Федеральное государственное бюджетное образовательное учреждение высшего образования

«Саратовский государственный технический университет имени Гагарина Ю.А.»

Кафедра «Математика и моделирование»

Расчётно-графическая работа

Численные методы

«Решение систем алгебраических уравнений»

Выполнил: Самарский И.А  
Группа: б-ПМИН-31

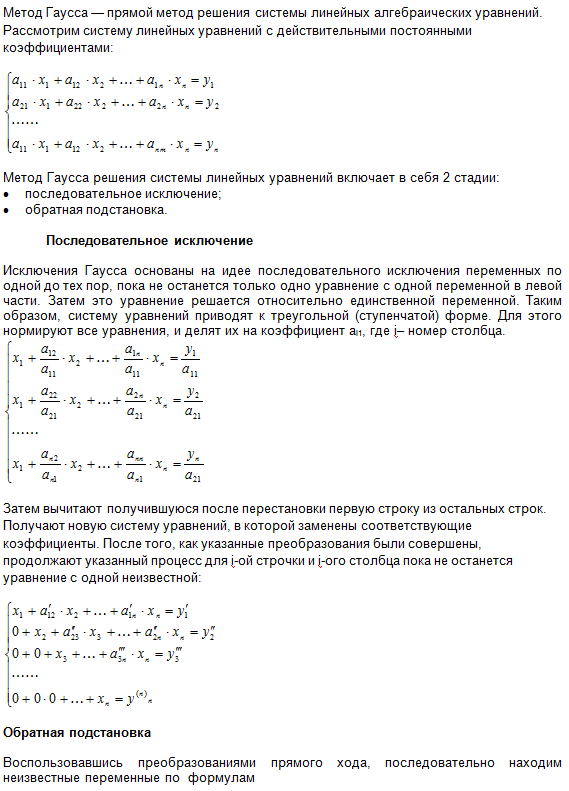
Проверил(а): к. ф - м. н,

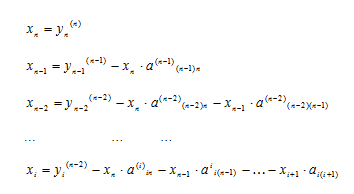
доцент каф. МиМ

Молоденкова Т.В

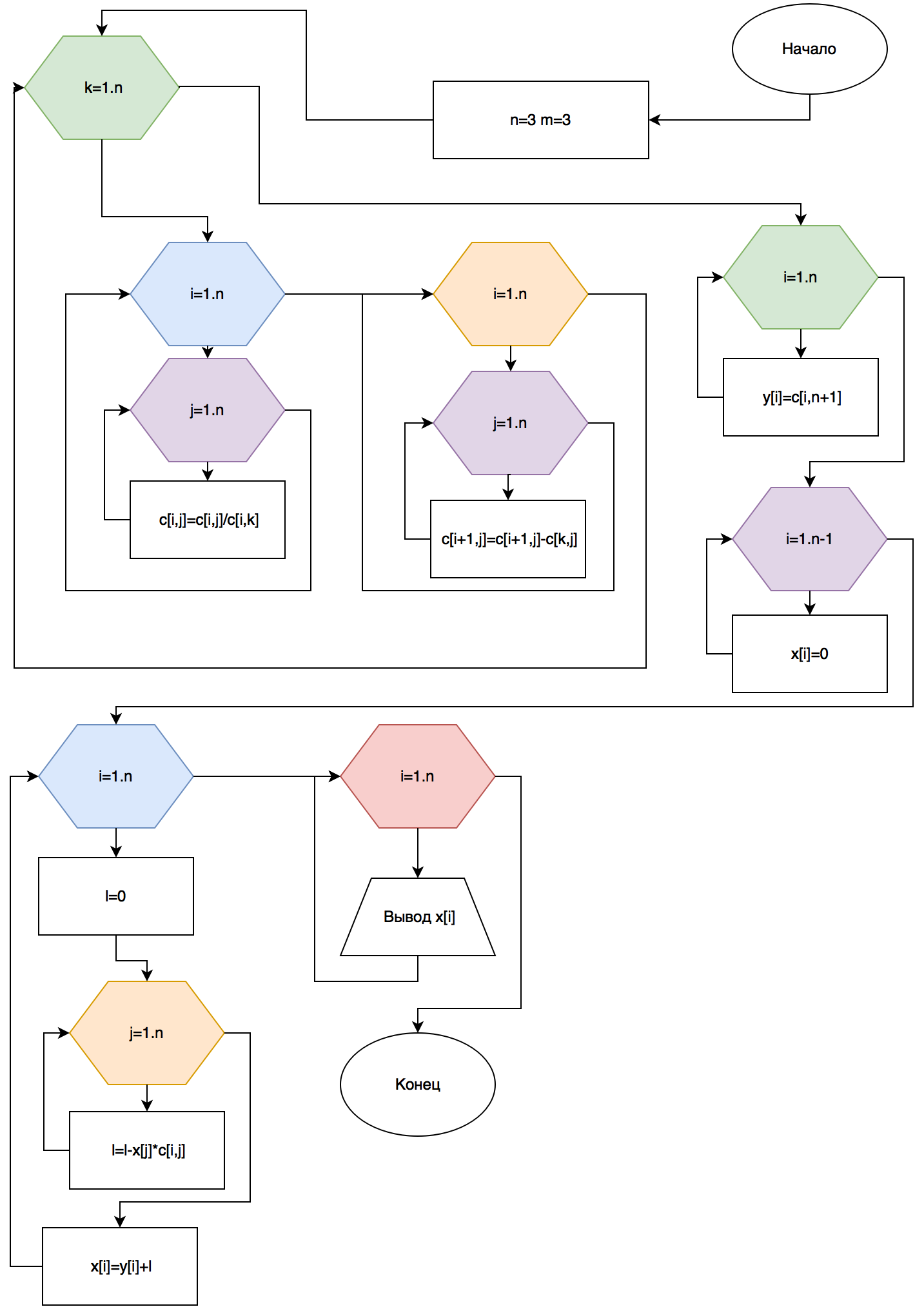
САРАТОВ 2017

Метод Гаусса



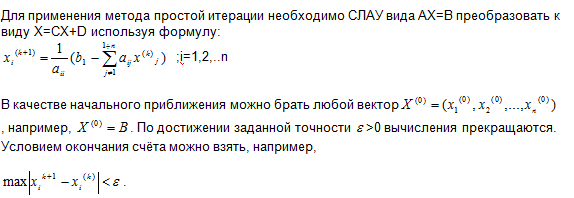


Блок-схема метода Гаусса

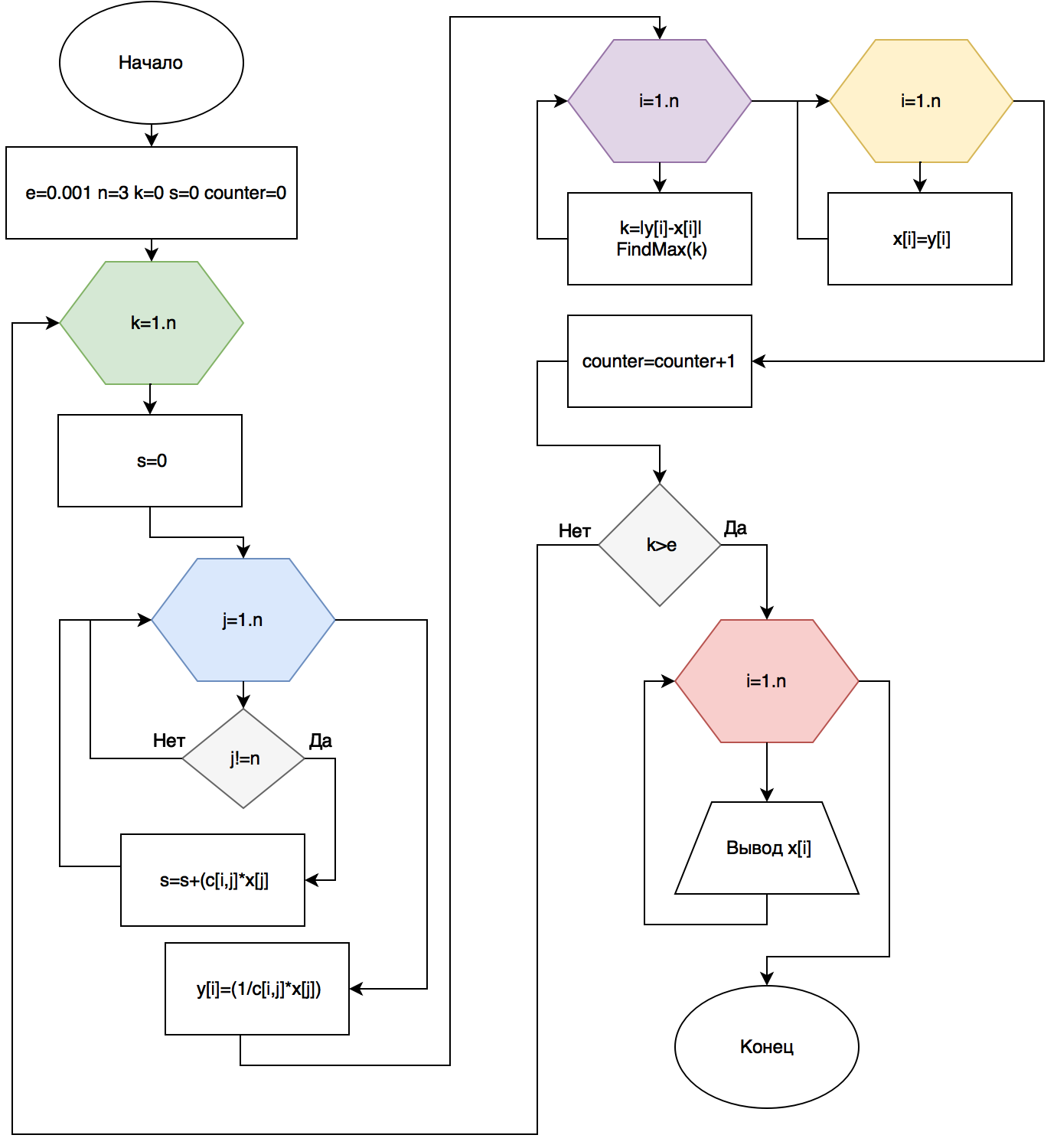
[](https://www.draw.io/?scale=2#G1KV_3l6UZDnyW0Zg8tlObLcYtBa12TIax)

using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
  
namespace Метод\_Гаусса  
{  
 class Program  
 {  
 static void Main(string[] args)  
 {  
 double[,] c = new double[5, 5];  
 double[] y = new double[5];  
 double[] x = new double[5];  
 int n = 3, m = 3;  
  
 c[1, 1] = 4;  
 c[1, 2] = 0.4;  
 c[1, 3] = 0.3;  
 c[1, 4] = 17.8;  
 c[2, 1] = 0.4;  
 c[2, 2] = 3;  
 c[2, 3] = 0.2;  
 c[2, 4] = 11;  
 c[3, 1] = 0.3;  
 c[3, 2] = 0.2;  
 c[3, 3] = 2;  
 c[3, 4] = 5.8;  
  
 Console.WriteLine("Исходная матрица:");  
 for (int i = 1; i <= n; i++)  
 {  
 for (int j = 1; j <= (m + 1); j++)  
 {  
 Console.Write("{0}\t", c[i, j]);  
 }  
 Console.WriteLine();  
 }  
  
 for (int k = 1; k <= n; k++)  
 {  
 for (int i = k; i <= n; i++)  
 {  
 for (int j = (m + 1); j >= k; j--)   
 {  
 c[i, j] = c[i, j] / c[i, k];  
 }  
 }  
  
 for (int i = k; i <= (n - 1); i++)   
 {  
 for (int j = k; j <= (m + 1); j++)   
 {  
 c[(i+1), j] = c[(i+1), j] - c[k, j];  
 }  
 }  
 }  
 Console.WriteLine();  
 for (int i = 1; i <= n; i++)  
 {  
 for (int j = 1; j <= (m + 1); j++)  
 {  
 //Console.Write("{0}\t", c[i, j]); показывает треугольный вид матрицы  
 }  
 Console.WriteLine();  
 }  
  
 for (int i = 1; i <= n; i++)  
 {  
 y[i] = c[i, (n + 1)];  
 }  
  
 for (int i = n; i >= 1; i--)  
 {  
 x[i] = 0;  
 }  
  
 for (int i = n; i >= 1; i--)  
 {  
 double l = 0;  
 for (int j = n; j >= 1; j--)  
 {  
 l = l - x[j] \* c[i, j];  
 }  
 x[i] = y[i] + l;  
 }  
 Console.WriteLine("Ответ:");  
 for (int i = 1; i <= n; i++)   
 {  
 Console.Write("{0} \t", x[i]);  
 }  
 Console.ReadKey();  
 }  
 }  
}

Метод простой итерации

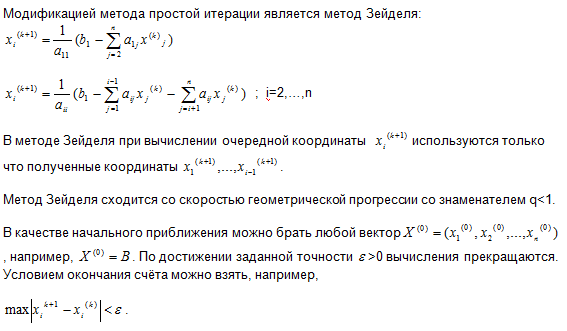


Блок-схема метода Простой итерации

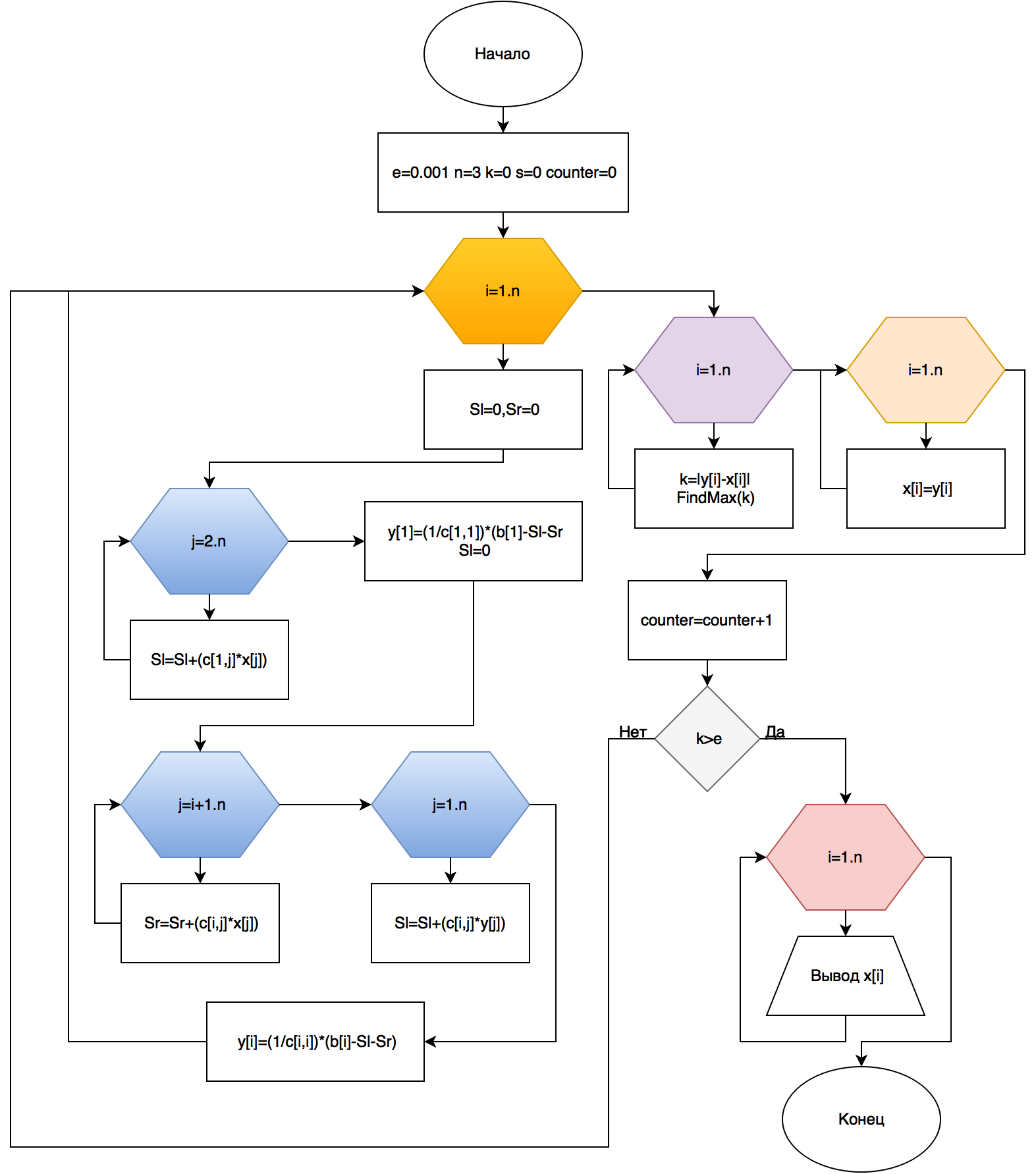
[](https://www.draw.io/?scale=2#G1GXvpU-388jJu42X08MvO2kO6r20j0teT)

using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
  
namespace Метод\_Простой\_Итерации{  
 class Program  
 {  
 static void Main(string[] args)  
 {  
 double[,] c = new double[5, 5];  
 double[] x = new double[5];  
 double[] y = new double[5];  
 double[] b = new double[5];  
  
 double e = 0.001, n = 3;  
   
  
 c[1, 1] = 4;  
 c[1, 2] = 0.4;  
 c[1, 3] = 0.3;  
 b[1] = 17.8;  
 c[2, 1] = 0.4;  
 c[2, 2] = 3;  
 c[2, 3] = 0.2;  
 b[2] = 11;  
 c[3, 1] = 0.3;  
 c[3, 2] = 0.2;  
 c[3, 3] = 2;  
 b[3] = 5.8;  
  
 x[1] = 17.8;x[2] = 11;x[3] = 5.8; // вы можете задать любой начальный вектор  
 //x[1] = 0; x[2] = 0; x[3] = 0;  
  
  
 Console.WriteLine("Матрица А");  
 for (int i = 1; i <= n; i++)  
 {  
 for (int j = 1; j <= n; j++)  
 {  
 Console.Write("{0}\t", c[i, j]);  
 }  
 Console.WriteLine();  
 }  
  
 Console.WriteLine();  
  
 Console.WriteLine("Матрица B");  
 for (int i = 1; i <= n; i++)  
 {  
 Console.Write("{0}\n", b[i]);  
 }  
  
 double k = 0, s = 0; int counter = 0;  
 do  
 {  
 //формула 17  
 for (int i = 1; i <= n; i++)  
 {  
 s = 0;  
  
 for (int j = 1; j <= n; j++)  
 {  
 if (j != i)  
 {  
 s = s + (c[i, j] \* x[j]);  
 }  
 }  
  
 y[i] = (1 / c[i, i]) \* (b[i] - s);  
 }  
  
 //проверка условий для окончания  
 for (int i = 1; i <= n; i++)  
 {  
 k = Math.Abs(y[i] - x[i]);  
 FindMax(k);//поиск максимума  
 }  
  
 for (int i = 1; i <= n; i++)  
 {  
 x[i] = y[i];  
 }  
  
 counter++;  
 } while (k > e);  
  
   
 Console.WriteLine();  
 Console.Write("Ответ: ");  
 for (int i = 1; i <= n; i++)  
 {  
 Console.Write("{0} ", x[i]);  
 }  
  
 Console.WriteLine("\nСчётчик = {0}",counter);  
 Console.ReadKey();  
 }  
  
 public static double FindMax(double k)  
 {  
 double max = -999999;  
 if (k > max)  
 {  
 max = k;  
 }  
 return k;  
 }  
 }  
}

Метод Зейделя

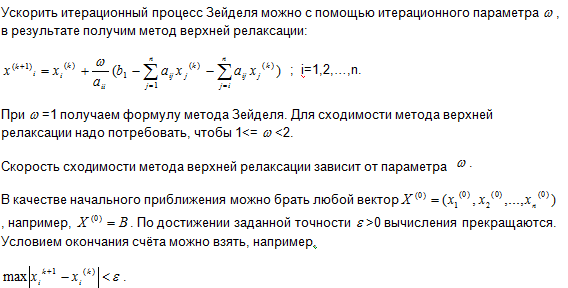


Блок-схема метода Зейделя

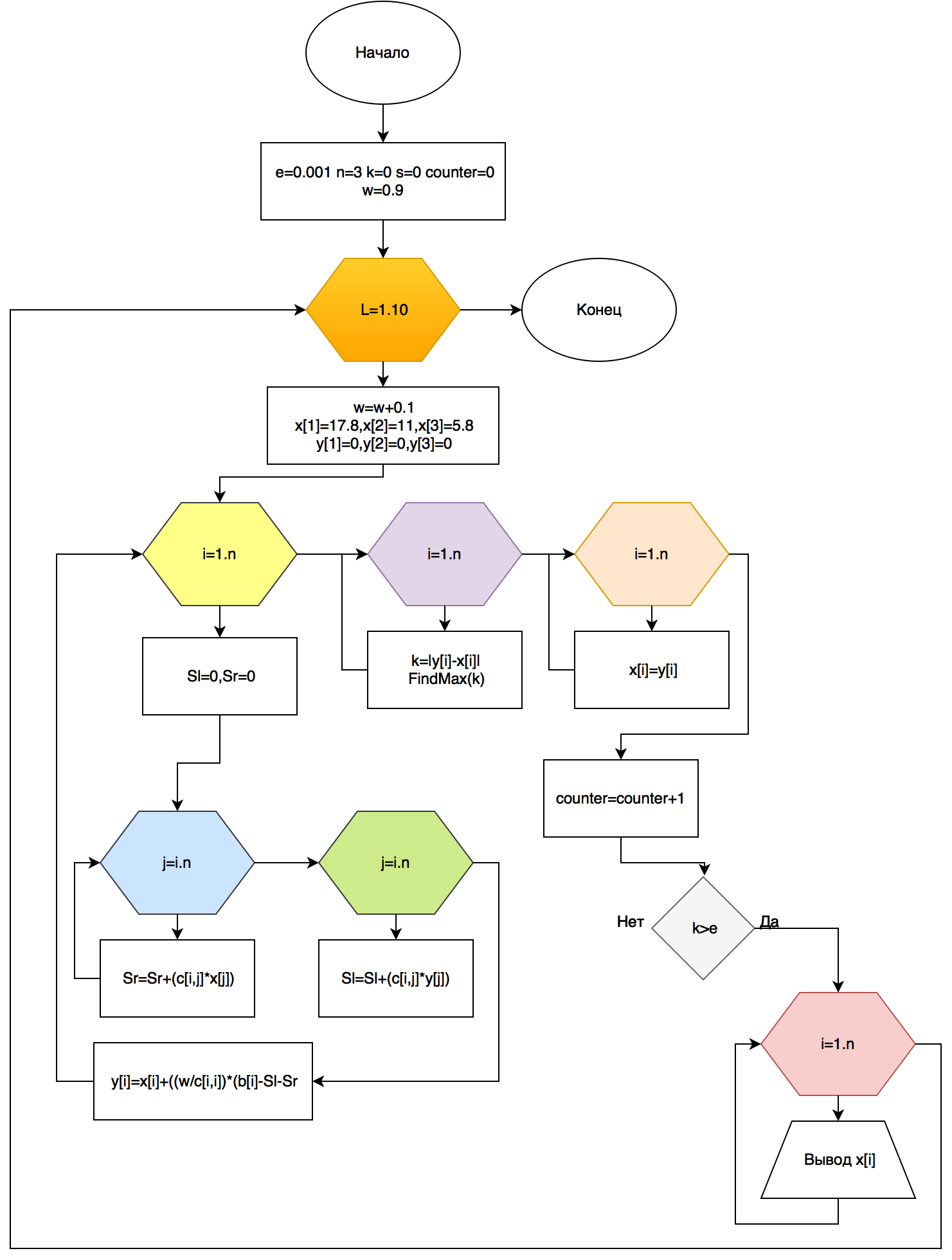
[](https://www.draw.io/?scale=2#G1WtnGKT6hyee5UnjFow4BXzuQr2QOfmak)

using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
  
namespace Метод\_Зейделя{  
 class Program  
 {  
 static void Main(string[] args)  
 {  
 double[,] c = new double[5, 5];  
 double[] x = new double[5];  
 double[] y = new double[5];  
 double[] b = new double[5];  
  
 double e = 0.001, n = 3;  
 int counter = 0, i = 1;  
  
 c[1, 1] = 4;  
 c[1, 2] = 0.4;  
 c[1, 3] = 0.3;  
 b[1] = 17.8;  
 c[2, 1] = 0.4;  
 c[2, 2] = 3;  
 c[2, 3] = 0.2;  
 b[2] = 11;  
 c[3, 1] = 0.3;  
 c[3, 2] = 0.2;  
 c[3, 3] = 2;  
 b[3] = 5.8;  
  
 x[1] = 17.8; x[2] = 11; x[3] = 5.8;  
 //x[1] = 0; x[2] = 0; x[3] = 0;  
  
 Console.WriteLine("Матрица А");  
 for (i = 1; i <= n; i++)  
 {  
 for (int j = 1; j <= n; j++)  
 {  
 Console.Write("{0}\t", c[i, j]);  
 }  
 Console.WriteLine();  
 }  
  
 Console.WriteLine();  
  
 Console.WriteLine("Матрица B");  
 for (i = 1; i <= n; i++)  
 {  
 Console.Write("{0}\n", b[i]);  
 }  
  
 double k = 0;  
  
 do  
 {  
 for (i = 1; i <= n; i++)  
 {  
 double Sl = 0, Sr = 0;  
  
 //формула 20  
 for (int j = 2; j <= n; j++)  
 {  
 Sl = Sl + (c[1, j] \* x[j]);  
 }  
 y[1] = (1 / c[1, 1]) \* (b[1] - Sl - Sr);  
  
 Sl = 0;  
 //формула 21  
 for (int j = i + 1; j <= n; j++) // прававое слагаемое  
 {  
 Sr = Sr + (c[i, j] \* x[j]);  
 }  
  
 for (int j = 1; j <= i - 1; j++)// левое слагаемое  
 {  
 Sl = Sl + (c[i, j] \* y[j]);  
 }  
 y[i] = (1 / c[i, i]) \* (b[i] - Sl - Sr); // коэффициент   
 }  
  
 //проверка условий для окончания  
 for (i = 1; i <= n; i++)  
 {  
 k = Math.Abs(y[i] - x[i]);  
 FindMax(k);//поиск максимума  
 }  
  
 for (i = 1; i <= n; i++)  
 {  
 x[i] = y[i];  
 }  
 counter++;  
 } while (k > e);  
   
 Console.WriteLine();  
 Console.Write("Ответ: ");  
 for (i = 1; i <= n; i++)   
 {  
 Console.Write("{0} ", x[i]);  
 }  
  
 Console.WriteLine("\nСчётчик = {0}", counter);  
 Console.ReadKey();  
 }  
  
 public static double FindMax(double k)  
 {  
 double max = -999999;  
 if (k > max)  
 {  
 max = k;  
 }  
 return k;  
 }  
 }  
}

Метод Релаксации

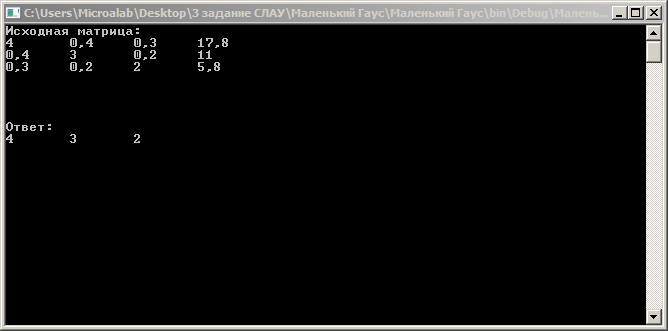


Блок-схема метода Верхней Релаксации

[](https://www.draw.io/?scale=2#G1_hB-2mHhyPLTNHp1dcUbYZCxuRLt1kDa)

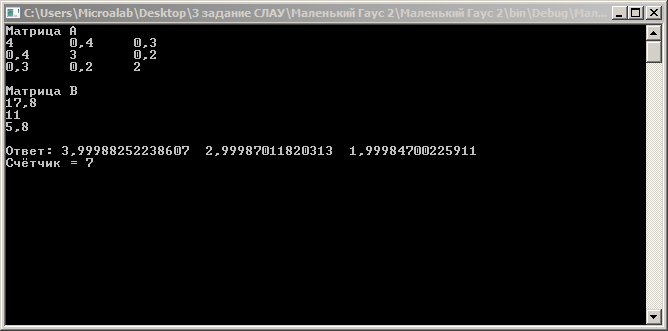
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
  
namespace Метод\_Рилаксации{  
 class Program  
 {  
 static void Main(string[] args)  
 {  
 double[,] c = new double[5, 5];  
 double[] x = new double[5];  
 double[] y = new double[5];  
 double[] b = new double[5];  
  
 double e = 0.001, k = 0, n = 3;  
 int counter = 0, i = 1;  
  
 c[1, 1] = 4;  
 c[1, 2] = 0.4;  
 c[1, 3] = 0.3;  
 b[1] = 17.8;  
 c[2, 1] = 0.4;  
 c[2, 2] = 3;  
 c[2, 3] = 0.2;  
 b[2] = 11;  
 c[3, 1] = 0.3;  
 c[3, 2] = 0.2;  
 c[3, 3] = 2;  
 b[3] = 5.8;  
  
 x[1] = 0; x[2] = 0; x[3] = 0;  
 //x[1] = 17.8; x[2] = 11; x[3] = 5.8;  
  
 Console.WriteLine("Матрица А");  
 for (i = 1; i <= n; i++)  
 {  
 for (int j = 1; j <= n; j++)  
 {  
 Console.Write("{0}\t", c[i, j]);  
 }  
 Console.WriteLine();  
 }  
  
 Console.WriteLine();  
  
 Console.WriteLine("Матрица B");  
 for (i = 1; i <= n; i++)  
 {  
 Console.Write("{0}\n", b[i]);  
 }  
  
 Console.WriteLine();  
  
 double w = 0.9;  
  
 for (int l = 1; l <= 10; l++)  
 {  
 w = w + 0.1;k = 0;  
 x[1] = 17.8; x[2] = 11; x[3] = 5.8;  
 y[1] = 0;y[2] = 0;y[3] = 0;  
   
 do  
 {  
 for (i = 1; i <= n; i++)  
 {  
 double Sl = 0, Sr = 0;  
 //правая часть  
 for (int j = i; j <= n; j++)  
 {  
 Sr = Sr + (c[i, j] \* x[j]);  
 }  
 //левая часть  
 for (int j = 1; j <= i - 1; j++)  
 {  
 Sl = Sl + (c[i, j] \* y[j]);  
 }  
 y[i] = x[i] + ((w / c[i, i]) \* (b[i] - Sl - Sr));  
 }  
  
 //проверка условий для окончания  
 for (i = 1; i <= n; i++)  
 {  
 k = Math.Abs(y[i] - x[i]);  
 FindMax(k);//поиск максимума  
 }  
  
 for (i = 1; i <= n; i++)  
 {  
 x[i] = y[i];  
 }  
   
 counter++;  
 } while (k > e);   
  
 Console.WriteLine();  
 Console.WriteLine("При w = {0}", w);  
 Console.Write("Ответ: ");  
 for (i = 1; i <= n; i++)  
 {  
 Console.Write("{0} ", x[i]);  
 }  
 Console.WriteLine("\nСчётчик = {0}", counter);  
  
 i = 1;   
 }   
 Console.ReadKey();  
 }  
  
 public static double FindMax(double k)  
 {  
 double max = -999999;  
 if (k > max)  
 {  
 max = k;  
 }  
 return k;  
 }  
 }  
}

Вывод: Необходимо решить систему уравнений 4-мя способами.

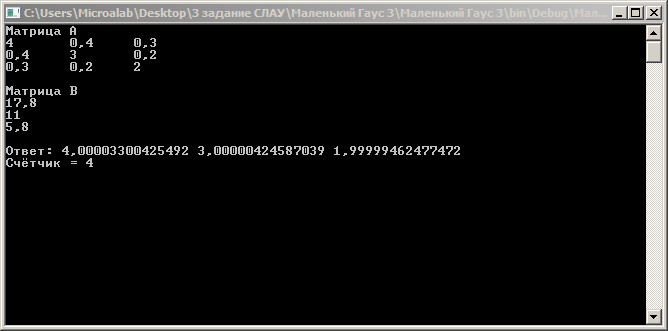
Для проверки итерационных методов, найдем точное решение системы с помощью метода Гаусса.  
  


Теперь воспользуемся итерационными методами:

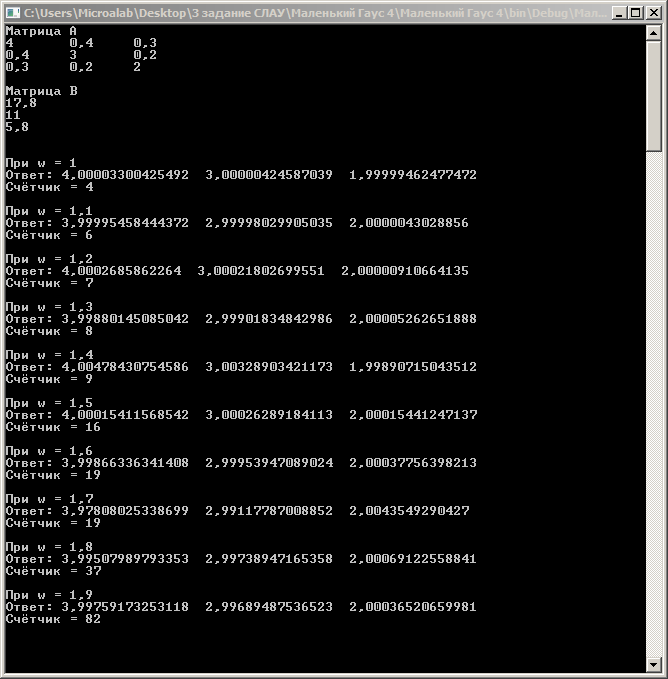
1) Метод простой итерации



2) Метод Зейделя



3) Метод Верхней Релаксации



Итого: Наилучшим методом является метод Зейделя при w = 1