**Министерство науки и высшего образования Российской Федерации**
**Федеральное государственное бюджетное образовательное учреждение высшего образования**
**«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»**
**(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ ИУ «Информатика и системы управления»

КАФЕДРА ИУ-1 «Системы автоматического управления»

# ОТЧЕТ

## по лабораторной работе №2

## «Численные методы решения СЛАУ»

## по дисциплине

## «Методы вычислений»

**Выполнил:** **Шевченко А.Д.**

**Группа:** **ИУ1-32Б**

**Проверил:** **Бобков А.В.**

**Работа выполнена:** **01.11.2022**

**Отчет сдан:** **01.11.2022**

**Оценка:** **сдана**

**Москва 2022**

**Оглавление**

**Цель работы**

Реализация прямых итерационных численных методов решения систем линейных алгебраических уравнений в среде MatLAB. Исследование приемтсвенности, сходимости методов и времени затрачиваемого на решение.

**Различные методы решения СЛАУ:**

1. **Метод Крамера**

```
function [X] = Kramer(A,B)
%Kramer решает СЛАУ методом крамера
%A - матрица системы
%B - столбец свободных переменных

N = length(B);
X=zeros(N,1);
d=det(A);


for i=1:N
    A1=A;
    for j=1:N
        A1(j,i)=B(j);
    end
    X(i)=det(A1)/d;
end

end
```

## 2. Метод Жордана – Гаусса

```
function [X]= Gauss(A,B)
%Gauss решает СЛАУ методом Гаусса
%A - матрица системы
%B - столбец свободных переменных

N = length(B);
C=zeros(N,N+1);
X=zeros(N,1);

for i=1:N
    for j=1:N
        C(i,j)=A(i,j);
    end
    C(i,N+1)=B(i);
end

for i=1:N %для каждого диагонального элемента
    if C(i,i)==0
        for j=i+1:N
            if C(i,j)~=0
                for k=1:N+1
                    C(i,k)=C(i,k)+C(j,k);
                end
                break;
            end
        end
        if C(i,i)==0
            X=[ ];
            return
        end

    end

obr=1/C(i,i);
for j=1:N+1
    C(i,j)=C(i,j)*obr;
end

%2 получаем нули
for j=1:N
    if j~=i
        m=C(j,i);
        for k=1:N+1
            C(j,k)=C(j,k)-C(i,k)*m;

        end
    end
end
for i=1:N
    X(i)=C(i,N+1);
end
end
end
```

# 3. Метод Холецкого

```
function [L]= Halecky(A)

N = length(A);
sum=0;
L= zeros(N,N);

for i=1:N
    for j=1:i
        if i==j
            sum=0;
            for k=1:i-1
                sum = sum+L(i,k)^2;
            end
            L(i,j)=sqrt(A(i,i)-sum);

        else
            sum=0;
            for k=1:j-1
                sum = sum+L(i,k)*L(j,k);
            end
            L(i,j)=(A(i,j)-sum)/L(j,j);

        end
    end
end

end
```

## 3.1 Прямой ход

```
function [Y]= Forvard(L,B)

N=length(L);
Y=zeros(N,1);

for i=1:N
    sum=0;
    for j=1:i-1
        sum=sum+L(i,j)*Y(j);
    end

    Y(i)= (B(i)-sum)/L(i,i);
end

Y=Y';
end
```

## 3.2 Обратный ход

```
function [X]= Backvard(L,Y)

N=length(L);
X=zeros(N,1);

for k=0:N-1
    sum=0;
    for j=0:k-1
        sum=sum+L(N-j,N-k)*X(N-j);
    end
    X(N-k)=(Y(N-k)-sum)/L(N-k,N-k);
end
end
```

## 4. Метод Якоби

```
function [X] = Jacoby(A,B,eps)

N=length(B);
X=zeros(N,1);
X1=X;

err = eps+1;
while err>eps

    for i=1:N
        Summa=0;
        for j=1:N
            if j~=i
                Summa=Summa+A(i,j)*X(j);
            end
        end
        X1(i)= 1/A(i,i)*(B(i)-Summa);
    end

    err=0;
    for i=1:N
        err=err+abs(X1(i)-X(i));

    end

    X=X1;
end

end
```

## 5. Метод Зйделя

```
function [X] = Zeidel(A,B,eps)

N=length(B);
X=zeros(N,1);
X1=X;
err = eps+1;

while err>eps

    for i=1:N
        Summa1=0;
        Summa2=0;
        for j=i+1:N
            Summa2=Summa2+A(i,j)*X(j);
        end

        for j=1:i-1
            Summa1=Summa1+A(i,j)*X1(j);
        end

        X1(i)= 1/A(i,i)*(B(i)-Summa1-Summa2);
    end
    err=0;
    for i=1:N
        err=err+abs(X1(i)-X(i));

    end

    X=X1;
end
end
```

## 6. Метод Релаксаций

```
function [X] = Relaxation(A,B,eps,omega)

N=length(B);
X=zeros(N,1);
X1=X;

err = eps+1;
while err>eps
    for k=1:N
        %for i=1:N
            Summa1=0;
            Summa2=0;

            for j=k+1:N
                Summa2=Summa2+A(k,j)*X(j);
            end

            for j=1:k-1
                Summa1=Summa1+A(k,j)*X1(j);
            end

            X1(k)= (1-omega)*X(k) +omega/A(k,k)*(B(k)-Summa1-Summa2);
    end

    err=0;
    for i=1:N
        err=err+abs(X1(i)-X(i));
    end
    X=X1;
end
end
```

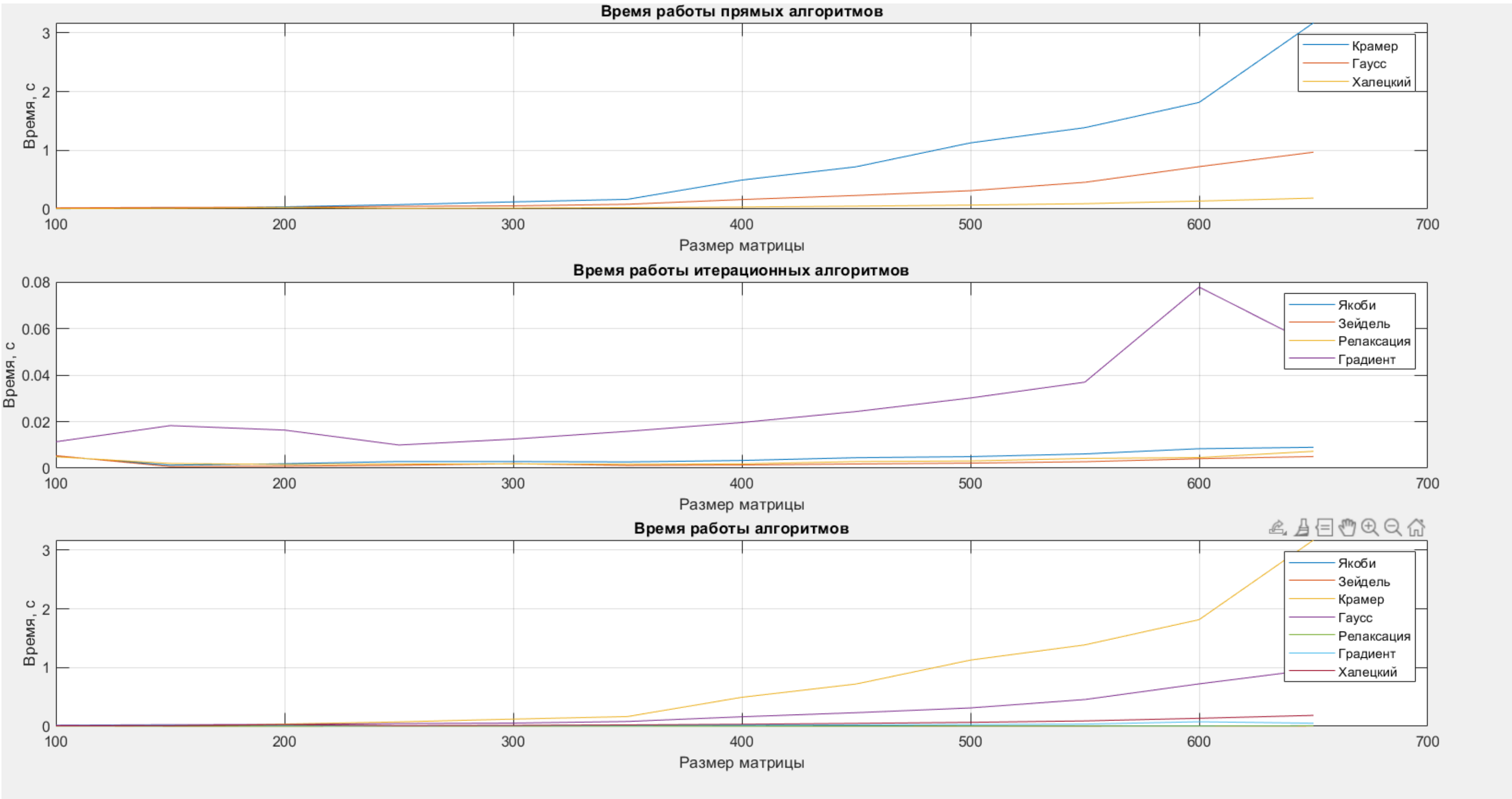
## 7. Метод Градиентного Спуска

```
function [X] = Gradient(A,B,eps,h)

N=length(B);
X=zeros(N,1);
X1=X;
E=zeros(N,1);
err = eps+1;

while err>eps
    for i=1:N
        Summa1=0;
        for j=1:N
            Summa1=Summa1+A(i,j)*X(j);
        end
        E(i)=Summa1-B(i);
    end
    Summa2=0;
    for k=1:N
        for i=1:N
            Summa2=Summa2+E(i)*A(i,k);
        end
        X1(k)=X(k)-h*Summa2;
    end
    err=0;
    for i=1:N
        err=err+abs(X1(i)-X(i));
    end
    X=X1;
end
end
```

## Сравнение эффективности методов




**Вывод:** метод Зейделя является наиболее эффективным методом среди представленных.

Метод Халецкого является самым эффективным среди прямых.

Метод Зейдяля является самым эффективным среди итерационных методов.