

1 MAPLE BÀSIC

El programa que utilitzarem en aquesta sessió de pràctiques és el MAPLE, que suposem conegut de l'assignatura *Eines d'informàtica*.

Els següent enllaç d'internet us pot ser d'utilitat:

- <http://www.maplesoft.com> (en particular, si aneu a Maple Application Centre o a Students trobareu informació online, però cal registrar-se).

1.1 Inicialització

Moltes vegades si una manipulació no funciona com esperada, és que alguna variable s'ha quedat assignada d'alguna manipulació anterior. És recomanable d'executar la comanda **restart** cada cop que comencem un càlcul nou, per a no tenir conflictes amb variables ja definides.

Per carregar llibreries especialitzades, com ara **LinearAlgebra** i **plots**, es fa servir la comanda **with()**.

1.2 Alguns exemples per recordar

Definim alguna expressió:

```
> f := (x-a) * (cos(theta)^2+sin(theta)^2)*(x-6);
```

MAPLE evita calcular abans que sigui necessari. Podem forçar una simplificació, però, matemàticament continua essent la mateixa expressió:

```
> f := simplify(f);
```

Preguntem quants constituents té la expressió, de quin tipus és, i en agafem el segon constituent:

```
> nops(f);  
> whattype(f);  
> op(2,f);
```

Si fem una expansió, ja és una altra expressió per a MAPLE:

```
> f := expand(f);  
> nops(f);  
> whattype(f);  
> op(3,f);
```

Si volem veure a com a variable principal, podem fer coses com ara:

```
> collect(f,a);  
> coeff(%,a);
```

(Reordeu que % denota sempre l'expressió anterior.)

D'altra banda, podem construir una equació que resollem respecte de x :

```
> eq := f=(16-a);
```

```
> sols := [solve(eq,x)];
```

Observeu que hem definit `sols` com una llista (jà que hem posat els claudàtors). Quants elements té la llista? Extraïm el segon element:

```
> whattype(sols);
> nops(sols);
> sols[2];
```

Volem substituir aquesta solució dins de l'equació per verificar que realment en és una solució. Primer obtenim una equació que no és gaire fàcil de llegir; i la resposta de la funció *eval boolean* no és fiable:

```
> check := subs(x=%,eq);
> evalb(check);
```

Cal primer expandir les expressions, per permetre a maple una comparació;

```
> check := expand(check);
> evalb(check);
```

Tornem al conjunt de les solucions: què passa si no posem els claudàtors? El resultat serà no pas una llista sinó una seqüència d'expressions:

```
> sols := solve(eq,x);
> whattype(sols);
> nops(sols);
```

L'error és degut al fet que una seqüència d'expressions no és una sola expressió (excepte si és una seqüència de longitud 1, és clar!), i per tant la comanda ve dos inputs al lloc de u! Tampoc no podem convertir la seqüència en una llista amb `convert`, perquè aquesta comanda tampoc accepta tants arguments:

```
> convert(sols,list);
```

La solució, i la gran utilitat de seqüències, és que les podem passar com si fossin molts arguments separats per comas. Així podem per exemple formar un conjunt o una llista:

```
> solset := {sols};
> sollist := [sols];
> convert(sollist,set);
```

Una manera de construir seqüències, i ulteriorment llistes, és amb la comanda `seq`,

```
> s := seq(n^2, n=1..10);
> L := [seq(coeff(f,x,i), i=0..degree(f,x))];
```

de la qual la sintaxi és molt semblant a la de `sum`:

```
> poly := sum( L[i]*x^(i-1), i=1..nops(L));
```

2 MATRIUS I VECTORS

En aquesta secció veurem com tractar amb MAPLE de matrius i vectors, amb èmfasi en les tècniques necessàries per al tractament de còniques i quàdriques. El paquet de comandes de MAPLE per a l'àlgebra lineal es diu `LinearAlgebra`. Començarem carregant aquest paquet (fixeu-vos en alguns del noms de les comandes noves que apareixen al carregar el paquet).

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

2.1 Construccions de matrius i vectors

Les comandes bàsiques per definir matrius i vectors 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 les entrades d'una matriu s'entren per files, és a dir, es van separant entre claudàtors `[ i ]` les diferents files de la matriu.

Per definir el vector

$$v = \begin{pmatrix} 1 \\ -2 \end{pmatrix}$$

cal escriure

```
> v := Vector([1,-2]);
```

Observeu que els vectors de MAPLE són vectors-columnes, tot i que l'input de la comanda s'escriu com una fila. Per a definir un vector-fila hem de fer

```
> w := Vector[row]([1,-2]);
```

Un vector és essencialment un cas particular d'una matriu, per tant podríem fer també

```
> w := Matrix([[1,-2]]);
```

La comanda `Matrix` serveix també per construir matrius més grans a partir de matrius existents, com ara

```
> M := Matrix([A,v],[w,89]);
```

$$M := \begin{bmatrix} 1 & 2 & 1 \\ 3 & 4 & -2 \\ 1 & -2 & 89 \end{bmatrix}$$

Hi ha moltes maneres d'introduir matrius — consulteu l'ajuda online. Una que resulta molt útil, per exemple per construir matrius ampliadas, és aquesta:

```
> S := <A|v>;
```

$$S := \begin{bmatrix} 1 & 2 & 1 \\ 3 & 4 & -2 \end{bmatrix}$$

Per a algunes formes de matrius especials existeix una sintaxi especial de la comanda:

```
> Id := Matrix(2,shape=identity);
```

```
> DD := Matrix(Vector([5,5,7,4]),shape=diagonal);
```

Per extreure submatrius la tècnica bàsica consisteix en indicar índexs entre claudàtors:

```
> A[2,1];
> DD[3,1];
> DD[1..3,2..4];
```

2.2 Operacions bàsiques

Addició, subtracció, i multiplicació per escalars, es fan simplement amb els símbols $+$, $-$ i $*$. Exemple, on l'escalar és un símbol formal:

```
> A-lambda*Id;
```

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

L'operador $*$ no funciona per multiplicació de matrius. Multiplicació de matrius es fa amb la comanda `Multiply`:

```
> Multiply(A,A);
> Multiply(A,Multiply(A,A));
```

Més pràctica és la notació amb un punt:

```
> A.A;
> A.v;
```

Per calcular la transposada d'una matriu, o la inversa d'una matriu quadrada, tenim les comandes

`Transpose` i `MatrixInverse`:

```
> Transpose(A);
> MatrixInverse(A);
```

Els invariants més importants d'una matriu són accessibles via comandes evidents:

```
> Determinant(A);
> Rank(A);
> Eigenvalues(A);
```

$$\begin{bmatrix} \frac{5}{2} + \frac{\sqrt{33}}{2} \\ \frac{5}{2} - \frac{\sqrt{33}}{2} \end{bmatrix}$$

Observeu que el resultat de la comanda `Eigenvalues` és una llista formada de tots els valors propis. La comanda `Eigenvectors` retorna un parell compost del vector format dels valors propis, seguit d'una matriu on les columnes són els vectors propis associats:

```
> Eigenvectors(A);
```

Per calcular el polinomi característic, cal indicar el nom de l'incògnita del polinomi com un segon argument a la comanda:

```
> CharacteristicPolynomial(A,t);
```

2.3 Eliminació de Gauss

Per resoldre sistemes d'equacions lineals, sabem que hem de fer operacions de files:

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

Totes aquestes operacions s'efectuen amb la comanda `RowOperation`:

- `RowOperation(A,[i,j])` intercanvia les files i i j de la matriu A .
- `RowOperation(A,i,m)` multiplica la fila i de la matriu A pel número m .
- `RowOperation(A,[i,j],m)` canvia la fila i de A per la mateixa fila a la que se li ha sumat m vegades la fila j .

Existen una comanda anàloga per realitzar les mateixes operacions per columnes: `ColumnOperation`.

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

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

Per exemple, la comanda

```
> RowOperation(B,[2,1],1);
```

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

```
> RowOperation(B,[3,1],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 B

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

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).

Les operacions de file evidentment són molt útils, però si l'objectiu és triangular una matriu, existeix evidentment una comanda que fa tot d'un cop:

```
> GaussianElimination(B);
```

Observeu que aquesta comanda no divideix les files per que els pivots és quedin iguals a 1.

2.4 Resoldre sistemes homogenis

Si una matriu representa un sistema d'equacions homogènies la podem resoldre fent servir eliminació de Gauss. Però, hi ha una altra opció que moltes vegades és més útil: com que el sistema és homogeni, resoldre'l vol dir calcular el nucli de l'aplicació associada a la matriu.

Hi ha una comanda que fa això: `NullSpace`.

Exemple: volem resoldre el sistema

$$\begin{cases} 4x_1 + 2x_2 - 7x_3 + 3x_4 = 0 \\ 2x_1 - x_2 + 3x_3 - 4x_4 = 0 \end{cases}$$

El representem per la matriu

```
> M := Matrix([4,2,-7,3],[2,-1,3,-4]);
```

i trobem la solució fent

```
> NS := NullSpace(M);
```

Es ve que el valor retornat és un conjunt de vectors que generen el nucli, es-a-dir una base de l'espai de les solucions.

El process invers és molt interessant: si tenim un subespai vectorial generat per alguns vectors, com trobar un sistema d'equacions d'aquest subespai? Escrivim els vectors com a columnes d'una matriu. Com exemple agafem els dos vectors resultats del càlcul anterior:

```
> N := Matrix(convert(NS,list));
```

Estem buscant els coeficients d'una equació

$$a_1x_1 + a_2x_2 + a_3x_3 + a_4x_4 = 0$$

que s'anul·la en les dues columnes de N . És a dir,

$$[a_1, a_2, a_3, a_4] \begin{bmatrix} N \end{bmatrix} = [0, 0]$$

Equivalentment, $N^T \mathbf{a}^T = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ Per tant, estem buscant el nucli de N^T (i després hem de transposar altra vegada):

```
> SS := NullSpace(Transpose(N));
> S := Transpose(Matrix(convert(SS,list)));
```

Aquesta matriu representa un sistema d'equacions que defineix el subespai. Observeu que no és exactament el mateix sistema que el nostre M original, però això és normal: hi ha moltes maneres de definir el mateix subespai. Si volem convertir la matriu S en un sistema d'equacions, hem de fer

```
> eqvect := S . Vector([x1,x2,x3,x4]);
> eqset := convert(eqvect,set);
```

2.5 Diagonalització de matrius simètriques

Donada una matriu simètrica A (sobre $\mathbf{R}$), sempre podem trobar una matriu C invertible tal que $C^T A C$ sigui diagonal.

El mètode per a trobar tal matriu C és el següent:

(a) Sota la matriu A , ajuntem la matriu identitat.

(b) Diagonalitzarem A pel mètode de Gauss amb la precaució que tota operació que afecti files s'ha de repetir igual afectant columnes (per exemple, si substituïm la fila 2 per la suma de les files 1 i 2, també hem de substituir la columna 2 per la suma de les columnes 1 i 2). La matriu identitat afegida queda afectada també per aquests canvis.

Observació: Si el pivot a_{ii} és zero, mirem els altres termes de la diagonal. Si a_{jj} és diferent de zero, i $i < j$, permutem la fila i amb la fila j i la columna i amb la columna j .

(c) La matriu C buscada serà la matriu que apareix, un cop A ha diagonalitzat, on era la matriu identitat.

Exemple

Donada la matriu simètrica $A = \begin{pmatrix} 3 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & 0 & -3 \end{pmatrix}$ trobem una matriu C invertible tal que $C^T A C$ sigui diagonal.

Seguim els passos descrits anteriorment:

```
> A := Matrix([[3,1,2],[1,2,0],[2,0,-3]]);
```

Ajuntem la matriu identitat — atenció de posar els claudàtors correctament!

```
> AI := Matrix([A],[Matrix(3,shape=identity)]);
```

Fem zeros sota primer element de la diagonal principal, i a la seva dreta

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

Fem zero sota segon element de la diagonal principal, i també a la seva dreta:

```
> RowOperation(%,[3,2],2/5);
> ColumnOperation(%,[3,2],2/5);
> DC := %;
```

Les matriu D i C que busquem són per tant

```
> diag := DC[1..3,1..3];
> C := DC[4..6,1..3];
```

$$C := \begin{pmatrix} 1 & -1/3 & -4/5 \\ 0 & 1 & 2/5 \\ 0 & 0 & 1 \end{pmatrix}.$$

No podem fer servir el símbol D com a variable en MAPLE: està reservat :-

Comprovem que efectivament $C^T A C$ és la matriu diagonal obtinguda D :

```
> Transpose(C).A.C;
```

3 PROGRAMACIÓ ELEMENTAL

Els exemples anteriors il·lustren que moltes vegades ens trobem amb tasques de càlculs repetitives. Després de fer tres o quatre cops el mateix tipus de repetició és millor fer una mica de programació per salvar temps, i també per evitar errors típics de copy-and-paste.

És molt fàcil programar en MAPLE. Per començar, escrivim una procediment que calcula el quadrat d'un nombre:

```
> sq := proc(n)
>   RETURN(n^2);
> end;
> sq(8);
```

Una mica més complicat: un proc que retorna el quadrat per als nombres parells i zero per al nombres senars:

```
> f := proc(n)
>   if n mod 2 = 0 then
>     RETURN(sq(n));
>   elif n mod 2 = 1 then
>     RETURN(0);
>   fi;
> end;
> f(-182);
> f(191);
```

El proc següent rep una llista (de nombres enters), i retorna el quadrat del primer nombre parell:

```
> fL := proc(L::list)
>   for n in L do
>     if n mod 2 = 0 then
>       RETURN( f(n) );
>     fi;
>   od;
> end;
```

(El warning no té importància avui.)

Ara passem a construir procediments més útils: un proc `RowColumnOperation` que aplica una operació de fila i després la mateixa operació de columna. Aquest proc té tres arguments¹: una matriu, una llista de posicions, i un coeficient, exactament com els procs `RowOperation` i `ColumnOperation`:

```
> RowColumnOperation := proc(M,pos,coef)
>   RowOperation(M,pos,coef);
>   RETURN(ColumnOperation(%,pos,coef));
> end;
> RowColumnOperation(A,[2,1],-1/3);
```

Si tornem a l'exemple de la matriu A , com sabem on aplicar el nostre proc `RowColumnOperation`? Bé, un cop que hem decidit quina entrada a la diagonal que volem pivotar, és només aplicar la proc a les altres

¹Malhauradament, els procs originals són una mica més sofisticats, per que permeten també la sintaxi on l'últim argument és omès: és el cas d'una operació que intercanvia dues files (o dues columnes). Definir tals procs és una mica més complicat...

entrades de la mateixa columna (i per tant també a les altres entrades de la mateixa fila). I com sabem quin coeficient hem de fer servir? Bé, si la entrada a la diagonal és d , i la l'altra entrada és p , llavors el coeficient ha de ser $-p/d$. Amb aquest raonament podem escriure un proc que pivota a la entrada $[i, i]$ d'una matriu M :

```
> sympivot := proc(M,i)
>   n := RowDimension(M);
>   d := M[i,i];
>   newM := M;
>   for j from 1 to n do
>     if i <> j then
>       coef := -newM[j,i]/d;
>       newM := RowColumnOperation(newM,[j,i],coef)
>     fi;
>   od;
>   RETURN(newM);
> end;
```

Observeu que M és la matriu original, newM és la còpia que modifiquem. La j corre de 1 a n que és la dimensió, i és clar, fem només alguna cosa quan $j \neq i$.

Ara podem sympivotar fàcilment:

```
> sympivot(A,1);
> sympivot(%,2);
```

Finalment, per evitar la repetició, escrivim un proc que sympivota en tota la diagonal:

```
> symdiag:=proc(M)
>   newM:=M;
>   for i from 1 to RowDimension(M) do
>     newM:=sympivot(newM,i);
>   od;
>   RETURN(newM);
> end;

> symdiag(A);
> S:=Matrix([[1,3,5,7,9],[3,2,6,0,1],[5,6,4,4,2],[7,0,4,1,1],[9,1,2,1,1]]);
> symdiag(S);
```

Ara podem diagonalitzar matrius simètriques molt ràpidament. Però, sovint, com en l'exemple de la secció anterior, interessa saber quina matriu C efectua la diagonalització!

Escrivim doncs un últim proc, que calcula C a partir d'una matriu simètrica M :

```
> CfromM := proc(M)
>   n := ColumnDimension(M);
>   MI := Matrix([[M],[Matrix(n,shape=identity)]]);
>   DI := symdiag(MI);
>   RETURN(DI[n+1..2*n,1..n]);
> end;
```

4 CÒNIQUES I QUÀDRIGUES

Una quàdrica de $\mathbb{R}^n$ és (el conjunt de zeros de) un polinomi de segon grau amb n indeterminades (i coeficients reals). Dos quàdriques Q_1 i Q_2 no buides són equivalents si existeix una afinitat bijectiva $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ tal que $f(Q_1) = Q_2$. Una quàdrica de $\mathbb{R}^2$ s'anomena *cònica*.

4.1 Polinomis quadràtics

Recordem que un polinomi de segon grau en n indeterminades, i a coeficients reals, és una expressió del tipus

$$Q(x_1, \dots, x_n) = \mathbf{x}^\top A \mathbf{x} + B \mathbf{x} + C$$

on A és una matriu real $n \times n$ simètrica, B és una matriu real $1 \times n$ (una fila), $C \in \mathbb{R}$, i $\mathbf{x}$ és la matriu $n \times 1$ (una columna) donada per

$$\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}.$$

Observem que tot polinomi de segon grau, $Q(\mathbf{x}) = \mathbf{x}^\top A \mathbf{x} + B \mathbf{x} + C$, es pot escriure com un producte de matrius:

$$Q(\mathbf{x}) = \mathbf{x}^\top A \mathbf{x} + B \mathbf{x} + C = (\mathbf{x}^\top \quad 1) \begin{pmatrix} A & B^\top/2 \\ B/2 & C \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}.$$

La matriu simètrica A s'anomena la matriu quadràtica de Q , i $\tilde{A} = \begin{pmatrix} A & B^\top/2 \\ B/2 & C \end{pmatrix}$ la matriu total de Q .

Exemple. El polinomi $Q(x, y, z) = x^2 - y^2 - 4xz - 8yz - 3y + z - 1$ es pot escriure com

$$\begin{aligned} Q(x, y, z) &= (x \ y \ z) \begin{pmatrix} 1 & 0 & -2 \\ 0 & -1 & -4 \\ -2 & -4 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + (0 \ -3 \ 1) \begin{pmatrix} x \\ y \\ z \end{pmatrix} - 1 \\ &= (x \ y \ z \ 1) \begin{pmatrix} 1 & 0 & -2 & 0 \\ 0 & -1 & -4 & -3/2 \\ -2 & -4 & 0 & 1/2 \\ 0 & -3/2 & 1/2 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}. \end{aligned}$$

Vegem amb Maple que la matriu total del polinomi $Q(x, y, z)$ està bé calculada:

```
> restart; with(LinearAlgebra):
> A := Matrix([[1,0,-2],[0,-1,-4],[-2,-4,0]]);
> B := Matrix([[0,-3,1]]);
> C := -1;
```

Construïm M , matriu total de $Q(x, y, z)$:

```
> M := Matrix([[A,Transpose(B/2)],[B/2,C]]);
```

Realitzem l'operació $(x \ y \ z \ 1) M \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$:

```
> PP := Vector([x,y,z,1]);
> Transpose(PP) . M . PP;
```

Efectivament dóna $Q(x, y, z)$.

4.2 Invariants

Recordem que l'índex $i(A)$ d'una matriu real simètrica A és el nombre

$$i(A) = \min\{r^+(A), r^-(A)\}.$$

Aquí $r^+(A)$ és el nombre de valors propis positius de A , i $r^-(A)$ és el nombre de valors propis negatius de A . Gràcies al **Teorema de Descartes** tenim que $r^+(A)$ és igual al nombre de canvis de signes dels coeficients no nuls consecutius del polinomi característic de A .

Donat un polinomi $Q(\mathbf{x}) = \mathbf{x}^\top A \mathbf{x} + B \mathbf{x} + C$, denotem

$$\rho = \text{rk}(A), \quad i = i(A)$$

$$\tilde{\rho} = \text{rk}(\tilde{A}), \quad \tilde{i} = i(\tilde{A}),$$

$$\text{on } \tilde{A} = \begin{pmatrix} A & B^\top/2 \\ B/2 & C \end{pmatrix}.$$

Diem llavors que ρ i i són respectivament el rang i l'índex quadràtic de Q , i que $\tilde{\rho}$ i $\tilde{i}$ són respectivament el rang i l'índex de $\tilde{Q}$. Es diu que aquest quatre números $(\rho, i, \tilde{\rho}, \tilde{i})$ són invariants associats al polinomi perquè no varien en canviar el polinomi per un d'equivalent.

Exemple. Calculem els invariants de $Q(x, y, z) = x^2 - y^2 - 4xz - 8yz - 3y + z - 1$ de l'exemple anterior.

> Rank(A);

Per tant, $\rho = 3$.

> CharacteristicPolynomial(A,t);

Fixant-nos en els canvis de signe dels coeficients del polinomi característic sabem que $r^+(A) = 2$. Així que $r^-(A) = \rho - r^+(A) = 3 - 2 = 1$ i per tant $i = 1$.

Calculem ara el rang i polinomi característic de la matriu total que havíem anomenat M :

> Rank(M);

> CharacteristicPolynomial(M,t);

Per tant $\tilde{\rho} = 4$, $r^+(M) = 2$ i com a conseqüència $r^-(M) = \rho - r^+(M) = 4 - 2 = 2$. L'índex de M és $\tilde{i} = 2$. Concloem llavors que els invariants de Q són $(\rho, i, \tilde{\rho}, \tilde{i}) = (3, 1, 4, 2)$.

De la classificació de les quàdriques de $\mathbb{R}^3$ (vegeu apèndix), concloem que Q és un hiperbolóide d'un full. És a dir, és equivalent a la quàdrica

$$x^2 - y^2 - z^2 + 1.$$

5 DIBUIXOS

MAPLE pot dibuixar corbes i superfícies definides de forma implícita — és a dir per equacions — mitjançant les comandes `implicitplot` i `implicitplot3D` que pertanyen al paquet `plots`. Hem de fer, doncs:

```
> restart; with(plots);
```

5.1 Dibuixos 2D

Per dibuixar una corba donada per una equació, hem de especificar, a més de l'equació, la part de $\mathbb{R}^2$ que volem que apareix al dibuix. Per dibuixar la hipèrbola $Q = x^2 - y^2 - 1$ amb valors de x i y entre -10 i 10 fem

```
> Q := x^2-y^2-1;
> implicitplot(Q, x=-10..10, y=-10..10);
```

Quan MAPLE fa un dibuix, evidentment ja no es tracta més de manipulació simbòlica. De fet, MAPLE simplement calcula un quantes punts, dibuixa cadascu i interpola entre ells. A vegades, el nombre de punts calculats per defecte (625) no dóna un bon dibuix. Podem millorar la qualitat amb una indicació opcional de tipus

```
> implicitplot(Q, x=-10..10, y=-10..10, numpoints=10000);
```

Per combinar més que una corba en un dibuix, la comanda accepta també una llista de corbes, al lloc d'una sola corba. En aquest cas es pot ajustar els colors:

```
> QQ := x^2+y^2-25;
> implicitplot([Q,QQ], x=-10..10, y=-10..10, color=[blue,red]);
```

5.2 Dibuixos 3D

Per fer dibuixos de superfícies de $\mathbb{R}^3$ donades per una equació, la sintaxi és la mateixa, però amb la comanda `implicitplot3d`:

```
> S := x^2-4*x*z-3*y-y^2-8*y*z+z-1;
> implicitplot3d(S, x=-10..10, y=-10..10, z=-10..10, numpoints=10000);
```

Per defecte, aquesta comanda no dibuixa els eixos de coordenades. Si volem que apareguin, hem de fer mitjançant l'opció `axes=normal` (o alguna de les seves variants, com `axes=framed` o `axes=boxed`):

```
> implicitplot3d(S,x=-10..10,y=-10..10,z=-10..10, numpoints=10000, axes=boxed);
```

És interessant recordar que moveu el ratolí per sobre de la figura mantenint el botó esquerre apretat, podeu moure el gràfic i observar la figura des de diferents punts de vista. (A més, si cliqueu sobre el gràfic amb el botó dret del ratolí apareix un menú que permet modificar algunes de les propietats del gràfic. Per exemple, proveu d'anar a l'opció `Axes` i proveu els diferents tipus d'eixos.)

5.3 Animacions

És fàcil, i moltes vegades útil, d'animar dibuixos en MAPLE. Es tracta de definir objectes geomètrics que depenen d'un paràmetre t , que interpretem com el temps, interpretar el objecte com a una família d'objectes per a cada valor de t , i veure com varien aquests membres amb el temps. La comanda relevant és `animate`.

Vegem un exemple simple:

```
> Pt := x^2-y^2-t;
> animate(implicitplot,[Pt,x=-4..4,y=-4..4,numpoints=10000],t=-1..1);
```

Com es ve, la comanda `animate` té tres arguments: el primer és el nom de la comanda de dibuix, en aquest cas `implicitplot`, el segon és una llista de tots els arguments necessaris a la primera comanda, i finalment un argument que posa una constricció al paràmetre t .

L'animació no comença automàticament. Cal primer fer clic al dibuix, i després a la barra de botons clicar el botó ▷.

L'interés de les animacions, és que veiem com les quàdriques degenerades, o singulars, apareixen com membres especials de famílies de quàdriques. Típicament, si la família varia linealment en el paràmetre t , llavors per a quasi tots els valors de t tenim una quàdrica no degenerada, i només per un o dos valors apareix una quàdrica degenerada.

Una família d'objectes geomètrics que varia amb un paràmetre t sovint es pot interpretar com a una interpolació entre dos objectes, diguem l'objecte de temps $t = 0$ i l'objecte de temps $t = 1$. Com exemple, agafem qualssevol dues còniques, i fem combinacions afins d'elles:

```
> Q0 := x^2-y^2-x;
> Q1 := x*y+y^2-1;
> Qt := (1-t)*Q0 + t*Q1;
```

Llavors per a $t = 0$ tenim Q_0 , per a $t = 1$ tenim Q_1 , i per a qualsevol valor de t la combinació també és una cònica, atès que la combinació dels dos polinomis de grau 2 també és un polinomi de grau 2.

```
> animate(implicitplot,[Qt,x=-4..4,y=-4..4,numpoints=10000],t=0..1);
```

De fet, aquesta tècnica té una interpretació geomètrica important: el conjunt de totes les còniques formen un espai afí $\mathbf{S}$ de dimensió 5: és l'espai de totes les coordenades possibles de còniques. (Més correctament és un espai *projectiu*, però aquí no fa mal pensar-lo com a un espai afí.) Cada cònica correpon doncs a un punt d'aquest espai $\mathbf{S}$, i quan definim una interpolació entre dues còniques no estem fent res més que construir la recta que passa per aquests dos punts de $\mathbf{S}$, o més exactament, un segment d'aquesta recta. Així tota recta de $\mathbf{S}$ defineix una família de còniques.

Ara passem a fer animacions en dimensió 3:

```
> animate(implicitplot3d,[x^2-y^2-t*z,x=-4..4,y=-4..4,z=-4..4,numpoints=10000],t=-1..1);
> animate(implicitplot3d,[x^2+y^2-z^2+t,x=-4..4,y=-4..4,z=-4..4,numpoints=10000],t=-1..1);
```

Observeu que es pot moure el dibuix amb el ratolí per aconseguir un angle de visió agradable abans de començar l'animació.

Fem una interpolació entre un hiperbolóide d'un full, i una paràbola hiperbòlica — vegueu l'apèndix, si no us recordeu bé de la classificació:

```
> Q0 := x^2-y^2-z^2+1;
> Q1 := x^2-y^2+z;
> Qt := (1-t)*Q0 + t* Q1;
> animate(implicitplot3d,[Qt,x=-4..4,y=-4..4,z=-4..4,numpoints=10000],t=0..1);
```

6 APÈNDIX

Teorema de classificació

Tot polinomi quadràtic en n indeterminades és equivalent a un i no més un dels polinomis següents:

$$\text{I. } r_I(x) = (x_1^2 - x_2^2) + \cdots + (x_{2i-1}^2 - x_{2i}^2) + x_{2i+1}^2 + \cdots + x_\rho^2,$$

$$\text{II. } r_{II}(x) = (x_1^2 - x_2^2) + \cdots + (x_{2i-1}^2 - x_{2i}^2) + x_{2i+1}^2 + \cdots + x_\rho^2 + 1,$$

$$\text{III. } r_{III}(x) = (x_1^2 - x_2^2) + \cdots + (x_{2i-1}^2 - x_{2i}^2) - x_{2i+1}^2 + \cdots - x_\rho^2 + 1,$$

$$\text{IV. } r_{IV}(x) = (x_1^2 - x_2^2) + \cdots + (x_{2i-1}^2 - x_{2i}^2) + x_{2i+1}^2 + \cdots + x_\rho^2 + x_{\rho+1},$$

on $\rho > 0$, $0 \leq 2i \leq \rho \leq n$ en els casos (I) i (II), $0 \leq 2i < \rho \leq n$ en el cas (III), i $0 \leq 2i \leq \rho < n$ en el cas (IV).

A més, per a cada cas els invariants són de la següent forma:

I	(ρ, i, ρ, i)	$2i \leq \rho \leq n$
II	$(\rho, i, \rho + 1, i)$	$2i \leq \rho \leq n$
III	$(\rho, i, \rho + 1, i + 1)$	$2i < \rho \leq n$
IV	$(\rho, i, \rho + 2, i + 1)$	$2i \leq \rho < n$

Com a conseqüència del teorema tenim que dos polinomis quadràtics són equivalents si i només si tenen els mateixos invariants $(\rho, i, \tilde{\rho}, \tilde{i})$.

Així, per exemple, el polinomi $Q(x, y, z) = x^2 - y^2 - 4xz - 8yz - 3y + z - 1$ és equivalent al polinomi

$$r_{III}(x, y, z) = x^2 - y^2 - z^2 + 1$$

ja que $(\rho, i, \tilde{\rho}, \tilde{i}) = (3, 1, 4, 2) = (\rho, i, \rho + 1, i + 1)$.

6.1 CLASSIFICACIÓ AFÍ DE LES QUÀDRQUES

Dues quàdriques no buides $\mathcal{Q}_1$ i $\mathcal{Q}_2$ d'un espai afí real $\mathbb{A}$ són equivalents si existeix una afinitat bijectiva f tal que $f(\mathcal{Q}_1) = \mathcal{Q}_2$.

Teorema (Equivalència entre polinomis i quàdriques)

Dues quàdriques no buides $Q(\mathbf{x})$ i $Q'(\mathbf{x})$ d'un espai afí real $\mathbb{A}$ són equivalents si i només si els polinomis que les defineixen, $Q(\mathbf{x})$ i $Q'(\mathbf{x})$, son equivalents.

Per tant, dos quàdriques no buides són equivalents quan els quatre invariants associats al polinomi quadràtic que defineix cada quàdrica, coincideixen.

Per a cada quàdrica podem trobar una quàdrica equivalent definida per un dels polinomis quadràtics que estableix el **Teorema de classificació**.

Cas $n = 2$:

	ρ	i	$\tilde{\rho}$	$\tilde{i}$	Equació	Nom
I	2	0	2	0	$x^2 + y^2 = 0$	Punt
I	2	1	2	1	$x^2 - y^2 = 0$	Dues rectes
I	1	0	1	0	$x^2 = 0$	Recta doble
II	2	1	3	1	$x^2 - y^2 + 1 = 0$	Hipèrbola
II	2	0	3	0	$x^2 + y^2 + 1 = 0$	Buit
II	1	0	2	0	$x^2 + 1 = 0$	Buit
III	2	0	3	1	$-x^2 - y^2 + 1 = 0$	El·lipse
III	1	0	2	1	$-x^2 + 1 = 0$	Dues rectes paral.
IV	1	0	3	1	$x^2 + y = 0$	Paràbola

Cas $n = 3$:

	ρ	i	$\tilde{\rho}$	$\tilde{i}$	Equació	Nom
I	3	0	3	0	$x^2 + y^2 + z^2 = 0$	Punt
I	3	1	3	1	$x^2 + y^2 - z^2 = 0$	Con
I	2	0	2	0	$x^2 + y^2 = 0$	Recta
I	2	1	2	1	$x^2 - y^2 = 0$	Dos plans
I	1	0	1	0	$x^2 = 0$	Pla doble
II	3	0	4	0	$x^2 + y^2 + z^2 + 1 = 0$	Buit
II	3	1	4	1	$x^2 + y^2 - z^2 + 1 = 0$	Hiperb. dos fulls
II	2	0	3	0	$x^2 + y^2 + 1 = 0$	Buit
II	2	1	3	1	$x^2 - y^2 + 1 = 0$	Cilindre Hiperb.
II	1	0	2	0	$x^2 + 1 = 0$	Buit
III	3	1	4	2	$x^2 - y^2 - z^2 + 1 = 0$	Hiperb. un full
III	3	0	4	1	$-x^2 - y^2 - z^2 + 1 = 0$	El·lipsoide
III	2	0	3	1	$-x^2 - y^2 + 1 = 0$	Cilindre El·lip.
III	1	0	2	1	$-x^2 + 1 = 0$	Plans paral.
IV	2	0	4	1	$x^2 + y^2 + z = 0$	Parab. El·líptic
IV	2	1	4	2	$x^2 - y^2 + z = 0$	Parab. Hiperbòlic
IV	1	0	3	1	$x^2 + z = 0$	Cilindre Parab.

Observació: Es pot veure que en el cas $n = 2$, prescindint de les còniques buides, la classificació es pot obtenir ràpidament calculant un parell de determinants: si denotem

$$d = \det A,$$

$$D = \det \begin{pmatrix} A & B^\top/2 \\ B/2 & C \end{pmatrix} = \det \tilde{A}.$$

$D \neq 0$	$d > 0$	El·lipse
	$d < 0$	Hipèrbola
	$d = 0$	Paràbola
$D = 0$	$d > 0$	Punt
	$d < 0$	Dues rectes que es tallen
	$d = 0$	Dues rectes paral·leles o coincidents