

Университет ИТМО, кафедра ВТ

**Лабораторная работа №1 по**  
**“Вычислительной математике”**  
**“Решение СЛАУ методом простых итераций”**

Работу выполнил  
студент группы Р3200

**Рогов Я. С.**

Преподаватель:

**Исаев И.В.**

Санкт-Петербург, 2016

## Описание метода

Метод простой итерации.

Этот метод широко используется для численного решения уравнений и их систем различных видов, например вида  $Ax=b$ , где  $A$  – коэффициенты при  $x$ , а  $b$  – свободные члены.

Метод записывается в векторно-матричном виде следующим образом:

$$x^{k+1} = (I - \tau A)x^k + \tau b, \text{ где } \tau - \text{некоторое число, } I - \text{единичная матрица.}$$

В качестве  $x^0$  берут некоторое приближение решения СЛАУ, и далее несколькими итерациями добивается необходимая точность итоговых значений, получение которых и скорость их получения зависит от  $\tau$  – параметр, непосредственно влияющий на шаг итерации для каждого значения.

## Код реализации метода на языке C#

```
public static Matrix SimpleIterMethod(
    Matrix m, double epsilon,
    out int iterations, out Matrix error){

    if (epsilon <= 0)
        throw new WrongEpsilonException(epsilon);
    if (m.Width-1 != m.Height)
        throw new MatrixSizeException(
            m.Width, m.Height, "solving SoLE");
    if(!m.IsSupremeDiag){
        m = m.ToSupremeDiag();
    }

    Matrix b = m.Extract(m.Width-1, 0, 1, m.Height);
    Matrix A = m.Extract(0, 0, m.Width-1, m.Height);
    Matrix I = new Matrix(m.Height, true);

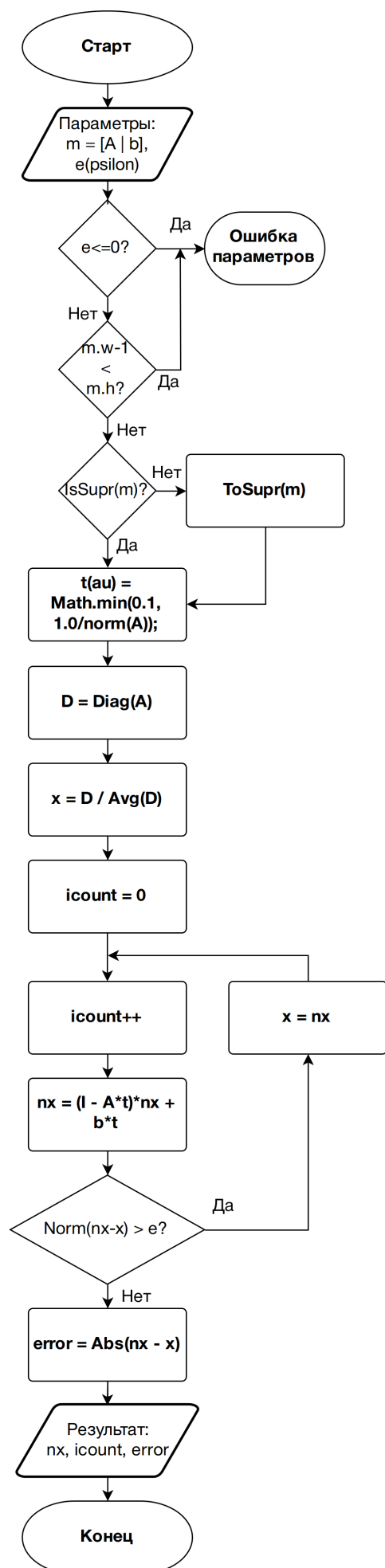
    double tau = Math.Min(0.1, 1.0/A.Norm());
    Matrix D = A.ExtractDiag();
    double[][] xdata = new double[D.Height][];
    double avg = D.Average();
    for(int r=0; r<D.Height; r++){
        xdata[r] = new double[1];
        xdata[r][0] = D[r][r] / avg;
    }

    int icount = 0;
    Matrix x, newx= new Matrix(xdata);
    do{
        x = newx;
        newx = ((I - (A * tau)) * x) + (b * tau);
        icount++;
    } while ((newx-x).Norm() > epsilon);

    iterations = icount;
    error = (newx - x).ToAbs();
    Console.WriteLine(iterations);

    return newx;
}
```

## Блок-схема метода



## Примеры:

Рассмотрим СЛАУ:

$$\begin{cases} 12x_1 + 2x_2 - 4x_3 + 3x_4 - 1x_5 = 20 \\ -2x_1 + 8x_2 + 1x_3 + 1x_4 - 2x_5 = -3 \\ 1x_1 - 1x_2 + 9x_3 - 2x_4 - 3x_5 = 21 \\ -2x_1 + 1x_2 + 1x_3 + 10x_4 - 4x_5 = -11 \\ -5x_1 - 2x_2 - 2x_3 - 2x_4 + 12x_5 = 33 \end{cases}$$

Или в матричном виде:

$$\begin{bmatrix} 12 & 2 & -4 & 3 & -1 \\ -2 & 8 & 1 & 1 & -2 \\ 1 & -1 & 9 & -2 & -3 \\ -2 & 1 & 1 & 10 & -4 \\ -5 & -2 & -2 & -2 & 12 \end{bmatrix} * X = \begin{bmatrix} 20 \\ -3 \\ 21 \\ -11 \\ 33 \end{bmatrix}$$

Корни данной СЛАУ:

$$X = \begin{bmatrix} 3 \\ 1 \\ 4 \\ 1 \\ 5 \end{bmatrix}$$

В файле с названием test2 хранятся следующие данные:

```

5
12 2 -4 3 -1 20
-2 8 1 1 -2 -3
1 -1 9 -2 -3 21
-2 1 1 10 -4 -11
-5 -2 -2 -2 12 33
0.001
  
```

Количество уравнений, матрица вида [A|b], и  $\epsilon$  соответственно.

Вывод программы с помощью команды  
"./Program.exe test2":

Решение системы:

```

3.0002780958446
1.00023641631309
4.0007630016409
1.00045582338928
5.00064443631392
  
```

Количество итераций:

29

Погрешности для каждого корня:

```

8.88004566519385E-05
7.60566903865723E-05
0.00024373635345043
0.000145217217665472
0.000205825105123125
  
```

**Вывод:** В ходе выполнения данной лабораторной работы я подробнее изучил метод простых итераций, а также закрепил свои навыки реализации матриц и базовых операций над ними.