Смоляков Адам

2 курс , 5 группа

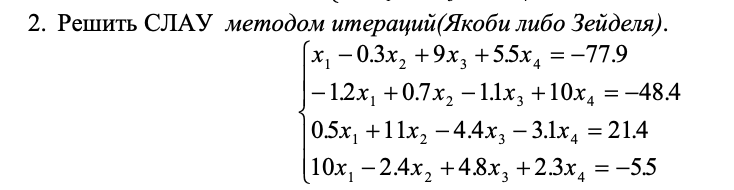
**Лабораторная работа № 1**

**Системы линейных алгебраических уравнений**

**Цель. Изучить метод Якоби для СЛАУ.**

**Вариант 3**

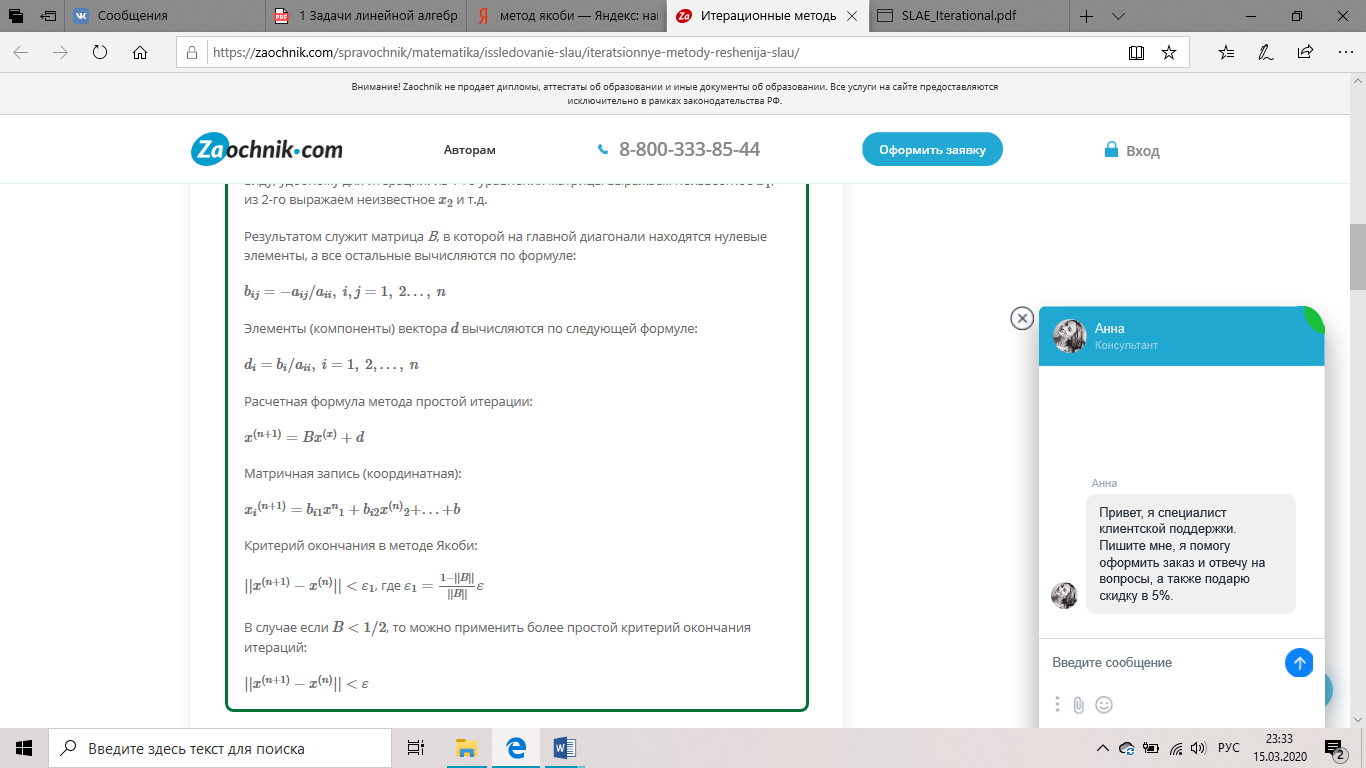
**Задание.**

****

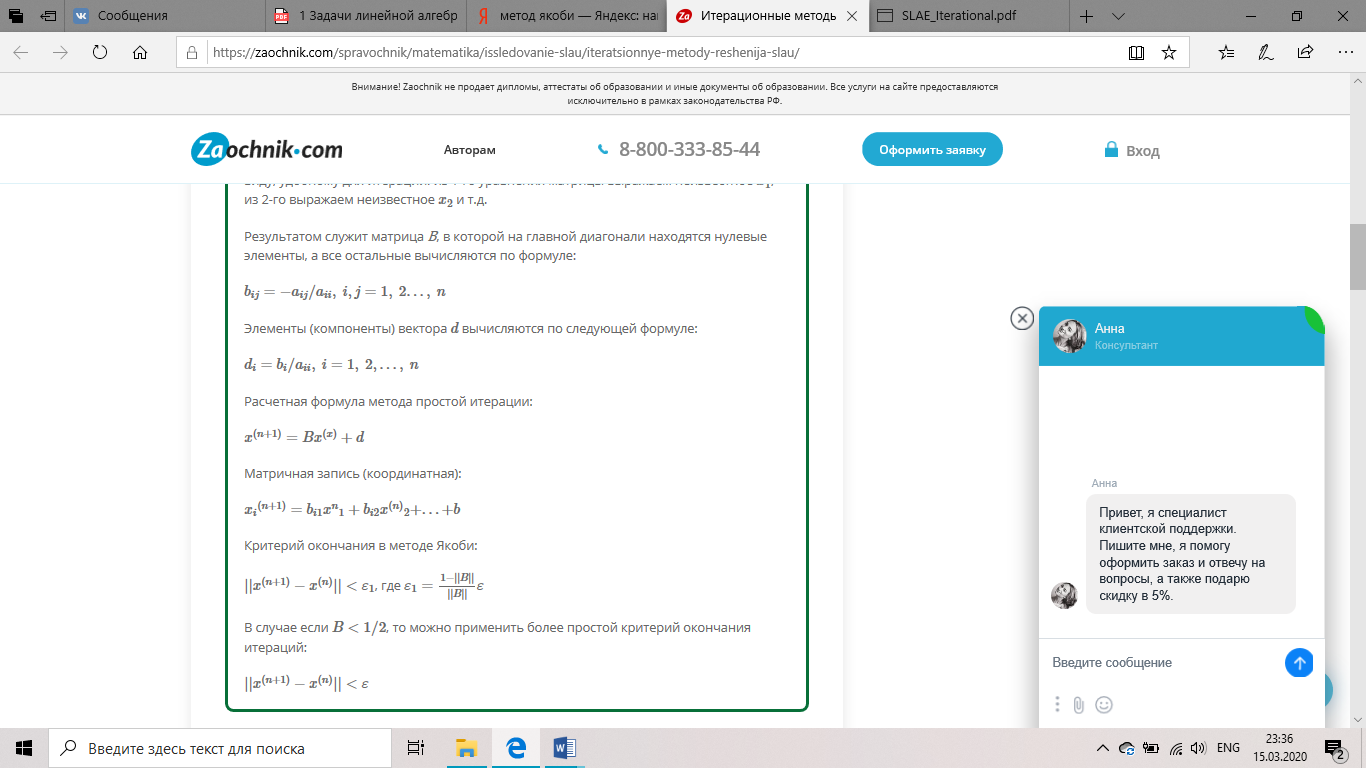
**Теория метода.**

**Метод Якоби** — один из наиболее простых методов приведения системы матрицы к виду, удобному для итерации: из 1-го уравнения матрицы выражаем неизвестное x1, из 2-го выражаем неизвестное x2и т.д.

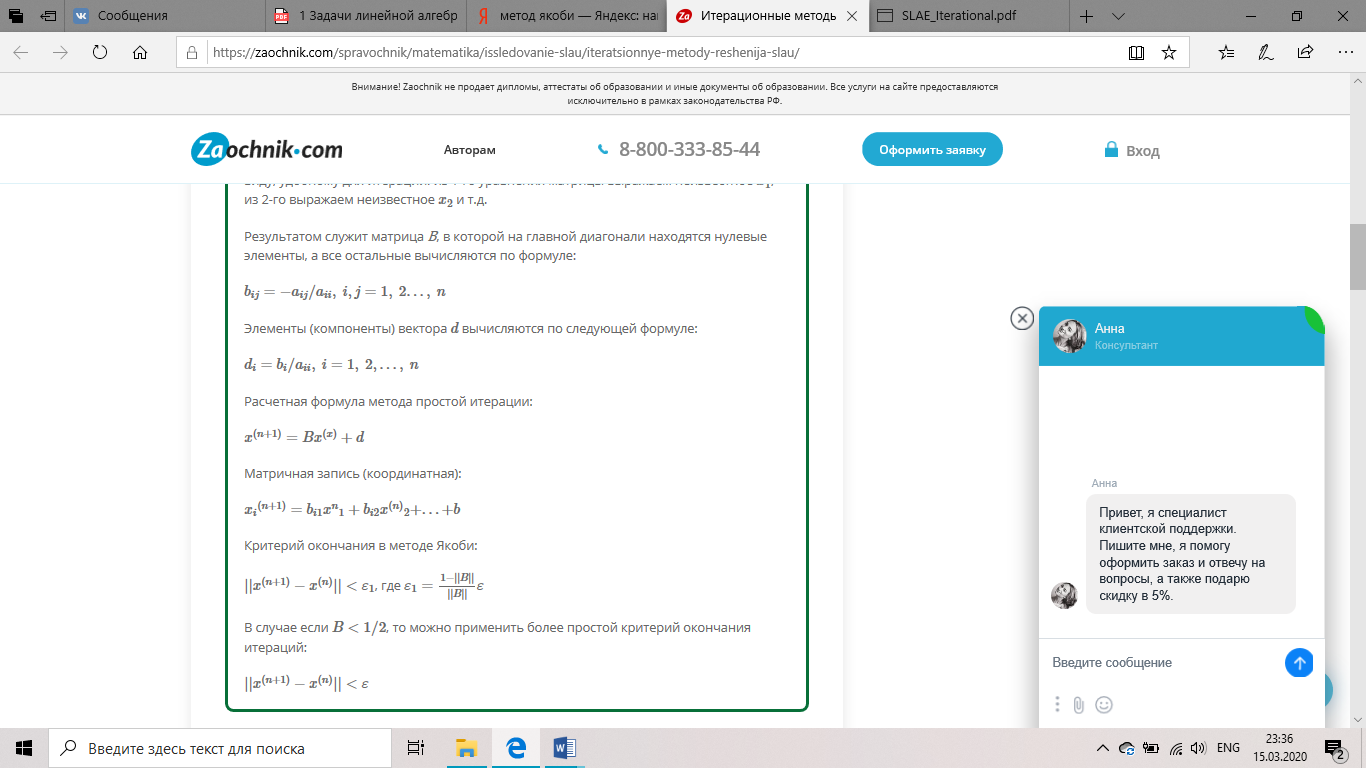
Результатом служит матрица *В*, в которой на главной диагонали находятся нулевые элементы, а все остальные вычисляются по формуле:



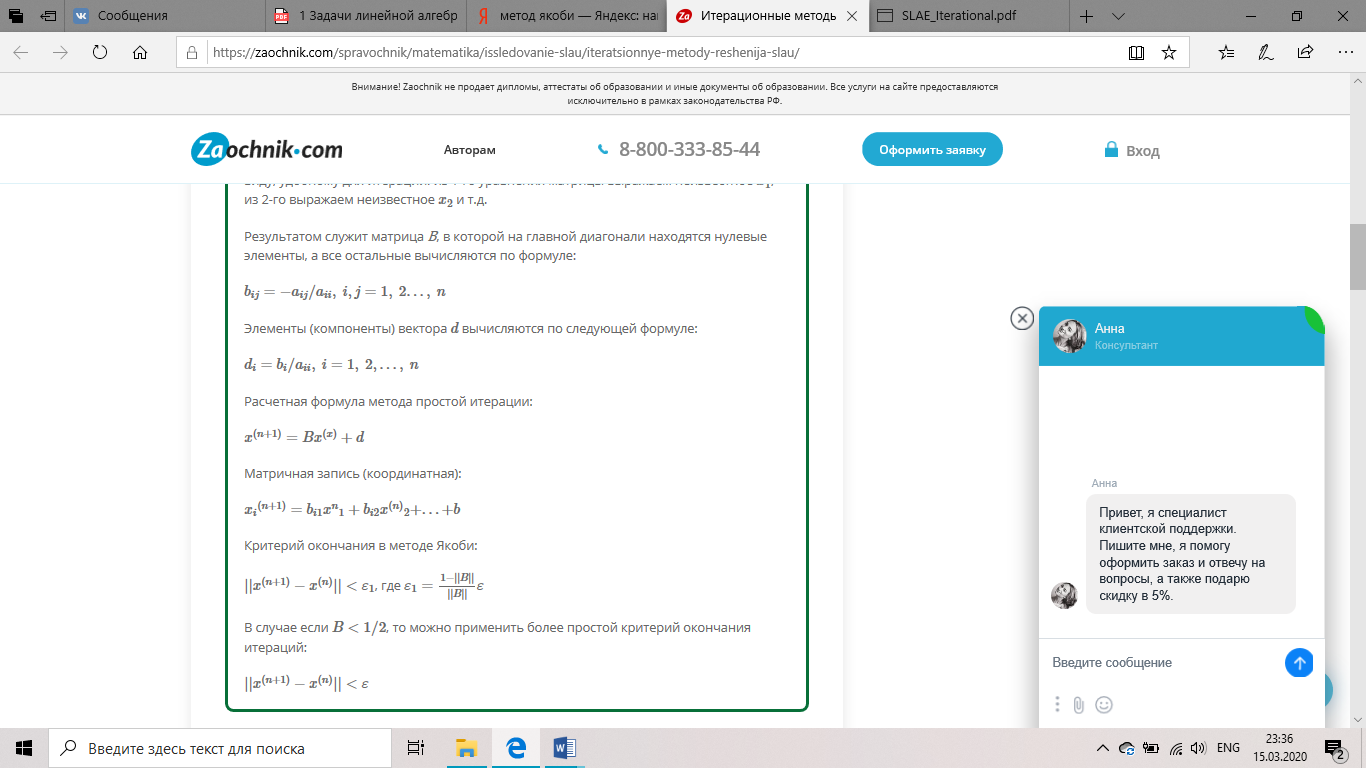
Элементы (компоненты) вектора d вычисляются по следующей формуле:



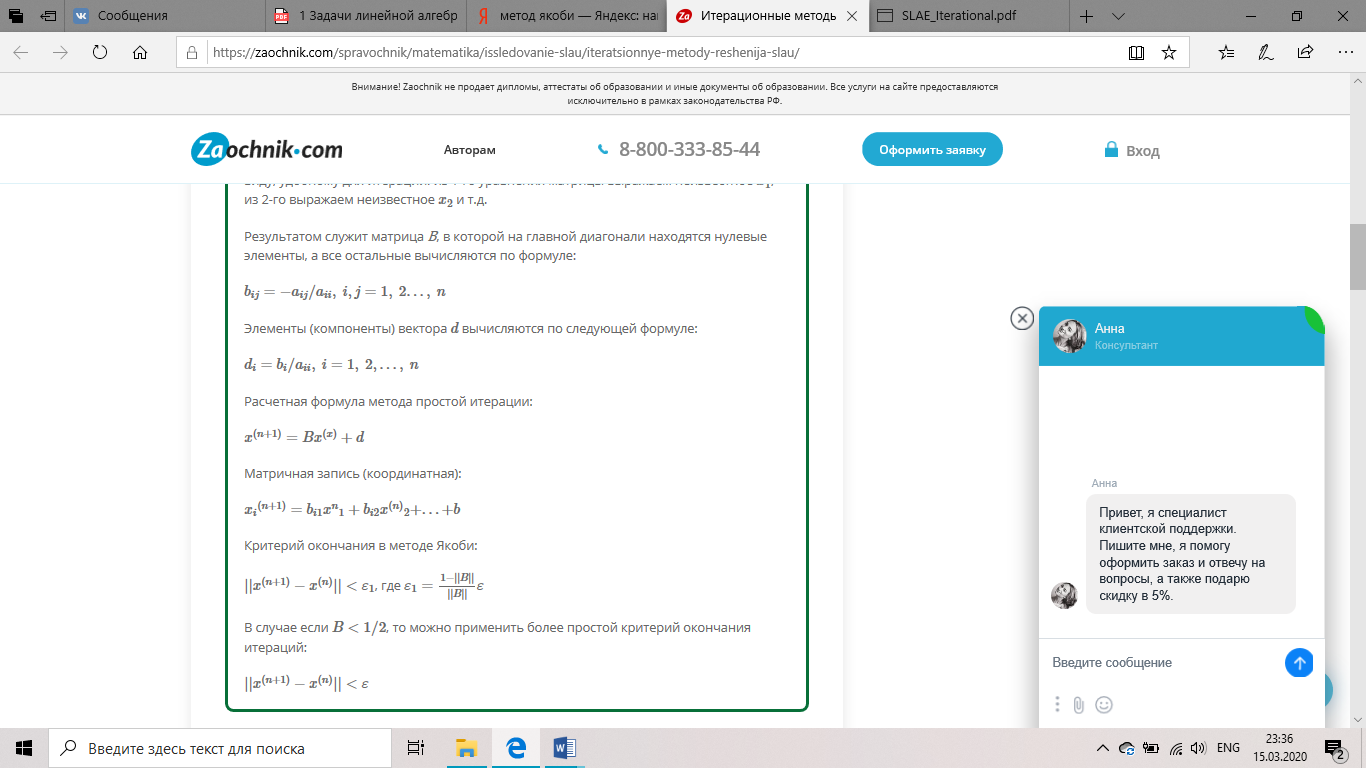
Расчетная формула метода простой итерации:



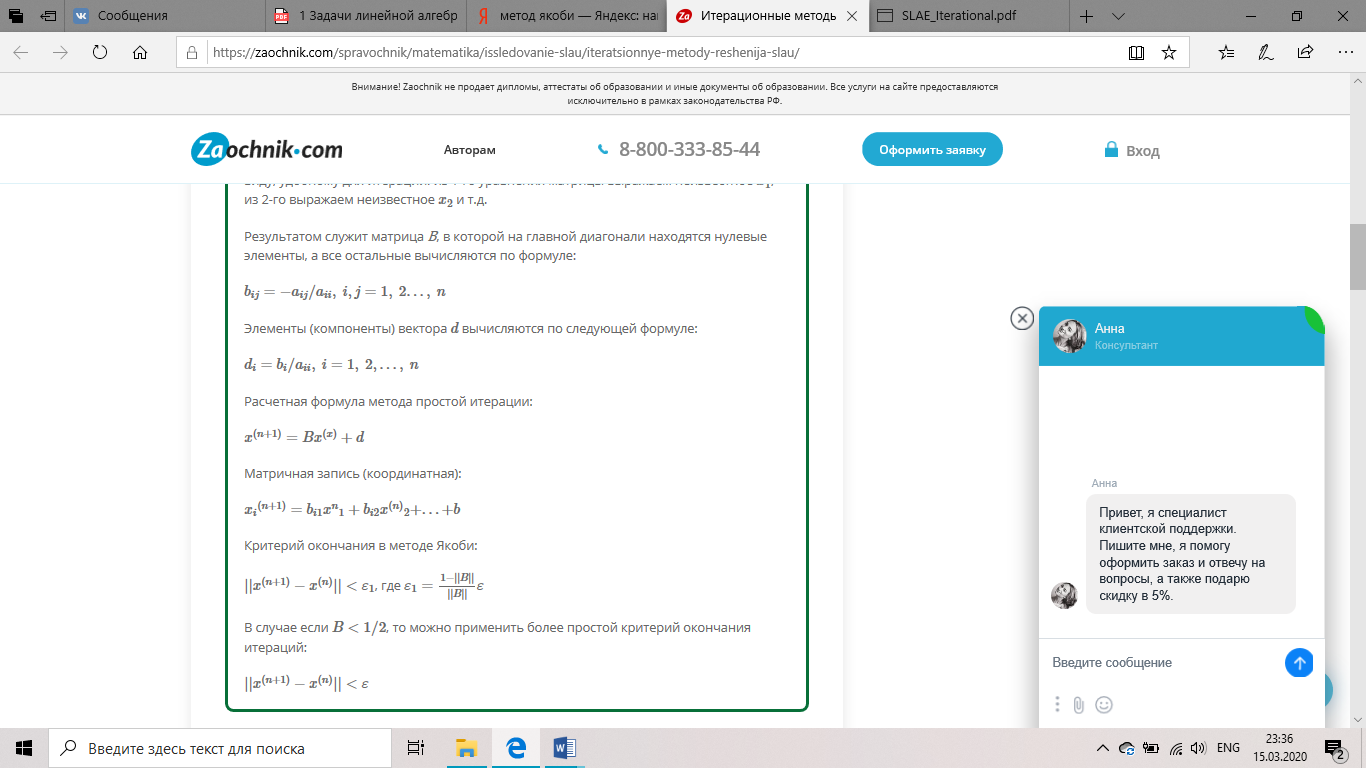
Матричная запись (координатная):



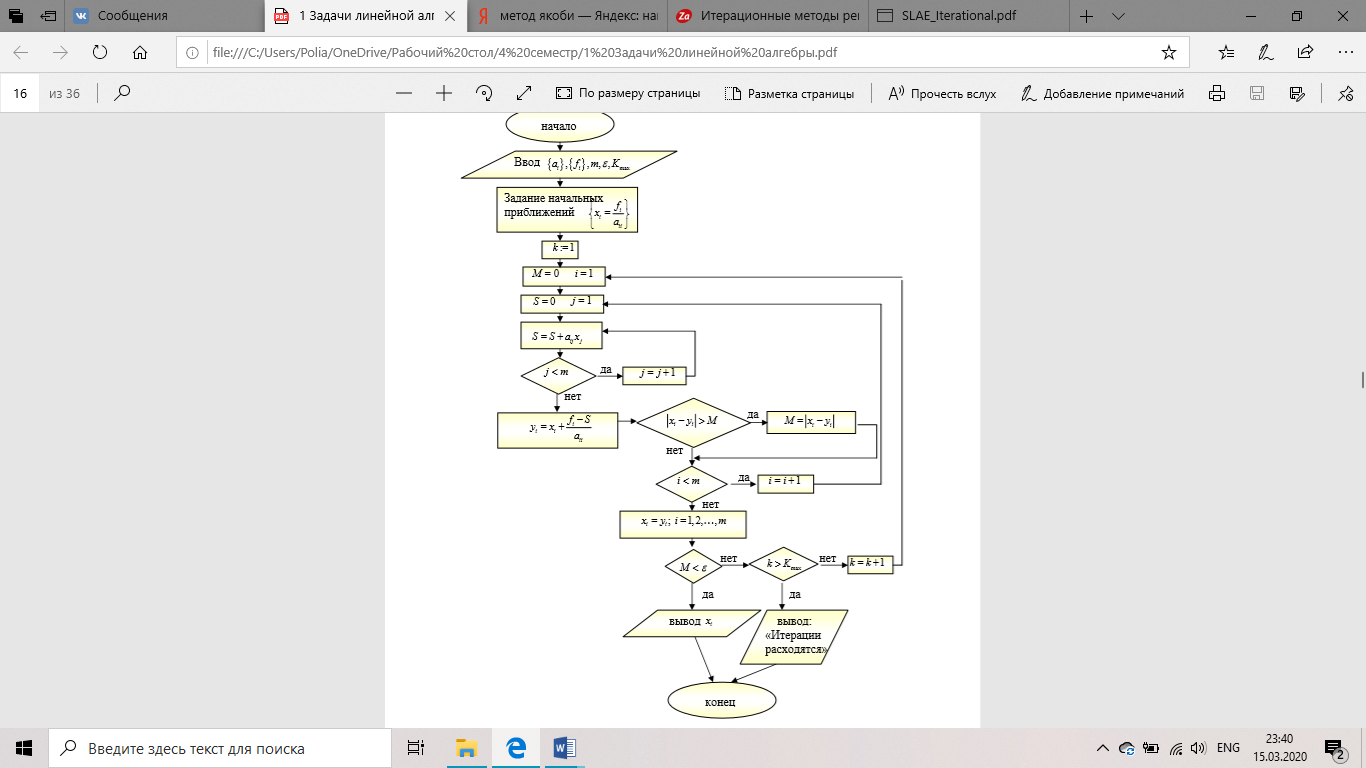
Критерий окончания в методе Якоби:



В случае если B<1/2B<1/2, то можно применить более простой критерий окончания итераций:



**Блок-схема метода Якоби:**



Метод Якоби характеризируется общим видом итерационной формулы, которая может быть представлена в простой форме

**(1)**

Слау вида сводится к виду . Данная Слау не может быть решена методом Якоби, так как она не удовлетворяет условию

Для того, чтобы наша слау решалась методом Якоби, надо поменять строчки местами.

После данного преобразования, **слау** имеет вид:



**Код программы**

**program** Project1;

{$APPTYPE CONSOLE}

{$R \*.res}

**uses**

System.SysUtils;

**var**

m,e,kmax,s:**double**;

a,a0: **array** [**1**..**4**,**1**..**4**] **of** **real**;

f,x,y,f0: **array** [**1**..**4**] **of** **real**;

i,j,k: **integer**;

**begin**

a[**1**][**1**] := **10**;

a[**1**][**2**] := -**2.4**;

a[**1**][**3**] := **4.8**;

a[**1**][**4**] := **2.3**;

a[**2**][**1**] := **0.5**;

a[**2**][**2**] := **11**;

a[**2**][**3**] := -**4.4**;

a[**2**][**4**] := -**3.1**;

a[**3**][**1**] := **1**;

a[**3**][**2**] := -**0.3**;

a[**3**][**3**] := **9**;

a[**3**][**4**] := **5.5**;

a[**4**][**1**] := -**1.2**;

a[**4**][**2**] := **0.7**;

a[**4**][**3**] := -**1.1**;

a[**4**][**4**] := **10**;

f[**1**] := -**5.5**;

f[**2**] := **21.4**;

f[**3**] := -**77.9**;

f[**4**] := -**48.4**;

k := **1**;

kmax := **1000**;

e := **0.0001**;

**for** i := **1** **to** **4** **do**

**begin**

f0[i] := f[i];

**for** j := **1** **to** **4** **do**

**begin**

a0[i,j] := a[i,j];

**end**;

**end**;

writeln('Èñõîäíàÿ ìàòðèöà');

**for** i := **1** **to** **4** **do**

**begin**

**for** j := **1** **to** **4** **do**

write(a[i][j]:**7**:**1**);

writeln;

**end**;

**for** i := **1** **to** **4** **do**

**begin**

x[i] := f[i] / a[i][i];

**end**;

**while** k <= kmax **do**

**begin**

m := **0**;

**for** i := **1** **to** **4** **do**

**begin**

s := **0**;

**for** j := **1** **to** **4** **do**

**begin**

s := s + a[i][j] \* x[j];

**end**;

y[i] := x[i] + (f[i] -s) / a[i][i];

**if** abs(x[i] - y[i]) > m **then**

m := abs(x[i] - y[i]);

x[i] := y[i];

**end**;

**if** (m <= e) **then**

**break**;

inc(k);

**end**;

**if** k >= kmax **then** writeln('Èòåðàöèè ðàñõîäÿòñÿ')

**else**

**begin**

writeln('âåêòîð-ñòîëáåö ðåøåíèé');

**for** i := **1** **to** **4** **do**

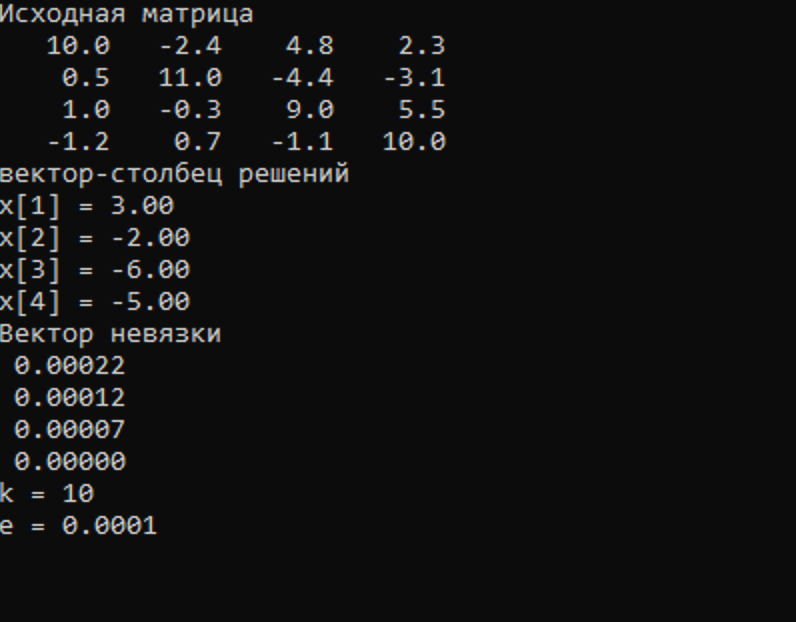
**begin**

writeln('x[',i, '] = ',x[i]:**2**:**2**);

**end**;

**end**;

**Результат:**

****

**Вывод:** Был изучен метод Якобы для решения СЛАУ. В Сравнении с Методом Гаусса, видно, что он менее точен так как вектор невязки в данном случае мал, но в методе Гаусса он был нулевой

Но преимущество, будет видно при работе с системами высокого порядка

Итерационные методы выгодны для систем высоких порядков, систем специального вида (например, со слабо заполненной матрицей). Для решения так называемых плохо обусловленных систем используются методы регуляризации.