

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«Кубанский государственный университет»

Факультет математики и компьютерных наук

## **Семестровая работа**

### **Метод Зейделя**

Специальность 02.03.01 Математика и компьютерные науки

Выполнил:

Студент 1 курса, 13А группы  
математического факультета  
КубГУ

специальности

«математика и компьютерные  
науки»

очной формы обучения

Акимов Вадим

## ЗАДАЧА

Используя метод Зейделя с заданной точностью решить систему линейных уравнений:

$$\text{№ 1. } \begin{cases} 2,7x_1 + 3,3x_2 + 1,3x_3 = 2,1; \\ 3,5x_1 - 1,7x_2 + 2,8x_3 = 1,7; \\ 4,1x_1 + 5,8x_2 - 1,7x_3 = 0,8. \end{cases}$$

## КРАТКОЕ ОПИСАНИЕ МЕТОДА

Метод Зейделя – это итерационный метод для решения систем линейных уравнений. Ключевой необходимостью для корректной работы данного метода является приведение полной матрицы с коэффициентами и свободными членами к виду с диагональным преобладанием.

В исходной матрице диагонального преобладания нет, и если мы будем работать с ней, то полученный результат не будет удовлетворительным и точным. Поэтому, с помощью ЭП(элементарных преобразований), необходимо привести матрицу к нужному виду. Для этого можно умножить первую строку матрицы на -1 и сложить ее со второй. count

## СПИСОК ИДЕНТИФИКАТОРОВ

*eps* – точность, *double*

*n* - размерность матрицы, *int*

*x* - неизвестные, *double[]*

*y* — *i*-ый ответ матрицы, *double*

*mat* – матрица коэффициентов и свободных членов, *double[][]*

*flag* - условие окончания приближения ответов, *int*

*count* – номер итерации, *int*

## ОПИСАНИЕ АЛГОРИТМА

Пользователь вводит матрицу и осуществляется вывод ее на консоль. Далее неизвестным присваиваются значения элементов столбца свободных членов. После к матрице применяется алгоритм Зейделя и последовательно находятся корни уравнения. В конце программы они выводятся на консоль.

## ТЕКСТ ПРОГРАММЫ

```
#include <iostream>
```

```
#include <iomanip>
```

```
#include <cmath>
```

```
using namespace std;
```

```
const double eps = 0.001;
```

```
int main() {
```

```
    cout.precision(4);
```

```
    cout.setf(ios::fixed);
```

```
    int n, flag = 0, count = 0;
```

```
    cout << "\nEnter the no. of equations\n";
```

```
    cin >> n;
```

```
    double mat[n][n + 1];
```

```
    double x[n] = {};
```

```
    double y;
```

```
    cout << "\nEnter the elements of the augmented matrix row-wise:\n";
```

```
    for (unsigned i = 0; i < n; i++)
```

```
        for (unsigned j = 0; j <= n; j++)
```

```
            cin >> mat[i][j];
```

```
    for (unsigned i = 0; i < n; i++)
```

```
        for (unsigned k = i + 1; k < n; k++)
```

```
            if (abs(mat[i][i]) < abs(mat[k][i]))
```

```

    for (unsigned j = 0; j <= n; j++) {
        double temp = mat[i][j];
        mat[i][j] = mat[k][j];
        mat[k][j] = temp;
    }

```

```

cout << "\nIter" << setw(10);

```

```

for(unsigned i = 0; i < n; i++)

```

```

    cout << "x" << i << setw(18);

```

```

cout << "\n-----\n";

```

```

do {

```

```

    flag = 0;

```

```

    cout << count + 1 << "." << setw(16);

```

```

    for (unsigned i = 0; i < n; i++) {

```

```

        y = x[i];

```

```

        x[i] = mat[i][n];

```

```

        for (unsigned j = 0; j < n; j++) {

```

```

            if (j != i)

```

```

                x[i] = x[i] - mat[i][j] * x[j];

```

```

        }

```

```

        x[i] = x[i] / mat[i][i];

```

```

        if (abs(x[i] - y) <= eps)

```

```

            flag++;

```

```

        cout << x[i] << setw(18);

```

```

    }

```

```

    cout << "\n";

```

```
    count++;  
} while(flag < n);  
  
cout << "\n The solution is as follows:\n";  
for (unsigned i = 0; i < n; i++)  
    cout << "x" << i << " = " << x[i] << endl;  
return 0;  
}
```

## РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРОГРАММЫ

Enter the no. of equations

3

Enter the elements of the augmented matrix row-wise:

2.7 3.3 1.3 2.1

3.5 -1.7 2.8 1.7

4.1 5.8 -1.7 0.8

Iter	x0	x1	x2
1.	0.1951	0.4767	0.6527
2.	-0.2086	0.5500	1.2018
3.	-0.0845	0.2321	0.8537
4.	0.2208	0.1194	0.4036
5.	0.1936	0.3190	0.5588
6.	-0.0244	0.4362	0.9025
7.	-0.0477	0.3199	0.8610
8.	0.0996	0.2157	0.6136
9.	0.1444	0.2765	0.5945
10.	0.0505	0.3609	0.7631
11.	0.0011	0.3349	0.8092
12.	0.0569	0.2711	0.7006
13.	0.1022	0.2768	0.6475
14.	0.0720	0.3223	0.7128
15.	0.0347	0.3272	0.7625
16.	0.0484	0.2964	0.7266
17.	0.0771	0.2870	0.6850
18.	0.0731	0.3067	0.7020
19.	0.0523	0.3170	0.7342
20.	0.0511	0.3053	0.7287
21.	0.0653	0.2959	0.7051
22.	0.0689	0.3022	0.7044
23.	0.0597	0.3100	0.7207
24.	0.0554	0.3071	0.7243
25.	0.0610	0.3011	0.7137
26.	0.0651	0.3020	0.7091
27.	0.0620	0.3063	0.7156
28.	0.0585	0.3066	0.7201
29.	0.0600	0.3036	0.7164
30.	0.0627	0.3028	0.7126
31.	0.0622	0.3047	0.7144
32.	0.0602	0.3056	0.7174
33.	0.0602	0.3045	0.7167
34.	0.0616	0.3036	0.7145
35.	0.0619	0.3043	0.7146

The solution is as follows:

x0 = 0.0619

x1 = 0.3043

x2 = 0.7146