ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**
(НИУ «БелГУ»)

ИНСТИТУТ ИНЖЕНЕРНЫХ И ЦИФРОВЫХ ТЕХНОЛОГИЙ

КАФЕДРА ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ

# Отчет по лабораторной работе по дисциплине: «Вычисления в инфокоммуникациях»

# Лабораторная работа №2. «Итерационные методы решения системы линейных алгебраических уравнений»

студент заочного отделения
2 курса 12002153 группы
Пасивенко А.Ю.
Проверил:
Балабанова Татьяна Николаевна

Белгород 2023

**Цель работы:** освоить реализацию алгоритмов решения систем линейных алгебраических уравнений итерационными методами.

**1. Задание:** Реализовать метод простых итераций решения системы линейных алгебраических уравнений. Система линейных уравнений имеет вид $X = DX + F$.

**Данные:**

| | | | | | |
|---|---|---|---|---|---|
| 7. | 0,1 | 0,3 | 0,1 | 0,2 | 9 |
| | 0,4 | 0,1 | 0,1 | 0,2 | 5 |
| | 0,1 | 0,2 | 0,2 | 0,1 | 2 |
| | 0,1 | 0,2 | 0,1 | 0,2 | 9 |

Суть метода заключается в последовательном приближении к искомому значению, зная только приблизительное значение при помощи итераций.

Т.е. в данном случае СЛАУ записана в виде $X = DX + F$, где

D - квадратичная числовая матрица n – порядка;

X - n – мерный вектор, неизвестная величина, которую требуется найти.

F - n – мерный вектор (известный, заданный столбец свободных членов).

Затем необходимо задать начальное приближение, чаще всего его задают нулями.

Была составлена блок-схема:

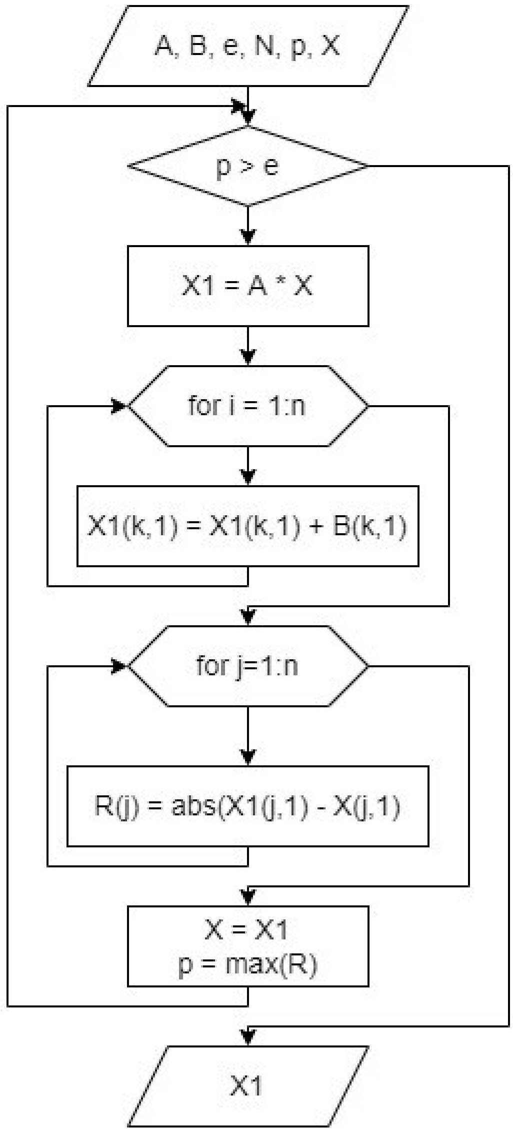


**Рисунок 1 – Блок-схема метода простых итераций**

## Листинг программы

```
clear, clc;
A=[0.1 0.4 0.1 0.1; 0.3 0.1 0.2 0.2; 0.1 0.1 0.2 0.1; 0.2 0.2 0.1 0.2];
B=[9; 5; 2; 9;];
e=0.001;
N=4;
p=1; %коэффициент апостериорной проверки
X=zeros(size(B)); %начальное приблежённое
while p>e
   X1=A*X; %вектор решений на предыдущем шаге
   for i=1:N
      X1(i,1)=X1(i,1)+B(i,1)
   end
   for j=1:N
      R(j)=abs(X1(j,1)-X(j,1)); %апостериорная проверка
```

```
    end
    X=X1; %вектор решений
    p=max(R); %апостериорная проверка
end
P=A*X+B %проверка
```

В результате выполнения программы были получены следующие результаты

```
X1 =

   17.4773
   17.1724
   16.3416
   12.2699
```

**Рисунок 2 – Результат вычислений**

При выполнении проверки были получены следующие результаты

```
P =

   17.4778
   17.1730
   16.3422
   12.2703
```

**Рисунок 2 – Результат проверки**

**Задание 2.** Реализовать метод Якоби решения СЛАУ. Система линейных уравнений имеет вид $AX = B$.

**Данные:**

| | | | | | |
|---|---|---|---|---|---|
| 7. | 1,4 | 0,1 | 0,1 | 0,2 | 5 |
| | 0,1 | 1,2 | 0,2 | 0,1 | 2 |
| | 0,1 | 0,2 | 1,1 | 0,2 | 9 |
| | 0,2 | 0,3 | 0,1 | 1,1 | 7 |

Для реализации метода Якоби необходимо понять суть метода.

Метод Якоби – один из наиболее простых методов приведения системы матрицы к виду, удобному для итерации: из 1-го уравнения матрицы выражаем неизвестное $x_1$, из 2-го выражаем неизвестное $x_2$ и т.д.

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

y(j)=(B(j)-A(j,1:j-1)*x(1:j-1)-A(j,j+1:n)*x(j+1:n))/A(j,j);

В ходе проверка должно выполняться условие, что максимальное значение из следующего выражения: |A*y-B|, не должен превышать заданного значения e.

Была составлена блок-схема:

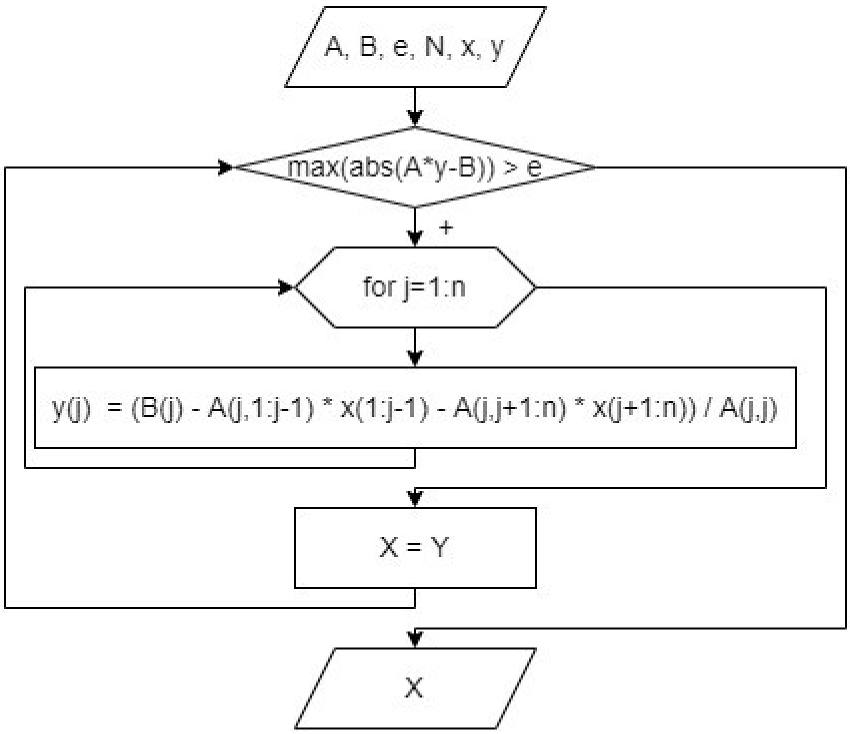


**Рисунок 4 – Блок-схема метода Якоби**

**Листинг программы**

```
clc; clear;
A=[1.4 0.1 0.1 0.2; 0.1 1.2 0.2 0.3; 0.1 0.2 1.1 0.1; 0.2 0.1 0.2 1.1];
```

```
B=[5; 2; 9; 7;];
e=0.001;
N=4;
x=zeros(size(B)); y=ones(size(B));
while max(abs(A*y-B))>e
        for j=1:N
            y(j)=(B(j)-A(j,1:j-1)*x(1:j-1)-A(j,j+1:N)*x(j+1:N))/A(j,j);
        end
    x=y
end
P=A\B %проверка
```

При вычислении были получены следующие результаты

```
x =

    4.4621
    1.5980
    2.9126
    1.6195
```

**Рисунок 5 – Результат вычислений**

При выполнении проверки были получены следующие результаты

```
P =

    4.4616
    1.5976
    2.9123
    1.6192
```

**Рисунок 6 – Результат проверки**

**Задание 3.** Реализовать метод Зейделя решения СЛАУ. Система линейных уравнений имеет вид $X = DX + F$.

**Данные:**

7. $\left|\begin{array}{cccc} 1{,}4 & 0{,}1 & 0{,}1 & 0{,}2 \\ 0{,}1 & 1{,}2 & 0{,}2 & 0{,}1 \\ 0{,}1 & 0{,}2 & 1{,}1 & 0{,}2 \end{array}\right\| \begin{array}{c} 5 \\ 2 \\ 9 \end{array}\right|$

$$\left|\begin{array}{cccc} 0{,}2 & 0{,}3 & 0{,}1 & 1{,}1 \end{array}\right\| \begin{array}{c} 7 \end{array} \right|$$

Метод Зейделя представляет собой некоторую модификацию метода простой итерации. Основная его идея заключается в том, что при вычислении (k+1)-го приближения неизвестной $x_i$ учитываются уже вычисленные ранее (k+1) – е приближения неизвестных x1, x2, ...,. И как в простом методе итерации необходимо сначала выбрать произвольно начальные приближения корней.

Следует обратить внимание на особенность метода Зейделя, которая состоит в том, что полученное в первом уравнении значение x1(l) сразу же используется во втором уравнении, а значения x1(1), x2(1) – в третьем уравнении и т. д. То есть все найденные значения x1(1) подставляются в уравнения для нахождения xi+1(1)

Была построена блок-схема.

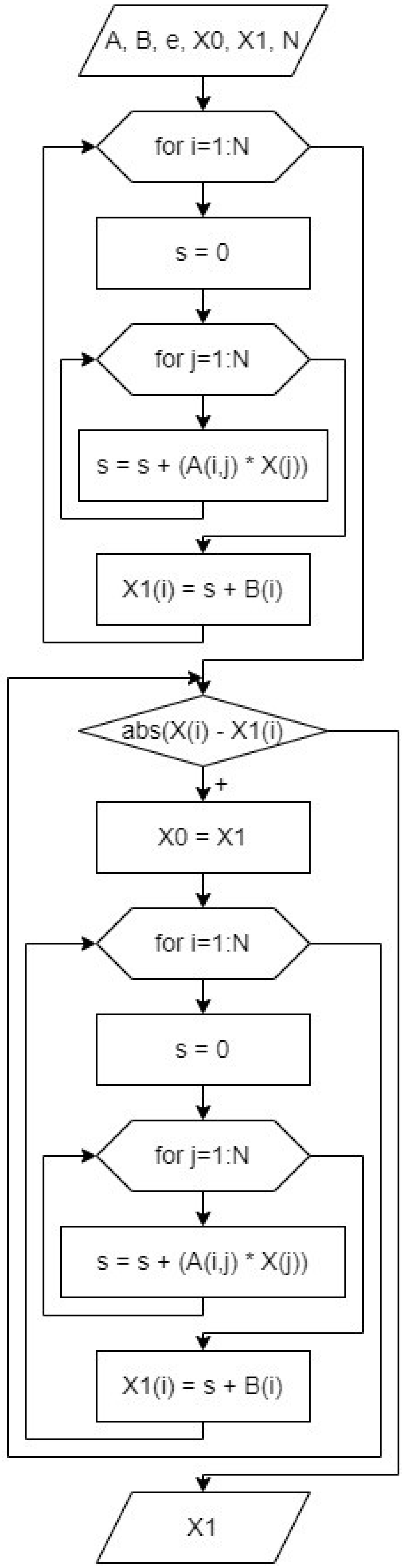


**Рисунок 7 – Блок-схема метода Зейделя**

**Листинг программы**

```
clc; clear;
A=[1.4 0.1 0.1 0.2; 0.1 1.2 0.2 0.3; 0.1 0.2 1.1 0.1; 0.2 0.1 0.2 1.1];
B=[5; 2; 9; 7;];
e=0.001;
x0=B;
x1=B;
N=4;
for i=1:1:N
   s=0;
   for j=1:1:N
      s=s+(A(i,j)*x0(j));
   end
   x1(i)=s+B(i);
end
while abs(x0(i)-x1(i))>e
   x0=x1;
   for i=1:1:N
      s=0;
      for j=1:1:N
         s=s+(A(i,j)*x0(j));
      end
      x1(i)=s+B(i)
   end
end
P=A*x1+B
```

По результатам выполнения программы были получены следующие результаты

```
x1 =

   17.4765
   17.1715
   16.3409
   12.2693
```

**Рисунок 8 – Результат вычислений**

При выполнении проверки были получены следующие результаты

```
P =

   17.4773
   17.1724
   16.3416
   12.2699
```

**Рисунок 9 – Результат проверки**

**Вывод:** В ходе выполнения лабораторной работы были изучены и реализованы алгоритмы решения систем линейных алгебраических уравнений итерационными методами. А именно были реализованы методы простой итерации, метод Якоби и метод Зейделя. Были составлены блок- схемы и реализованы в программе Matlab.