

FACTORIZACIÓN DIRECTA DE MATRICES

Para el desarrollo de ésta sección, se considera el sistema $Ax = b$ donde mediante la factorización directa de las matrices, se busca descomponer la matriz A siendo el resultado del producto de dos matrices triangulares L y U. Estas matrices son triangular inferior (L) y triangular superior (U).

Cabe mencionar, que para la solución de estos sistemas mediante factorización, la matriz A debe ser de tamaño $n \times m$, siendo esta una matriz cuadrada. A partir de esto, se considera el sistema de ecuaciones

$$\begin{pmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$$

Para este caso, se consideró una matriz 4×4 .

$$\begin{pmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{pmatrix} = \begin{pmatrix} 36 & 3 & -4 & 5 \\ 5 & -45 & 10 & -2 \\ 6 & 8 & 57 & 5 \\ 2 & 3 & -8 & -42 \end{pmatrix}$$

Con la imagen anterior, se muestra las respectivas matrices L y U que son igualadas a la matriz principal A. La finalidad de la factorización directa de matrices, es resolver este sistema encontrando las respectivas l_{ij} y u_{ij} , esto se logra mediante tres posibles soluciones. Dichas soluciones se conocen como factorización de Doolittle, factorización de Crout y factorización de Cholesky.

- La factorización de Doolittle se diferencia ya que los elementos de la diagonal de la matriz L son iguales a 1 así: $l_{ii} = 1$
- La factorización de Crout, se reconoce mediante su diagonal, ya que los elementos de la matriz U son iguales a 1 así: $u_{ii} = 1$
- Ahora la factorización de Cholesky, los elementos de la diagonal de L son iguales a los elementos de la diagonal de U así: $l_{ii} = u_{ii}$

Sabiendo esto, se procede al cálculo de los elementos de las matrices L y U como se mencionó anteriormente, esto se consigue dependiendo de cuál método se escoja. Para Doolittle, se empiezan multiplicando todos los elementos de la primera fila de L por todos los elementos de la primera columna de U ya que cuando se usa Doolittle, la fila 1 de L se conoce completamente. Por el contrario para Crout, los elementos que se conocen completamente son los elementos de la primera columna de U, por lo tanto, mediante Crout se procede multiplicando todas las filas de L por la primera columna de U.

Con lo anterior, para la factorización de una matriz $n \times n$, la etapa k del proceso se resuelve siguiendo las siguientes ecuaciones:

Para calcular todos los valores de l_{ik} tenemos que:

$$L_{(i)} * U^{(k)} = a_{ik}$$

Con esto, se pueden encontrar el valor de l_{kk} .

Ahora, para calcular los valores de la fila k de la matriz U tenemos que:

$$L_{(k)} * U^{(j)} = a_{kj}$$

Como conclusión, se tiene que para el elemento l_{ik} :

$$l_{ik} = a_{ik} - \sum_{p=1}^{k-1} l_{ip} * u_{pk}$$

Y para el elemento u_{kj} :

$$u_{kj} = \frac{a_{kj} - \sum_{p=1}^{k-1} l_{kp} * u_{pj}}{l_{kk}}$$

Después de tener completas las matrices L y U, se procede a encontrar el sistema $Lz = b$, al encontrar el vector "z", se continua con encontrar el sistema $Ux = z$. De este modo se logró encontrar el sistema inicial $Ax = b$ mediante factorización directa de matrices; en este caso, mediante Crout.

A continuación, se mostrara el código de todo este procedimiento, programado mediante "Octave" y utilizando las bases y conceptos adquiridos en clase.

CÓDIGO

FACTORIZACIÓN MEDIANTE DOOLITTLE

```
% FACTORIZACION LU DOOLITTLE
```

```
clc;
```

```
clear all;
```

```
    A=matrizA;
```

```
    b=vectorb;
```

```
    AB=[A,b];
```

```
    [n,m]=size(A);
```

```
%Hacer la factorizacion LU
```

```
if n==m
```

```
    for k =1:n;
```

```
        L(k,k)=2;
```

```
        suma1 = 0;
```

```
        for p=1:k-1
```

```
            suma1=suma1+L(k,p)*U(p,k);
```

```
        end
```

```
        U(k,k)=(A(k,k)-suma1);
```

```
        for i=k+1:n
```

```
            suma2=0;
```

```
            for p=1:(k-1)
```

```
                suma2=suma2+L(i,p)*U(p,k);
```

```
            end
```

```
            L(i,k)=(A(i,k)-suma2)/U(k,k);
```

```
        end
```

```
        for j=k+1:n
```

```
            suma3=0;
```

```
            for p=1:k-1
```

```
                suma3=suma3+L(k,p)*U(p,j);
```

```
            end
```

```

                U(k,j)=(A(k,j)-suma3)/L(k,k);
            end
        end
        fprintf("\n\n La matriz L es: \n\n")
        L
        fprintf("La matriz U es: \n\n")
        U
        L*U
    else
        fprintf("La matriz no es cuadrada \n\n")
    end
    %Encontrar el vector Z
    Z=b;
    for k=1:n
        suma1=0;
        for i=1:n
            if k==i
            else
                suma1 = suma1 +L(k,i)*Z(i,1);
            end
        end
        Z(k,1)=(b(k,1)-suma1)/L(k,k);
    end
    fprintf("La matriz Z es: \n\n")
    Z
    %Encontrar el vector X
    X=Z;
    for k=n:-1:1
        suma1=0;
        for i=n:-1:1

```

```

        if k==i
        else
            suma1 = suma1 +U(k,i)*X(i,1);
        end
    end
    X(k,1)=(Z(k,1)-suma1)/U(k,k);
end
%Mostrar los resultados de X
for k=1:n;
    fprintf("X%g = %g \n",k,X(k,1));
end
inv(A)*b

```

FACTORIZACIÓN MEDIANTE CROUT

```

A=matrizA;
b=vectorb;
[n,m]=size(A);
C=[A,b];
% la matriz C, representa la forma de la matriz aumentada [Ab]
if n==m
    for k=1:n
        %La instrucción iterativa for permite repetir estamentos a un
        %numero específico de veces
        u(k,k)=1; %princio del metodo
        suma=0;
        for p=1:k-1
            suma=suma+L(k,p)*u(p,k);
        end
        L(k,k)=(A(k,k)-suma);
    end
end

```

```

for i=k+1:n
    suma=0;
    for r=1:k-1
        suma=suma+L(i,r)*u(r,k);
    end
    L(i,k)=(A(i,k)-suma); %obtencion de la matriz L
end
for j=k+1:n
    suma=0;
    for s=1:k-1
        suma=suma+L(k,s)*u(s,j);
    end
    u(k,j)=(A(k,j)-suma)/L(k,k); %obtencion de la matriz U
end
end
memoriau=1; %calculo del determinante de u
memorial=1; %calculo del determinante inicial de L
for i=1:n
    memoriaL=memorial*L(i,i);
end
producto=memorial*memoriau; %calculo del determinante total
if producto~=0
for i=1:n
    suma=0;
    for p=1:i-1
        suma=suma+L(i,p)*z(p);
    end
    z(i)=(b(i)-suma)/L(i,i); %obtencion del vector Z
end
for i=n:-1:1

```

```

    suma=0;
    for p=(i+1):n
        suma = suma+u(i,p)*x(p);
    end
    x(i)=(z(i)-suma)/u(i,i); % solcion, calculos de las variables
end
else
    fprintf('\nEl determinante es igual a cero, por lo tanto el sistema tiene infinita o ninguna
solucion\n')
end
end
fprintf('\n Matriz Ab:\n')
disp(C)
fprintf('\n Matriz L:\n')
disp(L)
fprintf('\n Matriz U:\n')
disp(u)
fprintf('\n El vector Z:\n')
disp(z)
fprintf('\n\nLa solucion de X1 hasta Xn es:\n');
    %a continuacion de utiliza una instruccion for, para mostrar el usuario,
%los resultados de una manera mas ordenada
for i=1:n
    xi=x(1,i);
    fprintf('\nX%g=',i)
    disp(xi);
end

```

FACTORIZACIÓN MEDIANTE CHOLESKY

```
function []=cholesky(A,b)
clc;
clear all;
    A=matrizA;
    b=vectorb;
    AB=[A,b]
    [n,m]=size(A);
for k=1:n
    Acumulador = 0;
    for p=1:k-1
        Acumulador = Acumulador + L(k,p)*U(p,k);
    end
    L(k,k)= sqrt(A(k,k)- Acumulador);
    U(k,k) = L(k,k);

    for i=k:n
        Acumulador2 = 0;
        for p=1:k-1
            Acumulador2= Acumulador2 + L(i,p)*U(p,k);
        end
        L(i,k) = (A(i,k)- Acumulador2)/L(k,k);
    end
    for j= k:n
        Acumulador3 = 0;
        for p = 1:k-1
            Acumulador3 = Acumulador3 + L(k,p)*U(p,j);
        end
        U(k,j)= (A(k,j)- Acumulador3)/L(k,k);
    end
end
```

```

end
disp ('L=')
disp(L)
disp('U=')
disp (U)
[Lb]= Aumentada(L,b);
for i=1:n
    Sumatoria=0;
    for p=1:i-1
        Sumatoria=Sumatoria+Lb(i,p)*z(p,1);
    end
    z(i,1)=(Lb(i,n+1)-Sumatoria)/Lb(i,i);
end
[Uz]=Aumentada(U,z);
function [X]=SustitucionRegresiva(UB,n)
for i=n:-1:1
    Sumatoria=0;
    for p=i+1:n
        Sumatoria=Sumatoria+UB(i,p)*X(p,1);
    end
    X(i,1)=(UB(i,n+1)-Sumatoria)/UB(i,i);
end

disp('X=')
disp(X)
end

```

Ahora gracias a lo anterior, se mostrara ejecutándose el programa de Crout, el ejemplo tiene el siguiente sistema:

$$\begin{pmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{pmatrix} \begin{pmatrix} 1 & u_{12} & u_{13} & u_{14} \\ 0 & 1 & u_{23} & u_{24} \\ 0 & 0 & 1 & u_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 36 & 3 & -4 & 5 \\ 5 & -45 & 10 & -2 \\ 6 & 8 & 57 & 5 \\ 2 & 3 & -8 & -42 \end{pmatrix}$$

El vector b es igual a:

$$\begin{pmatrix} -20 \\ 69 \\ 96 \\ -32 \end{pmatrix}$$

Los resultados arrojados por el programa fueron:

```

Less
Matriz Ab:
36  3  -4  5  -20
 5 -45 10  -2  69
 6  8 57  5  96
 2  3 -8 -42 -32

Matriz L:
36.00000  0.00000  0.00000  0.00000
 5.00000 -45.41667  0.00000  0.00000
 6.00000  7.50000 59.40979  0.00000
 2.00000  2.83333 -7.11927 -41.99989

Matriz U:
1.00000  0.08333 -0.11111  0.13889
0.00000  1.00000 -0.23242  0.05933
0.00000  0.00000  1.00000  0.06264
0.00000  0.00000  0.00000  1.00000

El vector z:
-0.55556 -1.58043  1.87152  0.31160

La solución de X1 hasta Xn es:
líneas 1-24 -- (f)orward, (b)ack, (q)uit

```

En la imagen está representado el sistema $Ax = b$, la matriz L diagonal inferior, la matriz U diagonal superior. Luego hace el proceso de hallar el sistema $Lz = b$, mostrando el vector z .

```

La solución de X1 hasta Xn es:
X1=-0.29568
X2=-1.1685
X3= 1.8520
X4= 0.31160
líneas 9-32/32 (END) -- (f)orward, (b)ack, (q)uit_

```

Al final, la solución del sistema $Ux = z$, encontrando el vector x .