

EJERCICIO 1.

% Practica 4 CNU

n=3

x=[1;2;3]

**% En A se almacena la matriz original, y después de aplicar el algoritmo se deposita en A la matriz LU
% resultante.**

A=rand(n)

w=[0;0;0]

% En Z se guarda la matriz A original.

Z=A

% b es el vector de términos independientes.

b=A*x;

% ALGORITMO DE DESCOMPOSICION LU.

for k=1:n-1

if (A(k,k)==0) error ('A(k,k) es igual a 0.');

else

w(k+1:n)=A(k,k+1:n);

for i=k+1:n

u=A(i,k)/A(k,k);

A(i,k)=u;

for j=k+1:n, A(i,j)=A(i,j)-(u*w(j));

end;

end;

end;

end;

% Comprobación de los resultados.

U=triu(A);

L=tril(A);

L(1,1)=1; L(2,2)=1; L(3,3)=1;

y=L\b

x=U\y

EJERCICIO 2.

Sucede que el algoritmo falla, porque se produce una división por cero en la línea donde se calculan las a_i para eliminar la columna.

Se podría solucionar implementado un algoritmo que en cada momento obtenga el elemento máximo de la columna con la que estamos trabajando, que es el algoritmo que viene a continuación.

EJERCICIO 3.

% Practica 4 de CNU, ejercicio 2.

% Inicialización de las variables.

r=[1;2;3]

n=3

x=[1;2;3]

w=[0;0;0]

A=[0.7532 0.3296 0.4306; 0.5916 0.2321 0.4036; 0.8229 0.5365 0.9873]

Z=A

b=A*x

I=[1 0 0; 0 1 0; 0 0 1]

% ALGORITMO DE DESCOMPOSICION LU CON PIVOTACION PARCIAL

for k=1:n-1

aux=A(k,k); p=k

for i=k:n

if A(i,k)>=aux

```

aux=A(i,k)

p=i

end;

end;

r(k)=p;

for j=k:n

aux1=A(k,j);

A(k,j)=A(p,j);

A(p,j)=aux1;

end;

w(k+1:n)=A(k,k+1:n);

for i=k+1:n

u=A(i,k)/A(k,k);

A(i,k)=u;

for j=k+1:n, A(i,j)=A(i,j)-(u*w(j));

end;

end;

end;

end;

% Comprobación de los resultados.

L=tril(A);

U=triu(A);

L(1,1)=1;L(2,2)=1;L(3,3)=1;

% Aquí se halla la matriz de permutación P.

P=I;

for k=1:n

```

```
aux2=I;  
  
aux=aux2(1:n,k);  
  
aux2(1:n,k)=aux2(1:n,r(k));  
  
aux2(1:n,r(k))=aux;  
  
P=P*aux2;  
  
end;
```

EJERCICIO 4.

El hecho de que los resultados obtenidos con los dos algoritmos anteriores sean diferentes a los que se consiguen con la **lu** de MATLAB se debe a que en el primer algoritmo no se permitía que la diagonal principal contuviera ceros y además no se efectuaba ninguna pivotación, y en el segundo algoritmo la pivotación realizada era parcial, mientras que en la **lu** de MATLAB la pivotación que se hace es total.