Matemáticas II - Grado en Ing. Informática - Curso 2021/22

- Sesión 3 de Prácticas -

FLU, Determinate, Grafos y Espacio vectoriales

1. Factorización $L \cdot U$ de una matriz. Aplicación a la resolución de sistemas.

Ejercicio 1: Resolución de sistemas usando factorización $L \cdot U$ con Matlab. Resolveremos el siguiente S.E.L.

$$\begin{aligned} \text{4x-2y-z} &= 9 \\ \text{5x+y-z} &= 7 \\ \text{x+2y-z} &= 12 \end{aligned}$$

NOTA: Recordemos que para la obtención de la matriz U únicamente podemos utilizar transformaciones del tipo $F_{ij}(c)$ (es decir, matrices elementales de la forma $E_{ij}(c)$). La necesidad de utilizar otro tipo de transfomación es indicativo de la no existencia de factorización $L \cdot U$.

- Obtenemos la matriz U aplicando el método de Gauss a la matriz de coeficientes del sistema (utilizamos las matrices elementales descritas en la práctica 2):

$$C = \begin{pmatrix} 4 & -2 & -1 \\ 5 & 1 & -1 \\ 1 & 2 & -1 \end{pmatrix}$$

a) Introducimos dicha matriz en Matlab:

$\gg$ `C=[4 -2 -1; 5 1 -1; 1 2 -1]`

`C=`

$$\begin{matrix} 4 & -2 & -1 \\ 5 & 1 & -1 \\ 1 & 2 & -1 \end{matrix}$$

b) Creamos las matrices elementales para hacer ceros debajo del pivote de la primera fila.

$\gg$ `E1=eye(3); E1(2,:)= E1(2,:)-(5/4)*E1(1,:);`
$\gg$ `C=E1*C`

`C=`

$$\begin{matrix} 4 & -2 & -1 \\ 0 & 7/2 & 1/4 \\ 1 & 2 & -1 \end{matrix}$$

$\gg$ `E2=eye(3); E2(3,:)= E2(3,:)-(1/4)*E2(1,:);`
$\gg$ `C=E2*C`

`C=`

$$\begin{matrix} 4 & -2 & -1 \\ 0 & 7/2 & 1/4 \\ 0 & 5/2 & -3/4 \end{matrix}$$

```
≫ E3=eye(3); E3(3,:)= E3(3,:)-(5/7)*E3(2,:);
≫ C=E3*C
C=
```

$$\begin{matrix} 4 & -2 & -1 \\ 0 & 7/2 & 1/4 \\ 0 & 0 & -13/14 \end{matrix}$$

Por lo tanto hemos obtenido la matriz U:

$$U = \begin{pmatrix} 4 & -2 & -1 \\ 0 & 7/2 & 1/4 \\ 0 & 0 & -13/14 \end{pmatrix}$$

- Obtención de la matriz L:

Notemos que para obtener U hemos realizado en tres pasos la operación matricial:

$$E3 \cdot E2 \cdot E1 \cdot C = U$$

Para obtener la matriz L multiplicaremos por las inversas de las matrices elementales hasta despejar C. Así obtendremos la expresión:

$$C = E1^{-1} \cdot E2^{-1} \cdot E3^{-1} \cdot U$$

Luego $L = E1^{-1} \cdot E2^{-1} \cdot E3^{-1}$.

En Matlab haremos:

```
≫ L=inv(E1)*inv(E2)*inv(E3)
L=
```

$$\begin{matrix} 1 & 0 & 0 \\ 5/4 & 1 & 0 \\ 1/4 & 5/7 & 1 \end{matrix}$$

- Resolución del sistema:

Partimos de: $C \cdot X = b$,

siendo C la matriz de coeficientes del sistema, $X = (x, y, z)$ y $b = (9, 7, 12)^T$.

Descomponemos la matriz $C = L \cdot U$, así el sistema anterior queda:

$$L \cdot U \cdot X = b$$

Llamamos $U \cdot X = Y$, siendo $Y = (y_1, y_2, y_3)^T$. Con esto, el sistema anterior queda:

$$L \cdot Y = b$$

Introducimos el vector de términos independientes y obtenemos el valor de $Y = (y_1, y_2, y_3)^T$, con la orden $L \setminus b$, es decir:

≫ `b=[9 7 12]’ % las comillas hacen que b sea un vector columna. Tambien se consigue en la forma: b=[9;7;12].`
≫ `Y=L \ b`

`Y =`

$$\begin{matrix} 9 \\ -17/4 \\ 179/14 \end{matrix}$$

Resolvemos el sistema $U \cdot X = Y$, usando la misma orden:

≫ `X=U \ Y`

`X =`

$$\begin{matrix} -17/13 \\ -3/13 \\ -179/13 \end{matrix}$$

2. Grafos:representación a partir de matriz de adayacencia

Ejercicio 2: Representa el grafo correspondiente a la siguiente matriz de adyacencia e identifica si posee loops.

a): Introducimos la matriz de adyacencia en Matlab:

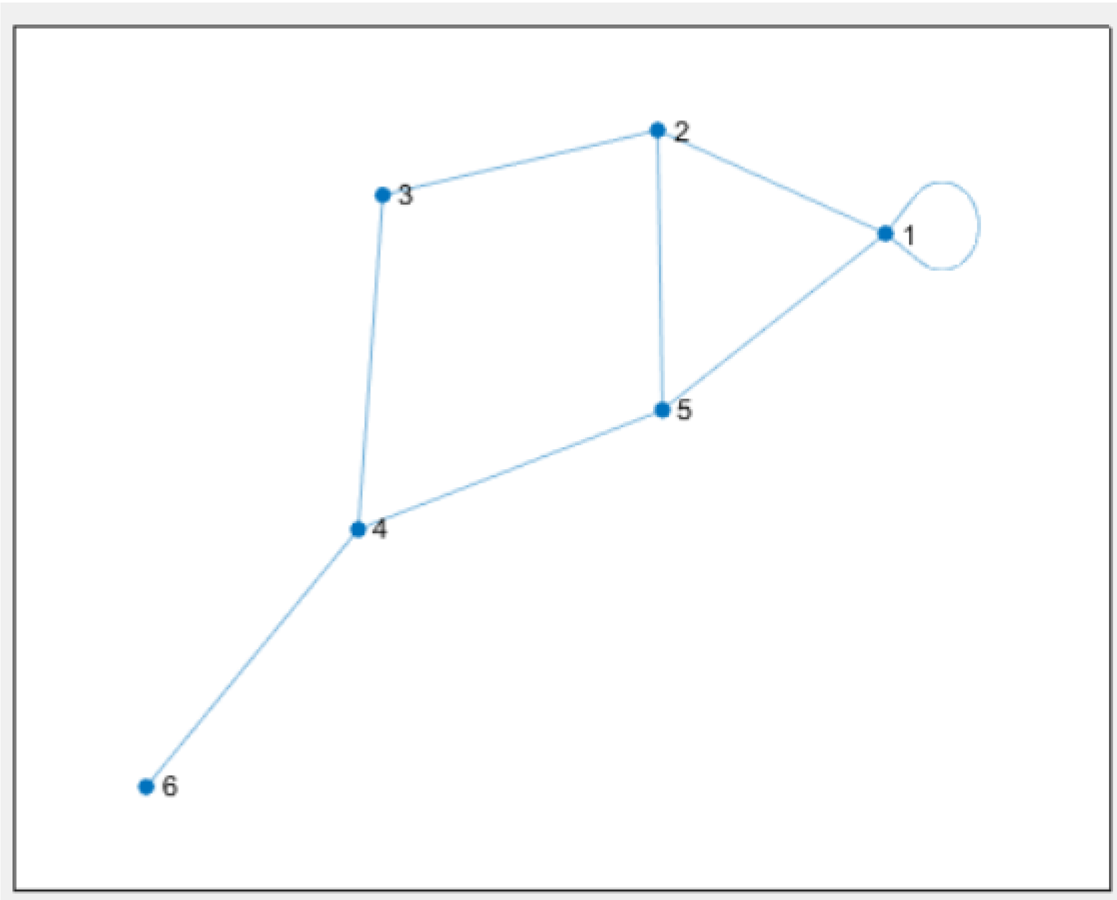
$$\begin{bmatrix} 2 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

b): Definimos el grafo usando el comando graph(A)

```
>> G=graph(A)

G =

  graph with properties:

    Edges: [8×2 table]
    Nodes: [6×0 table]
```

c): Dibujamos el grafo usando el comando plot

```
>> plot(G)
```



3. Determinantes: Submatrices, menores y menores principales.
Ejercicio 3: Cálculo de determinantes
Calcular el determinante de la siguiente matriz

$$R = \begin{pmatrix} -5 & 1 & -3 & 3 \\ 1 & 0 & 2 & -2 \\ 5 & -7 & 4 & 6 \\ 0 & -1 & -9 & 7 \end{pmatrix}$$

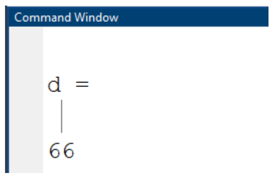
a): Iremos a la Moodle y descargaremos la carpeta de ficheros. Colocaremos el fichero sumac en la misma carpeta donde estamos guardando la información.

```
% Función matriz elemental: suma a la columna i la j por a
% --------------------------------------------------------
function E = sumac(i,j,a,n)
    E=sym(eye(n));
    E(:,i)=E(:,i)+a*E(:,j);
end
```

b): Abrimos un nuevo editor y lo guardamos en la mism carpeta o ruta de donde guardamos el fichero del paso anterior. Escribimos lo siguiente

```
A=sym([-5 1 -3 3;1 0 2 -2;5 -7 4 6;0 -1 -9 7])  % Matriz cuyo determinante pretendemos hallar
L1=sumac(1,2,5,4)*sumac(3,2,3,4)*sumac(4,2,-3,4);A1=A*L1      % Fila 1 de ceros excepto el 2,2
A1=A1(2:4,[1 3 4])                                           % Rebajamos el grado de la matriz
A1(2,:)=-A1(2,:)       % ¿Por qué lo hemos hecho? ¿Qué propiedad de los determinantes usamos?
L2=sumac(2,1,-2,3)*sumac(3,1,2,3);A2=A1*L2           % Primera fila de ceros excepto el 1,1
A2=A2(2:3,2:3)                                               % Rebajamos el grado de la matriz
d=A2(1,1)*A2(2,2)-A2(1,2)*A2(2,1)                                     % Valor del determinante
```

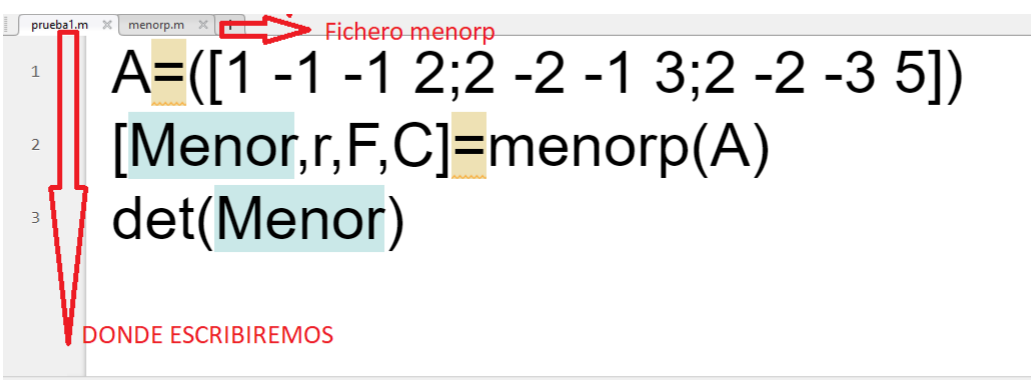
El determinante que nos debe dar es 66



Ejercicio 4: Localizar un menor principal en la matriz
Dado que el número de menores de una matriz es, en general, muy grande, el localizar una menor principal de una matriz es, a menudo, una labor harto tediosa. Por ello, se da (descargar de la plataforma Moodle) una función llamada *menorp* que localiza un menor principal en una matriz. Nota: Guardar el fichero *menorp* y el nuevo archivo donde editaremos en una misma carpeta o ruta.
En el nuevo editor localizar una menor principal de la matriz A

$$A = \begin{pmatrix} 1 & -1 & -1 & 2 \\ 2 & -2 & -1 & 3 \\ 2 & -2 & -3 & 5 \end{pmatrix}$$

En el nuevo editor es necesario escribir lo siguiente:



Recuerda que debes tener abiero el fichero menorp y el otro editor donde estÃ¡s escribiendo en Matlab. Asimismo, deben estar guardados en una misma carpeta, ya sea del escritorio o de descargas.

4. Espacios vectoriales: Hallar una base a partir de un sistema generador

Para obtener una base partiendo de un sistema generador usaremos el comando:

$\gg$ `linsolve`

Supongamos que tenemos $S = \langle \vec{v}_1, \vec{v}_2, \cdots, \vec{v}_k \rangle$. Con este comando podemos ir comprobando si cada vector del sistema generador depende linealmente de sus predecesores. Cuando esto ocurre eliminamos dicho vector del sistema. Lo vemos por pasos:

• Tomamos los dos primeros vectores del sistema $\vec{v}_1, \vec{v}_2$ (si los dos primeros vectores del sistema son proporcionales eliminamos uno de ellos y tomamos el siguiente). Comprobaremos si el tercer vector $\vec{v}_3$ es combinación lineal de $\vec{v}_1, \vec{v}_2$ usando el comando `linsolve` de la siguiente forma:

$\gg$ `linsolve([v1 v2], v3)`

Utilizando esta orden le estamos pidiendo a Matlab que nos resuelva el sistema:

$$\vec{v}_3 = \alpha_1 \vec{v}_1 + \alpha_2 \vec{v}_2$$

Matlab nos mostrará una de las siguientes respuestas:

1. Nos muestra las soluciones para los coeficientes α_1 y α_2. En este caso deduciremos que $\vec{v}_3$ es combinación lineal de $\vec{v}_1, \vec{v}_2$ y lo eliminaremos del sistema.
2. Nos muestra el mensaje:
   `Warning: Solution does not exist because the system is inconsistent.`

   En este caso concluiremos que $\vec{v}_3$ no es combinación lineal de $\vec{v}_1, \vec{v}_2$ y por lo tanto el sistema $\{\vec{v}_1, \vec{v}_2, \vec{v}_3\}$ es linealmente independiente.

• Continuamos el proceso comprobando si $\vec{v_4}, \vec{v_5}, \ldots$ son combinación lineal de sus predecesores hasta terminar con todos los vectores del sistema generador del subespacio S.

Ejercicio 5: Dado el subespacio vectorial H generado por los vectores $\{\vec{v}_1 = (1, -2, 5, -1), \vec{v}_2 = (0, 3, -2, 2),$
$\vec{v}_3 = (1, 0, 1, -1), \vec{v}_4 = (3, 5, 5, 3)\}$, obtener una base de H.

$\gg$ % Introducimos, en modo simbólico, los vectores del sistema en Matlab (se deben introducir como vectores columna).
$\gg$ v1=sym([1 -2 5 -1]'); v2=sym([0 3 -2 2]'); v3=sym([1 0 1 -1]'); v4=sym([3 5 5 3]');
$\gg$ % Como v1 y v2 son linealmente independientes (no son proporcionales)
$\gg$ % comprobamos si v3 es c.l. de v1 y v2.
$\gg$ linsolve([v1 v2], v3)

Warning: The system is inconsistent. Solution does not exist.
> In symengine (line 57)
In sym/privBinaryOp (line 903)
In sym/linsolve (line 63)
ans =

Inf
Inf

$\gg$ % Como v1, v2, v3 son l.i., continuamos comprobando si v4 es l. i. con
$\gg$ % los anteriores.
$\gg$ linsolve([v1 v2 v3], v4)

ans =

2
3
1

≫ % Concluimos que v4=2*v1+3*v2+1*v1 y debemos eliminar v4 del sistema por ser combinación lineal de los anteriores.

Por lo tanto, podemos decir que una base de H es

$$B_H = \{\vec{v}_1 = (1, -2, 5, -1), \vec{v}_2 = (0, 3, -2, 2), \vec{v}_3 = (1, 0, 1, -1)\}$$

.