

PRÁCTICA 7-CRISTIAN PINELA PANIAGUA.

Resolver por el método de LU el siguiente sistema de ecuaciones lineales $Ax=b$:

$$\begin{bmatrix} 2 & 1 & -1 & 1 & -3 \\ 1 & 0 & 2 & -1 & 1 \\ 0 & -2 & -1 & 1 & -1 \\ 3 & 1 & -4 & 0 & 5 \\ 1 & -1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 7 \\ 2 \\ -5 \\ 6 \\ 3 \end{bmatrix}$$

Cuya solución es conocida y es (1.91813, 1.96491, -0.988304, -3.19298, -1.1345)

CÓDIGO FUENTE DEL PROGRAMA

```
////////////////////////////////////
//Cristian Pinela Paniagua. 15/10/2013. Práctica 7-Método de LU////////
////////////////////////////////////

//Librerías

#include <iostream>

#include <cmath>

#include <fstream>

using namespace std;

//Establecemos la dimensión de la matriz como una constante para simplificar el trabajo

const int n=5;

//Función

void LU(double A[n][n], double b[n], double x[n], const int n);

int main()

{

double A[n][n]={2,1,-1,1,-3,1,0,2,-1,1,0,-2,-1,1,-1,3,1,-4,0,5,1,-1,-1,-1,1}; //Matriz de
parámetros

double b[n]={7,2,-5,6,3}; //Matriz de soluciones

double x[n]; //Matriz de incógnitas

//Lo que tenemos que abordar en este problema es que al multiplicar  $A*x=b$  tenemos una matriz
cuyos //valores desconocemos, tenemos

//que averiguarla para cumplir la igualdad.

cout<<"La matriz A introducida es: "<<endl;

for (int i=0; i<n; i++)
```

```

for (int j=0; j<n; j++)
{
    cout<<A[i][j]<<" ";
    if (j==n-1)
    {cout<<endl;}
}

cout<<endl<<"Y el espacio b de soluciones es: "<<endl;

for (int i=0; i<n; i++)
{
    cout<<b[i]<<" "<<endl;
}

LU(A,b,x,n);

cout<<endl<<"Y la matriz x tal que  $A \cdot x = b$  es: "<<endl;

for (int i=0; i<n; i++)
{
    cout<<"x"<<i+1<<"= "<<x[i]<<endl;
}

cout<<"Terminado!"<<endl;

//Líneas comunes en cada programa con dev c++

system ("PAUSE");

return 0;
}

//FUNCIÓN LU

void LU(double A[n][n], double b[n], double x[n], const int n)
{
    //Tenemos primero que definir dos matrices, la L y la U:

    double L[n][n], U[n][n], z[n];

    double sumU, sumL, sumz,sumx;

    //Ahora que ya están definidas tenemos que darles valores, para eso utilizaremos un loop.

    //Loops utilizados para hallar LU

    for(int i=0; i<=n-1;i++)

    for(int j=0; j<=n-1;j++)

    {

        //Caso general de L y U

```

```

if (i<=j) //Condicional para hacer U
{
    if (i<j) //Condicional del caso triángulo superior de L
    {L[i][j]=0;}

    if (i==j) //Condicional del caso diagonal de L
    {L[i][j]=1;}

    sumU=0;

    for(int k=0; k<=i-1; k++) //Loop con respecto a sumU
    {sumU=sumU+L[i][k]*U[k][j];}

    U[i][j]=A[i][j]-sumU;
}

if (i>j) //Condicional para hacer L
{
    U[i][j]=0;

    sumL=0;

    for(int k=0; k<=j-1; k++) //Loop con respecto a sumL
    {sumL=sumL+L[i][k]*U[k][j];}

    L[i][j]=(A[i][j]-sumL)/U[j][j];
}

}

//Loops utilizados para hallar z
for(int i=0; i<=n-1;i++) //Loop principal
{sumz=0;

    for(int j=0; j<=i-1;j++) //Loop utilizado para hallar el sumatorio de z
    {

        sumz=sumz+L[i][j]*z[j];

    }

    z[i]=b[i]-sumz;
}

//Loops utilizados para hallar x
for(int i=n-1; i>=0; i--) //Loop principal
{

    double cx=0, sumx=0;

    for (int j=i+1; j<=n-1; j++) //Loop utilizado para hallar el sumatorio

```

```

{
sumx=sumx+U[i][j]*x[j];
}
x[i]=(z[i]-sumx)/U[i][i];
}
}

```

RESULTADO.

```

C:\Users\Portatil\Documents\Proyecto.exe
La matriz A introducida es:
2 1 -1 1 -3
1 0 2 -1 1
0 -2 -1 1 -1
3 1 -4 0 5
1 -1 -1 -1 1

Y el espacio b de soluciones es:
7
2
-5
6
3

Y la matriz x tal que A*x=b es:
x1= 1.91813
x2= 1.96491
x3= -0.988304
x4= -3.19298
x5= -1.1345
Terminado!
Presione una tecla para continuar . . .

```

COMENTARIOS.

- Ha sido una práctica sencilla, no conocía el método en cuestión y muy instructiva para entender mejor los loops.
- He entregado la práctica un poco desordenada al comienzo porque todo el código lo terminé en 1h, pero al ejecutarlo me daba constantemente un error de “debe cerrarse el programa” y he estado cerca de 1,30h intentado resolverlo, lo he resuelto poco antes de terminar la clase y era simplemente que cuando creo un loop, puse i en vez de j en la condición y por eso daba error.
- Computacionalmente no es muy pesado, por lo que creo que resulta útil, además, resulta muy fácil cambiar el programa a otra matriz o incluso hacer todo como inputs, no encuentro limitaciones salvo en el caso de que el sistema no sea lineal.
- He utilizado dev c++ esta vez por probar y me ha gustado más que el software de mac, ya que es mucho más complicado.