max(max_difference, abs(residuals[i]))
            return x_values
        except:
            Result.isMethodApplicable = False
            return []

    def isDDM(n, matrix):
        for i in range(n):
            sum_of_abs_row_elements = 0
            for j in range(n):
                sum_of_abs_row_elements = sum_of_abs_row_elements + abs(matrix[i][j])
            if sum_of_abs_row_elements >= 2 * abs(matrix[i][i]):
                return False
        return True
```

Примеры работы

Пример 1

Ввод	Вывод
3 8 4 2 10 3 5 1 5 3 -2 10 4 0.001	1.037880831316406 0.3457776268257813 0.157791275970234 43

Пример 2

Ввод	Вывод
3 1 2 2 3 1 2 1 3 2 3 5 1 0.0001	The system has no diagonal dominance for this method. Method of the Gauss-Seidel is not applicable.

Пример 3

Ввод	Вывод
3 4 3 1 3 3 5 2 2 2 5 7 10 0.001	The system has no diagonal dominance for this method. Method of the Gauss-Seidel is not applicable.

Пример 4

Ввод	Вывод
4	0.450100513214754 97
10 1 1 2 3	0.266771161961815 83
2 23 12 0 2	-0.419690934761474 1
5 4 15 3 -5	-0.674076979904923
3 3 2 42 -27	
0.0001	

Пример 5

Ввод	Вывод
2	2.888888888888889
3 1 9	0.3333333333333333
0 12 4	
0.00001	

Вывод

Точность работы алгоритма зависит от **epsilon**, это хорошо ведь его можно настраивать по мере необходимости.

Метод Гаусса-Зейделя отличается от метода простых итераций тем, что для вычисления значений текущей итерации используются уже полученные на этой итерации значения. Из-за этого скорость сходимости метода может быть быстрее, но при этом его сложнее параллелизовать. Также у этого метода условие сходимости строже, чем у метода простых итераций.

Алгоритм применим только для СЛАУ, матрицы которых имеют диагональное преобладание.

Сложность алгоритма $O(k*n^2)$, где k – количество итераций метода