Университет ИТМО

Факультет программной инженерии и компьютерной техники

**Лабораторная работа №1**

по «Вычислительной математике»

Решение системы линейных алгебраических уравнений СЛАУ

Выполнил:

Студент группы P32151

Кортыш А.О.

Преподаватель:

Малышева Т.А.

Санкт-Петербург

2023

## Цель работы.

Составление программы для решение СЛАУ при помощи метода простой итерации.

## Описание метода, расчетные формулы.

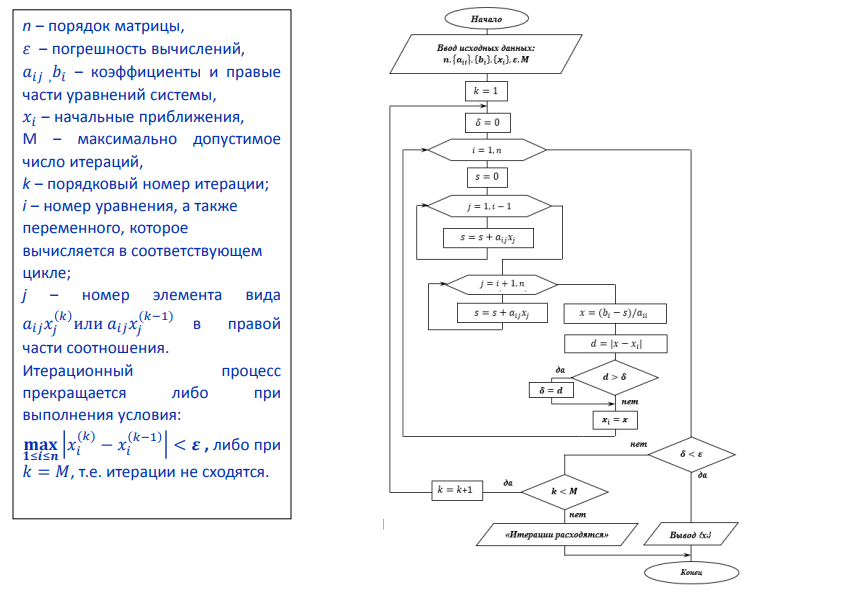
Суть методы состоит в построении последовательности векторов решений , которая стремится к точному решению.

1. Для этого сначала приведем матрицу коэффициентов A в исходной системе Ax = B к матрице с диагональным преобладанием.
2. Далее приведем систему к виду x = Cx + D,
3. Тогда рабочая формула метода простой итерации:
4. За нулевое приближение можно взять либо нулевой вектор , либо вектор свободных коэффициентов .

## Листинг программы.

def solution(dim, matrix, precision):  
 A = [[matrix[j][i] for i in range(dim)] for j in range(dim)]  
 B = [matrix[i][-1] for i in range(dim)]  
 A = get\_diagonal\_dominance(A)  
 if not check\_diagonal\_dominance(A):  
 raise Exception  
 coefficients = [[0 if i == j else -A[i][j] / A[i][i] for j in range(dim)] for i in range(dim)]  
 free\_coefficients = [B[i] / A[i][i] for i in range(dim)]  
 x = free\_coefficients.copy()  
 cur\_error = 1e9  
 iterations = 0  
 cur\_error\_vector = []  
 while cur\_error > precision:  
 new\_x = [free\_coefficients[i] + sum(coefficients[i][j]\*x[j] for j in range(dim)) for i in range(dim)]  
 cur\_error\_vector = get\_error\_vector(x, new\_x)  
 cur\_error = max(cur\_error\_vector)  
 x = new\_x  
 iterations += 1  
 return iterations, x, cur\_error\_vector

## Блок схема



## Примеры и результат работы программы

### Пример 1.

Ввод из файла matrix2.txt:

Содержимое файла:

4

7 1 1 1 2

1 8 1 1 3

1 1 9 1 4

1 1 1 10 5

Работа программы:

Введите 0 для ввода с клавиатуры или 1 для ввода из файла: 1

Введите название файла: matrix2.txt

Всего итераций 6

Вектор неизвестных: 0.137610749415717 0.2608218663065172 0.3532284205847078 0.42509891415804457

Вектор погрешностей: 0.0009672045740222457 0.0008786369227039015 0.0008043363906613532 0.0007414736459233628

### Пример 2.

Ввод с клавиатуры:

Введите 0 для ввода с клавиатуры или 1 для ввода из файла: 0

Введите размерность: 3

Введите матрицу:

8 1 -4 6

2 -6 1 -9

-1 1 4 5

Введите точность: 0.001

Всего итераций 8

Вектор неизвестных: 0.9999911932297695 2.0000841117199557 1.000160546950352

Вектор погрешностей: 0.0003122871304735986 0.0004228191611208576 0.00026283735110432804

### Пример 3.

Ввод некорректных данных:

Введите 0 для ввода с клавиатуры или 1 для ввода из файла: 0

Введите размерность: 3

Введите матрицу:

asd

Неверные данные.

## Выводы.

Реализация на компьютере итерационных методов очень проста и не требует большого количества времени или кода. С другой стороны задача преобразования матрицы к виду с диагональным преобразованием, если выйти за рамки перестановки строк/столбцов, скорее всего является не слишком очевидной, что может представлять основную сложность в реальных задачах. При сравнении с точными методами решения СЛАУ такими как метод Гаусса и метод Гаусса с выбором главного элемента, несомненным преимуществом итерационных методов является возможность вычислить результат с (практически) любой заданной заранее точностью. Однако по времени точные методы выигрывают у итерационных, за счет меньшего количества используемых операций (их количество в итерационных методах вообще не ограничивается ничем кроме требуемой точности).