***Algoritmo de Choleski***

function [ L , Lt ] = choleski( A )

%Factorizacion LLt - Algoritmo de Choleski

if (positiva(A))

***Anexo: Matiz positiva***

function [ sal ] = positiva( A )

%Regresa '1' si la función es …

%positiva o '0' si no lo es.

[a b]=size(A);

if (a==b)

for i=1:a

if (det(A(1:i,1:i))<=0)

sal=0;

break;

else

sal=1;

end

end

else

sal=0;

end

end

[n x]=size(A);

L=zeros(n);

L(1,1)=sqrt(A(1,1));

for j=2:n

L(j,1)=A(j,1)/L(1,1);

end

for i=2:n-1

for k=1:i-1

L(i,i)=L(i,i)+L(i,k)^2;

end

L(i,i)=sqrt(A(i,i)-L(i,i));

for j=i+1:n

for k=1:i-1

L(j,i)=L(j,i)+L(j,k)\*L(i,k);

end

L(j,i)=(A(j,i)-L(j,i))/L(i,i);

end

end

for k=1:n-1

L(n,n)=L(n,n)+L(n,k)^2;

end

L(n,n)=sqrt(A(n,n)-L(n,n));

Lt=transpose(L);

else

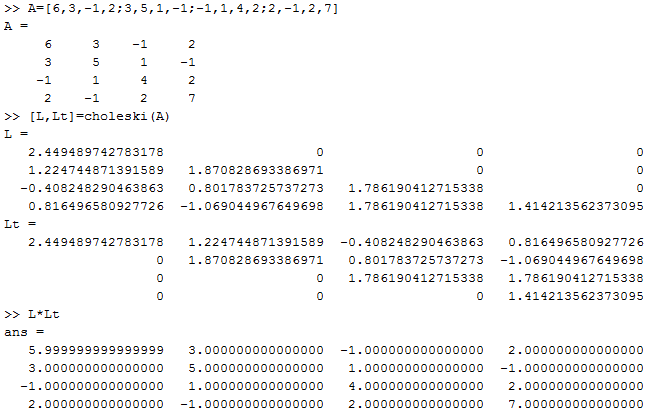
disp('la matriz no es positiva');

L=[];

Lt=[];

end

end

******

***Factorización LDLt***

function [ L , D , Lt ] = LDLt( A )

%Factorizacion LLt - Algoritmo de Choleski

if (positiva(A))

[n x]=size(A);

L=eye(n);

D=zeros(n);

for i=1:n

for j=1:i-1

V(j)=L(i,j)\*D(j,j);

D(i,i)=D(i,i)+L(i,j)\*V(j);

end

D(i,i)=A(i,i)-D(i,i);

for j=i+1:n

for k=1:i-1

L(j,i)=L(j,i)+L(j,k)\*V(k);

end

L(j,i)=(A(j,i)-L(j,i))/D(i,i);

end

end

Lt=transpose(L);

else

disp('la matriz no es positiva');

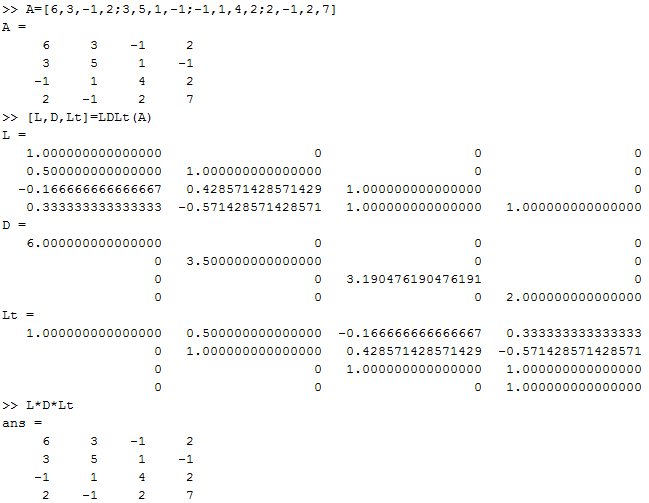
L=[];

D=[];

Lt=[];

end

end

******