

Curs pràctic de Maple

Pràctica 8

8 Bàsics sobre àlgebra lineal: el paquet “LinearAlgebra”.

Tot i que Maple pot solucionar sistemes d'equacions lineals utilitzant la comanda `solve()`, aquesta manera de treballar ens amagarà els càlculs que cal realitzar per a resoldre un sistema d'equacions lineals i en alguns casos pot portar-nos a solucions incorrectes. Per exemple:

Si volem resoldre el sistema d'equacions homogeni dependent del paràmetre a donat per les equacions

$$\begin{cases} ax + (a - 1)y = 0 \\ x - ay = 0 \end{cases}$$

utilitzant la comanda `solve` farem:

```
> equacions:={a*x+(a-1)*y=0,x-a*y=0};
> solve(equacions,{x,y});
```

però si considerem el cas en el que a val $\frac{-1+\sqrt{5}}{2}$ resulta que tenim

```
> caspart:= subs(a=(-1+sqrt(5))/2,equacions);
```

Que té una infinitat de solucions, com es pot veure amb:

```
> solve(caspart,y);
```

I encara que considerem el paràmetre a com una altra incògnita el resultat que ens dona `solve()` no és del tot satisfactori.

```
> solve(equacions);
```

Per tant és molt important conèixer quins són aquests càlculs i saber com funcionen per a tractar aquest tipus de problemes. També cal tenir en compte que per a entendre conceptes més profunds com són els relacionats amb els espais vectorials cal saber solucionar sistemes d'equacions lineals de tota mena. Els diferents tipus d'operacions amb matrius són la maquinaria bàsica per a resoldre sistemes d'equacions lineals. Per a poder realitzar tots aquests tipus de càlculs és possible utilitzar les funcions que defineix el paquet de Maple **LinearAlgebra**. Aquest paquet de Maple no és l'únic que existeix per a realitzar aquest tipus de càlculs (per exemple, en la instal·lació normal de Maple també trobareu el paquet **linalg**) i les mateixes operacions es poden realitzar amb instruccions de nom i sintaxi diferents per a cada paquet. A continuació presentem les que són més fonamentals per a treballar amb problemes típics d'àlgebra lineal utilitzant **LinearAlgebra**.

Com sempre, comencem *carregant* el paquet:

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

8.1 Introducció de les dades.

Els objectes sobre els que treballen les funcions del paquet **LinearAlgebra** són les matrius i els vectors. Per a introduir una matriu o un vector podem utilitzar la notació `< , , , ... >` o `< | | | ... >` segons si volem fer referència a les files de l'objecte (notació amb comes) o a les columnes (notació amb barres).

Exemple 8.1

La matriu $M = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$ es pot introduir per files com

```
> Mf:=< < 1 | 2 | 3 >,< 4 | 5 | 6 > >;
```

o per columnes com

```
> Mc:= < < 1 , 4> | < 2 , 5 > | < 3 , 6 > >;
```

Fixeu-vos que en els dos casos es veu el mateix resultat exactament.

Aquestes construccions només són formes abreujades i simplificades d'introduir les matrius o vectors, podeu consultar a l'ajuda del programa les funcions per a escriure les matrius i els vectors que són `Matrix()` i `Vector()` i podreu veure totes les possibilitats que tenen (i també que és una mica més llarg tenir definida una matriu a partir d'aquestes comandes).

Per a extreure una submatriu d'una matriu ja definida podem utilitzar la sintaxi usual a les llistes, per exemple, si A és una matriu, $A[i,j]$ és el coeficient de la fila i , columna j . Tenint en compte que tant l'argument i com el j poden ser rangs i llistes.

Exemple 8.2 Considereu la matriu M_f de l'exemple anterior i proveu:

```
> Mf[2,1];
> Mf[1..2,2..3];
> Mf[[2,1],1..3];
```

També heu de saber que el paquet **LinearAlgebra** té funcions per a introduir molts tipus especials de matrius amb poc esforç (matrius diagonals, triangulars superiors, amb tots els coeficients iguals, ...). Pot ser ara és un bon moment per a fer un cop d'ull al joc de funcions d'aquest paquet mirant la secció de l'ajuda del programa dedicada al paquet **LinearAlgebra** on hi trobareu la llista completa.

8.2 Les operacions entre matrius.

Les operacions bàsiques entre matrius són la suma (i la diferència), el producte per un escalar i el producte de matrius. Quan utilitzem el paquet **LinearAlgebra** totes aquestes operacions estan disponibles de forma immediata. Podeu comprovar-ho en els exemples següents.

Exemple 8.1

```
> M1:= < <1,2,-1>|<0,3,3>|<-1,1,-3> >;
> M2:= < <0,1,-1>|<-2,3,5>|<1,1,0> >;
> M1+M2;
> M1-M2;
> 3*M1;
> 2*M1-4*M2;
```

Només cal tenir en compte que, com que el producte de matrius no es fa multiplicant cada una de les components dels factors, l'operador `*` (asterisc) no produeix el resultat que esperariem en un principi (de fet dóna un missatge d'error). Per a obtenir el vertader producte de matrius cal utilitzar l'operador `.` (punt).

Exemple 8.2

```
> M1.M2;
> M1*M2;
> M2.M1;
```

El producte de matrius no sempre es pot efectuar. Si les matrius que intenteu multiplicar no són dels tamanys adequats, Maple us donarà un missatge d'error.

Exemple 8.3

- > M3 := < <1, 2> | <2, -1> | <1, -1> >;
- > M1.M3;

En canvi sí que es pot realitzar l'operació amb l'ordre invers.

- > M3.M1;

Per a practicar una mica més amb la introducció de dades en vectors i matrius i en la realització dels càlculs elementals amb matrius aquí teniu un petit exercici.

Exercici 8.1

Introduïu la matriu $P_1 = \begin{pmatrix} \frac{4}{7} & \frac{1}{7} & -\frac{1}{7} \\ -\frac{5}{21} & \frac{4}{21} & \frac{1}{7} \\ -\frac{3}{7} & \frac{1}{7} & -\frac{1}{7} \end{pmatrix}$, el vector $v_1 = \begin{pmatrix} 1 \\ -\frac{11}{3} \\ -2 \end{pmatrix}$ i el vector $v_2 = \begin{pmatrix} 1 \\ -1 \\ 2 + 2\sqrt{2} \end{pmatrix}$. Utilitzant la matriu $M_1 = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 3 & 1 \\ -1 & 3 & -3 \end{pmatrix}$, que ja hauríeu de tenir introduïda en l'exemple 8.3 d'aquest apartat, calculeu els productes $(M_1) \cdot (P_1)$, $(M_1) \cdot (v_i)$, $(P_1) \cdot (v_i)$ (per a $i = 1$ i 2). Quina relació creieu que hi ha entre els productes $(M_1) \cdot (v_i)$, $(P_1) \cdot (v_i)$ i els vectors v_i ? Què passa si intenteu realitzar el producte $(v_1) \cdot (M_1)$? Busqueu en l'ajuda de Maple com es pot transposar un vector o una matriu i calculeu el producte del vector fila amb les mateixes components que v_1 per la matriu M_1 .

8.3 La matriu d'un sistema d'equacions lineals.

Quan es vol discutir i resoldre un sistema d'equacions lineals, el procediment usual consisteix en obtenir un sistema equivalent (és a dir, amb el mateix conjunt de solucions) que tingui una forma *reduïda*. (El significat de reduït depèn del context. En general podem dir que un sistema és reduït si podem dir sense cap mena de dubte si té solucions o no i en el cas que en tingui puguem llegir, sense fer càlculs addicionals, quines són aquestes solucions). Aquest procediment de *reducció* sempre es podrà realitzar utilitzant tres tipus bàsics de transformacions:

- Intercanviar dues de les equacions entre si.
- Multiplicar una de les equacions per una constant no nul·la.
- Sumar a una de les equacions un múltiple d'una de les altres.

Recordeu que per a poder fer aquestes operacions només necessitem treballar amb els *coeficients* de les equacions, sense utilitzar per res les *incògnites*. Així, per exemple, si es vol treballar sobre el sistema d'equacions $\begin{cases} 3x - 2y = 2 \\ 2x - y = 7 \end{cases}$ bastarà que treballem sobre la matriu $\begin{pmatrix} 3 & -2 & 2 \\ 2 & -1 & 7 \end{pmatrix}$.

Maple ens permet fer aquest procés de forma automàtica. Per això el paquet **LinearAlgebra** proporciona la comanda `GenerateMatrix()` (i la comanda inversa `GenerateEquations()`).

Exemple 8.1

Per a obtenir la matriu de coeficients i el terme independent del sistema d'equacions

$$\begin{cases} x - y + z - 2t = 1 \\ y - 2z = 6 \\ 2x + 4y + t = 8 \\ 44x - y + 34z - 21t = 98 \end{cases}$$

posarem

```
> sistema1:=[x-y+z-2*t=1, y-2*z=6, 2*x+4*y+t=8,44*x-y+34*z-21*t=98];
> vars:=[x,y,z,t];
> S1:= GenerateMatrix(sistema1,vars);
```

Noteu que **S1** és una llista de dues matrius, la primera amb els coeficients i l'altra amb el terme independent. Si el que voleu és la matriu ampliada del sistema podeu fer

```
> Sa1:= <S1[1]|S1[2]>;
```

o bé utilitzar l'opció de la comanda `GenerateMatrix()` que fa el procés anterior d'un sol cop

```
> GenerateMatrix(sistema1,vars, augmented);
```

Noteu que és molt important especificar quines són les incògnites ja que, si no es fa, Maple no seria capaç de distingir en un sistema d'equacions lineals que depèn d'un paràmetre si les indeterminades que hi surten són totes incògnites o no.

Exemple 8.2

Si tenim el sistema d'equacions, dependent del paràmetre λ , donat per $\begin{cases} x - 2y = \lambda \\ 2x - 3y = 2 - \lambda \end{cases}$, per a obtenir la seva matriu de coeficients haurem de fer

```
> GenerateMatrix([x-2*y=lambda,2*x-3*y=2-lambda],[x,y]);
```

encara que dins d'aquest sistema la dependència respecte λ també sigui lineal.

Si no posem quines són les incògnites Maple donarà un missatge d'error.

```
> GenerateMatrix([x-2*y=lambda,2*x-3*y=2-lambda]);
```

L'exemple següent mostra el procediment per a generar un sistema d'equacions a partir de la matriu de coeficients utilitzant `GenerateEquations()`. Els detalls sobre la comanda `GenerateEquations()` els podreu trobar a l'ajuda de Maple en l'apartat corresponent a la funció `GenerateEquations` del paquet **LinearAlgebra**.

Exemple 8.3

El sistema d'equacions lineals que té per matriu de coeficients $\begin{pmatrix} 1 & -2 & 3 \\ 2 & 3 & 1 \\ -1 & 1 & -5 \end{pmatrix}$ i per terme independent

$\begin{pmatrix} 2 \\ -1 \\ 7 \end{pmatrix}$ respecte les incògnites $\begin{pmatrix} x \\ y \\ z \end{pmatrix}$ s'obtindrà amb la comanda:

```
> eqs:=GenerateEquations(<<1,2,-1|<-2,3,1|<3,1,-5>>,[x,y,z],<2,-1,7>);
```

El resultat serà una llista amb les tres equacions que volíem, de la que podem extreure cada un dels seus elements amb la notació usual `eqs[i]` per a $i=1..3$.

8.4 Les operacions de reducció.

La comanda de Maple `RowOperation()` permet realitzar **totes** les operacions de reducció de la matriu d'un sistema d'equacions lineals. En els exemples que venen tot seguit podeu veure una mostra de cada un dels

tipus aplicat sobre la matriu $\begin{pmatrix} 2 & 1 & -1 \\ 3 & 5 & -2 \\ 4 & 8 & 16 \end{pmatrix}$.

```
> matriu:= <<2|1|-1>,<3|5|-2>,<4|8|16>>;
```

Exemple 8.1

Primer intercanviem la primera fila amb la tercera:

```
> RowOperation(matriu,[1,3]);
```

Fixeu-vos que els arguments són: la matriu sobre la que volem actuar i una llista amb el parell files que voleu intercanviar.

Exemple 8.2

També podem multiplicar tota la tercera fila per $\frac{1}{4}$:

```
> RowOperation(matriu,3,1/4);
```

Aquí heu de notar que els arguments són: la matriu sobre la que volem actuar, la fila sobre la que volem fer la operació i finalment el nombre pel que volem multiplicar.

Exemple 8.3

I finalment podem sumar a la segona fila la primera multiplicada per $-\frac{3}{2}$ (cosa que *elimina* la primera incògnita en la segona equació).

```
> RowOperation(matriu,[2,1],-3/2);
```

En aquest cas els arguments són: la matriu sobre la que actuem, una llista amb dos nombres que designen, el primer, la fila sobre la que volem realitzar l'operació i, el segon, la fila que utilitzem per actuar sobre la primera i, com a últim argument, el nombre pel que volem multiplicar.

Cal dir que en tots aquests casos també es podria afegir un argument opcional (`inplace`) que té l'efecte de realitzar les operacions i modificar la mateixa matriu sobre la que les fem d'acord amb el resultat obtingut (fixeu-vos que utilitzar aquesta opció té el perill d'equivocar-se amb l'operació que es vol realitzar i no saber, o poder, desfer el desastre).

Exemple 8.4

Podem modificar `matriu` per a deixar una matriu reduïda realitzant les operacions següents:

```
> RowOperation(matriu,[2,1],-3/2,inplace);
> RowOperation(matriu,[3,1],-2,inplace);
> RowOperation(matriu,[3,2],-12/7,inplace);
> RowOperation(matriu,1,1/2,inplace);
> RowOperation(matriu,2,2/7,inplace);
> RowOperation(matriu,3,7/132,inplace);
```

També es pot dir que existeix la comanda `ColumnOperation()` amb les mateixes característiques que la comanda de les operacions per files però aplicant les operacions a les columnes.

Ara ja podeu mirar de fer els següents exercicis:

Exercici 8.2

Donat el sistema d'equacions
$$\begin{cases} 2x_1 - 4x_2 + 3x_3 - 5x_4 + x_5 = 3 \\ x_1 - 4x_2 - 2x_3 - 7x_5 = 2 \\ 5x_1 - 3x_3 - 2x_4 + 2x_5 = 1 \\ x_2 - 4x_3 + 3x_4 - x_5 = \frac{3}{2} \end{cases}$$
 genereu la seva matriu ampliada, realitzeu les operacions de reducció necessàries en aquesta matriu i determineu si té solucions o no en té. Si en té, com són aquestes solucions?

Exercici 8.3

En la fabricació d'un cert producte alimentari s'utilitzen quatre components diferents. Una unitat de cada un dels ingredients conté les quantitats de vitamines (miligrams) i de kilocalories que s'expressen en la taula següent:

	Comp1	Comp2	Comp3	Comp4
vit A	1	1	1	2
vit B	1	2	1	3
vit C	1	3	2	1
Kilocal	2	2	1	1

Quines quantitats de cada un dels ingredients s'han de barrejar per a produir un producte final que aportí, exactament, 300 mg de vitamina A, 430 mg de vitamina B, 310 mg de vitamina C i tingui un valor energètic de 250 kilocalories?

És possible fabricar un producte final en el que la combinació de vitamines i d'aport energètic sigui qualsevol?

Exercici 8.4

Estudieu, per als diferents valors dels paràmetres b i t , el sistema d'equacions lineals respecte $[x, y, z]$.

$$\begin{cases} 3x + 2y + z = t \\ x - y + 2z = 1 + t^2 \\ 3x + 7y - 4z = -1 - t - t^2 - t^3 \\ 2x + y + bz = t^3 \end{cases}$$

Sovint no serà necessari realitzar les operacions de reducció sobre la matriu d'un sistema d'equacions lineals una per una. La comanda de Maple `Pivot()` és en aquests casos força útil. Quan executem la comanda `Pivot(A,i,j)`, se sumen a totes les files de la matriu A múltiples adequats de la fila que ocupa la posició i per tal que els coeficients en la columna j siguin tots nuls (naturalment la fila i no es modifica, tampoc es pot realitzar la operació si el coeficient en la columna j de la fila i és 0). Aquesta comanda també té l'opció `inplace` per a realitzar els càlculs modificant la matriu sobre la que estem treballant.

Exemple 8.5

Si es considera la matriu donada per $A = \begin{pmatrix} 1 & 3 & 0 \\ 2 & -1 & -5 \\ 0 & 1 & 3 \end{pmatrix}$

```
> A:= <<1|3|0>,<2|-1|-5>,<0|1|3>>;
```

Podem *pivotar* sobre l'element [3, 2] amb

```
> Pivot(A,3,2);
```

Però no ho podem fer sobre l'element [3, 1].

```
> Pivot(A,3,1);
```

8.5 Reduccions automàtiques.

Quan no hi ha cap indeterminació en el camí per a fer la reducció d'un sistema d'equacions lineals es pot demanar a Maple que redueixi d'un sol cop la matriu del sistema.

La comanda de Maple `GaussianElimination()` és la que realitza la reducció *Gaussiana* (dóna com a resultat una matriu reduïda amb zeros a la part inferior)

Exemple 8.1

Apliquem `GaussianElimination()` sobre la matriu $A = \begin{pmatrix} 1 & 3 & 0 \\ 2 & -1 & -5 \\ 0 & 1 & 3 \end{pmatrix}$ (que és la de l'exemple anterior).

```
> GaussianElimination(A);
```

Per a reduir completament la matriu del sistema d'equacions lineals tenim `ReducedRowEchelonForm()` que a part de la reducció Gaussiana torna com a resultat una matriu en la que els *pivots* són 1 i tots els altres coeficients de la columna d'un d'aquests pivots són 0.

Exemple 8.2

```
> ReducedRowEchelonForm(A);
```

Finalment, si el que es vol és determinar la solució d'un sistema d'equacions lineals donat per la seva matriu i pel terme independent, es pot utilitzar la comanda `LinearSolve()`.

Exemple 8.3

La solució del sistema d'equacions amb matriu de coeficients A i terme independent $\begin{pmatrix} 1 \\ 2 \\ -5 \end{pmatrix}$ es pot obtenir

amb

```
> LinearSolve(A,<1,2,-5>);
```

Encara que el sistema sigui indeterminat, la comanda `LinearSolve()` també funciona

```
> LinearSolve(<<1|2|1>,<-2|1|-5>>,<2,-1>);
```