

CODIGOS IMPLEMETADOS

Código Descomposición LU:

```
function [L,U] = descomlu(A)

    [n,n]=size(A);
    L=eye(n); %eye= matriz identidad
    U=eye(n);

    for k=1:n
        L(k,k)=1;
        U(1,k)=A(1,k);

        for i=k:n
            U(k,i)=A(k,i)-L(k,1:k-1)*U(1:k-1,i);

        end

        for i=k+1:n
            L(i,k)=(A(i,k)-L(i,1:k-1)*U(1:k-1,k))/U(k,k);

        end
    end
end
```

Código Solución LU:

```
function [L,U,D,X] = solu_lu(A,B)

    [n,n]=size(A);
    L=eye(n); %eye= matriz identidad
    U=eye(n);

    for k=1:n
        L(k,k)=1;
        U(1,k)=A(1,k);

        for i=k:n
            U(k,i)=A(k,i)-L(k,1:k-1)*U(1:k-1,i);

        end

        for i=k+1:n
            L(i,k)=(A(i,k)-L(i,1:k-1)*U(1:k-1,k))/U(k,k);

        end
    end

    end

    %hacemos sustitución hacia adelante
```

```

D=B*0; %Vector de ceros para guardar la solución
if L(n,n)==0
    fprintf('No hay solución única, puede haber o no solución');

else
    for i=1:1:n %sustituyendo "desde atras hacia adelante"
        suma=0;

        for j=(1):i-1
            suma=suma+L(i,j)*D(j);

        end
        D(i)=(B(i)-suma);

    end
end

%hacemos sustitución hacia atrás

X=B*0; %Vector de ceros para guardar la solución
if U(n,n)==0
    fprintf('No hay solución única, puede haber o no solución');

else
    X(n)=D(n)/U(n,n); %para que los for de abajo queden bonitos

    for i=(n-1):-1:1 %sustituyendo "desde atras hacia adelante"
        acum=0;

        for j=(i+1):n
            acum=acum+U(i,j)*X(j);

        end
        X(i)=(D(i)-acum)/U(i,i);

    end
end
end
end

```

Código de la inversa de una matriz

```

function [L,U,D,X, invA] = inv_lu(A)

[n,n]=size(A);
L=eye(n); %eye= matriz identidad
U=eye(n);

for k=1:n
    L(k,k)=1;
    U(1,k)=A(1,k);

    for i=k:n
        U(k,i)=A(k,i)-L(k,1:k-1)*U(1:k-1,i);
    end
end
end

```

```

end

for i=k+1:n
    L(i,k)=(A(i,k)-L(i,1:k-1)*U(1:k-1,k))/U(k,k);

end

end

%hacemos sustitución hacia adelante
B=eye(n);
D=B*0; %Vector de ceros para guardar la solución
X=B*0; %Vector de ceros para guardar la solución

for k=1:n
    if L(n,n)==0
        fprintf('No hay solución única, puede haber o no solución');

    else
        for i=1:1:n %sustituyendo "desde atras hacia adelante"
            suma=0;

            for j=(1):i-1
                suma=suma+L(i,j)*D(j,k);

            end
            D(i,k)=(B(i,k)-suma);

        end
    end

    %hacemos sustitución hacia atrás

    if U(n,n)==0
        fprintf('No hay solución única, puede haber o no solución');

    else
        X(n,k)=D(n,k)/U(n,n); %para que los for de abajo queden
bonitos

        for i=(n-1):-1:1 %sustituyendo "desde atras hacia adelante"
            acum=0;

            for j=(i+1):n
                acum=acum+U(i,j)*X(j,k);

            end
            X(i,k)=(D(i,k)-acum)/U(i,i);

        end
    end
end
invA=X;
end

```