

ÀLGEBRA LINEAL

En aquest tema veurem com tractar amb MAPLE els conceptes i tècniques bàsiques de l'àlgebra lineal que hem estudiat a les classes de teoria, al mateix temps que repassarem i resolldrem exercicis. Començarem amb les manipulacions elementals de matrius i vectors, el mètode de Gauss i la resolució de sistemes lineals d'equacions. Aplicarem aquestes tècniques a l'estudi de posicions relatives de plans i rectes a l'espai.

Finalment, l'última part està dedicada a l'estudi d'espais vectorials i aplicacions lineals, amb especial atenció a la diagonalització de matrius.

Els paquets de comandes de MAPLE que ens permetran treballar amb matrius i desenvolupar les tècniques d'àlgebra lineal són `linalg` i `LinearAlgebra`. Al llarg de la pràctica utilitzarem principalment el paquet `linalg`, i al final us donarem les equivalències amb les comandes del paquet `LinearAlgebra` (veure Secció 10).

1 Matrius i vectors

Gran part de les comandes amb les que treballarem es troben a la llibreria `linalg`, per això començarem carregant aquest paquet (fixeu-vos en alguns del noms de les comandes noves que apareixen al carregar el paquet).

```
> restart; with(linalg);
```

1.1 Definicions i operacions bàsiques

Els objectes principals amb els que treballarem són les matrius i els vectors. Començarem veient com es defineixen i es realitzen operacions elementals amb ells. Les comandes bàsiques per definir una matriu i un vector en MAPLE són: `matrix` i `vector`. Així, per definir la matriu

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix},$$

cal escriure

```
> A:=matrix([[1,2],[3,4]]);
```

Fixeu-vos que amb aquesta comanda les matrius s'entren per files, és a dir, es van separant entre corxets `[i]` les diferents files de la matriu. Per definir el vector

$$u = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

cal escriure

```
> u:=vector([1,-1]);
```

Les operacions aritmètiques bàsiques són: suma (+), resta (-) i producte (&*). El símbol * només es fa servir per multiplicar una matriu per un número. Recordeu que el Maple és un manipulador algebraic i quan

opera amb matrius i vectors deixa les operacions indicades. Per a obtenir (o evaluar) el resultat final hem d'utilitzar la comanda `evalm`. Comproveu el següent exemple,

```
> B:=matrix([[3],[ -1]]);
> 5*A;
> evalm(5*A);
> A &* u;
> evalm(A &* u);
> evalm(A &* B);
> evalm(B &* A);
```

Què passa quan executeu l'última comanda de l'exemple? El producte de matrius també es pot calcular amb l'operador `.`, per exemple,

```
> evalm(A.u);
> evalm(A.B);
```

I també es pot fer amb la comanda `multiply`,

```
> multiply(A,B);
```

Per algunes matrius que tenen una forma especial, hi ha comandes específiques. Per exemple, podem definir matrius diagonals (tots els valors fora de la diagonal són zero) amb la comanda `diag`, en la qual només cal indicar els valors de la diagonal i automàticament col·loca zeros a la resta.

```
> diag(1,2,3,-1,-2,-3);
> diag(0,1,0,1);
```

Així doncs, podem definir la matriu identitat de qualsevol dimensió de manera molt fàcil.

```
> Identitat5:=diag(1,1,1,1,1);
> Identitat3:=diag(1,1,1);
```

Finalment, donada una matriu A , la seva trasposada és la que s'obté intercanviant files per columnes. I la comanda que ho realitza automàticament és `transpose`.

```
> A;
> transpose(A);
```

Exercicis

1. Calcula el producte de la matriu següent A

$$A = \begin{pmatrix} -1 & 3 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 3 \end{pmatrix}$$

per cadascun del tres vectors b_1 , b_2 i b_3 que es donen.

$$b_1 = \begin{pmatrix} 1 \\ 7 \\ 9 \end{pmatrix}, b_2 = \begin{pmatrix} 8 \\ 6 \\ 5 \end{pmatrix}, b_3 = \begin{pmatrix} 4 \\ 2 \\ 3 \end{pmatrix}.$$

2. Donades les matrius $A = \begin{pmatrix} -1 & 3 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 3 \end{pmatrix}$, $B = \begin{pmatrix} 1 & 8 & 4 \\ 7 & 6 & 2 \\ 9 & 5 & 3 \end{pmatrix}$, calcula $A + B$, $A \cdot B$ i $B \cdot A$, tot comprovant que el producte de matrius no és commutatiu ($AB \neq BA$).

3. Amb les matrius $A = \begin{pmatrix} 2 & 5 & -1 \\ 4 & -1 & 2 \\ 6 & 4 & 1 \end{pmatrix}$, $B = \begin{pmatrix} 3 & 1 & 5 \\ 2 & 4 & 1 \\ -4 & 2 & -9 \end{pmatrix}$, calcula $(3A + B) \cdot (A - \frac{1}{3}B)$.

1.2 Manipulacions amb files i columnes

El MAPLE és un manipulador algebraic, i com a tal permet construir matrius a partir d'altres. D'aquesta manera es poden ajuntar matrius (horitzontalment o verticalment), eliminar columnes o files d'una matriu, intercanviar files o columnes d'una matriu,...

- La llibreria `linalg` conté, entre d'altres, algunes comandes bàsiques:

<code>concat</code>	Ajunta dues matrius horitzontalment	<code>concat(A,B)</code>
<code>stackmatrix</code>	Ajunta dues matrius verticalment	<code>stackmatrix(A,B)</code>
<code>diag</code>	Ajunta dues matrius diagonalment	<code>diag(A,B)</code>
<code>col(row)</code>	Extrau una columna (fila) de la matriu com a vector	<code>col(A,i)</code>
<code>delcols(delrows)</code>	Elimina columnes (files) de la matriu	<code>delcols(A,i..j)</code>
<code>swapcol(swaprow)</code>	Intercanvia columnes (files) d'una matriu	<code>swapcol(A,i,j)</code>

Per exemple, amb les matrius A i B de l'exercici 3 anterior $A = \begin{pmatrix} 2 & 5 & -1 \\ 4 & -1 & 2 \\ 6 & 4 & 1 \end{pmatrix}$, $B = \begin{pmatrix} 3 & 1 & 5 \\ 2 & 4 & 1 \\ -4 & 2 & -9 \end{pmatrix}$, feu

```
> stackmatrix(A,B);
> diag(A,B);
> col(A,1)
> delcols(B,2..3);
> swaprow(A,1,2);
```

Pots buscar a l'ajuda de MAPLE un fitxer d'exemple de MAPLE per veure com usar aquestes comandes per fer l'exercicis següents:

Exercici

Donades les matrius $A = \begin{pmatrix} 2 & 5 & -1 \\ 4 & -1 & 2 \\ 6 & 4 & 1 \end{pmatrix}$ i $B = \begin{pmatrix} 3 & 1 & 5 \\ 2 & 4 & 1 \\ -4 & 2 & -9 \end{pmatrix}$

1. Construeix les matrius $\begin{pmatrix} A & B \end{pmatrix}$, $\begin{pmatrix} A \\ B \end{pmatrix}$ i $\begin{pmatrix} A & B \\ B & A \end{pmatrix}$.
2. Fes una matriu que tingui només les dues primeres files de la matriu A .
3. Fes una matriu que tingui només les dues últimes columnes de la matriu B , però intercanviades.

2 El mètode de Gauss

Quan volem discutir i resoldre un sistema d'equacions lineals, un procediment que podem utilitzar consisteix en obtenir un sistema equivalent de forma reduïda més senzill.

El procés de reducció es podrà realitzar utilitzant tres tipus bàsics de transformacions:

- intercanviar dues equacions,
- multiplicar una equació per un número diferent de zero, i
- sumar a una equació un múltiple d'una altra.

Per realitzar aquestes operacions només necessitem treballar amb els coeficients, per això es treballa amb la matriu dels coeficients i els termes independents i no amb les equacions senceres. Les operacions anteriors amb equacions es corresponen amb les corresponents transformacions per files a les matrius associades al sistema. És a dir, donada la matriu ampliada del sistema d'equacions lineals, les transformacions permeses són:

- intercanviar dues files,
- multiplicar una fila per un número diferent de zero, i
- sumar a una fila un múltiple d'una altra.

2.1 Triangulació de matrius

Les comandes que ens permetran realitzar les transformacions elementals amb les matrius es troben al paquet `linalg`.

```
> restart; with(linalg);
```

- Aquestes comandes bàsiques que corresponen a les transformacions elementals de les matrius són:

1. `swaprow(A,i,j)` intercanvia les files i i j de la matriu A .
2. `mulrow(A,i,m)` multiplica la fila i de la matriu A pel número m .
3. `addrow(A,i,j,m)` canvia la fila j de A per la mateixa fila a la que se li ha sumat m vegades la fila i .

És important que sapigueu que existeixen comandes anàlogues per realitzar les mateixes operacions per columnes: `swapcol`, `mulcol` i `addcol`.

Donada la matriu $A = \begin{pmatrix} -1 & 3 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 3 \end{pmatrix}$, comproveu el resultat de fer les operacions següents:

```
> swaprow(A,1,2);
> mulrow(A,1,-3);
```

Per exemple, la comanda

```
> addrow(A,1,2,1);
```

a la fila 2 li suma la fila 1, i la comanda

```
> addrow(A,1,3,2);
```

a la fila 3 li suma dues vegades la fila 1.

Observeu en el següent exemple com la successió de comandes triangulen la matriu A :

```
> evalm(A);
> addrow(% ,1,2,1);
> addrow(% ,1,3,2);
> addrow(% ,2,3,-7/3);
```

En general, donada una matriu i aplicant les transformacions elementals successivament, podem obtenir una matriu triangular superior (és a dir, amb zeros sota la diagonal).

Exercici

Triangula, pas a pas utilitzant les transformacions elementals que acabem de veure, les matrius següents:

$$\begin{pmatrix} 18 & 10 & 30 & -4 \\ 8 & 22 & 24 & -22 \\ -5 & -6 & 12 & 17 \end{pmatrix} \quad \begin{pmatrix} 3 & 1 & 5 \\ 2 & 4 & 1 \\ -4 & 2 & 9 \end{pmatrix}$$

2.2 La comanda gausselim

En el paquet `linalg` també hi ha una comanda que permet obtenir la forma triangular o reduïda directament, és la comanda `gausselim`.

Exercici

Comprova que no t'has equivocat a l'exercici de la secció anterior obtenint la matriu triangular amb la comanda `gausselim` (recorda que a vegades el MAPLE no dóna la versió més simplificada del resultat i cal utilitzar `simplify`). I diagonalitza la següent matriu:

$$\begin{pmatrix} 1 & -3 & 1 & 4 \\ 5 & 2 & -6 & 1 \\ -3 & -8 & 8 & 7 \\ 7 & -13 & 15 & 10 \end{pmatrix}$$

2.3 El rang d'una matriu

Donada una matriu A , el seu rang és el número de files diferents de zero que conté la seva matriu triangular reduïda per files obtinguda aplicant el mètode de Gauss. El rang també es pot calcular de manera idèntica fent servir operacions per columnes i triangulant la matriu. Recordeu que el rang és sempre més petit o igual al mínim entre el número de files i el de columnes. També es compleix que una matriu i la seva trasposada tenen el mateix rang.

Exemple

Com a exemple anem a determinar el rang de la matriu

$$A = \begin{pmatrix} 5 & 6 & 18 & 15 \\ 8 & 22 & 24 & -22 \\ -5 & -6 & 12 & 17 \end{pmatrix}$$

```
> restart; with(linalg);
> A:=matrix([[5,6,18,15],[8,22,24,-22],[-5,-6,12,17]]);
> gausselim(A);
```

Observeu que el rang de la matriu és 3.

El Maple també conté una comanda que determina el rang directament. És la comanda `rank`.

```
> rank(A);
```

Exercici

Determineu el rang de la matriu $A = \begin{pmatrix} 1 & 8 & 4 & 7 \\ 7 & 6 & 2 & -2 \\ 9 & 5 & 3 & 1 \\ 5 & 7 & -1 & 2 \end{pmatrix}$ amb els dos mètodes vistos anteriorment. Comproveu que el rang de la matriu trasposada A^t també és el mateix.

2.4 La matriu inversa

Donada una matriu A , la seva matriu inversa A^{-1} és aquella tal que al multiplicar-les en qualsevol ordre ens dóna sempre la matriu identitat ($AA^{-1} = A^{-1}A = Id$). No totes les matrius tenen inversa. Han de ser quadrades i tenir rang màxim (és a dir, igual al número de files o columnes de la matriu).

Per a calcular la matriu inversa es pot utilitzar el mètode de Gauss.

- Comencem afegint una matriu identitat al costat de la matriu A de la qual volem calcular la inversa.

$$(A \mid Id)$$

- Si apliquem qualsevol operació per files a la matriu ampliada anterior, obtenim una altra matriu de la forma

$$(A' \mid B)$$

amb la propietat que sempre es compleix que $BA = A'$. És a dir, la matriu B 'recorda' totes les operacions que hem fet i les podem executar totes d'una vegada només multiplicant B per A i obtenim el resultat A' de fer totes les operacions.

- Per tant, una manera de calcular la matriu inversa és realitzar operacions per files fins que la matriu A queda transformada en la identitat,

$$(Id \mid B)$$

Com que es compleix que $BA = Id$, la matriu B que surt al final és la inversa de A .

En aquest exemple calcularem la matriu inversa de la següent matriu

$$A = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ -2 & 0 & 1 & 1 \\ -1 & -1 & 4 & 1 \end{pmatrix}$$

Comencem definint la matriu B formada per la matriu A a la que hem ajuntat una matriu identitat a la dreta. Podem aprofitar les comandes que sabem de manipulació de matrius,

```
> M:=concat(A,diag(1,1,1,1));
```

$$0 := \begin{bmatrix} 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 1 & 0 & 0 \\ -2 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ -1 & -1 & 4 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Mitjançant operacions elementals transformarem la matriu A en la matriu identitat.

Sumem a la fila 2 la fila 1, a la fila 3 la fila 1 multiplicada per 2 i a la fila 4 la fila 1.

```
> addrow(M,1,2,1):addrow(%,1,3,2):addrow(%,1,4,1);
```

$$\begin{bmatrix} 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 1 & 1 & 0 & 0 \\ 0 & -2 & 1 & 1 & 2 & 0 & 1 & 0 \\ 0 & -2 & 4 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Important: fixeiu-vos què passa quan multipliquem la matriu que ens ha quedat a la dreta per la matriu original A ,

```
> B:=delcols(%,1..4);
> evalm(B &* A);
```

El resultat que obteniu és el mateix que la matriu obtinguda després d'aplicar les operacions per files, és a dir, el bloc de l'esquerra. Això passa sempre. Per tant, per aconseguir la matriu inversa hem d'aconseguir tenir la identitat a l'esquerra i la matriu inversa serà el bloc de la dreta. Anem a veure-ho. Tornem a començar,

```
> addrow(M,1,2,1):addrow(%,1,3,2):addrow(%,1,4,1);
```

Sumem a la fila 1 la fila 2, a la fila 3 la fila 2 multiplicada per 2 i a la fila 4 la fila 2 multiplicada per 2.

```
> addrow(%,2,1,1):addrow(%,2,3,2):addrow(%,2,4,2);
```

$$\begin{bmatrix} 1 & 0 & -1 & 0 & 2 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 4 & 2 & 1 & 0 \\ 0 & 0 & 2 & 1 & 3 & 2 & 0 & 1 \end{bmatrix}$$

Multipliquem la tercera fila per -1 .

```
> mulrow(%,3,-1);
```

$$\begin{bmatrix} 1 & 0 & -1 & 0 & 2 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & -4 & -2 & -1 & 0 \\ 0 & 0 & 2 & 1 & 3 & 2 & 0 & 1 \end{bmatrix}$$

Sumem a la fila 1 la fila 3, a la fila 2 la fila 3 i a la fila 4 la fila 3 multiplicada per -2 .

```
> addrow(%,3,1,1):addrow(%,3,2,1):addrow(%,3,4,-2);
```

$$\begin{bmatrix} 1 & 0 & 0 & -1 & -2 & -1 & -1 & 0 \\ 0 & 1 & 0 & -1 & -3 & -1 & -1 & 0 \\ 0 & 0 & 1 & -1 & -4 & -2 & -1 & 0 \\ 0 & 0 & 0 & 3 & 11 & 6 & 2 & 1 \end{bmatrix}$$

Multipliquem la quarta fila per $\frac{1}{3}$.

```
> mulrow(%,4,(1/3));
```

$$\begin{bmatrix} 1 & 0 & 0 & -1 & -2 & -1 & -1 & 0 \\ 0 & 1 & 0 & -1 & -3 & -1 & -1 & 0 \\ 0 & 0 & 1 & -1 & -4 & -2 & -1 & 0 \\ 0 & 0 & 0 & 1 & \frac{11}{3} & 2 & \frac{2}{3} & \frac{1}{3} \end{bmatrix}$$

Sumem a la fila 1 la fila 4, a la fila 2 la fila 4 i a la fila 3 la fila 4.

```
> addrow(%,4,1,1):addrow(%,4,2,1):addrow(%,4,3,1);
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \frac{5}{3} & 1 & \frac{-1}{3} & \frac{1}{3} \\ 0 & 1 & 0 & 0 & \frac{2}{3} & 1 & \frac{-1}{3} & \frac{1}{3} \\ 0 & 0 & 1 & 0 & \frac{-1}{3} & 0 & \frac{-1}{3} & \frac{1}{3} \\ 0 & 0 & 0 & 1 & \frac{11}{3} & 2 & \frac{2}{3} & \frac{1}{3} \end{bmatrix}$$

La matriu inversa és la matriu en la que s'ha transformat la identitat que hem afegit a la dreta de la matriu al llarg d'aquest procés.

```
> Ainv:=delcols(%,1..4);
```

Per acabar, podem comprovar que la matriu obtinguda és realment la matriu inversa.

```
> evalm(A &* Ainv);evalm(Ainv &* A);
```

Anem a veure un altre exemple tret de la llista de problemes.

```
> restart;with(linalg):
```

```
> A:=matrix([[a,0,0,0],[1,a,0,0],[0,1,a,0],[0,0,1,a]]);
```

```
> B:=matrix([[1,0,0,0],[0,1,0,0],[0,0,1,0],[0,0,0,1]]);
```

```
> AB:=concat(A,B);
```

```
> mulrow(AB,1,1/a);
```

```
> addrow(%,1,2,-1):mulrow(%,2,1/a);
```

```
> addrow(%,2,3,-1):mulrow(%,3,1/a);
```

```
> addrow(%,3,4,-1):mulrow(%,4,1/a);
```

La matriu inversa és

```
> Ainv:=delcols(%,1..4);
```

Exercici

Calculeu pas a pas la inversa de la matriu $A = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ -2 & 0 & 1 & 1 \\ -1 & -1 & 4 & 1 \end{pmatrix}$.

Finalment, Maple conté una comanda que calcula directament la inversa d'una matriu, sempre i quan sigui possible. És la comanda `inverse`.

```
> inverse(A);
```

Exemple d'aplicació a la resolució de sistemes d'equacions lineals

El càlcul de la matriu inversa ens permet trobar la solució de certs sistemes d'equacions. Per exemple,

donat el sistema d'equacions $\begin{cases} 2x + 3y + z = 1 \\ x - y - 2z = 2 \\ x + 2y - z = 4 \end{cases}$ podem trobar una solució si l'escrivim matricialment

com $Ax = b$.

$$\begin{pmatrix} 2 & 3 & 1 \\ 1 & -1 & -2 \\ 1 & -2 & -1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix}$$

Aleshores podem aïllar el vector d'incògnites x i trobar la solució com $x = A^{-1}b$ (sempre i quan A tingui inversa).

```
> A:=matrix([[2,3,1],[1,-1,-2],[1,2,-1]]);b:=vector([1,2,4]);
> inverse(A);
> evalm(inverse(A)&* b);
```

En general, però, utilitzarem el mètode de Gauss per resoldre equacions lineals.

Exercici

Resoleu l'equació matricial (és a dir, trobeu el valor de X) següent:

$$X \cdot \begin{pmatrix} 1 & 2 & -3 & 1 \\ 3 & 2 & -4 & 2 \\ 2 & -1 & 0 & 3 \\ 0 & 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & -3 & 0 & 1 \\ 10 & 2 & 7 & 0 \\ 10 & 7 & 8 & -2 \end{pmatrix}$$

2.5 Resolució de sistemes d'equacions amb el mètode de Gauss

Donat un sistema d'equacions lineals, podem construir la matriu associada al sistema d'equacions per tal de reduir-lo o simplificar-lo realitzant operacions de files a la matriu obtinguda.

```
> restart; with(linalg);
```

- La comanda `genmatrix(eqns, vars)` construeix la matriu dels coeficients d'un sistema on `eqns` són les equacions i `vars` són les variables. Si volem com a resultat la matriu ampliada hem d'escriure `flag` com a opció. Vegeu el següent exemple,


```
> genmatrix({x+3*y-z=1,2*x+8*z=9}, [x,y,z]);
> genmatrix({x+3*y-z=1,2*x+8*z=9}, [x,y,z], flag);
```
- La comanda `geneqns(A, vars, b)` dóna les equacions d'un sistema que té a la matriu A com a matriu de coeficients, b és la columna dels termes independents i `vars` són les variables o incògnites del sistema.


```
> A:=matrix([[[-1,3,2],[1,0,1],[2,1,4]]]);
> b:=vector([1,2,5]);
> eqns:=geneqns(A, [x,y,z], b);
```

Fixeu-vos que obtenim el sistema d'equacions

$$\begin{cases} -x + 3y + 2z = 1 \\ x + z = 2 \\ 2x + y + 4z = 5 \end{cases}$$

i podem recuperar les matrius amb

```
> genmatrix(eqns, [x,y,z]);
> Ab:=genmatrix(eqns, [x,y,z], flag);
> col(Ab,4);
```

Ara ja podem simplificar el sistema d'equacions obtenint un sistema equivalent

```
> G:=gausselim(Ab);
> Af:=delcols(G,4..4); bf:=col(G,4);
> geneqns(Af, [x,y,z], bf);
```

Observeu com aquest sistema d'equacions és molt més senzill de resoldre. Fins i tot en aquesta situació el MAPLE té un comanda que permet obtenir el resultat final de la resolució del sistema. És la comanda `backsub` que s'ha d'aplicar a la matriu esglaonada del sistema: resol l'última equació i va substituint enrera (però cal que la matriu estigui en forma reduïda triangular).

```
> backsub(G);
```

És important que recordeu que ja vam veure una comanda que ens permetia resoldre equacions i també sistemes: la comanda `solve`.

```
> solve(eqns, {x, y, z});
```

La llibreria `linalg` conté una altra comanda que resol sistemes d'equacions lineals donats per la matriu dels coeficients A i la dels termes independents b , és a dir, $Ax = b$. És la comanda `linsolve`. Observeu com funciona en l'exemple següent on utilitzem les matrius A i b definides al començament d'aquesta secció,

```
> linsolve(A,b);
```

Exercicis

1. Resol el sistema d'equacions
$$\begin{cases} x - 3y + 2z = 6 \\ 2x + y - 5z = -4 \\ 2x - 13y + 13z = 28 \end{cases}$$
 usant el mètode de Gauss i comprovant el resultat amb la comanda `solve`.

2. Triangulant les matrius pas a pas, usant el mètode de Gauss, digues quina és la solució dels sistemes

$$\text{d'equacions: } \begin{cases} 12x + 14y - 15z + 23t + 27u = 5 \\ 16x + 18y - 22z + 29t + 37u = 8 \\ 18x + 20y - 21z + 32t + 41u = 9 \\ 10x + 12y - 16z + 20t + 23u = 4 \end{cases} \quad \begin{cases} 10x + 23y + 17z + 44t = 25 \\ 15x + 25y + 26z + 69t = 40 \\ 25x + 57y + 42z + 108t = 65 \\ 30x + 69y + 51z + 133t = 95 \end{cases}$$

3 Geometria en tres dimensions: posició relativa de plans i rectes

3.1 La recta

Una recta ve determinada per un punt per on passa $P = (p_1, p_2, p_3)$ i el vector director que indica la direcció de la recta, $v = (v_1, v_2, v_3)$. Les equacions paramètriques són:

$$\begin{cases} x = p_1 + tv_1 \\ y = p_2 + tv_2 \\ z = p_3 + tv_3 \end{cases}$$

on els punts de la recta s'obtenen donant valors al paràmetre t . Per exemple, si $t = 0$ obtenim el punt $P = (p_1, p_2, p_3)$.

MAPLE té comandes que permeten representar equacions paramètriques a l'espai de tres dimensions. Una d'elles és la comanda `spacecurve` que es troba al paquet de comandes `plots`. Vegeu el seu funcionament en el següent exemple on representarem la recta que passa pel punt $P = (1, 1, 2)$ i té direcció $v = (0, 1, -1)$. Les equacions paramètriques són

$$\begin{cases} x = 1 \\ y = 1 + t \\ z = 2 - t \end{cases}$$

i la representació gràfica és

```
> restart; with(plots):
> spacecurve([1,1+t,2-t],t=-3..3,axes=normal);
```

Una recta també pot venir donada en forma cartesiana, és a dir, com a solució d'un sistema de dues equacions (o geomètricament, com la intersecció de dos plans). Anem a veure com s'obté la forma cartesiana de la recta anterior: només cal aïllar el paràmetre t d'una de les equacions i substituir-lo a les altres dues. Així $t = y - 1$ de la segona equació i al substituir a les altres dues obtenim $x = 1$, $z = 3 - y$. és a dir

$$\begin{cases} x - 1 = 0 \\ z + y - 3 = 0 \end{cases}$$

Per a tornar a trobar la forma paramètrica, només hem de resoldre el sistema anterior respecte les variables x , y i z .

```
> solve({x-1=0,z+y-3=0},{x,y,z});
```

3.2 El pla

Un pla ve determinat per un punt contingut en ell $P = (p_1, p_2, p_3)$ i dos vectors que indiquen les dues direccions del pla, $u = (u_1, u_2, u_3)$ i $v = (v_1, v_2, v_3)$. Les equacions paramètriques del pla venen determinades per dos paràmetres,

$$\begin{cases} x = p_1 + tv_1 + su_1 \\ y = p_2 + tv_2 + su_2 \\ z = p_3 + tv_3 + su_3 \end{cases}$$

on els punts del pla s'obtenen donant valors als paràmetres t i s .

Un pla també queda determinat si coneixem un punt per on passa $P = (p_1, p_2, p_3)$ i un vector perpendicular a ell, $n = (n_1, n_2, n_3)$. Aleshores podem obtenir l'equació cartesiana del pla que s'obté de desenvolupar

$$\langle (x - p_1, y - p_2, z - p_3), (n_1, n_2, n_3) \rangle = 0,$$

és a dir, el producte escalar de $(x - p_1, y - p_2, z - p_3)$ i n igualat a zero.

La comanda pel producte escalar es troba al paquet `linalg` i es diu `dotprod`.

Per exemple l'equació del pla que passa per $P = (1, 0, 1)$ i té vector perpendicular $n = (1, 1, 1)$ és

```
> with(linalg): dotprod([x-1,y-0,z-1],[1,1,1])=0;
```

MAPLE conté una comanda al paquet `plots` que permet dibuixar solucions d'una equació, com és el cas de l'equació cartesiana d'un pla: és la comanda `implicitplot3d`. Vegeu com funciona amb el pla de l'exemple anterior, $x + y + z - 2 = 0$.

```
> implicitplot3d(x+y+z-2=0,x=-5..5,y=-5..5,z=-5..5,axes=normal);
```

3.3 Posició relativa de dues rectes

Donades dues rectes en forma cartesiana (és a dir, cadascuna ve determinada per dues equacions), podem estudiar la seva posició relativa resolent el sistema d'equacions obtingut amb les quatre equacions. Per exemple, anem a estudiar la posició relativa de les rectes

$$\begin{cases} 2x + 3y - 5z + 1 = 0 \\ x - 2y + z - 3 = 0 \end{cases}$$

i

$$\begin{cases} 3x + y - 3z - 2 = 0 \\ x + y + z - 1 = 0 \end{cases}$$

El primer que hem de fer és resoldre el sistema format per les quatre equacions. Si té una única solució aleshores les dues rectes es tallen en un punt. Si té infinites solucions, aleshores les dues rectes són la mateixa. Si no té solució, aleshores no es tallen i cal determinar si són paral·leles o no (es creuen).

```
> r1:=2*x+3*y-5*z+1=0,x-2*y+z-3=0;
> r2:=3*x+y-3*z-2=0,x+y+z-1=0;
> solve({r1,r2},{x,y,z});
```

Observeu que no hi ha solucions. Aleshores vol dir que no es tallen i cal veure si els seus vectors directores són paral·lels o no. Primer simplifiquem les equacions de les rectes per obtenir les paramètriques,

```
> solve({r1},{x,y,z});
> solve({r2},{x,y,z});
```

Fixeu-vos que podem obtenir un vector director mirant el coeficient del paràmetre lliure. Així, obtenim $v_1 = (1, 1, 1)$ per a r_1 i $v_2 = (2, -3, 1)$ per a la recta r_2 . És fàcil veure que aquests dos vectors no són paral·lels (un no és un múltiple de l'altre) i per tant, les dues rectes no són paral·leles i es creuen.

Anem a veure-ho gràficament.

```
> dib1:=spacecurve([1+z,z-1,z],z=-3..3,axes=normal):
> dib2:=spacecurve([1/2+2*z,1/2-3*z,z],z=-3..3,axes=normal):
> display(dib1,dib2);
```

Què observeu? Feu la representació canviant la longitud de l'interval del paràmetre.

```
> dib1:=spacecurve([1+z,z-1,z],z=-1/2..1/2,axes=normal):
> dib2:=spacecurve([1/2+2*z,1/2-3*z,z],z=-1/2..1/2,axes=normal):
> display(dib1,dib2);
```

3.4 Posició relativa de dos plans

Donats dos plans en forma cartesiana, podem determinar la seva posició relativa resolent el sistema d'equacions format amb les dues equacions. Si no té solució aleshores són paral·lels. Si té infinites solucions amb un grau de llibertat aleshores es tallen en una recta. Si té infinites solucions amb dos graus de llibertat aleshores són el mateix pla.

Per exemple, considerem els plans $3x - y - 3z - 2 = 0$ i $x + y + z - 1 = 0$.

```
> solve({3*x-y-3*z-2=0,x+y+z-1=0},{x,y,z});
```

Es tallen en una recta que té equacions paramètriques $x = \frac{3}{4} + \frac{1}{2}t$, $y = \frac{1}{4} - \frac{3}{2}t$, $z = t$.

```
> dib1:=implicitplot3d(3*x-y-3*z-2=0,x=-5..5,y=-5..5,z=-5..5,color=red,axes=normal):
> dib2:=implicitplot3d(x+y+z-1=0,x=-5..5,y=-5..5,z=-5..5,color=blue,axes=normal):
> display(dib1,dib2);
```

Ara, considereu $x + y + z + 1 = 0$ i $2x + 2y + 2z + 2 = 0$.

```
> solve({2*x+2*y+2*z+2=0,x+y+z+1=0},{x,y,z});
```

Són el mateix pla ja que la solució té dos graus de llibertat.

3.5 Posició relativa d'un pla i una recta

Donats un pla i recta en equacions cartesianes, podem determinar la seva posició relativa resolent el sistema format per les tres equacions. Si no hi ha solució aleshores no es tallen i són paral·lels. Si el sistema té infinites solucions amb un grau de llibertat aleshores la recta està continguda al pla. Si el sistema té una única solució aleshores es tallen en un punt.

Per exemple, sigui r la recta determinada per les equacions $x = 1 + 2t$, $y = 3t$, $z = -1 + 4t$. Comproveu que les equacions cartesianes són

$$\begin{cases} 3x - 2y - 3 = 0 \\ 4y - 3z - 3 = 0 \end{cases}$$

Ara considerem el pla $x + y + 2z + 1 = 0$. Podem saber la seva posició relativa resolent el sistema corresponent,

```
> solve({3x-2y-3=0,4y-3z-3=0,x+y+2z+1=0},{x,y,z});
```

Observeu que es tallen en un sol punt perquè obtenim una única solució i ho podem observar en el següent gràfic.

```
> dib1:=spacecurve([1+2*t,3*t,-1+4*t],t=-5..5,axes=normal):
> dib2:=implicitplot3d(x+y+2*z+1=0,x=-5..5,y=-5..5,z=.5..5,axes=normal):
> display(dib1,dib2);
```

Exercicis

1. Estudieu la posició relativa de la recta $\begin{cases} x = 1 + 2t \\ y = 3t \\ z = -1 + 4t \end{cases}$ i el pla $2x - 4y + 2z = 0$ i feu-ne una representació gràfica.

4 Dependència i independència lineal de vectors

Donat un vector v i un conjunt de vectors $\{v_1, \dots, v_m\}$ de \mathbb{R}^n , direm que v és combinació lineal de v_1, \dots, v_m si existeixen números $\lambda_1, \dots, \lambda_k$ tals que podem escriure v com $v = \lambda_1 v_1 + \dots + \lambda_k v_k$. Matricialment l'equació anterior es pot escriure com

$$(v_1 \mid \dots \mid v_k) \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_k \end{pmatrix} = v$$

Per tant, a la pràctica només hem de determinar si el sistema que determina l'anterior equació $A\lambda = v$ és compatible o no, on A és la matriu que té els vectors v_1, \dots, v_k per columnes.

Exemple

En el següent exemple determinarem si el vector $b = (7, -2, 1)$ és combinació lineal dels vectors $a_1 = (2, 3, 5)$, $a_2 = (3, 7, 8)$ i $a_3 = (1, -6, 1)$.

```
> restart; with(linalg):
> b:=vector([7,-2,1]);
> v1:=vector([2,3,5]); v2:=vector([3,7,8]); v3:=vector([1,-6,1]);
> A:=concat(v1,v2,v3);
```

Aprofitarem la ocasió per recordar una comanda que permet resoldre sistemes d'equacions lineals $Ax = b$ donant com a paràmetres les matrius A i b . És la comanda `linsolve`.

```
> linsolve(A,b);
```

Observeu que el sistema no té solució. Per tant, el vector b no és combinació lineal dels altres.

Per determinar si una família de vectors v_1, \dots, v_m de \mathbb{R}^n és linealment independent o no, cal resoldre el sistema d'equacions lineals que resulten de

$$\lambda_1 v_1 + \lambda_2 v_2 + \dots + \lambda_k v_m = 0.$$

Si aquest sistema és compatible determinat, aleshores l'única solució és $\lambda_1 = \lambda_2 = \dots = \lambda_m = 0$ i els vectors són linealment independents. Si el sistema és compatible indeterminat aleshores existeixen combinacions no zero d'aquests vectors que donen el vector zero. Per tant, tot es redueix a estudiar el sistema homogeni $A\lambda = 0$ on A és la matriu que té els vectors v_1, \dots, v_m per columnes.

Exemple

En el següent exemple determinarem si els vectors $v_1 = (1, 1, -1)$, $v_2 = (2, 3, 1)$ i $v_3 = (4, 2, -1)$ són linealment independents o no.

```
> v1:=vector([1,1,-1]);v2:=vector([2,3,1]); v3:=vector([4,2,-1]);
> A:=concat(v1,v2,v3);
> b:=vector([0,0,0]);
```

```
> linsolve(A,b);
```

Com que la única solució és la $(0, 0, 0)$, això vol dir que aquests vectors són linealment independents.

Recordeu que el rang d'una matriu és el nombre màxim de files o columnes linealment independents que conté. Aleshores també podríem resoldre el problema amb la comanda `rank`.

```
> rank(A);
```

Exercici

Determineu si $v = (4, -1, 5, 6)$ és combinació lineal de $v_1 = (4, -5, 2, 6)$, $v_2 = (2, 2, -1, 3)$ i $v_3 = (6, -3, 3, 9)$. Són v_1 , v_2 i v_3 linealment independents?

5 Dimensió i base d'un subespai vectorial donat per un conjunt de vectors generadors

Sigui E un subespai vectorial de \mathbb{R}^n generat per un conjunt de vectors v_1, \dots, v_k . La dimensió de E és el mínim número de vectors generadors linealment independents de E . Una base és un conjunt de vectors generadors i linealment independents de E . Per a determinar una base cal estudiar les relacions de dependència lineal entre els vectors generadors. Veurem diferents mètodes per fer-ho i els analitzarem mitjançant exemples.

- El primer mètode utilitza el mètode de Gauss per files per a determinar quins d'aquests vectors són linealment independents entre sí. Estudiarem el subespai de \mathbb{R}^3 generat pels vectors $v_1 = (1, 0, -1)$, $v_2 = (-2, 0, 3)$, $v_3 = (3, 1, 0)$ i $v_4 = (0, 0, 1)$. Sigui A la matriu que conté aquests vectors en les columnes. Recordeu que el rang de A és la dimensió del subespai generat per les columnes de A .

```
> restart;with(linalg):
> v1:=vector([1,0,-1]); v2:=vector([-2,0,3]);v3:=vector([3,1,0]);
> v4:=vector([0,0,1]); A:=concat(v1,v2,v3,v4);
> addrow(A,1,3,1);
> swaprow(% ,2,3);
```

Observem que la dimensió del subespai que generen és 3 ja que el rang de la matriu és 3. A més, mirant els pivots de la matriu veiem que els vectors v_1 , v_2 i v_3 són linealment independents i, per tant, formen una base. Fixeu-vos que el vector v_4 s'obté fent $2v_1 + v_2$.

- El segon mètode utilitza el mètode de Gauss per columnes de manera que a cada pas anem obtenim un nou conjunt de vectors generadors. Així al final n'obtenim un en que els vectors són linealment independents i, per tant, base. Estudiarem el subespai de \mathbb{R}^4 generat pels vectors $v_1 = (0, 0, 0, 1)$, $v_2 = (1, -1, 0, 1)$, $v_3 = (-1, 1, 0, 3)$, $v_4 = (0, 0, 1, 7)$. Sigui A la matriu que conté aquests vectors en les columnes. Per a poder fer operacions per columnes utilitzarem la comanda `addcol` que té un funcionament a la comanda `addrow` que ja vam aprendre a fer anar a la pràctica 3.

```
> v1:=vector([0,0,0,1]); v2:=vector([1,-1,0,1]);v3:=vector([-1,1,0,3]);
> v4:=vector([0,0,1,7]); A:=concat(v1,v2,v3,v4);
> swapcol(A,1,4);
> addcol(% ,4,1,-7);addcol(% ,4,2,-1); addcol(% ,4,3,-3);
> swapcol(% ,1,3);
> addcol(% ,2,1,1);
```

Per tant, els vectors $(1, -1, 0, 0)$, $(0, 0, 1, 0)$, $(0, 0, 0, 1)$ formen una base del subespai generat pels vectors v_1 , v_2 , v_3 i v_4 .

Observeu que en els dos exemples és fàcil determinar quins vectors cal afegir per obtenir una base de l'espai vectorial total \mathbb{R}^n . En el primer exemple els tres vectors obtinguts ja formen una base i en el segon exemple només caldria afegir el vector $(1, 0, 0, 0)$.

Exercici

Completeu el conjunt $\{(0, 1, 2, 3, 0, 1), (1, 2, 3, 4, 1, 0), (0, 0, 0, 1, 1, 1)\}$ a una base de \mathbb{R}^6 .

Finalment, Maple té una comanda que, donada una matriu, ens retorna una base del subespai generat per les columnes de la matriu. És la comanda `colspace`. Hi ha una comanda anàloga per files, `rowspace`.

Exercici

Apliqueu la comanda `colspace` a les matrius dels dos exemples anteriors per obtenir una base.

6 Aplicacions lineals

Una aplicació lineal entre dos espais vectorials $\mathbb{R}^m \rightarrow \mathbb{R}^n$ ve donada per multiplicació per una matriu A , és a dir, $x \rightarrow Ax$.

El nucli d'una aplicació lineal és el subespai vectorial format pels vectors v tal que $Av = 0$, és a dir, les solucions del sistema homogeni. Així doncs, una base del nucli i la seva dimensió s'obtenen de solucionar el sistema homogeni $Ax = 0$.

La imatge d'una aplicació lineal és el subespai vectorial generat per les columnes de la matriu A . Així tenim que una base de la imatge i la seva dimensió es poden obtenir com a la secció anterior. També, si volem determinar si un vector v és de la imatge de l'aplicació lineal, cal resoldre el sistema $Ax = v$.

Exemple

Anem a estudiar el nucli i la imatge de l'aplicació lineal donada per la matriu A .

$$A = \begin{pmatrix} 2 & 1 & -3 & 3 \\ -1 & 3 & -5 & 3 \\ -4 & 6 & -6 & 4 \end{pmatrix}$$

- Per calcular el nucli hem de resoldre el sistema homogeni $Ax = 0$.

```
> linsolve(A,vector([0,0,0]));
```

Fixeu-vos que l'espai de solucions té dimensió 1 ja que és un sistema compatible indeterminat amb un grau de llibertat. Donant el valor 1 al paràmetre podem obtenir una base, $(-8, -11, 1, 10)$.

- Per calcular una base i la dimensió de la imatge, només cal trobar una base de l'espai vectorial generat per les columnes de A

```
> colspace(A);
```

Per tant, la imatge és un espai vectorial de dimensió 3 amb base formada pels vectors $(1, 0, 0)$, $(0, 1, 0)$ i $(0, 0, 1)$ de \mathbb{R}^3 .

Hi ha una altra manera que ens permet obtenir tota la informació de cop aplicant el mètode de Gauss per columnes. Consisteix en afegir una matriu identitat a sota de la matriu A per tal de recordar els vectors de l'espai vectorial de sortida que corresponen a cada columna de la matriu A . Anem a veure com funciona amb un exemple.

```
> restart;with(linalg);
> A:=matrix([[2,1,-3,3],[-1,3,-5,3],[-4,6,-6,4]]);
> AB:=stackmatrix(A,diag(1,1,1,1));
> swapcol(AB,2,4);
> addcol(%,4,3,1);addcol(%,4,2,-4/6);addcol(%,4,1,4/6);
> addcol(%,3,2,1/2);addcol(%,3,1,1/2);
> addcol(%,2,1,-5/4);
```

Aleshores una base del nucli està formada pel vector $v = (1, 11/8, -1/8, -5/4)$ (comproveu que $Av = 0$) i per tant, és de dimensió 1. Una base de la imatge està formada pels vectors $(4/3, 0, 0)$, $(-2, -2, 0)$ i $(1, 3, 6)$ i és de dimensió 3.

7 El determinant d'una matriu

El paquet `linalg` conté una comanda que calcula el determinant d'una matriu, és la comanda `det`. El seu funcionament és molt senzill com podeu veure en el següent exemple.

```
> restart; with(linalg):
> A:=matrix([[1,2],[4,3]]);
> det(A);
```

Exercici:

Utilitzeu aquesta comanda per a calcular el determinant de les següents matrius:

$$A = \begin{pmatrix} 0 & 1 & 2 & -1 & -2 \\ 2 & 0 & 1 & 3 & 1 \\ 0 & 0 & 2 & 1 & -1 \\ 4 & 4 & 0 & 2 & 0 \\ 0 & 3 & -1 & -1 & -2 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 0 & 0 & -1 \\ 4 & 4 & 3 & -3 \\ 3 & 3 & 0 & -1 \\ 8 & 8 & 6 & -6 \end{pmatrix}$$

8 Polinomi característic i valors propis d'una matriu

Recordeu que el polinomi característic d'una matriu A és el polinomi en una variable indeterminada x que s'obté quan calculem el determinant de la matriu $A - x \cdot Id$, on x és una variable indeterminada.

Per tant, per a determinar el polinomi característic d'una matriu A podem senzillament aplicar la definició amb les comandes que ja coneixem: construir la matriu $A - x \cdot Id$ i calcular el determinant.

```
> restart; with(linalg);
> A:=matrix([[3,4],[4,-3]]); Id:=diag(1,1);
> evalm(A-x*Id);
```



```
> det(A-x*Id);
```

Ara bé, hi ha una comanda que ens dóna aquesta informació directament: és la comanda `charpoly`. Observeu que té dos paràmetres: la matriu A i la variable que volem utilitzar en el polinomi característic.

```
> charpoly(A,x);
```

Un cop calculat el polinomi característic, podem calcular els valors propis. Recordeu que els valors propis d'una matriu són les arrels del polinomi característic, és a dir, les solucions de resoldre l'equació que s'obté al igualar el polinomi característic a zero.

```
> solve(=0,x);
```

Quins són els valors propis de la matriu A ? Recordeu també que un valor propi λ és un número tal que existeix algun vector $v \neq 0$ amb $Av = \lambda v$. La feina de calcular els vectors amb aquesta propietat (són els vectors propis associats a cada valor propi) la farem al següent apartat. Ara bé, també tenim una comanda que ens dóna directament tots els valors propis de la matriu directament, és la comanda `eigenvalues`.

```
> eigenvalues(A);
```

Observeu que en aquest cas obtenim els valors $x = 5, -5$.

9 Vectors propis i matrius diagonalitzables

Un vector propi d'una matriu A associat al valor propi x , és un vector $\vec{v} \neq \vec{0}$ tal que $A\vec{v} = x\vec{v}$ (és a dir, $\vec{v} \in \text{Nuc}(A - xI)$).

Continuarem treballant amb la matriu A de l'exemple anterior. Recordeu que el polinomi característic és $x^2 - 25 = (5 - x)(-5 - x)$, i per tant, tenim dues arrels (valors propis) que són 5 i -5 amb multiplicitat 1.

Tornem a la matriu de l'exemple anterior. Un dels valors propis de la matriu A és 5 i té multiplicitat algebraica 1. Per a trobar el subespai de vectors propis per aquest valor propi 5 podem fer:

```
> b:=vector([0,0]);
```

```
> linsolve(A-5*Id,b);
```

Les solucions són de la forma $(2t, t)$ (observeu que tenim un grau de llibertat), i per tant, una base per aquest subespai ve donada pel vector $(2, 1)$ (que s'obté al fer $t = 1$).

Ara fem el mateix pel vector propi de valor propi -5 :

```
> b:=vector([0,0]);
```

```
> linsolve(A+5*Id,b);
```

Les solucions són de la forma $(t, -2t)$, i per tant, una base per aquest subespai ve donada pel vector $(1, -2)$ (que s'obté al fer $t = 1$).

Diem que la matriu diagonalitza si obtenim tants vectors propis independents com la dimensió de la matriu. Fixeu-vos que aquest és el cas: la dimensió de la matriu és 2 i hem obtingut dos vectors linealment independents $\{(2, 1), (1, -2)\}$. Aquest nom ve del fet que es compleix la següent relació: si D és la matriu diagonal que només té els valors propis a la diagonal i C és la matriu que té per columnes els vectors propis associats als valors propis aleshores $CDC^{-1} = A$ i $C^{-1}AC = D$.

```
> C:=matrix([[2,1],[1,-2]];Diag:=diag(5,-5);
```

```
> evalm(C&*Diag&*inverse(C));
```

```
> evalm(inverse(C)&*A&*C);
```

- A l'apartat anterior hem vist com calcular els valors propis associats a una matriu: un escalar λ és valor propi d' A si i només si és arrel del polinomi característic $p(x) := \det(A - xI)$. La multiplicitat de λ com a arrel del polinomi característic s'anomena **multiplicitat algebraica**.

- Per cada valor propi λ , volem calcular l'espai vectorial dels seus vectors propis. Aquest és el subespai vectorial de les solucions del sistema homogeni $A - \lambda \cdot I$. La dimensió d'aquest subespai vectorial s'anomena **multiplicitat geomètrica**.
- Una matriu **diagonalitza** si, per cada valor propi, la multiplicitat algebraica i la multiplicitat geomètrica coincideixen.

Finalment també podem fer servir la comanda `eigenvectors` que ens dóna tota la informació de cop. És a dir, el valor propi amb la seva multiplicitat algebraica i amb una base dels vectors propis associats. La multiplicitat geomètrica d'una valor propi serà el número de vectors que conté la base dels vectors propis associats.

```
> eigenvectors(A);
```

Exercici:

1. Calculeu els valors propis i vectors propis de les matrius següents i discutiu si són diagonalitzables

$$A = \begin{pmatrix} 2 & 1 & -3 \\ 0 & 0 & 6 \\ 0 & 0 & 2 \end{pmatrix} \quad B = \begin{pmatrix} 2 & 1 & -1 \\ 0 & 0 & 6 \\ 0 & 0 & 2 \end{pmatrix} \quad C = \begin{pmatrix} 1 & -3 & 3 \\ 3 & -5 & 3 \\ 6 & -6 & 4 \end{pmatrix} \quad D = \begin{pmatrix} 5 & -2 & -4 \\ 4 & -1 & -1 \\ 1 & 2 & 2 \end{pmatrix}$$

2. Discutiu si les següents matrius són diagonalitzables o no:

$$A = \begin{pmatrix} 1/2 & -1/2 & -1/2 & -1/2 & -1/2 & 0 & 1/2 \\ 7/2 & 9/2 & 3/2 & 7/2 & -3/2 & -1 & -1/2 \\ -1/2 & -5/2 & 1/2 & -1/2 & 3/2 & -2 & 1/2 \\ -5 & -4 & 0 & -6 & 6 & 4 & -1 \\ 0 & 1 & 1 & -2 & 3 & 3 & -1 \\ 7/2 & -9/2 & -1/2 & -11/2 & 11/2 & 3 & -1/2 \\ -3/2 & -3/2 & 5/2 & -11/2 & 17/2 & 4 & -3/2 \end{pmatrix}$$

$$B = \begin{pmatrix} 1/2 & -1/2 & -1/2 & -1/2 & -1/2 & 0 & 1/2 \\ 7/2 & 9/2 & 3/2 & 7/2 & -3/2 & -1 & -1/2 \\ -1/2 & -5/2 & 1/2 & -1/2 & 3/2 & -2 & 1/2 \\ -4 & -3 & 1 & -5 & 6 & 3 & -1 \\ 1 & 2 & 2 & -1 & 3 & 2 & -1 \\ -5/2 & -7/2 & 1/2 & -9/2 & 11/2 & 2 & -1/2 \\ 1/2 & 1/2 & 9/2 & -7/2 & 17/2 & 2 & -3/2 \end{pmatrix}$$

10 El paquet LinearAlgebra

Al llarg de la pràctica hem utilitzat sempre el paquet `linalg` però hi ha un altre paquet de comandes que ens permet fer les mateixes operacions de matrius i sistemes d'equacions lineals i que és compatible amb la manera automàtica que té el MAPLE 10 d'introduir matrius amb les pestanyes de l'esquerra de la pantalla. Tot seguit us donem una llista d'equivalències de comandes. Per saber més sobre el seu funcionament podeu recórrer a l'ajuda del programa.

linalg	LinearAlgebra
matrix	Matrix
vector	Vector
diag	DiagonalMatrix
transpose	Transpose
col (row)	Column (Row)
delcols (delrows)	DeleteColumn (DeleteRow)
swaprow(A,i,j) (swapcol)	RowOperation(A,[i,j]) (ColumnOperation)
mulrow(A,i,m) (mulcol)	RowOperation(A,i,m) (ColumnOperation)
addrow(A,i,j,m) (addcol)	RowOperation(A,[i,j],m) (ColumnOperation)
gausselim	GaussianElimination
rank	Rank
inverse	MatrixInverse
genmatrix(eqns,vars,flag)	GenerateMatrix(eqns,var,augmented)
geneqns	GenerateEquations
backsub	BackwardSubstitute
linsolve	LinearSolve
rowSpace	RowSpace
colSpace	ColumnSpace
det	Determinant
charpoly	CharacteristicPolynomial
eigenvalues	Eigenvalues
eigenvectors	Eigenvectors