Федеральное государственное автономное образовательное учреждение
высшего образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Отчет

по лабораторной работе №1
«Решение системы линейных алгебраических уравнений СЛАУ»

по дисциплине «Вычислительная математика»
вариант 15

Выполнил: Черноморов Кирилл, группа P3209
Преподаватель: Наумова Н. А.

Санкт-Петербург
~ 2024 ~

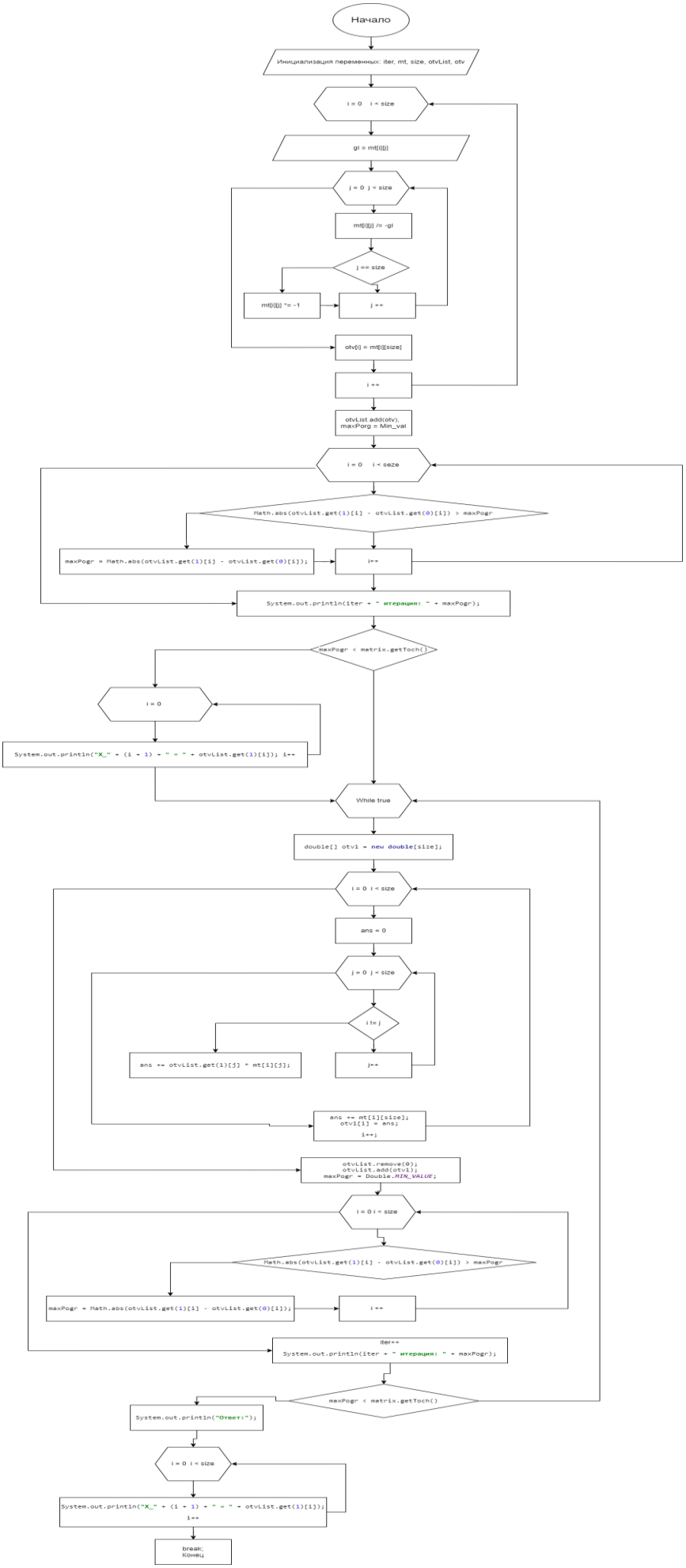
Цель работы: изучить на практике метод простых итераций и написать код для решение задач такого типа.

Задание:

1. № варианта определяется как номер в списке группы согласно ИСУ.
2. В программе численный метод должен быть реализован в виде отдельной подпрограммы/метода/класса, в который исходные/выходные данные передаются в качестве параметров.
3. Размерность матрицы n<=20 (задается из файла или с клавиатуры - по выбору конечного пользователя).
4. Должна быть реализована возможность ввода коэффициентов матрицы, как с клавиатуры, так и из файла (по выбору конечного пользователя).

• Точность задается с клавиатуры/файла
• Проверка диагонального преобладания (в случае, если диагональное преобладание в исходной матрице отсутствует, сделать перестановку строк/столбцов до тех пор, пока преобладание не будет достигнуто). В случае невозможности достижения диагонального преобладания - выводить соответствующее сообщение.
• Вывод вектора неизвестных: $x1, x2, \dots, xn$
• Вывод количества итераций, за которое было найдено решение.
• Вывод вектора погрешностей: $|xi\hat{}(k) - xi\hat{}(k-1)|$

Блок-схема реализованного алгоритма


Начало
Инициализация переменных: iter, mt, size, otvList, otv
i = 0 i < size
gl = mt[i][i]
j = 0 j < size
mt[i][j] /= -gl
j == size
mt[i][j] *= -1
j ++
otv[i] = mt[i][size]
i ++
otvList.add(otv),
maxPorg = Min_val
i = 0 i < seze
Math.abs(otvList.get(1)[i] - otvList.get(0)[i]) > maxPogr
maxPogr = Math.abs(otvList.get(1)[i] - otvList.get(0)[i]);
i++
System.out.println(iter + " итерация: " + maxPogr);
maxPogr < matrix.getToch()
i = 0
System.out.println("X_" + (i + 1) + " = " + otvList.get(1)[i]); i++
While true
double[] otv1 = new double[size];
i = 0 i < size
ans = 0
j = 0 j < size
i != j
ans += otvList.get(1)[j] * mt[i][j];
j++
ans += mt[i][size];
otv1[i] = ans;
i++;
otvList.remove(0);
otvList.add(otv1);
maxPogr = Double.MIN_VALUE;
i = 0 i < size
Math.abs(otvList.get(1)[i] - otvList.get(0)[i]) > maxPogr
maxPogr = Math.abs(otvList.get(1)[i] - otvList.get(0)[i]);
i ++
iter++
System.out.println(iter + " итерация: " + maxPogr);
maxPogr < matrix.getToch()
System.out.println("Ответ:");
i = 0 i < size
System.out.println("X_" + (i + 1) + " = " + otvList.get(1)[i]);
i++
break;
Конец

- Реализация (код) численного метода:

```
public void iter(Matrix matrix) {
    int iter = 0;
    double[][] mt = matrix.getMatrix();
    int size = matrix.getSize();
    ArrayList<double[]> otvList = new ArrayList<>();
    otvList.add(new double[size]);
    double[] otv = new double[size];
    for (int i = 0; i < size; i++) {
        double gl = mt[i][i];
        for (int j = 0; j <= size; j++) {
            mt[i][j] /= -gl;
            if (j == size) {
                mt[i][j] *= -1;
            }
        }
        otv[i] = mt[i][size];
    }
    otvList.add(otv);
    double maxPogr = Double.MIN_VALUE;
    for (int i = 0; i < size; i++) {
        if (Math.abs(otvList.get(1)[i] - otvList.get(0)[i]) > maxPogr) {
            maxPogr = Math.abs(otvList.get(1)[i] - otvList.get(0)[i]);
        }
    }
    System.out.println(iter + " итерация: " + maxPogr);
    if (maxPogr < matrix.getToch()) {
        System.out.println("Ответ:");
        for (int i = 0; i < size; i++) {
            System.out.println("X_" + (i + 1) + " = " + otvList.get(1)[i]);
        }
        return;
    }
    while (true) {
        double[] otv1 = new double[size];
        for (int i = 0; i < size; i++) {
            double ans = 0;
            for (int j = 0; j < size; j++) {
                if (i != j) {
                    ans += otvList.get(1)[j] * mt[i][j];
                }
            }
            ans += mt[i][size];
            otv1[i] = ans;
        }
        otvList.remove(0);
        otvList.add(otv1);
        maxPogr = Double.MIN_VALUE;
        for (int i = 0; i < size; i++) {
            if (Math.abs(otvList.get(1)[i] - otvList.get(0)[i]) > maxPogr)
{
                maxPogr = Math.abs(otvList.get(1)[i] - otvList.get(0)[i]);
            }
        }
        iter++;
        System.out.println(iter + " итерация: " + maxPogr);
```

```
        if (maxPogr < matrix.getToch()) {
            System.out.println("Ответ:");
            for (int i = 0; i < size; i++) {
                System.out.println("X_" + (i + 1) + " = " +
otvList.get(1)[i]);
            }
            break;
        }
    }
}
```

- Ссылка на GitHub с основной реализацией

https://github.com/Gallade901/---1

- Пример работы программы

Запуск программы 'Метод простых итераций'
Хотите сгенирировать матрицу? (Да) / (если нет 'любой ввод')
да
Введите точность
1
Введите количество переменных
3
-17,62536 9,37729 -8,14807 -4,02128
-5,85109 -7,88079 -1,92970 5,99357
3,43822 -8,97872 -12,51694 3,09477
Точность: 1.0
Матрица приведенная к диагональному виду
-17,62536 9,37729 -8,14807 -4,02128
-5,85109 -7,88079 -1,92970 5,99357
3,43822 -8,97872 -12,51694 3,09477
Точность: 1.0
0 итерация: 0.7605285058155986
Ответ:
X_1 = 0.2281533171749485
X_2 = -0.7605285058155986
X_3 = -0.24724684995062626

Вывод:

Я научился применять метод простых итераций на практике и написал код для решений задач этого типа.