

Факультет программной инженерии и компьютерной техники

Вычислительная математика

Лабораторная работа №1

Метод Гаусса-Зейделя

Выполнил:

Макаров Ньургун Михайлович

Группа:

P3214

Преподаватель:

Малышева Татьяна Алексеевна

Санкт-Петербург

2022 г.

*Цель работы:*

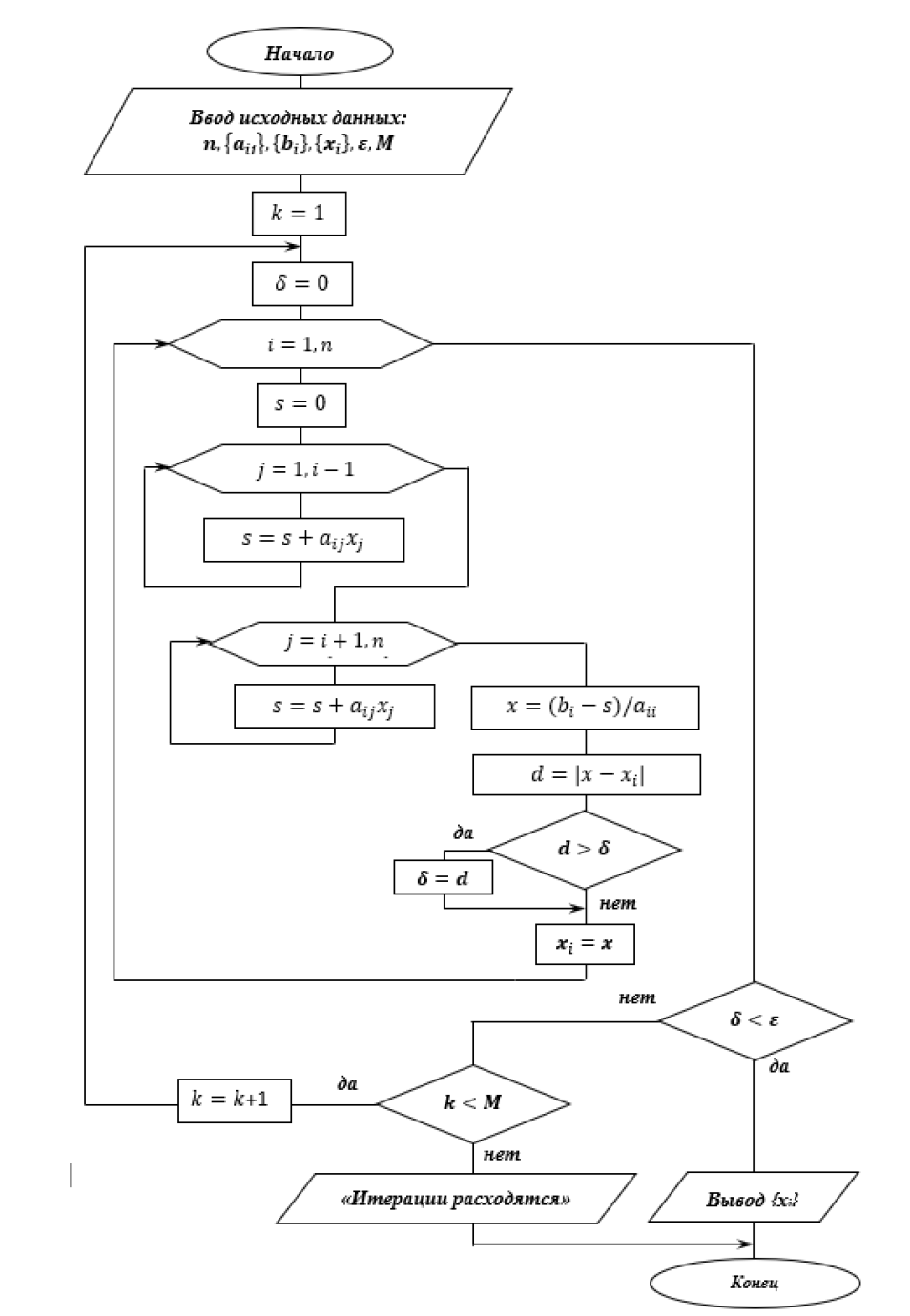
Реализовать и протестировать программу для решения системы линейных уравнений методом Гаусса-Зейделя, размерность до 20 включительно неизвестных.

*Описание метода:*

Изображение выглядит как текст

Автоматически созданное описание

*Блок – схема метода Гаусса-Зейделя:*

**

*Листинг программы (код численного метода):*

public class LinearSystemSolver {

double EPS = 0.0000012;

int approximation = 1;

public double[] findSolution(double[][] matrix) {

return findSolution(matrix, EPS);

}

public double[] findSolution(double[][] matrix, double eps) {

int size = matrix.length;

double[] previousValue = new double[size];

for (int i = 0; i < size; i++) {

previousValue[i] = 0.0;

}

if (diagonal(size, matrix)) {

while (true) {

double[] currentValue = new double[size];

for (int i = 0; i < size; i++) {

currentValue[i] = matrix[i][size];

for (int j = 0; j < size; j++) {

if (j < i)

currentValue[i] -= matrix[i][j] \* currentValue[j];

if (j > i)

currentValue[i] -= matrix[i][j] \* previousValue[j];

}

currentValue[i] /= matrix[i][i];

}

double error = 0.0;

for (int i = 0; i < size; i++) {

error = Math.abs(currentValue[i] - previousValue[i]);

}

if (error < eps)

break;

previousValue = currentValue;

approximation++;

}

return previousValue;

} else

System.out.println("Не выполняется преобладание диагоналей");

return previousValue;

}

public int getApproximation() {

return approximation;

}

private boolean diagonal(int size, double[][] matrix) {

int i, j, k = 0;

double sum;

for (i = 0; i < size; i++) {

sum = 0;

for (j = 0; j < size; j++) {

sum = sum + Math.abs(matrix[i][j]);

sum = sum - Math.abs(matrix[i][i]);

if (sum > matrix[i][i]) {

System.out.println(matrix[i][i] + " < " + sum);

} else {

k++;

System.out.println(matrix[i][i] + " > " + sum);

}

}

}

return k != size \* 3;

}

private boolean diagonalDominating(int size, double[][] matrix) {

double sum = 0;

for (int i = 0; i < size; i++) {

for (int j = 0; j < size; j++) {

if (i != j) {

sum += Math.abs(matrix[i][j]);

System.out.println(matrix[i][j] + " <>" + sum);

}

}

}

for (int i = 0; i < size; i++) {

if (Math.abs(matrix[i][i]) < sum) {

System.out.println(matrix[i][i] + "<> " + sum);

break;

}

}

return true;

}

}

Полный код можно найти по ссылке:

<https://github.com/Rvze/computional_mathematics_lab1>

*Пример*

--------GAUSS-SEIDEL METHOD--------

----------------MENU----------------

Press 1, if you want input matrix

:0

1

How do you want to enter the size?

1 - Input from keyboard

2 - Input from file

1

Input matrix size:

3

How do you want to enter the matrix values?

1 - Input from keyboard

2 - Input from file

2

Input the file name:

file.txt

Matrix:

5,00 -1,00 3,00 5,00

1,00 -4,00 2,00 20,00

2,00 -1,00 5,00 10,00

5.0 > 0.0

5.0 > -4.0

5.0 > -6.0

-4.0 < -3.0

-4.0 < -3.0

-4.0 > -5.0

5.0 > -3.0

5.0 > -7.0

5.0 > -7.0

[-0.7352950444172247, -4.485292976234234, 1.397059422520043]

[3.978291760020625E-6, 4.294440202556871E-6, 0.0]

итераций: 11

*Вывод:*

Метод Гаусса – Зейделя вычисляет за меньшее число итераций, что дает нам выигрыш по времени. Но и у него есть недостатки так как весь массив приходится хранить в оперативной памяти компьютера. Происходит накопление погрешности в процессе решения, поскольку на любом этапе используют результаты предыдущих операций.