Curs pràctic de Maple
(febrer–maig del 2002)

G. Guasp

Contingut

Introducció

Aquests textos són el recull de les pràctiques realitzades en el curs *Curs pràctic de Maple* organitzat pel Dept. de Matemàtiques i la Fac. de Ciències de la UAB i realitzat entre el febrer i el maig del 2002. Poden servir per a fer de nou un curs d'aquest mateix tipus o, fins i tot, com a guia d'auto-aprenentatge del programa Maple. El que no es pretén és que siguin un *manual* en el que trobar quina és la comanda adequada per a resoldre un problema determinat. De fet, el menú d'ajuda del mateix programa és prou complet com per a poder considerar-lo com un vertader manual i, per tant, una de les activitats sovint recomanades al llarg de totes aquestes pràctiques és la d'anar consultant, en l'apartat d'ajuda, les diferents opcions de les comandes que es presenten, amb la finalitat d'acostumar als seguidors del curs a aquesta tasca imprescindible per a dominar en profunditat el programa.

La majoria d'aquestes pràctiques també s'han fet servir en l'assignatura *Pràctiques Integrades* de la llicenciatura de matemàtiques durant aquest mateix curs. També han estat responsables d'aquesta assignatura N. Castellana, H. Pantazi, F. Reverter i A. Ruiz i tots ells han aportat alguna cosa als textos que es recullen.

No tot el material que apareix aquí és totalment original, una part d'aquestes pràctiques no són res més que adaptacions d'alguns dels materials que es poden trobar lliurement a `http://www.maplesoft.com` i n'hi ha d'altres que estan directament inspirades en alguns dels capítols del llibre *The Maple Book* de F. Garvan (publicat per Chapman & Hall/CRC, ISBN:1-58488-232-8).

1 Càlculs Numèrics

En aquesta secció utilitzareu Maple per a fer alguns càlculs numèrics estàndard. L'habilitat de Maple per a donar resultats exactes, a més de les aproximacions numèriques, us donarà més opcions a l'hora de resoldre problemes.

1.1 Realitzant càlculs aritmètics exactes amb Maple

Utilitzar Maple per a realitzar càlculs numèrics és molt fàcil. Només s'ha d'introduir l'expressió numèrica i acabar la línia amb un **punt i coma (;)**. Prement [**Retorn**] executareu la línia i el resultat apareixerà en color blau en el centre de la pantalla.

Exemple 1.1
Introduïu els càlculs simples següents i premeu [**Retorn**].

```
> 2+4;
> 12*34567890;
```

Cada línia vermella és "viva" i es pot modificar en qualsevol moment. Canvieu el "4" de la línia anterior per un "8" i premeu [**Retorn**]. Notareu com el resultat de color blau s'actualitza automàticament per a mostrar el nou resultat.

Exemple 1.2
Pel nostre exemple següent calculem 134^{39}.

```
> 134^39;
```

A diferència de la vostra calculadora, Maple us donarà la resposta exacta d'aquest problema, les 83 xifres que té.

Exemple 1.3
Maple pot fer càlculs amb fraccions sense convertir-les en expressions decimals:

```
> 3/5 + 5/9 + 7/12;
```

Exemple 1.4
Per a introduir l'arrel quadrada d'un nombre utilitzeu `sqrt( )` :

```
> sqrt(24);
```

Noteu que Maple ha simplificat $\sqrt{24}$ però que ha deixat el resultat en forma exacta. En el proper apartat aprendreu com obtenir una aproximació decimal per a aquest nombre.

Exemple 1.5
Maple té introduïdes totes les constants matemàtiques importants. Per a introduir π escriviu `Pi`. Fixeu-vos que es necessita un asterisc `*` per a indicar la multiplicació.

```
> 4*(3+Pi);
```

Una altra vegada Maple fa el càlcul però deixa el resultat en una forma exacta.

Exemple 1.6
A diferència de la vostra calculadora, Maple dóna la solució exacta quan s'aplica a les funcions trigonomètriques.

```
> sin(5*Pi/3);
> sec(Pi/4);
```

Per a obtenir l'invers del sinus d'un nombre utilitzeu la funció `arcsin( )`:

```
> arcsin(-1);
```

Si demaneu a Maple que calculi un valor *indeterminat* respondrà amb un missatge d'error:

```
> tan(Pi/2);
```

Exemple 1.7
Per a introduir la funció exponencial e^x en Maple escriviu: `exp(x)`.

```
> exp(x);
```

I per a obtenir el nombre e escriviu: `exp(1)` .

```
> exp(1);
```

Exemple 1.8
Per a introduir la funció valor absolut $|x|$ en Maple escriviu: `abs(x)`. Notareu que Maple dóna la solució correcta i exacta per a la tercera línia ja que: $e - \pi < 0$

```
> abs(x);
> abs(-3);
> abs(exp(1)-Pi);
```

Exemple 1.9
Maple té moltes comandes per a realitzar càlculs específics amb nombres. Les podreu aprendre així que es vagin necessitant. Aquí presentem un últim exemple per ara: si teniu un nombre enter i voleu obtenir la seva descomposició en factors primers podeu utilitzar la comanda de Maple `ifactor( )`. Podeu practicar tot el que vulgueu canviant el nombre que hi ha aquí sota.

```
> ifactor(31722722304);
```

Exemple 1.10
Pot haver hi molts moments en els que vulgueu introduir més d'una comanda en la mateixa línia. Això es pot fer en Maple, només cal assegurar-se d'acabar **cada una** de les comandes amb un punt i coma (`;`). També ajudarà posar espais en blanc entre les comandes. Quan premeu [**Retorn**] totes les expressions s'executaran i els resultats apareixeran, en ordre, en un únic camp de resultats.

```
> sin(Pi/3);   cos(Pi/3);   tan(Pi/3);
```

Exemple 1.11
Per a calcular i mostrar una successió de nombres utilitzeu la comanda `seq( )`. Aquí calculem els quadrats dels 100 primers nombres naturals

```
> seq(k^2,k=1..100);
```

1.2 Aproximacions numèriques utilitzant la comanda `evalf`

Recordeu que en l'apartat anterior hem demanat a Maple que fes la suma de tres fraccions i que ha donat com a resultat una altra fracció. Aquest tipus d'aritmètica exacta és molt útil però hi ha vegades que preferirem una resposta en forma d'expressió decimal. La comanda de Maple `evalf( )` realitza aquesta tasca per nosaltres.

Exemple 1.12
Compareu els resultats de les dues línies següents

```
> 3/5+5/9+7/12;
> evalf(3/5+5/9+7/12);
```

Exemple 1.13
Donar un nom al resultat d'un determinat càlcul fa que sigui més fàcil utilitzar-lo en un càlcul posterior. Per assignar un nom utilitzem uns **dos punts (:)** seguit del **signe igual (=)** (és a dir `nom := resultat`). En la línia que ve a continuació hem assignat a la lletra `k` el resultat del càlcul anterior. Després apliquem `evalf( )` a `k`.

```
> k:=3/5+5/9+7/12;
> evalf(k);
```

Nota important de Maple: Maple distingeix entre majúscules i minúscules. Per tant Maple considera que `k` i `K` són variables diferents.

```
> k;
> K;
```

Per cert, també es poden utilitzar paraules com a nom de variables.

```
> josep:=2^5;
> sqrt(josep);
```

Exemple 1.14
Si volem més o menys dígits de precisió en comptes dels 10 que s'utilitzen per defecte, podem afegir un argument extra a la comanda `evalf( )` tal i com es mostra a continuació.

```
> w:=4*(3+Pi);
> evalf(w);
> evalf(w,4);
> evalf(w,45);
```

Exemple 1.15
Si introduïu nombres amb un punt decimal Maple donarà de forma automàtica un resultat decimal. Compareu els dos resultats que apareixen a continuació.

```
> sqrt(34);
> sqrt(34.0);
```

Aquí hi ha un altre exemple:

```
> 4-1/3;
> 4.0-1/3;
```

Exemple 1.16
Podem aplicar la comanda `evalf( )` a una successió de nombres: aquí sota generem primer les arrels quadrades exactes dels 10 primers nombres naturals, després apliquem `evalf( )` per a obtenir les aproximacions decimals.

```
> result:=seq(sqrt(k),k=1..10);
> evalf(result);
```

Drecera de Maple: referir-se de manera ràpida a l'últim resultat
Quan utilitzeu Maple hi ha moltes vegades que voldreu encadenar una sèrie de càlculs. En comptes de donar un nom a cada un dels resultats que aneu obtenint, podeu utilitzar el signe tant per cent (`%`) per referir-vos a **l'última expressió que Maple ha calculat.** Aquí hi ha alguns exemples de com funciona.

```
> 3/5+5/9+7/12;
> evalf(%);
> Pi;
```

```
> evalf(%);
> %+5;
```

Per a més informació de com fer servir aquest símbol mireu l'exercici 1.4 que ve tot seguit.

Exercici 1.1
Utilitzeu Maple per a calcular el nombre 37^{43}.

Exercici 1.2
Calculeu $\sqrt{34}$ amb 18 xifres.

Exercici 1.3
Obteniu una aproximació numèrica per a l'expressió : $\dfrac{3+\pi}{7-\sqrt{13}}$

Exercici 1.4
El símbol tant per cent (`%`) és una drecera molt útil però pot donar en alguns casos resultats inesperats. Aquí hi ha un exemple:

Primer executeu cada una de les tres línies següents. Heu de poder dir quin serà el resultat per endavant.

```
> 4+Pi;
> evalf(%);
> %+10;
```

Ara torneu enrera i torneu a executar l'última línia (i.e. `%+10;`). Fixeu-vos que el resultat canvia de 17.14159265 a 27.14159265
Podeu explicar el perquè?

1.3 "Netejant" Variables

Des del moment en que definiu una variable, Maple recordarà quin és el seu valor durant tota la sessió de treball. Si voleu introduir un nou valor a la variable, només heu de fer la nova assignació.

Per exemple, cada una de les assignacions que hi ha aquí baix redefineix el valor de la variable `h`. (Nota: per a verificar el valor actual d'una variable només cal escriure el seu nom seguit d'un punt i coma).

```
> h;
> h:=56;
> h;
> h:=sqrt(Pi);
> h;
```

A vegades voldreu "netejar" una variable de la memòria per a poder utilitzar-la en una situació nova. Aquí hi ha un exemple. Primer introduïm en `x` el valor 65.

```
> x:=65;
```

Ara suposem que comencem un problema nou i que volem introduir l'expressió algebraica genèrica $x^2-4\,x+7$ i assignar-li el nom `w`. Si només fem això, Maple substitueix automàticament el valor anterior de `x`.

```
> w:=x^2-4*x+7;
```

Per a fer que `x` sigui una variable genèrica un altre cop hem de "netejar" (és a dir esborrar la memòria de Maple) el nostre antic valor per a `x`. Això s'aconsegueix introduint `x:='x';`. Fixeu-vos que aquí utilitzem una cometa (apòstrof). Executeu les dues línies següents per a veure com funciona.

```
> x:='x';
> w:=x^2-4*x+7;
```

Netejant totes les variables de cop: la comanda `restart`.

La comanda `restart` neteja la memòria de Maple de totes les definicions que hagueu fet. És com si es comencés una nova sessió de Maple. Si heu de començar un problema totalment nou podeu utilitzar la comanda `restart` per assegurar que no queden definicions del vostre treball anterior. Abans d'executar la segona línia d'aquí sota, endevineu ràpidament el resultat.

```
> p:=4;
> p;  x;  h;
```

Probablement recordareu que `p` era 4 i que `x` s'havia reassignat a la variable genèrica `x`, però poder no haureu recordat que `h` s'havia assignat al valor $\sqrt{\pi}$. És per això que és una bona idea utilitzar `restart` per a eliminar totes les definicions d'un sol cop. (Si aneu seguint aquestes pràctiques veureu que quan comenceu moltes seccions noves es recomana utilitzar una comanda `restart`).

```
> restart;
> p;  x;  h;
```

2 Càlculs Algebraics

Maple és un "C.A.S" , i.e. un **"Computer Algebra System"** (en català es diu normalment que és un *Manipulador Algebraic).* Això significa que Maple coneix totes les regles algebraiques que vosaltres sabeu. Al mateix temps que aneu progressant en el Càlcul, les Equacions diferencials i l'Àlgebra lineal veureu que Maple també conté les operacions essencials d'aquests temes introduïdes en el seu gran conjunt de comandes.

En aquesta secció aprendreu com introduir una expressió algebraica i a donar valors a les seves variables. Després aprendreu les comandes que us permetran expandir, factoritzar i simplificar expressions.

```
> restart;
```

2.1 La comanda `subs`

Exemple 2.1

Com a primer exemple comencem amb l'expressió $3\,x^2+8$ i assignem-li com a nom `W`.

```
> W:=3*x^2+8;
```

Ara suposem que volem substituir la variable `x` de l'expressió $3\,x^2+8$ pel valor 4. La forma més ràpida de fer-ho és utilitzar la comanda de Maple `subs( )`. Aquí veiem com

```
> subs(x=4,3*x^2+8);
```

De forma alternativa podem aplicar la comanda `subs( )` a `W`.

```
> subs(x=4,W);
```

Exemple 2.2

La comanda `subs( )` també funciona bé amb valors simbòlics:

Per a substituir x per $5+2\,u$ en l'expressió $3\,x^2+8$ executem la línia següent (En aquest cas posem l'etiqueta `M` al resultat).

```
> W:=3*x^2+8;
> M:=subs(x=5+2*u,W);
```

I ara per a fer que Maple "multipliqui" aquesta expressió utilitzem la comanda `expand( )` .

```
> expand(M);
```

Exemple 2.3

La comanda `subs( )` és molt versàtil. Podem utilitzar-la per a avaluar expressions en les que intervenen més d'una variable. Aquí substituïm la x per 7 i la y per 12 en l'expressió $U = \frac{2\,x^2}{5} + 3\,y$.

```
> U:=(2/5)*x^2+3*y;
> subs(x=7,y=12,U);
> evalf(%);
```

Exemple 2.4

També podem utilitzar la comanda `subs( )` per a substituir un valor en una equació. Això és el tipus de coses que cal fer per a verificar si un determinat valor "satisfà" l'equació. En les línies que venen a continuació substituirem diferents valors en l'equació $x^3-5\,x^2+7\,x-12=0$. És algun d'aquests valors una solució de l'equació?

Noteu que utilitzem "`:=`" per assignar el nom i només "`=`" per a l'equació mateixa.

```
> eqn:=x^3-5*x^2+7*x-12=0;
> subs(x=3,eqn);
> subs(x=4,eqn);
```

```
> subs(x=5,eqn);
```

Exercici 2.1
Assigneu el nom `k` a l'expressió $x^2 + 4\,x - 3$. Un cop fet això dieu `M` a l'expressió $k^2 - 9$. Finalment feu que Maple calculi `3*M+6;`. Nota: per fer que Maple multipliqui les expressions utilitzeu la comanda `expand( )`. És a dir, introduïu `expand(3*M+6);`. Aprendreu més coses sobre la comanda `expand( )` en l'apartat següent.

Exercici 2.2
Desenvolupeu $(1 + x)^4$ utilitzant la comanda `expand( )`.

Exercici 2.3
Sigui $P = a\,x^3 + b\,x^2 + c\,x + d$. Calculeu P si $x = 0.01$, $a = -\dfrac{1}{5}$, $b = \dfrac{2}{5}$, $c = 0$, i $d = \dfrac{13}{15}$

Exercici 2.4
Utilitzeu la comanda `subs( )` per a verificar si algun dels nombres 1, 2 o 3 és una solució de l'equació: $x^3 - 16\,x^2 + 51\,x - 36 = 0$

2.2 La comanda `expand`

L'ús principal de la comanda `expand( )` és el de "fer les multiplicacions" en els productes d'expressions polinòmiques. També es pot utilitzar per a desenvolupar funcions trigonomètriques i altres tipus de funcions més generals.

Exemple 2.5
Utilitzem la comanda `expand( )` per efectuar les multiplicacions en $(x + 2)^2\,(3\,x - 3)\,(x + 5)$.

```
> k:=(x+2)^2*(3*x-3)*(x+5);
> expand(k);
```

Exemple 2.6
Maple aplica algunes de les identitats trigonomètriques més comuns per a desenvolupar $\sin(2\,x)$ i $\cos(2\,x)$.

```
> expand(sin(2*x));
> expand(cos(2*x));
```

Feu alguna prova desenvolupant el sinus i el cosinus d'alguns altres múltiples enters de x. Per exemple: $\sin(3\,x)$, $\cos(6\,x)$,....

Exemple 2.7
Aquí hi ha un exemple final. Feu que Maple multipliqui l'expressió $x^{(1/2)}\,(x^{(3/2)} + x^{(-1/2)})$

```
> h:=x^(1/2)*(x^(3/2)+x^(-1/2));
> expand(h);
```

Exercici 2.5
Desenvolupeu $(x + 1)^n$ per a $n = 2, 3$ i 4.

2.3 La comanda `factor`

Exemple 2.8
Factoritzem l'expressió: $3\,x^2 - 10\,x - 8$

```
> w:=3*x^2-10*x-8;
> factor(w);
```

O podem fer el mateix en una sola línia:

```
> factor(3*x^2-10*x-8);
```

Exemple 2.9
Primer desenvolupem l'expressió $2\,(x-2)\,(2\,x^2+5\,x+2)\,(x+4)$. Després apliquem al resultat la comanda `factor( )`. Podeu explicar per què el resultat final sembla diferent de l'expressió original?

```
> H:=2*(x-2)*(2*x^2+5*x+2)*(x+4);
> ans:=expand(H);
> factor(ans);
```

Exemple 2.10
Maple pot factoritzar expressions amb més d'una variable.
Factoritzem l'expressió: $x^2\,y + 2\,xy + y$

```
> h:=x^2*y+2*x*y+y;
> factor(h);
```

Exemple 2.11
Si Maple no pot factoritzar una expressió util·litzant nombres racionals (és a dir, enters i fraccions) donarà com a resultat el mateix que heu introduït sense cap canvi.

```
> factor(3*x^2-10*x-9);
```

Exemple 2.12
La comanda `factor( )` no està limitada als polinomis. Es pot utilitzar per a factoritzar altres formes.
Factoritzeu $(\sin x)^2 - (\cos x)^2$.

```
> factor((sin(x))^2-(cos(x)^2));
```

Exemple 2.13
Si la comanda `factor( )` s'utilitza amb una expressió racional, es factoritza el numerador i el denominador i els factors comuns es cancel·len per a simplificar l'expressió:

```
> A:=(x^3-7*x^2+15*x-9)/(x^2+4*x+4);
> factor(A);
> B:=(x^3-7*x^2+15*x-9)/(x^2-4*x+3);
> factor(B);
```

L'exemple següent us permet veure la forma factoritzada sense les simplificacions.

Exemple 2.14
Les comandes de Maple `numer( )` i `denom( )` us permeten aïllar tant el numerador com el denominador d'una fracció. Aquí utilitzem aquestes comandes per examinar els factors del numerador i el denominador per separat (és a dir, abans de les simplificacions dels factors comuns).

```
> B:=(x^3-7*x^2+15*x-9)/(x^2-4*x+3);
> factor(numer(B));  factor(denom(B));
```

Exercici 2.6
Factoritzeu l'expressió $3\,x^4 - 2\,x^3 + 22\,x^2 - 18\,x - 45$.

Exercici 2.7
Factoritzeu l'expressió $x^{(1/2)} - x^{(3/2)}$ i després utilitzeu la comanda `expand( )` per a comprovar el resultat.

2.4 La comanda `simplify`

Exemple 2.15
Considereu l'expressió $(\cos x)^5 + (\sin x)^4 + 2\,(\cos x)^2 - 2\,(\sin x)^2 - \cos(2\,x)$. Maple pot aplicar identitats per a simplificar expressions matemàtiques llargues, com ara expressions trigonomètriques.

```
> V:=cos(x)^5 + sin(x)^4 + 2*cos(x)^2 - 2*sin(x)^2 - cos(2*x);
> simplify(V);
```

Exemple 2.16
Les expressions trigonomètriques amb arguments donats com a múltiples d'algun angle queden simplificades en termes de funcions trigonomètriques d'aquest angle si és possible:

```
> simplify(sin(5*t)+sin(3*t));
```

Exemple 2.17
La comanda `simplify( )` es pot utilitzar per a sumar expressions racionals. A continuació reescrivim la suma $\dfrac{1}{x+1} + \dfrac{x}{x-1}$ com una única fracció.

```
> M:=(1/(x+1))+(x/(x-1));
> simplify(M);
```

Exercici 2.8
Simplifiqueu l'expressió $\dfrac{7}{x+2} + \dfrac{3\,x}{(x+2)^2}$

Exercici 2.9
Com simplifica Maple $\sin(3\,t) - \sin(7\,t)$? Si aquesta expressió "simplificada" és o no és útil dependrà del que es tingui planejat fer amb ella.

3 Gràfics

En aquesta secció aprendreu com dibuixar el gràfic d'una funció definida per una expressió. Altres temes que hi podreu trobar inclouen: combinar gràfics de diferents expressions en un únic dibuix, representar punts, i combinar diferents estructures gràfiques en un únic dibuix.

Recordeu fer

```
> restart;
```

si veniu de fer altres càlculs.

3.1 Representar una expressió: la comanda `plot`

Exemple 3.1
Utilitzem la comanda `plot( )` per a dibuixar el gràfic de $3\,x^2 - 8$ per a x entre -5 i 5.

```
> plot(3*x^2-8,x=-5..5);
```

Observeu que Maple posa l'escala de l'eix de les y de manera automàtica, triant l'escala de les y que mostra tot el gràfic que correspon al domini que s'ha especificat. Podem eliminar l'elecció automàtica de l'escala de les y especificant a part del rang de les x un rang per a les y. En la línia següent hem limitat el rang de les y a l'interval $[-20, 40]$.

```
> plot(3*x^2-8,x=-5..5,y=-20..40);
```

Si feu clic amb el botó esquerre del ratolí, el gràfic queda seleccionat i la barra d'opcions inferior es modifica. Ara quan feu clic en el gràfic, les coordenades del punt del lloc on ho feu es veuran en la finestra de l'esquerra. El botó $\boxed{1{:}1}$ fa que les escales de les x i de les y siguin iguals. Podeu experimentar sobre el gràfic anterior les diferents opcions possibles.

Exemple 3.2
L'escala automàtica és una característica útil però hi ha cops en els que necessitarem especificar manualment el rang de les y. Per exemple l'escalat automàtic no és apropiat per a gràfics amb asímptotes verticals. Compareu els dos gràfics següents. Noteu que hem fixat els límits per a les y a l'interval $[-20, 20]$ en la segona comanda `plot( )`.

```
> plot(x/(x-2),x=-5..5);
> plot(x/(x-2),x=-5..5,y=-20..20);
```

Exemple 3.3
Dibuixarem el gràfic de $y = x^3 + 1 - e^x$ en el domini $[-8, 8]$. Triarem un rang per a les y que permeti veure els quatre talls amb l'eix de les x. Primer doneu un cop d'ull al gràfic amb l'escala automàtica per a les y.

```
> plot(x^3+1-exp(x),x=-8..8);
```

Com que els valors de les y a prop de 8 són negatius i molt grans en valor absolut, l'escala vertical ha hagut de ser massa gran i no es poden veure clarament els talls amb l'eix de les x. S'obté una visió molt millor restringint els límits en el rang de les y.

```
> plot(x^3+1-exp(x),x=-8..8,y=-5..15);
```

Exercici 3.1
Dibuixeu $y = \sin(x)$ per a dos períodes complets.

Exercici 3.2
Dibuixeu $y = 3\,x^4 - 6\,x^2$ per al domini $[-10, 10]$ amb escala automàtica per a les y. Després d'observar el gràfic, editeu el domini i el recorregut per tal de veure els talls amb l'eix de les x clarament. Feu una estimació dels talls amb l'eix de les x amb el cursor.

3.2 Representar diferents expressions

Per a mostrar més d'un gràfic en el mateix dibuix feu-ne la llista dins de claudàtors `[ ]` separant-los per comes.

```
> plot([cos(x),x^2],x=-1..4,y=-4..4);
```

Noteu que cada un dels gràfics es mostra utilitzant un color diferent. Podeu especificar els colors per a cada funció afegint una opció de color al final de la comanda. Els colors s'assignaran en el mateix ordre que el de les funcions. Fixeu-vos que la llista dels colors també es fa amb uns claudàtors `[ ]`. Aquí hi ha un exemple.

```
> plot([cos(x),x^2],x=-1..5,y=-4..4,color=[blue,black]);
```

Aquests són els colors disponibles en Maple (s'han d'escriure en anglès, no val posar `groc`).

aquamarine	coral	green	magenta	plum	turquoise	yellow
black	cyan	gray	maroon	red	violet	
blue	brown	grey	orange	sienna	wheat	
navy	gold	khaki	pink	tan	white	

Taula 1: Colors disponibles per a comandes `plot( )`

Exercici 3.3
Feu junts els gràfics de les funcions $y = x^2 - 5\,x + 6$ i $y = \dfrac{1}{(x-2)^2}$. Experimenteu amb diferents rangs per a les y de forma que es puguin veure dibuixos complets dels dos gràfics.

3.3 Representar punts

La comanda `plot( )` pot dibuixar també un o més punts.

Exemple 3.4
Dibuixem el punt $(2, 3)$. Noteu en la línia següent que utilitzem dos jocs de claudàtors.

```
> plot([ [2,3] ],style=point);
```

Exemple 3.5
Podem controlar la mesura dels rangs per a les x i per a les y afegint aquesta informació a la comanda com en la línia següent.

```
> plot([ [2,3] ],x=-7..7,y=-7..7,style=point);
```

Exemple 3.6
Per a dibuixar més d'un punt fem una llista dins la comanda `plot( )` (observeu les comes). Recordeu que s'ha de posar un parell de claudàtors per cada punt i un parell extra envoltant la llista.

```
> plot([ [2,3],[-2,5],[1,-4] ],x=-7..7,y=-7..7,style=point);
```

Exemple 3.7
Canviant l'estil a "`line`" es connecten els punts conservant l'ordre de la llista.

```
> plot([ [2,3],[-2,5],[1,-4] ],x=-7..7,y=-7..7,style=line);
```

Exemple 3.8
Es poden utilitzar declaracions opcionals per a especificar el color dels punts i el símbol que es fa servir (per exemple "`diamond`", "`circle`" i "`cross`", que és el que hi ha per defecte) per a representar-los.

```
> plot([[3,2],[-2,3],[2,-1]],style=point,color=blue,symbol=circle);
```

Exercici 3.4
Dibuixeu els punts següents utilitzant el color vermell i el símbol "`diamond`": [1, 4], [−2, −3], [4, −5] i [−6, 5]. Després connecteu els punts amb línies rectes amb una comanda `plot( )` a part.

3.4 Combinar gràfics d'expressions i punts: la comanda `display`

Un *paquet* especial de dibuix anomenat **plots** conté moltes més possibilitats gràfiques. Per a utilitzar les seves comandes, necessiteu executar la línia següent que *carrega* **plots**. Recordeu que els dos punts al final de la línia fa que la línia es pugui executar sense mostrar cap resultat. Per a poder veure el contingut de **plots** podeu substituir els dos punts per un punt i coma.

```
> with(plots):
```

La comanda `display( )` permet combinar gràfics d'expressions i de punts en el mateix dibuix. El primer pas consisteix en nomenar individualment cada un dels components del dibuix. **Important:** Ens hem d'assegurar que utilitzem uns **dos punts** al final de cada línia per a suprimir la presentació dels resultats (mireu les tres primeres línies que hi ha a sota). La comanda `display( )` s'utilitza ara per a fer el dibuix que volem (acaba amb un punt i coma).

```
> pict1:=plot([-3*x+5,9-x^2],x=-3..5,color=[green,red]):
> pict2:=plot([[-1,8],[4,-7]],style=point,color=blue,symbol=circle):
> display([pict1,pict2]);
```

De forma alternativa podem fer la llista d'aquestes tres comandes `plot( )` relacionades en un sol grup d'execució escrivint MAJÚSCULES-RETORN al final de cada línia.

```
> pict1:=plot([-3*x+5,9-x^2],x=-3..5,color=[green,red]):
  pict2:=plot([[-1,8],[4,-7]],style=point,color=blue,symbol=circle):
  display([pict1,pict2]);
```

Exercici 3.5
Feu un gràfic que contingui a l'hora el gràfic de la funció $y = x^2 + x - 6$ i les seves interseccions amb l'eix de les x i el de les y marcades amb un cercle.

4 Resoldre equacions

A aquesta pràctica aprendreu a aplicar la comanda de Maple `solve( )` per a obtenir les solucions **exactes** d'equacions (quan això sigui possible). Recordeu que en nombrosos casos no és possible obtenir solucions exactes de les equacions i que s'ha de deixar la feina als *solucionadors numèrics* per a poder calcular solucions **aproximades**. Més endavant en aquesta mateixa secció utilitzareu la comanda de Maple `fsolve( )` per a obtenir aproximacions decimals de solucions. També es discutirà la resolució de sistemes d'equacions lineals. Comenceu reinicialitzant les variables i carregant el paquet **plots**.

```
> restart:
> with(plots):
```

4.1 Introduir i manipular equacions: les comandes `lhs` i `rhs`

Exemple 4.1
Es pot donar un nom a una equació igual que es fa amb qualsevol altre expressió. En la línia següent introduïm l'equació $x^3 - 5\,x^2 + 23 = 2\,x^2 + 4\,x - 8$ i li posem com a nom `eqn1`.

```
> eqn1:=x^3-5*x^2+23=2*x^2+4*x-8;
```

Exemple 4.2
Podem aïllar la part esquerra i la part dreta de l'equació utilitzant les comandes `lhs( )` i `rhs( )`.

```
> lhs(eqn1);
> rhs(eqn1);
```

Exemple 4.3
Utilitzem les comandes `lhs( )` i `rhs( )` per a obtenir una equació que és equivalent a l'equació original `eqn1` però que té com a part dreta un 0. Posem-li com etiqueta `eqn2`.

```
> eqn2:=lhs(eqn1)-rhs(eqn1)=0;
```

4.2 Obtenir solucions exactes: la comanda `solve`

Primer considerem equacions polinòmiques. Existeixen algoritmes per a calcular les solucions **exactes** per a polinomis que tenen fins a **grau 4** utilitzant únicament radicals. La comanda de Maple `solve( )` coneix aquests algoritmes.

Exemple 4.4
Per a obtenir les solucions exactes de l'equació polinomial $3\,x^3 - 4\,x^2 - 43\,x + 84 = 0$ utilitzem la comanda `solve( )`. Noteu que el segon argument de la comanda li diu a Maple que x és la incògnita respecte de la que volem obtenir la solució.

```
> solve(3*x^3-4*x^2-43*x+84=0,x);
```

Aquí Maple ha trobat les tres solucions i les ha ensenyades.

Exemple 4.5
A vegades voldrem seleccionar una solució de la llista de solucions i utilitzar-la en un altre càlcul. Podem fer això assignant primer un nom (en aquest cas utilitzem la lletra `N`) al resultat de la comanda `solve( )`. Aleshores `N[1]` és el primer nombre de la llista, `N[2]` és el segon i així successivament (fixeu-vos en els claudàtors).

```
> N:=solve(x^2-5*x+3=0,x);
> N[1];
```

Exemple 4.6
Quan es treballa amb la comanda `solve( )` sovint és convenient començar donant un nom a l'equació. Noteu que utilitzem "`:=`" per assignar el nom i només "`=`" per a l'equació pròpiament dita.

```
> eqn1:=7*x^3-11*x^2-27*x-9=0;
```

Seguidament resolem l'equació utilitzant la comanda `solve( )` i assignem el nom `H` al resultat.

```
> H:=solve(eqn1,x);
```

Per a practicar comproveu que cada un d'aquests valors satisfà l'equació. Això es pot fer fàcilment utilitzant la comanda `subs( )`.

```
> subs(x=H[1],eqn1);
```

```
> subs(x=H[2],eqn1);
```

```
> subs(x=H[3],eqn1);
```

Exemple 4.7
A vegades les solucions "exactes" són massa complicades per a poder ser realment útils. En les dues línies següents resolem l'equació $x^3 - 34\,x^2 + 4 = 0$.

```
> eqn1:=x^3-34*x^2+4=0;
```

```
> H:=solve(eqn1,x);
```

Com podreu veure, llegir aquestes solucions exactes és realment un repte! Noteu que I representa $\sqrt{-1}$. Quan una solució és així de complicada pot ser més útil mirar una solució aproximada utilitzant `evalf( )`.

```
> evalf(H);
```

Una bona alternativa a la comanda `solve( )` en una situació d'aquest tipus és la comanda `fsolve( )` que discutirem en la secció següent.

La comanda `solve( )` també es pot utilitzar per a determinar solucions exactes d'equacions **no polinòmiques**. Hi ha una llista d'alguns exemples simples a baix. En tot cas, si les equacions són complicades, per exemple si combinen exponencials, polinomis i expressions trigonomètriques, normalment no es podrà disposar de les solucions exactes. Un altre cop la comanda `fsolve( )` és una alternativa.

Exemple 4.8
Resoleu l'equació: $5\,e^{(x/4)} = 43$

```
> solve(5*exp(x/4)=43,x);
```

Exemple 4.9
A vegades Maple no mostra **totes** les solucions possibles. Com podríeu utilitzar el resultat d'aquí sota per a poder escriure el conjunt de totes les solucions de l'equació?

```
> solve(sin(x)=1/2,x);
```

Exercici 4.1
Resoleu l'equació $x^3 - 11\,x^2 + 7\,x + 147 = 0$. Per què Maple produeix només dues solucions diferents per a aquesta equació cúbica? Per què una d'elles està escrita dues vegades? (**Indicació:** Factoritzeu la part esquerra de l'equació).

4.3 Obtenir solucions aproximades: la comanda `fsolve`

Es pot utilitzar per a aproximar solucions de qualsevol tipus d'equació. Per a equacions polinòmiques `fsolve( )` produeix una **llista completa** de totes les solucions reals en un sol pas (mireu l'exemple 4.10). Per a altres tipus d'equacions, `fsolve( )` es pot utilitzar per a obtenir les solucions **d'una en una** (mireu els exemples 4.11 i 4.12).

Exemple 4.10
La comanda de Maple `fsolve( )` calcularà una aproximació numèrica per a cada una de les solucions reals d'una equació polinòmica. Aproximem totes les solucions reals de l'equació: $x^4 - x^3 - 17\,x^2 - 6\,x + 2 = 0$.

```
> eqn:=x^4-x^3-17*x^2-6*x+2=0;
> fsolve(eqn,x);
```

Les quatre solucions que es mostren a sota ens donen una llista completa de les solucions de l'equació polinòmica.

Exemple 4.11
Determinem **totes?** les solucions reals de l'equació $x^3 + 1 - e^x = 0$ utilitzant la comanda `fsolve( )`.

```
> eqn:=x^3+1-exp(x)=0;
> fsolve(eqn,x);
```

Maple ens dóna una solució real. Aquest cop Maple no ens ha explicat la historia completa. Hi ha altres solucions? Com es poden trobar? En l'exemple 4.12 es presenta un procediment sistemàtic per a determinar les solucions que resten.

Exemple 4.12
Buquem les altres solucions reals de l'equació $x^3 + 1 - e^x = 0$.

El primer pas per a trobar les altres solucions és fer un dibuix del gràfic de la part esquerra de l'equació. **Nota:** Recordeu que els talls amb l'eix de les x de $y = x^3 + 1 - e^x$ es corresponen exactament amb les solucions de l'equació $x^3 + 1 - e^x = 0$.

```
> plot(x^3+1-exp(x),x=-3..5,y=-5..15);
```

El gràfic mostra **quatre** interseccions amb l'eix de les x. Una d'elles correspon a la solució que hem obtingut en l'exemple 4.11. Quina? Com trobaríeu les altres que falten?

Podem estendre la comanda `fsolve( )` per a que miri de trobar una solució en un interval particular. Per exemple per a trobar la solució negativa li demanem a Maple que busqui en l'interval $[-1, -0.2]$ ja que podem veure a partir del gràfic que hi ha exactament una solució en aquest interval.

```
> fsolve(eqn,x=-1..-.2);
```

Per a determinar les altres dues solucions utilitzem `fsolve( )` un altre cop, aquesta vegada buscant a l'interval $[1, 2]$ i a l'interval $[4, 5]$.

```
> fsolve(eqn,x=1..2);
> fsolve(eqn,x=4..5);
```

Què passa si demaneu a Maple que busqui una solució en un interval on no hi ha solucions? Proveu-ho. A partir del gràfic és clar que no hi ha talls amb l'eix de les x (i per tant no hi ha solucions) entre 2 i 4.

```
> fsolve(eqn,x=2..4);
```

Noteu que Maple simplement respon amb la línia que originalment heu introduït, sense cap canvi, quan no pot trobar una solució a l'interval donat.

Hi ha altres solucions? Per exemple, hi ha alguna solució per a x més gran que 5? Podem comprovar això fent més gran l'interval sobre el que fem el gràfic. En la línia següent allarguem l'interval fins al $[-3, 50]$. No apareixen nous talls amb l'eix de les x. El gràfic confirma el que podem esperar mirant els termes de l'equació, és a dir el terme exponencial domina i fa que el gràfic vagi baixant a la llarga.

```
> plot(x^3+1-exp(x),x=-3..50,y=-10..15);
```

Alternativament podem utilitzar la comanda `fsolve( )`, ara buscant en aquest interval més gran.

```
> fsolve(eqn,x=5..50);
```

Tal com esperàvem Maple no haurà trobat solucions.

D'una forma semblant podem comprovar si hi ha solucions **cap a l'esquerra**. Aquí busquem si hi ha solucions a l'interval $[-50, -1]$.

```
> fsolve(eqn,x=-50..-1);
```

Cap ni una tampoc!

Ara tenim una llista completa de les solucions aproximades de l'equació original $x^3 + 1 - e^x = 0$. Són: $-.8251554597$, 0, 1.545007279 i 4.567036837.

Exemple 4.13
Utilitzem `fsolve( )` per a calcular les solucions aproximades de l'equació: $\frac{x^2}{20} - 10\,x = 15\cos(x+15)$. Com en l'últim exemple utilitzarem un gràfic per ajudar-nos a determinar el nombre i la situació aproximada de les solucions. La nostra feina se simplifica si comencem convertint l'equació que tenim en una d'equivalent que té com a part dreta un 0. Així resoldrem l'equació equivalent:

```
> x^2/20-10*x-15*cos(x+15) = 0;
```

Si ara dibuixem el gràfic de la part esquerra d'aquesta equació obtindrem un altre cop solucions en cada un dels talls amb l'eix de les x.

```
> eqn:=x^2/20-10*x-15*cos(x+15)=0;
> plot(lhs(eqn),x=-10..10);
```

A partir del gràfic sembla que hi ha una solució a l'interval $[1, 2]$. Ara dirigirem Maple cap a aproximar una solució en aquest interval.

```
> fsolve(eqn,x=1..2);
```

Hem trobat totes les solucions d'aquesta equació? De fet hi ha una altra solució! Per a trobar-la comenceu estirant l'interval sobre el que heu fet el gràfic. Aleshores utilitzeu `fsolve( )` per a obtenir una aproximació numèrica d'aquesta segona solució.

Exercici 4.2
Determineu totes les solucions de l'equació $x^5 - 4\,x^3 + 3\,x^2 + 7\,x - 1 = 0$. Comenceu mirant un gràfic significatiu.

Exercici 4.3
Determineu totes les solucions de l'equació $x^2 - 2 = \ln(x + 5)$. Utilitzeu el gràfic **d'una** expressió per a localitzar les solucions. Comproveu cada una de les solucions substituint-la en la'equació original.

Exercici 4.4
Els gràfics de $y = 10 - x^2$ i de $y = 4\sin(2\,x) + 5$ es troben dues vegades sobre l'interval $[-5, 5]$.

a) Feu el gràfic de les dues equacions juntes i estimeu amb el ratolí els punts d'intersecció.

b) Escriviu una equació que es pugui resoldre per a determinar les coordenades x dels punts d'intersecció.

c) Utilitzeu `fsolve( )` per a resoldre aquesta equació.

d) Utilitzeu els resultats de la part c) per a estimar les coordenades y dels punts d'intersecció.

e) Sembla que les corbes es poden trobar en un tercer punt a prop de $(1, 9)$. Utilitzeu `fsolve( )` i/o un gràfic significatiu per a demostrar que no hi ha cap punt d'intersecció en aquest lloc.

4.4 Resoldre equacions formals

Sovint Maple pot resoldre equacions formals per a qualsevol de les variables.

Suposeu que volem obtenir la solució per a la variable g de l'equació: $4 - v = 2\,T - k\,g$. La comanda `solve( )` funciona bé en aquest cas.

```
> solve(4-v=2*T-k*g,g);
```

Aquí hi ha una manera més maca per a mostrar el mateix resultat:

```
> g=solve(4-v=2*T-k*g,g);
```

Exercici 4.5
Editeu l'última comanda per a obtenir solucions per a les altres lletres T, k i v.

Exercici 4.6
Resoleu l'equació $x^2 + y^2 = 9$ per a la variable y. Assigneu el conjunt de solucions a una variable que es digui S. Quina relació hi ha entre les solucions `S[1]` i `S[2]`?

4.5 Resoldre sistemes d'equacions lineals utilitzant la comanda `solve`

Recordeu que cal reinicialitzar les variables abans de continuar. Carregueu també el paquet **plots**:

```
> restart:
> with(plots):
```

La comanda `solve( )` també es pot utilitzar per a resoldre un sistema de m equacions lineals amb n incògnites. En direm un **sistema lineal** m **per** n perquè sigui més curt.

Exemple 4.14
Resoldre el sistema 2 per 2: $3\,x + 2\,y = 3$ i $x - y = -4$

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

Un gràfic de les dues funcions subjacents mostrarà que la solució correspon al punt d'intersecció en $(-1, 3)$. Però primer necessitem obtenir la forma explícita de cada una de les dues funcions lineals abans de poder fer-ne el dibuix. Per tant resolem cada una de les equacions respecte y.

```
> y1:=solve(3*x+2*y=3,y);
> y2:=solve(x-y=-4,y);
```

Ara construïm un gràfic format de dues parts: la "`part1`" conté els gràfics de les dues equacions i la "`part2`" dibuixa el punt que és la solució que hem trobat. Aquest punt ha de ser el punt d'intersecció de les dues línies. Ho és efectivament?

```
> part1:=plot([y1,y2],x=-5..5):
> part2:=plot([[-1,3]],style=point,color=blue,symbol=circle):
> display([part1,part2]);
```

Exemple 4.15
Aquí hi ha un exemple de solució d'un sistema **3 per 3** amb incògnites x, y i z. Resolem el sistema 3 per 3: $\{x + y + z = 1,\ 3\,x + y = 3,\ x - 2\,y - z = 0\}$

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

Exercici 4.7
Determineu les solucions del sistema: $4\,x + 3,\ y = 12$, $5\,x - 7\,y = 35$ Comproveu la solució substituint els valors que s'obtenen en les dues equacions del sistema.

Sistemes lineals amb un nombre de solucions infinit

Quan un sistema té més **incògnites** que **equacions** sovint trobem no una si no un nombre infinit de solucions. Aquí hi ha un exemple.

Exemple 4.16
Resolem el sistema: $\{x + y + z = 1, 3\,x + y = 3\}$.

```
> solns:=solve({x+y+z=1, 3*x+y=3});
```

Noteu que aquest cop no obtenim un únic conjunt valors numèrics per a x, y i z. En comptes d'això Maple ens diu com han d'estar relacionats els valors de x, y i z per a construir una solució típica. En particular l'expressió `x=x` que surt en la resposta anterior indica que l'incògnita `x` pot ser **qualsevol** nombre. Ens referirem a aquesta incògnita com la *variable lliure* de la solució. Per a obtenir qualsevol **solució particular** (entre el nombre infinit que hi ha) trieu un valor per a la `x` i utilitzeu-lo per a calcular els valors corresponents de la `y` i de la `z`. Per exemple si considerem `x=4`.

```
> subs(x=4,solns);
```

Una solució és: `x=4`, `y=-9` i `z=6`. Preneu-vos un minut de temps i verifiqueu a ma que aquests tres nombres satisfan realment les equacions originals: $x + y + z = 1$ i $3\,x + y = 3$. Mirem ara la solució generada quan prenem `x=2`.

```
> subs(x=2,solns);
```

Així que dues de les infinites solucions que hi ha són: `((x,y,z)=(4, -9, 6)` i `(x,y,z=)(2, -3, 2)`.

Exercici 4.8
Resoleu el sistema: $\{x + 2\,y + z = 2, 3\,x + y = 1\}$ i doneu com a mínim tres solucions particulars.

5 Funcions i eines bàsiques de programació

5.1 Definir i treballar amb funcions

En aquesta secció aprendreu com definir una funció en Maple. Per funció s'ha d'entendre una nova comanda `f` per a la que és possible demanar quin valor adquireix per a diferents arguments `x` utilitzant la notació `f(x)`. La resta de la secció tracta de com avaluar funcions, resoldre equacions on hi ha funcions, i de com fer gràfiques de funcions.

5.1.1 Definir i esborrar una funció en Maple

Per a distingir entre una funció i una expressió, Maple necessita una notació especial quan es defineix una funció. Per exemple, la funció `f` que fa correspondre a un paràmetre `x` el valor `cos(Pi*x)+3` s'introdueix en Maple com:

```
> f:=x->cos(Pi*x)+3;
```

Preneu nota de la sintaxi que utilitzem. És **absolutament necessari** introduir la fletxa " `->`" construïda amb el signe "menys" i un símbol "més gran que". **Maple no definirà una funció** si introduïu una comanda del tipus `f(x) := cos(Pi*x) + 3`.

A continuació podeu veure una comparació entre una expressió i una funció. Fixeu-vos amb la diferència de la sintaxi i en el resultat que dóna Maple per a cada un dels dos casos.

```
> y:=(x + 2)/(x^3 + 5*x + 2);
> f:=x->(x + 2)/(x^3 + 5*x + 2);
```

Les funcions necessiten sempre una fletxa quan s'han d'introduir, en el resultat que dóna Maple també hi ha d'haver una fletxa. Verifiqueu sempre que en el resultat hi ha la fletxa per a confirmar que heu definit una funció.

Exercici 5.1
Definiu en Maple la funció `h` que actua sobre `x` com $h(x) = x^3 \sin(2\,x + 1)$.

Quan ja heu definit una funció, Maple recorda aquesta funció durant tota la sessió de treball. Si voleu substituir la funció per una nova definició, simplement reescriviu la definició. Per exemple, si voleu substituir la funció anterior `f` per la funció que sobre `x` dóna com a resultat `ln(cos(5*x))`, escriviu:

```
> f:=x->ln(cos(5*x));
```

Podeu confirmar el valor actual de la funció `f` fent

```
> f(x);
```

Si voleu esborrar la funció `f` sense definir-la de nou, escriviu:

```
> f:='f';
```

com si es netegés el valor d'una variable qualsevol. Sempre és una bona idea esborrar les funcions que tingueu quan comenceu un problema nou. De forma alternativa també podeu utilitzar la comanda `restart;` per a netejar tot el que hi hagi en la memòria de la sessió.

5.1.2 Avaluar una funció

Quan heu definit una funció, podeu avaluar-la per a diferents valors o per a expressions literals utilitzant la notació funcional. Abans de definir la funció netegem el valor de `f`.

```
> f:='f';
> f:=x->3*x+x^2;
> f(-1);
```

```
> f(2+sqrt(5));
> evalf(f(2+sqrt(5)));
> f(x+4);
> simplify(%);
> (f(x+h)-f(x))/h;
> simplify(%);
```

Si intervenen més d'una funció, la composició de funcions és fàcil de fer.

```
> g:=x->cos(x)+1;
> f(g(Pi/3));
> j:=x->g(f(x));
> j(x);
```

Exercici 5.2
Definiu la funció `s` que actuï com $s(t) = \dfrac{3+t^2}{\sqrt{3\,t+1}}$ i feu que Maple calculi `s(2)`, `s(t-3)` i `s(t)-s(3)` simplificant els resultats. No oblideu la notació de la fletxa!

5.1.3 Resoldre equacions en les que intervenen funcions

Quan heu definit una funció, podeu resoldre equacions amb aquesta funció de manera exacta o aproximada:

```
> g:='g';
> g:=t->t^3-6*t^2+6*t+8;
> solve(g(t)=0,t);
> fsolve(g(t)=0,t);
```

5.1.4 Gràfics de funcions

La comanda `plot` funciona igualment per a funcions:

```
> h:='h';   y:='y';  x:='x';
> h:=x->x*exp(-x);
> plot(h(x),x=-1..4,y=-2..1);
```

Es poden dibuixar gràfics de diferents funcions simultàniament de la mateixa manera que es fa per a les expressions.

Si es considera una funció que doni $f(x) = \dfrac{2}{(x^2+1)}$ es pot fer el gràfic d'aquesta funció i dels seus desplaçaments horitzontals $f(x+1)$, $f(x-3)$ i $f(x-6)$ amb

```
> f:=x->2/(x^2+1);
> plot([f(x),f(x+1),f(x-3),f(x-6)],x=-5..10,y=-1..3);
```

Podeu identificar cada un d'aquests gràfics?

Exercici 5.3
Definiu la funció que doni $f(x) = 2\,x - \left|x^2 - 5\right|$ i després responeu a les qüestions següents:

a) Quin és el valor de $f(6.5)$?

b) Simplifiqueu el valor de $f(z-4)$, on z és una variable.

c) Feu el gràfic de $f(x)$.

d) Determineu els valors de x per als que $f(x) = 0$.

Exercici 5.4
Definiu les funcions g i h tals que $g(x) = 5\,e^{x/2}$ i $h(x) = x + 10$ i feu el següent:

a) Dibuixeu un gràfic que mostri les dues funcions. Experimenteu amb diferents valors per al domini i per al recorregut.

b) Feu una estimació de les coordenades del punt d'intersecció d'aquests dos gràfics utilitzant el botó esquerra del ratolí.

c) Utilitzeu la comanda `fsolve( )` per a resoldre l'equació $g(x) = h(x)$. Com es relaciona la solució d'aquesta equació amb el que heu obtingut en l'apartat b)?

Exercici 5.5
Definiu la funció $k(x) = x + 3\,\sin(2\,x)$, després feu el següent:

a) Dibuixeu el gràfic d'aquesta funció en el domini $[-1, 8]$.

b) Modifiqueu el dibuix de l'apartat a) per tal que inclogui la línia horitzontal $y = 4$. Utilitzeu aquest nou gràfic per a estimar el nombre i els valors aproximats dels x tals que $k(x) = 4$.

c) Quina funció única hauríeu de dibuixar per a obtenir la mateixa informació que en l'apartat b)?

d) Utilitzeu la comanda de Maple `fsolve( )` per a aproximar totes les solucions de l'equació $k(x) = 4$.

5.2 Programació: lògica, iteracions i procediments

Alguns cops el Maple no tindrà les comandes que necessitem per a algun treball concret. En aquests casos necessitarem definir les nostres pròpies comandes i necessitarem saber com incloure funcions lògiques, iteracions i procediments. Comencem amb funcions lògiques:

5.2.1 Lògica: `if-then-else-end if`

La comanda bàsica és `if` i podem veure el funcionament al següent exemple:

Exemple 5.1
Suposem que volem definir una funció `f(x)` que valgui -1 si `x<0`, 0 si `x=0` i 1 si `x>0`. Podríem fer-ho de la següent manera:

```
> restart;
> f:=x-> if x<0 then -1 elif x=0 then 0 else 1 end if;
```

Comproveu que funciona executant:

```
> f(-.5);f(0);f(.5);
```

Nota: el que hem definit ja estava definit al Maple amb el nom de `signum`.

El que hem fet és definir la funció `f` on hauríem de traduir `if` per "*si*", `elif` per "*altrament si*" i `end if` per "*hem acabat*". Llavors podem pensar la comanda `if` com una comanda en la que cal tenir en compte tres elements:

- El primer que hem de veure és la sintaxi de la funció. Hem d'entrar els arguments `if ... then`, `elif ... then`, `else`, i `end if` sempre i en aquest ordre.

- El segon és la introducció de condicions lògiques. Això són expressions com $x > 0$, ... i que en Maple s'han d'introduir com:

`<`	més petit que,
`<=`	més petit o igual que,
`>`	més gran que,
`>=`	més gran o igual que,
`=`	igual,
`<>`	diferent.

Quan Maple es troba amb alguna d'aquestes expressions ho avalua com una "*expressió Booleana*", o sigui, com una expressió que pot ser "*certa*" o "*falsa*". Aquest tipus d'expressions també es poden avaluar, fora d'un `if`, mitjançant la funció `evalb( )`:

```
> evalb(7>5);evalb(3=4);evalb(3<>4);
```

Podem fer construccions més complicades, combinant amb les comandes lògiques `and`, `or`, `xor` i `not`. Podem consultar la ajuda per a més detalls: `?boolean`.

- El tercer element són les comandes que s'han d'executar a cada un dels casos. En l'exemple tant sols havíem d'introduir -1, 0 i 1, però es poden introduir expressions més complexes: per a definir una funció `g` que valgui `1/(x^2+1)` quan s'aplica sobre un `x<0` i `cos(x)` quan s'aplica sobre un `x>=0` farem

```
> g:=x-> if x<0 then 1/(1+x^2) else cos(x) end if;
```

Podem comprovar que funciona amb

```
> g(-1.5);g(2.5);
```

Tot i això, què passa quan fem:

```
> g(Pi);
```

El missatge d'error diu que no ha pogut avaluar la condició booleana que hem introduït. Això passa degut a que `Pi` no és exactament un número. Obtenim exactament el mateix error quan avaluem `g` a una indeterminada `a`:

```
> g(a);
```

Una manera de arreglar el problema seria utilitzant la comanda `evalf( )` i avaluar `g` en un valor aproximat de `Pi`.

```
> g(evalf(Pi));
```

Però si ara intentem dibuixar el gràfic de la funció `g` que hem definit:

```
> plot(g(x),x=-10..10);
```

Tornem a obtenir errors en les expressions booleanes. El problema és que quan substitueix `x` a la comanda `g` encara no és un número i per això retorna l'error. En aquest cas el que podem fer és introduir la "*notació d'operadors*" en la comanda `plot( )`, o sigui, no escriure la variable `x`:

```
> plot(g,-10..10);
```

Però si volem utilitzar la funció `g` per a definir altres funcions, per exemple, si volem calcular la seva derivada, també tindrem problemes:

```
> diff(g(x),x);
```

La funció `if` és molt útil en segons quins contexts, però no és la més apropiada per a definir funcions a trossos. Per a fer això és més versàtil la funció `piecewise( )`. Podem redefinir la funció `g` com:

```
> g:=x->piecewise(x<0, 1/(1+x^2),x>=0, cos(x));
```

Ara podem treballar amb `g` com amb qualsevol altre funció, per exemple:

```
> plot(g(x),x=-5..5);
```

i calcular la seva derivada:

```
> diff(g(x),x);
```

Per tant per a definir funcions a trossos utilitzarem la funció `piecewise`, mentre que reservarem `if - then - elif - end if` per a utilitzar-la dins de iteracions, ...

Exercici 5.6
Utilitzeu la funció `piecewise( )` per a definir amb Maple la funció que avalua la següent funció i feu la gràfica a l'interval $(-4, 4)$.

$$f(t) = \begin{cases} \dfrac{(2+t)^3}{6} & \text{si } t \in [-2,-1) \\ \dfrac{-3t^3-6t^2+4}{6} & \text{si } t \in [-1,0) \\ \dfrac{3t^3-6t^2+4}{6} & \text{si } t \in [0,1) \\ \dfrac{-(t-2)^3}{6} & \text{si } t \in [1,2) \\ 0 & \text{si } t \notin [-2,2) \end{cases}$$

Segons com feu la definició haureu d'introduir rangs amb límit inferior i límit superior. Per a això hem d'introduir dues desigualtats en la mateixa condició, i això ho hem de fer mitjançant la comanda `and`.

5.2.2 Iteracions: `for`, `do`, `end do` i `while`

Moltes comandes a Maple contenen iteracions: per exemple, les comandes `sum( )`, `fsolve( )`, ... Tot i això algun cop necessitarem definir les nostres pròpies iteracions. Com a exemple, calculem la suma dels enters entre 1 i 100 elevats al quadrat:

```
> sum(i^2,i=1..100);
```

En alguna part del procés Maple ha aplicat una iteració per a fer el càlcul. Fet "a mà" hauríem pogut fer:

Iteració suma:

Decidim primer el nom de la variable, i la inicialitzem a zero.

```
> resp:=0;
```

Llavors cal introduir la iteració. Per a això utilitzem les comandes `for..from..by..to..while..do..end do`. Les comandes que s'han d'executar s'han d'introduir entre el `do` i el `end do`.

```
> for i from 1 to 100 do
  resp:=resp + i^2;
  end do;
```

Per a evitar la sortida de tots els passos, podem canviar el punt i coma (`;`) per uns dos punts (`:`) i demanar el valor de la variable `resp` al final del procediment.

```
> resp:=0;
```

```
> for i from 1 to 100 do
  resp:=resp + i^2;
  end do:
> resp;
```

Exercici 5.7
Calculeu la suma de tots els nombre senars entre 1 i 99 mitjançant la comanda `for` (consulteu l'ajuda per veure la opció `by` dins de la comanda `for`).

Més iteracions:

A vegades no coneixerem el nombre d'iteracions que volem fer. La informació que tindrem serà que hem d'iterar fins que alguna condició fixada es doni. Per exemple, què passa si premeu indefinidament la tecla "*cosinus*" de la vostra calculadora, començant pel valor 5? A l'exemple ho farem 20 vegades:

```
> x:=5.;
> for i from 1 to 20 do
  x:=cos(x);
  end do;
```

Podeu observar que sembla que la successió es va acostant a un valor determinat. De fet, podeu comprovar que aquest valor ha de complir l'equació $\cos(x) = x$. Tot i això, podeu veure que amb 20 cops no n'hi ha prou per a obtenir una successió que no canvia de valor (8 dígits correctes), per tant, el que volem fer és iterar el procediment fins que la diferència entre dos passos sigui inferior a un número fixat, per exemple, 10^{-8}. Necessitem dues variables, una serà `x` i la segona serà `xold`, que en tot moment serà el valor de la iteració anterior (haurem de donar-li un valor inicial diferent del que donem a `x`).

```
> x:=5.;xold:=0;
```

Llavors hem d'iterar mentre es compleixi que `| x - xold |> 10^(-8)`. La comanda la podríem entrar com:

```
> for i from 1 while abs(x-xold)>1e-8 do
  tempx:=cos(x):
  xold:=x:
  x:=tempx:
  end do;
```

Observem que la "`i`" que hi ha dins del `for` no s'ha utilitzat al procediment. De fet en qualsevol iteració l'única part obligatòria és el `do..end do`. Les parts `for` i `while` són opcionals. Com a exemple podríem escriure:

```
> x:=0.; xold:=1;
  while abs(x-xold)>1e-8 do
  tempx:=cos(x):
  xold:=x:
  x:=tempx:
  end do;
```

5.2.3 Procediments

Quan volem definir una funció poder necessitarem incloure iteracions, condicions lògiques, altres funcions... i en general potser necessitarem dues o més instruccions de Maple encadenades per a fer els càlculs. També és bastant comú que ens aquests casos utilitzem variables que només tenen sentit dins el càlcul de la funció que estem definint i no a la resta del full de treball. Això és el que anomenarem **variables locals**, mentre que les que són vàlides a tot el full de treball les anomenarem **variables globals**.

Per a definir funcions que inclouen variables locals i globals hem d'utilitzar la comanda `proc( )`.

Exemple 5.2
Mireu l'estructura de la següent definició:

```
> f:=proc(x,y)
  local a,b,z;
  global d;
  a:=3.;b:=17.;
  z:=a*b*x*y*d;
  end proc;
```

Podem veure com s'executa aquesta definició provant:

```
> f(5,5);
> d:=2.;
> f(5,5);
> d:='d';
> f(5,5);
> a;
> b;
> z;
```

Exemple 5.3
La funció següent avalua el cosinus d'un angle en graus:

```
> cosdeg:=proc(theta)
  local resultat,phi;
  phi:=theta*Pi/180;
  resultat:=cos(phi);
  end proc;
```

Proveu-la calculant el cosinus de 45 graus.

Si voleu veure els passos que segueix un procediment, ho podeu fer amb la funció `trace( )`. Aquesta funció permet "*activar*" per pantalla els passos que hi ha dins de la funció `proc( )`.

Exemple 5.4
Per activar aquesta opció per a la funció `cosdeg( )` hem de fer:

```
> trace(cosdeg);
```

llavors, quan executem, tindrem:

```
> cosdeg(45);
```

Si volem desactivar aquesta la visualització podem fer:

```
> untrace(cosdeg);
```

5.2.4 Exercicis

Els següents exercicis són perquè practiqueu els diferents conceptes de "*programació*" que hem introduït, però també pretenen il·lustrar alguns conceptes matemàtics. No us oblideu d'interpretar els resultats que aneu obtenint.

Exercici 5.8
Considereu la funció:

$$\mathtt{aprcos}(\mathtt{x},\mathtt{n}) = \sum_{\mathtt{i}=0}^{\mathtt{n}} (-1)^{\mathtt{i}} \frac{\mathtt{x}^{2\mathtt{i}}}{(2\mathtt{i})!}$$

on `x` és un paràmetre real i `n` és un enter positiu.

Dibuixeu en on sol gràfic les funcions `cos(x)`, `aprcos(x,1)`, `aprcos(x,2)` i `aprcos(x,4)` per a `x` entre -2π i 2π.

Exercici 5.9

Definiu una funció que depengui de tres paràmetres `aprox(funcio,x,iteracions)` que retorni una llista amb els valors `[x+1/i,funcio(x+1/i)]` per a `i=1..iteracions`.

Avalueu `aprox` per a la funció $\frac{\sin(x)}{x}$, amb `x=0` i `iteracions=50`.

Modifiqueu la funció `aprox` per a que a més retorni un dibuix dels punts calculats.

6 Llistes, conjunts, seqüències, taules i "matrius"

El programa Maple permet treballar amb *conjunts* de dades. Aquests *conjunts* es poden entrar de diverses maneres i podem canviar l'estructura segons ens interessi.

6.1 Llistes

Una llista és un conjunt ordenat d'expressions on es permeten repeticions. L'entrada al Maple es fa entre claudàtors (`[ ]`), separant els elements amb comes.

Exemple 6.1
Definim la variable a com la llista $[1, 2, 3, 2, 1, 2]$:

```
> a:=[1,2,3,2,1,2];
```

Els elements d'una llista es poden cridar un a un mitjançant la posició entre claudàtors. Si utilitzem nombres negatius, retorna l'element començant per la cua.

Exemple 6.2
Si volem el tercer element de la llista a és:

```
> a[3];
```

Mentre que si volem l'últim és:

```
> a[-1];
```

De la mateixa manera també podem extreure una subllista. Si volem una subllista amb els elements entre les posicions i i j, cal introduir l'argument `i..j`:

```
> a[2..4];
```

També podem construir una llista com la unió de vàries. Per a això necessitem abans "*treure*" els claudàtors i afegir-los després. La funció que ens permet treure els claudàtors és la funció `op( )`. Un cop hem tret els claudàtors podem concatenar dues llistes:

Exercici 6.1
Definiu la funció `concatenar`, depenent de dues variables, que retorni la unió de dues llistes donades.

Comproveu que la definició que heu fet és correcta fent la prova sobre les llistes `a:=[1,2,3,2,1]` i `b:=[3,2,4]`. El resultat ha de ser la llista `c:=[1,2,3,2,1,3,2,4]`.

6.2 Conjunts

En un conjunt no hi ha repeticions i les dades no tenen un ordre prefixat (Maple pot variar l'ordre, depenent del context). L'entrada al Maple es fa entre claus (`{ }`), separant els elements per comes. La majoria d'opcions que hem vist per a les llistes són també vàlides per als conjunts, tenint en compte les possibles diferències.

Exemple 6.3
Introduïu en la variable `b` el conjunt $\{1, 2, 5, 1, 3\}$:

```
> b:={1,2,5,1,3};
```

Observeu la sortida que us torna.

Els elements d'un conjunt també es poden cridar mitjançant els claudàtors, tenint en compte que, a priori, no coneixem l'ordre en que Maple guarda els elements.

Exemple 6.4
Per a cridar el tercer element del conjunt `b` fem:

```
> b[3];
```

Observeu que el tercer element no coincideix amb el que hem introduït quan hem definit `b`.

Exercici 6.2
Definiu `a` i `b` com els conjunts $\{x, y, z\}$ i $\{t, u\}$. Definiu el conjunt `c` unió de `a` i `b`.

Exercici 6.3
Definiu una funció que, a partir de dos conjunts, doni com a resultat la seva unió. Proveu-la sobre els conjunts $A = \{1, 4, 9, 16\}$ i $B = \{2, 4, 6, 8, 10\}$.

6.3 Seqüències

Una seqüència d'elements és un conjunt ordenat però que s'interpreta com n arguments (i no una sola llista), on n és el nombre d'elements de la seqüència. L'entrada al Maple és sense cap delimitador.

Exemple 6.5
Si volem definir com `a` la seqüència $1, 4, 9, 16$ posem

```
> a:=1,4,9,16;
```

Una altra manera d'aconseguir seqüències és mitjançant la comanda `seq( )` que vam introduir a la primera pràctica.

Exemple 6.6
La llista `c` amb els primers 10 quadrats enters s'obté amb

```
> c:=seq(i^2,i=1..10);
```

Igual que en el cas de les llistes podem treure un element d'una seqüència, unir seqüències,...

Exercici 6.4
Definiu la funció de dues variables $f(x, y) = x^2 - y^2$. Considereu $a = (5, 7)$. Com s'ha d'introduir a per aconseguir que $f(a)$ doni com a resultat el valor -24?

6.4 Taules (`table`)

Una taula és una llista d'objectes "indexada" en la que els *índexs* que determinen les *posicions* poden ser qualsevol tipus d'expressió (és a dir, els índexs no tenen perquè ser nombres enters consecutius). Per exemple:

```
> T:=table([a,b]);
```

Genera un objecte `T` de tipus taula amb valors per a `T[1]` i `T[2]`

```
> T[1];T[2];
```

Però també podem generar un objecte del mateix tipus on els índexs per als que tenim valors associats siguin indeterminades. Per exemple:

```
> S:= table();
> S[k]:= sin(k*Pi);
> op(S);
> S[1];
> S[k];
```

6.5 Matrius tipus `array`

Per a Maple els `array` són un tipus especial de `table` i són el tipus d'objecte bàsic en el que poden guardar dades de tipus matricial (és a dir que depenen d'un multiíndex).

```
> M:= array(1..2,1..2,[[a,b],[c,d]]);
> M[1,2];
> op(M);
> M;
> A:= array(1..3,1..2);
> op(A);
> A[1,1]:= 3:  A[1,2]:= 4:
> A[2,1]:= 5:  A[2,2]:= 2:
> A[3,1]:= 1:  A[3,2]:= sin(Pi/7):
> op(A);
> B:= array(1..2,1..3,1..2);
> op(B);
```

també podem veure el contingut de `B` amb

```
> print(B);
```

6.6 Comptar elements de llistes i conjunts

La funció que permet comptar el nombre d'elements d'una llista o conjunt és la funció `nops( )`.

Exemple 6.7
Volem saber quants elements diferents té la llista $a = [3, 4, 2, 1, 3, 5, 3, 4, 3, 4, 5, 3, 2, 2, 4]$. La funció `nops( )` aplicada directament a la llista, com en

```
> a:=[3,4,2,1,3,5,3,4,3,4,5,3,2,2,4];
> nops(a);
```

retorna el nombre d'elements de la llista (amb repeticions incloses). Una manera de saber quants elements diferents té és passar primer la llista a conjunt `b` i comptar el nombre d'elements del conjunt:

```
> b:=convert(a,'set'); nops(b);
```

Exercici 6.5
Quants elements té la llista `a:=[[1,2,3],[x,y],{2, 3,4}, x, 1, 2]`?

6.7 Canvis de format entre llistes, seqüències i conjunts

Maple permet transformar una expressió que té guardada com una llista, seqüència o conjunt als altres formats.

A l'apartat de llistes ja hem vist com podem canviar una llista d'expressions en una seqüència, mitjançant la comanda `op( )`.

Exemple 6.8
Podem convertir la llista `a:=[1,2,3,2,1]` en una seqüència `b` fent

```
> a:=[1,2,3,2,1];
> b:=op(a);
```

Un cop tenim una seqüència, podem transformar-la un altre cop en una llista afegint els delimitadors.

Exemple 6.9
Si volem recuperar la seqüència `b` en format llista podem afegir els claudàtors:

```
> [b];
```

Podem fer el mateix procediment amb els conjunts i les seqüències. Hem de tenir en compte que quan passem una seqüència o una llista a format conjunt perdem les repeticions i l'ordre que hi teníem.

Una altra manera de fer conversions entre llistes, seqüències i conjunts és amb la comanda `convert( )`. Aquesta comanda s'utilitza per a canvis en altres contexts dins de Maple (taules a matrius,...).

Per a passar d'un format a l'altre tan sols cal conèixer la terminologia de Maple corresponent a cada un d'ells: `set` per a conjunts i `list` per a llistes.

Exemple 6.10
Considerem la llista formada pels elements $a = [3, 2, 1, 4, 3]$. Podem passar-la a format conjunt mitjançant:

```
> a:=[3,2,1,4,3];
> b:=convert(a,'set');
```

I podem tornar al format llista mitjançant:

```
> convert(b,'list');
```

7 Problemes per a practicar

Aquesta pràctica conté únicament una llista de petits exercicis per a consolidar el que s'ha treballat fins a aquest punt.

Exercici 7.1
Doneu el nom `y` al nombre $\dfrac{2\,\pi}{3}$ i després determineu el valor exacte i una aproximació decimal per a: y^2, $\sqrt{y}$ i $\cos(y)$.

Exercici 7.2
Introduïu la funció $f(x) = 20\,x + 30\,x^2 - \sqrt{46 - x^2}$ i determineu el valor aproximat de $f(3.29) + f(-3.1)$ amb 20 decimals.

Exercici 7.3
Determineu el valor aproximat de T en la fórmula $T = \sqrt{\dfrac{2\,a - 3\,b^2}{c - 20}}$ quan $a = 4.6$, $b = -3.8$ i $c = 2.9$.

Exercici 7.4
Si $f(x) = x^2 - 2\,x + 3$, calculeu i simplifiqueu $f(3\,t + 2)$.

Exercici 7.5
Quan multipliquem l'expressió $(x - 4)^2\,(x + 1)^3$ s'obté un polinomi de grau cinc. Quin és el coeficient de x^2 en aquest polinomi? Busqueu amb quina instrucció podem obtenir aquest coeficient. Fabriqueu una llista amb tots els coeficients d'aquest polinomi ordenats des del terme independent fins al de grau 5.

Exercici 7.6
Feu el gràfic de les expressions $\cos(x)$ i $\cos(x)\sin(10\,x)$ sobre l'interval $[0,\, 2\,\pi]$ i amb eix d'ordenades entre -2 i 2. Després feu el gràfic de les mateixes expressions a l'interval $[0,\, 4\,\pi]$. Determineu els punts de tall d'aquestes gràfiques de forma exacta (el més simple possible) i aproximada.

Exercici 7.7
Feu el gràfic de la funció $f(x) = \sec(x) + 4$ a l'interval $[0,\, 2\,\pi]$. L'escala automàtica no produeix un dibuix gaire útil. Especifiqueu un rang per a les y que doni una bona visió d'aquesta funció en aquest interval.

Exercici 7.8
Recordeu que quan els nombres racional tenen una expressió decimal infinita, a partir d'un lloc d'aquesta expressió les xifres s'han de repetir. Un exemple prou familiar és la fracció $\dfrac{1}{3} = .33333333\ldots$ on el dígit 3 es repeteix. Una mica més interessant és l'expressió decimal de $\dfrac{33}{14} = 2.3571428571428\ldots$ on la part que es repeteix és 571428. Ara fixeu vos amb l'expressió decimal de la fracció $\dfrac{2}{19}$. Podeu identificar el període? Investigueu totes les longituds possibles de períodes de fraccions $\dfrac{a}{19}$ amb a enter.

Exercici 7.9
a) Dibuixeu en un gràfic els punts següents: $(1, 0.53)$, $(1.5, 0.65)$, $(2, 0.91)$, $(2.5, 0.95)$ i $(3, 1.10)$

b) Feu un dibuix que contingui els punts anteriors i, a més, les gràfiques de les funcions $f(x) = \sin\left(\dfrac{x}{2}\right)$ i $g(x) = \dfrac{x^2}{5}$.
Utilitzeu el vostre dibuix per a decidir quina de les dues funcions s'ajusta millor a aquest conjunt de punts.

Exercici 7.10
Aproximeu les solucions reals de l'equació $x^4 - 4\,x^3 + 3\,x - 12 = 0$.

Exercici 7.11
Aproximeu totes les solucions reals de l'equació $x^4 - 4\,x^3 = \cos(3\,x) + 3$.

Exercici 7.12
Els gràfics de $f(x) = 20 - x$ i $h(x) = 1.012^x$ s'intersequen en un punt. Utilitzeu les habilitats de solució numèrica de Maple per aproximar les coordenades d'aquest punt d'intersecció. Comenceu introduint una equació apropiada per a resoldre. Comproveu la vostra resposta fent un dibuix que mostri la intersecció dels gràfics.

Exercici 7.13
Resoleu respecte r l'equació: $r\,(p\,k - 18\,m) = \dfrac{32\,(2 - p\,r\,m)}{m^2}$. Assegureu-vos de que heu introduït correctament l'equació. En particular, comproveu que heu utilitzat un `*` per a totes les multiplicacions.

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 depenent del paràmetre a donat per les equacions

$$\left\{ \begin{array}{c} a\,x + (a-1)\,y = 0 \\ x - a\,y = 0 \end{array} \right.$$

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 $\dfrac{-1+\sqrt{5}}{2}$ 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 dóna `solve( )` no és del tot satisfactori.

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

Per tant és molt important conéixer 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 instalació 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, comenceu netejant la memòria de Maple i *carregant* el paquet amb el que treballareu:

```
> 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 `Mf` 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 8.3

```
> 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 esperaríem en un principi (de fet dóna un missatge d'error). Per a obtenir el vertader producte de matrius cal utilitzar l'operador `.` (punt).

Exemple 8.4

```
> 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 8.5

```
> 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) . (P_1), (M_1) . (v_i), (P_1) . (v_i) (per a $i = 1$ i 2). Quina relació creieu que hi ha entre els productes (M_1) . (v_i), (P_1) . (v_i) i els vectors v_i? Qué passa si intenteu realitzar el producte (v_1) . (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 $\left.\begin{aligned} 3\,x - 2\,y = 2 \\ 2\,x - y = 7 \end{aligned}\right\}$ 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 (i la comanda inversa `GenerateEquations( )`).

Exemple 8.6
Per a obtenir la matriu de coeficients i el terme independent del sistema d'equacions

$$\left.\begin{aligned} x - y + z - 2\,t = 1 \\ y - 2\,z = 6 \\ 2\,x + 4\,y + t = 8 \\ 44\,x - y + 34\,z - 21\,t = 98 \end{aligned}\right\}$$

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 8.7
Si tenim el sistema d'equacions, depenent del paràmetre λ, donat per $\left.\begin{array}{r} x-2\,y=\lambda \\ 2\,x-3\,y=2-\lambda \end{array}\right\}$, 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 8.8
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` prenent valors entre 1 i 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 8.9
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 8.10
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 8.11
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 8.12
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
$$\left.\begin{aligned} 2\,x_1 - 4\,x_2 + 3\,x_3 - 5\,x_4 + x_5 &= 3 \\ x_1 - 4\,x_2 - 2\,x_3 - 7\,x_5 &= 2 \\ 5\,x_1 - 3\,x_3 - 2\,x_4 + 2\,x_5 &= 1 \\ x_2 - 4\,x_3 + 3\,x_4 - x_5 &= \frac{3}{2} \end{aligned}\right\}$$
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 aporti, 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'aportació energètica sigui qualsevol?

Exercici 8.4
Estudieu, per als diferents valors dels paràmetres b i t, el sistema d'equacions lineals respecte $[x,\ y,\ z]$.

$$\left.\begin{aligned} 3\,x+2\,y+z&=t\\ x-y+2\,z&=1+t^2\\ 3\,x+7\,y-4\,z&=-1-t-t^2-t^3\\ 2\,x+y+b\,z&=t^3 \end{aligned}\right\}$$

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 8.13
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 podrem 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 8.14
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.

```
> 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 8.15
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>);
```

i dóna el conjunt de totes les solucions possibles en funció dels paràmetres necessaris.

9 Les eines bàsiques per al càlcul infinitesimal

9.1 Límits

La comanda de Maple que permet calcular el límit d'una expressió qualsevol és `limit( )`. Per utilitzar aquesta comanda s'ha d'introduir `limit(expr,x=lim)`, on `expr` serà l'expressió de la que volem calcular el límit, `x` la variable respecte de la que es calcula el límit i `lim` el valor de la variable on volem calcular el límit. A continuació podeu veure alguns exemples concrets.

Exemple 9.1

```
> limit((x^2-4)/(x-2),x=2);
> limit(sin(x)/x,x=0);
> limit(sin(x)/x,x=infinity);
> limit((1+1/x)^x,x=infinity);
> limit(ln(1+x)/x,x=0);
```

Normalment la variable respecte la que es calcula el límit serà una variable real i, per tant, quan volem calcular el límit d'una successió (que depèn d'una variable `n` que és un nombre natural) us podeu trobar amb alguna petita sorpresa.

Exemple 9.2
Si `n` és un nombre natural i voleu calcular $\lim_{n\to\infty} \sin(n\pi)$ (que dóna òbviament 0) i feu

```
> limit(sin(n*Pi),n=infinity);
```

veureu que el resultat és un xic estrany.

Aquest petit inconvenient es pot evitar fent que Maple *tingui en compte* que la variable `n` només pren valors naturals. La comanda `assume( )` permet declarar que una una variable és d'un cert tipus. També és possible fer que, mentre es realitza un càlcul concret, Maple assumeixi alguna condició sobre les variables però que aquestes variables segueixin tenint un tipus general fora d'aquest càlcul. Això es fa amb la comanda `assuming`. Fixeu-vos en la sintaxi que s'utilitza en els exemples següents.

Exemple 9.3
Podeu fer

```
> assume(n, posint);
> limit(sin(n*Pi),n=infinity);
```

però mentre no doneu algun valor concret a `n` o *netejeu* el contingut d'aquesta variable amb `n:='n'` (o amb un `restart`) Maple pensarà que `n` només por ser un nombre enter positiu.

Pot ser convenient fer el mateix càlcul amb

```
> restart;
> limit(sin(n*Pi),n=infinity) assuming n::posint;
```

D'aquesta forma obteniu el mateix resultat però `n` segueix lliure de restriccions.

Ara és un bon moment per a donar un cop d'ull a l'ajuda de Maple per a veure les possibilitats de les comandes `assume( )` i `assuming` i també per a mirar el tipus de restriccions que es poden donar a una variable.

Per acabar la secció dedicada al límit, un petit exercici:

Exercici 9.1
Utilitzeu `limit( )` per a conéixer els límits de les successions:

a) $x_n = \dfrac{1 + 2^\alpha + 3^\alpha + \cdots + n^\alpha}{n^{(\alpha+1)}}$ per a diferents valors (positius) del paràmetre α.

b) $x_n = \dfrac{1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}}{\ln(n)}$.

c) $x_n = \dfrac{1}{n+1} + \dfrac{1}{n+2} + \cdots + \dfrac{1}{2\,n}$.

d) $x_n = \left(\dfrac{\ln(n+1)}{\ln(n)}\right)^{(n\,\ln(n))}$.

e) $x_n = (\sqrt{n+1} - \sqrt{n} + 1)^{\sqrt{n}}$.

Calculeu el valor dels x_n per a algun n bastant gran (amb un `evalf( )` amb molts decimals) comparant els valors obtinguts amb els resultats que heu obtingut abans.

9.2 Derivades

Amb Maple es pot obtenir fàcilment la derivada d'una expressió `expr` qualsevol. Com ja deveu saber la comanda que fa aquesta feina és `diff(expr,var)` on `expr` és l'expressió que es vol derivar i `var` és la variable respecte la que es farà la derivada.

Exemple 9.4
Podem fer la derivada de l'expressió $x^3 - 3x^2 + 5x + 4$ respecte x amb

```
> diff(x^3-3*x^2+5*x+4,x);
```

i si tenim definida una funció també podrem fer el mateix

```
> f:= x-> x^3-3*x^2+5*x+4;
> diff(f(x),x);
```

Teniu en compte, però, que segons com us poseu a fer els càlculs podeu tenir sorpreses. Podeu preveure quin serà el resultat de l'operació següent després d'haver definit la funció `f`?

```
> diff(f,x);
```

Noteu que Maple necessita conèixer respecte quina variable s'ha de fer la derivada ja que en una expressió hi poden haver diferents paràmetres (o, si ho voleu dir d'una altra forma, l'expressió pot ser una funció de múltiples variables i es poden calcular les derivades parcials respecte cada una d'elles). Per exemple, les expressions

Exemple 9.5

```
> diff(x^2*cos(b)-exp(y^2)*tan(x),x);
> diff(x^2*cos(b)-exp(y^2)*tan(x),y);
> diff(x^2*cos(b)-exp(y^2)*tan(x),b);
```

produeixen, òbviament, tres resultats diferents.

Quan treballem amb la comanda `diff( )` sovint el que ens interessa és definir una nova funció que és la derivada d'una funció que ja tenim definida. Com que `diff( )` treballa amb expressions i dóna com a resultat expressions, aquest procés no és tan immediat com es podria pensar. Si proveu de fer

```
> f:= x-> (x^3-4)/(x^2+3);
> g:= x-> diff(f(x),x);
> g(x);
> g(1);
```

veureu que els resultats semblen no tenir massa lògica.

Quan tenim una funció `f` que depèn d'una variable podem obtenir una nova funció `g` que és la derivada de la primera amb la comanda `D( )`.

Exemple 9.6
Si definim la funció $f(x) = x^3 - 4x$ i volem tenir la funció $g(x) = f'(x)$ (que òbviament ha de ser $g(x) = 3x^2 - 4$) es pot fer

```
> f:= x-> x^3-4*x;
> g:= D(f);
> g(x);
> g(1);
```

Quan es té definida una funció `f` que depèn de més d'una variable es pot especificar respecte quina de les variables es vol fer la derivada utilitzant `D[i](f)` on `i` és un enter que indica que es vol fer la derivada de `f` respecte la variable que ocupa el lloc `i`.

Exemple 9.7
Si tenim definida la funció $f(x, y, b) = x^2 \cos(b) - \exp(y^2)\tan(x)$ la primera variable és la x, la segona és la y i la tercera és la b. Per tant, fent

```
> f:= (x,y,b)-> x^2*cos(b)-exp(y^2)*tan(x);
> g1:= D[1](f);
> g2:= D[2](f);
> g3:= D[3](f);
```

obtindrem tres funcions `g1, g2, g3` que són les derivades de `f` respecte `x, y, b` respectivament.

Exercici 9.2
Definiu una funció `h` que sigui $h(x, y) = x\dfrac{\exp(xy)\sin(y^2)}{\ln(x^2 + y^2 + 2)}$ i calculeu

a) La funció `h1` derivada de `h` respecte `x`.

b) La funció `h2` derivada de `h` respecte `y`.

c) Els valors de `h1` i `h2` per a `x=1` i `y=Pi/4`.

d) La funció `h11` derivada segona de la funció `h` respecte `x`.

És clar que la resposta a l'últim apartat de l'exercici anterior pot ser

```
> h11:= D[1](D[1](h));
```

Com que això no és massa pràctic hi ha dreceres que ens permeten fer el mateix amb més facilitat. Una expressió de la forma `x$n` és equivalent a una successió de `n` còpies de l'expressió `x`. Per exemple

```
> x$4;
```

produeix `x,x,x,x`. D'aquesta forma en les comandes `diff` o `D` podem fer

```
> diff(sin(x)-x*(cos(x))^3,x$4);
> D[1$4](cos)(x);
```

9.3 Integrals

Maple disposa d'algoritmes per a calcular integrals definides i indefinides (primitives). Gairebé qualsevol problema d'integració (que tingui una solució raonable) es pot resoldre simplement amb la comanda `int( )`. Veieu a continuació alguns exemples de les diferents formes en les que es pot aplicar aquesta funció.

Exemple 9.8

```
> int(x^2/sqrt(1-x^3),x);
> int(1/(x*sqrt(x^2-1)),x);
> int(cos(x),x=0..Pi);
> int(1/(x*sqrt(x^2-1)),x=1..2/sqrt(3));
> int(r*exp(-r^2),r=0..infinity)
```

Noteu que fins i tot es poden determinar integrals que són impròpies.

Fins i tot quan no és possible determinar un valor exacte per a una integral es pot mirar de determinar ne una aproximació amb la comanda `evalf( )` ja que Maple també disposa dels algoritmes d'aproximació numèrica necessaris.

Exemple 9.9

```
> a:=int(sqrt(1+x^6),x=1..3);
> evalf(a);
```

9.4 Polinomis i sèries de Taylor

La comanda per a obtenir l'aproximació de Taylor de grau `n` d'una funció `f` al voltant del punt `c` és `taylor(f(x),x=c,n)` (noteu que la comanda `taylor` s'aplica a una expressió `f(x)` i no a la funció `f`).

Exemple 9.10

```
> taylor(cos(x),x=0,8);
> taylor(ln(x),x=1,10);
```

El resultat de la comanda `taylor( )` està format per una component polinòmica i una component de la forma `O((x-c)^n)`. La part polinòmica és el polinomi aproximador pròpiament dit, més endavant veurem com es pot interpretar la part `O((x-c)^n)`. Si es vol extreure el polinomi del resultat d'una comanda `taylor( )`, s'utilitza `convert(expr, polynom)`.

Exemple 9.11
El polinomi de Taylor de grau 5 al voltant del 0 per a la funció $f(x) = \dfrac{1}{1-x}$ s'obté amb

```
> expr:=taylor(1/(1-x),x=0,5);
> poli:=convert(expr, polynom);
```

Exercici 9.3
Realitzeu un procediment que, donats $f(x)$, c i n, doni com a resultat només el polinomi de Taylor de grau n de la funció $f(x)$ al voltant de c. Modifiqueu el procediment anterior per tal que retorni en una llista els polinomis de grau 1 fins a n.

Exercici 9.4
Considereu la funció $f(x) = e^x$ i el punt $c = 0$. Feu un gràfic en el que es vegin al mateix temps el gràfic de $f(x)$ i el del seu polinomi de Taylor per a $n = 2$. Què observeu?

Repetiu l'exercici per a $f(x) = \ln(x^2 + 1)$ i per a $f(x) = \arctan(x)$.

Noteu doncs que es pot dir que el polinomi de Taylor de grau 1 ($n = 2$) és la recta tangent al gràfic de $f(x)$ en $x = c$ (això s'expressa sovint dient que *l'ordre de contacte entre els dos gràfics és* 2).

En l'exercici següent podreu veure com el concepte d'ordre de contacte s'estén a graus superiors.

Exercici 9.5
Preneu $f(x) = \ln(x^2 + 1)$ i $p(x)$ el polinomi de Taylor amb $n = 8$ de $f(x)$ al voltant de 0. Calculeu els valors de $\dfrac{f(x) - p(x)}{x^7}$ per a $x = \pm\dfrac{1}{2^n}$ (amb n des de 1 fins a 20 i amb 25 decimals com a mínim). Quin sembla que serà el límit $\lim\limits_{x\to 0} \dfrac{f(x) - p(x)}{x^7}$?

Com es poden obtenir els coeficients del polinomi de Taylor d'una funció a partir d'aquesta funció? Per a veure quina és aquesta relació recordeu que la comanda que dóna el coeficient d'una expressió polinòmica `expr` respecte una altra expressió `var` és `coeff(expr,var)` i que si es vol buscar el coeficient respecte `var^k` també es pot fer `coeff(expr,var,k)`. Per exemple:

Exemple 9.12

```
> taylor(exp(-x^2),x=0,9);
> coeff(taylor(exp(-x^2),x=0,9),x,4);
> taylor(sin(x),x=Pi/2,8);
> coeff(%,(x-Pi/2),4);
```

Exercici 9.6
Donada $f(x) = \arctan(x)$ determineu les expressions de les derivades successives $f^{(n)}(x)$ i avalue-les en $x = 0$ per a n des de 1 fins a 10. Feu el mateix amb $p(x)$, el polinomi de Taylor de grau 10 de $f(x)$ al voltant de 0.

Exercici 9.7
Compareu els valors que heu obtingut en l'exercici anterior amb els coeficients del polinomi de Taylor $p(x)$. (Feu el quocient entre uns i altres i si no veieu res mireu què passa si $f(x) = e^x$).

Al principi de la pràctica diu que el polinomi de Taylor permet aproximar una funció per un polinomi. L'exercici següent mostra com funciona aquest tipus d'aproximació.

Exercici 9.8
Feu un procediment que faci un dibuix dels gràfics (en diferents colors) d'una funció donada i dels seus polinomis de Taylor, fins a un grau n especificat en els arguments, al voltant d'un punt també donat entre els arguments.

Apliqueu l'anterior a la funció $f(x) = \dfrac{1}{(2x+1)^4}$ al voltant del punt $x = 2$ fins al grau 7.

Exercici 9.9
Les coses no van sempre tan bé com en tots els exemples anteriors. Proveu de determinar l'expressió de qualsevol polinomi de Taylor de $f(x) = e^{(-1/x^2)}$ al voltant de 0.

Tot i que `taylor` no aconsegueix fer gaire cosa amb aquest problema, hauríeu de poder determinar quin és el polinomi de Taylor de qualsevol grau per a la funció anterior al voltant de 0. I encara que es tingui el polinomi de Taylor, noteu que el resultat que s'obté no és gaire interessant!

10 Gràfics "avançats"

En una pràctica anterior, ja heu vist que la funció bàsica per a fer un gràfic d'una expressió és `plot`, que aquesta comanda també permet dibuixar punts del pla, i que en el paquet **plots** hi ha la funció `display` que permet agrupar diferents gràfics en un de sol. En aquesta pràctica veurem algunes aplicacions més potents de la comanda `plot` i del paquet **plots** així com les comandes per a fer gràfics a l'espai.

10.1 Gràfics al pla d'expressions paramètriques

Tot i que ja donen força joc, els gràfics del pla de la forma $y = f(x)$ no deixen de ser els més simples de tots. En general, una corba del pla es descriu per una expressió de la forma $(x(t), y(t))$ on t és un paràmetre real. Per exemple, les el·lipses de centre l'origen i eixos els eixos de coordenades són les corbes de la forma $(a\cos(t), b\sin(t))$ amb $t \in [0, 2\pi]$ i a, b valors positius. Per a fer un gràfic d'una d'aquestes el·lipses es pot utilitzar la comanda `plot` de la forma següent:

```
> plot([2*cos(t),3*sin(t),t=0..2*Pi],scaling=constrained);
```

per a dibuixar una amb semieixos 2 i 3, o posar

```
> plot([4*cos(t),4*sin(t),t=0..2*Pi],scaling=constrained);
```

per a dibuixar la circumferència de radi 4.

Exercici 10.1
Quina corba descriu la parametrització $\{x(t) = t\cos(2\pi t), \quad y(t) = t\sin(2\pi t)\}$

10.1.1 Coordenades polars

Per a determinar corbes del pla s'utilitzen sovint les coordenades polars (r = distància d'un punt a l'origen, θ = angle entre la línia horitzontal i la recta que uneix l'origen i el punt). Concretament, sovint tenim una corba descrita com $r = f(\theta)$. Dins el paquet **plots** disposeu de la comanda `polarplot( )` que fa el gràfic corresponent sense cap altre manipulació. Per exemple, l'espiral $r = 2\theta$ es pot fer amb:

```
> with(plots):
> polarplot(2*theta,theta=0..3*Pi);
```

Teniu en compte que, com que les coordenades rectangulars s'obtenen a partir de les coordenades polars d'una forma prou simple ($x = r\cos(\theta)$, $y = r\sin(\theta)$), també es podria haver fet

```
> plot([2*theta*cos(theta),2*theta*sin(theta),theta=0..3*Pi]);
```

per a obtenir el mateix gràfic.

També resulta fàcil combinar diferents gràfics en un de sol per la comanda `polarplot( )`. Per exemple

```
> with(plots):
> polarplot([2*theta,10*cos(5*theta)],theta=0..3*Pi);
```

dibuixarà l'espiral anterior i una flor de cinc pètals a sobre.

10.2 Gràfics al pla definits implícitament

Quan es vol fer el gràfic d'una corba del pla determinada per una equació de la forma $f(x, y) = 0$ i no es pot aïllar una de les variables en funció de l'altre, ni tampoc es disposa d'una parametrització, també es pot obtenir un gràfic amb la comanda `implicitplot( )` del paquet **plots**. Per exemple la hipèrbola $x^2 - y^2 = 1$ es pot veure amb

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

No cal que espereu miracles respecte aquesta funció, hi pot haver moltes corbes que siguin difícils de representar a partir de la seva equació. Per exemple

```
> implicitplot(y^2-x^3-x^2,x=-1..1,y=-1..1);
```

no dóna un gràfic que passi per $(0,0)$ sense una mica més d'ajuda.

10.3 Opcions de la comanda `plot`

En alguns dels exemples d'utilització de la comanda `plot` ja heu vist que després de l'expressió que es vol representar i els límits de la variable i els valors es poden afegir *opcions*. Les opcions que poden sortir en un gràfic del pla són:

`adaptive`	`filled`	`legend`	`scaling`	`tickmarks`
`axes`	`font`	`linestyle`	`style`	`title`
`axesfont`	`labels`	`numpoints`	`symbol`	`titlefont`
`coords`	`labeldirections`	`resolution`	`symbolsize`	`view`
`discont`	`labelfont`	`sample`	`thickness`	`xtickmarks`

Taula 2: Opcions de la comanda `plot( )`

Busqueu en l'ajuda de Maple (`?plot[options]`) quina és la funció de cada una d'elles. En particular, podreu trobar com aconseguir que `implicitplot( )` dibuixi una mica millor alguns dels gràfics que li donen problemes.

10.4 Gràfics a l'espai

La comanda bàsica pera fer el gràfic d'una funció de dues variables $f(x,y)$ és `plot3d( )`. S'utilitza de la mateixa manera que `plot( )` i l'únic que s'ha de tenir en compte és que no es pot deixar d'explicitar els marges de variació de les dues variables (x,y). A continuació podeu provar els següents exemples:

Exemple 10.1

```
> plot3d(exp(-(x^2+y^2-1)),x=-2..2,y=-2..2);
> plot3d(sin(x*y),x=-Pi..Pi,y=-Pi..Pi);
```

Observareu que, igual que en el cas dels gràfics en el pla, quan seleccioneu un gràfic dels que acabeu de fabricar, apareixen nous botons en el menú del programa que permeten canviar les opcions de visualització de la figura. És particularment interessant la possibilitat de *girar* el gràfic per a observar-lo des de diferents punts de vista (seleccionar i moure el cursor). També podreu observar que, amb el gràfic seleccionat, el *botó dret* fa que surti un menú des del que també es poden realitzar aquestes modificacions. La majoria d'aquests efectes també es poden obtenir afegint *opcions* a la comanda `plot3d( )` que executeu.

10.4.1 Gràfics paramètrics

Quan l'objecte de l'espai que voleu dibuixar no és el gràfic d'una funció sinó que el que teniu és una parametrització respecte dues variables $(x(u,v), y(u,v), z(u,v))$ també es pot utilitzar `plot3d( )` de la forma següent:

Exemple 10.2

```
> plot3d([sin(u)*cos(v), sin(u)*sin(v), cos(u)]
> ,u=-Pi..Pi,v=-Pi/2..Pi/2);
> plot3d([u*cos(v),u*sin(v),v],u=-4..4,v=-2*Pi..2*Pi);
```

Exercici 10.2
Feu un gràfic en el que surti l'esfera de radi 1 centrada a l'origen i al mateix temps el cilindre obtingut considerant els punts (x, y, z) amb (x, y) a la circumferència de centre $(0, \frac{1}{2})$ i radi $\frac{1}{2}$ i la coordenada z arbitrària. Com és la intersecció d'aquestes dues superfícies?

Si el que es vol aconseguir és una corba de l'espai $(x(t), y(t), z(t))$, s'ha de carregar el paquet **plots** i utilitzar la comanda `spacecurve( )`.

Exemple 10.3

```
> spacecurve([3*cos(v),3*sin(v),v],v=-2*Pi..2*Pi);
```
fa el gràfic d'una hèlix.

10.4.2 Corbes de nivell

Un altre tipus de gràfics que dóna informació interessant sobre les funcions de dues variables és el gràfic de les seves corbes de nivell. Dins el paquet **plots** es pot fer servir la comanda `contourplot( )` com en l'exemple següent:

Exemple 10.4

```
> contourplot(sin(x)+sin(y),x=-2*Pi..2*Pi,y=-2*Pi..2*Pi,
> filled=true,coloring=[red,green],scaling=constrained);
```
on s'observen les corbes de nivell de l'expressió $\sin(x) + \sin(y)$ colorejades en diferents tons de vermell i verd segons els diferents valors que va prenent l'expressió en cada una de les regions.

Relacionada amb les corbes de nivell, també es disposa de la comanda `contourplot3d( )` en el paquet **plots** que marca sobre el gràfic tridimensional d'una funció les diferents corbes de nivell.

Exemple 10.5

```
> contourplot3d(sin(x)+sin(y),x=-2*Pi..2*Pi,y=-2*Pi..2*Pi,
> filled=true,coloring=[red,green],scaling=constrained);
```

10.4.3 Superfícies definides implícitament

A vegades les superfícies que es volen visualitzar no tenen un parametrització senzilla sinó que venen donades de forma implícita. Com per les corbes del pla determinades implícitament, podem dibuixar una aproximació del gràfic de la superfície determinada per $f(x, y, z) = 0$ amb la comanda `implicitplot3d( )` del paquet **plots**.

Exemple 10.6

```
> implicitplot3d(x^2+y^2-z^2=1,x=-2..2,y=-2..2,z=-2..2);
> implicitplot3d(x^2-y^2-z=1,x=-2..2,y=-2..2,z=-2..2);
```

Nota: Com en el cas de la comanda `implicitplot( )`, no espereu miracles. Si voleu dibuixar una regió d'una superfície en la que hi ha singularitats o *porqueries* complicades el resultat pot ser que no sigui del tot satisfactori.

Exercici 10.3
Feu un gràfic del con $x^2 + y^2 - z^2 = 0$ en el que es vegi la punxa en $(0, 0, 0)$.

10.4.4 Opcions dels gràfics a l'espai

Com en el cas de gràfics al pla, ja heu vist en els exemples anteriors que es poden modificar aspectes d'un gràfic de l'espai modificant algunes de les opcions de visualització. La llista de totes les opcions comuns a tots els gràfics a l'espai és:

`ambientlight`	`filled`	`labels`	`projection`	`thickness`
`axes`	`font`	`light`	`scaling`	`tickmarks`
`axesfont`	`grid`	`lightmodel`	`shading`	`title`
`color`	`gridstyle`	`linestyle`	`style`	`titlefont`
`contours`	`labeldirections`	`numpoints`	`symbol`	`view`
`coords`	`labelfont`	`orientation`	`symbolsize`	

Taula 3: Opcions de la comanda `plot3d( )`

Mireu en l'ajuda de Maple que és el que controla cada una d'aquestes opcions (`?plot3d[options]`).

10.5 Animació de gràfics

Quan es vol veure com evoluciona una corba o una superfície que depèn d'un paràmetre, una bona solució és fer una *animació* d'aquesta evolució. Les comandes `animate( )` i `animate3d( )` del paquet **plots** fan aquesta feina.

Exemple 10.7
La comanda

```
> animate([r*cos(u),r*sin(u),u=0..2*Pi],r=0..6,frames=20,
> scaling=constrained);
```

dibuixarà una animació d'un circumferència centrada a l'origen de radi creixent, des de 0 fins a 6 (realitzant 20 *fotogrames* de la pel·lícula). Per a veure efectivament l'animació heu de seleccionar el gràfic que surt, en aquest moment apareixeran nous botons en la barra d'eines del programa que permeten arrencar, parar, anar més ràpid o més lent, ... (amb el botó dret també apareix un menú des del que es pot controlar l'animació).

Per a generar animacions de gràfics a l'espai s'utilitza la comanda `animate3d( )`.

Exemple 10.8

```
> animate3d([r*cos(t+a), r*sin(t+a), r^2*cos(2*t)], r=0..2, t=0..2*Pi,
> a=0..3,frames=20);
```

Exercici 10.4
Feu una animació que mostri una esfera de radi 1 que va *botant*, movent el seu centre des del $(0, 0, 1)$ fins al $(0, 0, 0)$ i tornant a pujar.

10.6 Comandes del paquet **plots**

Ja s'ha comentat en altres apartats algunes de les funcions que proporciona el paquet **plots** (com ara `display( )`, `polarplot( )`, `implicitplot( )`). El llistat complet de les funcions que defineix aquest paquet és el de la taula que ve tot seguit. En l'ajuda de Maple podreu trobar què fa i com s'utilitza cada una d'elles (`?plots`).

`animate`	`densityplot`	`listplot`	`polyhedra_supported`
`animate3d`	`display`	`listplot3d`	`replot`
`animatecurve`	`display3d`	`loglogplot`	`rootlocus`
`arrow`	`fieldplot`	`logplot`	`semilogplot`
`changecoords`	`fieldplot3d`	`matrixplot`	`setoptions`
`complexplot`	`gradplot`	`odeplot`	`setoptions3d`
`complexplot3d`	`gradplot3d`	`pareto`	`spacecurve`
`conformal`	`implicitplot`	`pointplot`	`sparsematrixplot`
`contourplot`	`implicitplot3d`	`pointplot3d`	`sphereplot`
`contourplot3d`	`inequal`	`polarplot`	`surfdata`
`coordplot`	`listcontplot`	`polygonplot`	`textplot`
`coordplot3d`	`listcontplot3d`	`polygonplot3d`	`textplot3d`
`cylinderplot`	`listdensityplot`	`polyhedraplot`	`tubeplot`

Taula 4: Comandes del paquet **plots**

11 Equacions diferencials

La comanda de Maple que permet solucionar una equació diferencial és `dsolve( )`. Per a escriure una equació diferencial, es pot utilitzar indistintament la comanda `diff( )` o la comanda `D( )`, com es pot veure en els exemples següents:

Exemple 11.1
Si es vol resoldre l'equació diferencial més simple $y' = y$ posarem

```
> dsolve(diff(y(x),x)=y(x),y(x));
```

I si volem afegir-hi una condició inicial bastarà fer

```
> dsolve({diff(y(x),x)=y(x),y(0)=3},y(x));
```

En comptes d'utilitzar `diff( )` també es pot utilitzar `D( )` seguint els convenis equivalents.

```
> dsolve({D(y)(x)=y(x),y(0)=3},y(x));
```

Per a practicar, podeu resoldre les equacions diferencials següents:

Exercici 11.1

a) $y' - y\tan(x) = 3\exp(-\sin(x))$

b) $xy' + y = y^2 x\ln(x)$

c) $y' = x^2 - 2xy + y^2$

d) $y' = \dfrac{x-y+1}{x+y+3}$

e) $y' = \dfrac{x-y-1}{2x-2y+1}$

f) $(\exp(y) - 2xy)y' = y^2$

g) $(xy' - y)\cos(\frac{2y}{x}) = -3x^4$

h) $(2\cos y)y' + \sin y = x^2 \operatorname{cosec}(y)$

Substituint els valors que surten en cada una de les equacions podeu comprovar que les solucions que dóna la comanda `dsolve( )` són correctes.

I si voleu un exercici amb enunciat (trivial si s'aplica el sentit comú), podeu resoldre el següent:

Exercici 11.2
Els rajos que surten d'una font lluminosa situada a l'origen de coordenades del pla es reflecteixen en una certa corba d'aquest pla i tornen al punt d'emissió. Quina és aquesta corba?

Tots els exemples anteriors són equacions diferencials *de primer ordre* i amb una sola incògnita, com ja deveu suposar la comanda `dsolve( )` també pot treballar amb equacions d'ordres superiors i amb sistemes d'equacions diferencials amb més d'una incògnita.

Exemple 11.2
El sistema $y' = z, z' = -y$ es pot solucionar amb

```
> dsolve({D(y)(x)=z(x), D(z)(x)=-y(x)},{y(x),z(x)});
```

I si es vol que $y(0) = 0, z(0) = 1$ resultarà

```
> dsolve({D(y)(x)=z(x), D(z)(x)=-y(x),y(0)=0,z(0)=1},{y(x),z(x)});
```

(fet que és ben conegut!).

Per a les equacions d'ordre superior (en les que intervenen derivades d'ordre dos o més gran) es pot treballar també amb `diff( )` o amb `D( )`. Veieu a continuació un parell d'exemples (notareu que en la comanda `dsolve( )` no s'especifica al final la funció respecte de la que volem trobar les solucions, sovint no és necessari).

Exemple 11.3
Primer utilitzant `diff( )`

```
> dsolve(y(x)*diff(y(x),x$2)-diff(y(x),x)^2-x*y(x)^2=0);
```

I ara utilitzant `D( )` en un altre exemple

```
> dsolve(y(x)*(D@@2)(y)(x)+(D(y)(x))^2=0);
```

Observareu que per a designar la funció segona derivada de y hem utilitzat l'operador `@@`. Cal dir que, en general, la comanda `f@@n` és una manera ràpida de representar la funció resultant de composar `n` cops la funció `f` amb ella mateixa (en l'exemple `(D@@2)(y)` és sinònim de `D(D(y))`).

Exercici 11.3
Quina és la solució de l'equació diferencial de tercer ordre

$$y''' - 3y'' + y' + y = \exp(-x)(10 - 4x)$$

tal que $y(0) = 5, y'(0) = 6, y''(0) = 3$?

Hi ha moltes equacions diferencials que no tenen solucions en les que la incògnita es pugui expressar de forma simple en funció de la variable independent. L'opció `implicit` de la comanda `dsolve( )` fa que els procediments de solució de l'equació no intentin aïllar la incògnita en funció de la variable i es conformin amb una expressió implícita.

Exemple 11.4
Proveu per un costat

```
> dsolve((3*y(x)^2+exp(x))*diff(y(x),x)+exp(x)*(y(x)+1)+cos(x)=0,implicit);
```

i per un altre

```
> dsolve((3*y(x)^2+exp(x))*diff(y(x),x)+exp(x)*(y(x)+1)+cos(x)=0);
```

Exercici 11.4
a) Feu un gràfic de la solució de l'equació diferencial de l'exemple anterior que passa pel punt $(3, 5)$.

b) Feu un gràfic en el que es vegin 20 solucions diferents de l'equació de l'exercici anterior.

Una altra forma d'obtenir les solucions d'una equació diferencial és en forma paramètrica (s'obtenen totes les variables com a funcions d'un paràmetre) i d'aquesta forma es pot descriure la *trajectòria* d'una solució. Aquest tipus de solució es pot obtenir combinant les opcions `implicit` i `parametric` en un `dsolve( )`. Per exemple

Exemple 11.5
Proveu `dsolve` amb l'equació $yy' = -x$ i opcions `parametric` i `implicit`. Feu un gràfic d'algunes de les solucions que s'obtenen. Comproveu que les solucions són les circumferències amb centre a l'origen de coordenades (això serà especialment obvi si deixeu només l'opció `implicit`).

11.1 Gràfics d'equacions diferencials

A part dels gràfics que es poden obtenir amb la comanda `plot( )` o amb `implicitplot( )` disposem també de funcions específiques per a fer gràfics a partir d'equacions diferencials dins del paquet **DEtools**. La comanda bàsica és `DEplot( )` que fa un gràfic del camp de direccions que determina l'equació diferencial com en l'exemple següent:

Exemple 11.6

```
> with(DEtools):
> DEplot(D(y)(x)=4*x/y(x),y(x),x=-2..2,y=-2..2);
```

Podeu veure que s'ha d'escriure l'equació diferencial, la incògnita i la *finestra* en la que s'ha de representar el camp.

Aquesta mateixa comanda també té opcions per a representar solucions concretes de l'equació diferencial (més d'una a l'hora) marcant dins de la comanda la/les condició/ons inicials que ha/n de verificar.

Exemple 11.7
Per a veure una solució marcada sobre el camp es fa

```
> DEplot(D(y)(x)=4*x/y(x),y(x),x=-2..2,[[y(0)=1]],y=-2..2);
```

I si s'en volen veure més es pot fer

```
> DEplot(D(y)(x)=4*x/y(x),y(x),x=-2..2,[[y(0)=1],[y(0)=0.5],[y(0)=0.25]],
> y=-2..2);
```

També es pot fer un dibuix d'un sistema

```
> DEplot({D(x)(t)=y(t),D(y)(t)=-x(t)},{x(t),y(t)},t=0..2*Pi,
> [[x(0)=0,y(0)=1]],x=-2..2,y=-2..2,scaling=constrained);
```

Si només interessa dibuixar algunes de les solucions de l'equació diferencial i eliminar el dibuix del camp, es pot utilitzar l'opció `arrows= none` (l'opció `arrows` també pot tenir assignats el valors `small, medium, large` que canviaran la grandària de les fletxes que representen el camp).

Podeu trobar més opcions d'aquesta comanda amb `?DEplot`.

11.2 Solucions numèriques d'equacions diferencials

Amb la mateixa comanda `dsolve( )` es poden obtenir solucions numèriques d'una equació diferencial. És suficient afegir l'opció `numeric` com en l'exemple següent:

```
> solnum:= dsolve({D(y)(x)=x^2+y(x)^3,y(0)=0},y(x),numeric);
```

Exemple 11.8
Veureu que en realitat la resposta de Maple no és cap valor numèric concret sinó un *procediment* que calcula aquests valors aproximats de la solució demanada.

```
> seq(solnum(i/10),i=0..10);
```

Es poden fer directament gràfics d'aquestes aproximacions numèriques de solucions d'equacions diferencials amb la comanda `odeplot( )` del paquet **plots**.

Exemple 11.9

```
> with(plots):
> odeplot(solnum,[x,y(x)],x=-1..1);
```

Finalment, no deixeu de consultar l'ajuda de Maple sobre la comanda `dsolve( )` (`?dsolve`) i les seves opcions ni la del paquet **DEtools** (`?DEtools`) on podreu veure totes les funcions que posa a la vostra disposició per al tractament de les equacions diferencials.

12 Endomorfismes, vectors i valors propis, diagonalització. . .

Un dels problemes típics quan es treballa amb endomorfismes d'un espai vectorial de dimensió finita és el de determinar bases respecte de les que la matriu de l'endomorfisme és el més simple possible. Per a poder fer aquest estudi hi juga un paper fonamental la determinació dels vectors i valors propis i dels subespais associats. En el paquet **LinearAlgebra** de Maple es poden trobar funcions per a realitzar totes les operacions necessàries.

12.1 Polinomi característic i valors propis

Per a determinar el polinomi característic d'una matriu M podem realitzar els càlculs construint la matriu $M - x\,\mathrm{Id}$ i demanant el determinat d'aquesta nova matriu

```
> with(LinearAlgebra):
> M:=< <1,1,3> | <2,7,-2> | <1,3,-6> >;
> Mx:= simplify(M-x*IdentityMatrix(3));
> de:=Determinant(Mx);
```

o demanant directament que Maple faci el càlcul i doni directament el polinomi amb la comanda `CharacteristicPolynomial`

```
> CharacteristicPolynomial(M,x);
```

(triar una cosa o l'altre només és una qüestió de gust personal ja que els càlculs que s'han de realitzar internament són essencialment els mateixos).

Pera determinar els valors propis (i la seva multiplicitat) caldrà determinar les arrels del polinomi característic. També es pot fer directament amb la comanda `solve` o amb la comanda específica del paquet **LinearAlgebra** que fa aquesta feina (`Eigenvalues( )`).

```
> M1:=< <1,1,3> | <-1,7,-2> | <1,3,-6> >;
> solve(Determinant(M1-x*IdentityMatrix(3)));
> Eigenvalues(M1);
```

Com podreu veure, la diferència entre els dos resultats és ben poca.

12.2 Vectors propis

Un cop es coneixen els valors propis associats a una matriu A, la determinació dels espais de vectors $\vec{v}$ per als que es verifica alguna de les condicions $A\vec{v} = \lambda\vec{v}$ es redueix a la solució de sistemes d'equacions lineals homogenis. La comanda `NullSpace( )` del paquet **LinearAlgebra** permet fer la feina.

Exemple 12.1
La matriu

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

té per valors propis 1 i -1 tal i com es pot comprovar amb

```
> A:= <<-2|2|-3|-1>,<2|-1|2|0>,<3|-2|4|1>,<-2|0|-2|-1>>;
> Eigenvalues(A);
```

Els espais de vectors propis amb valor propi 1 o -1 es podran determinar amb

```
> B1:= A-IdentityMatrix(4);
> E1:=NullSpace(B1);
> B2:= A+IdentityMatrix(4);
> E2:=NullSpace(B2);
```

Podreu veure que `E1` i `E2` contenen cada una un parell de vectors que generen l'espai de vectors propis corresponent (multipliqueu A per cada una de les columnes que surten en `E1` o `E2` per a comprovar que es tenen, efectivament, vectors propis amb valor propi 1 o -1). A la vista dels resultats es pot afirmar que la matriu A és diagonalitzable i es pot confirmar aquest fet considerant

```
> E:=<op(E1)|op(E2)>;
> E^(-1).A.E;
```

Tot el procés de calcular els vectors propis associats a cada un dels valors propis d'una matriu té una comanda específica en el paquet **LinearAlgebra** que es crida amb `Eigenvectors( )`. Si considereu

```
> EE:=Eigenvectors(A);
```

obtindreu una seqüència amb dos elements, el primer és una columna amb els valors propis (repetits tants cops com la multiplicitat) i el segon és una matriu que té per columnes les components dels vectors propis ordenades de la mateixa manera que els valors propis (si us hi fixeu, veureu que no surten exactament els mateixos vectors que quan apliqueu `NullSpace( )`). Si feu la multiplicació

```
> EE[2]^(-1).A.EE[2];
```

tornareu obtenir la matriu diagonal equivalent a la matriu A.

Amb tot el que heu vist fins ara, l'única dificultat de l'exercici següent hauria de ser la introducció de la matriu M.

Exercici 12.1
Considereu la matriu

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

a) Calculeu el polinomi característic de M.

b) Determineu els seus valors propis i les multiplicitats corresponents.

c) Determineu els espais de vectors propis associats a cada un dels valors propis.

d) Comproveu que la matriu M és diagonalitzable.

e) Calculeu una matriu C tal que $C^{-1} \cdot M \cdot C$ sigui diagonal.

En aquest altre exercici podreu veure una matriu amb el mateix polinomi característic que la matriu M però que no és diagonalitzable.

Exercici 12.2
Considereu

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

calculeu el seu polinomi característic i comproveu que no és diagonalitzable.

Una de les propietats bàsiques del polinomi característic d'una matriu és que *anul·la* a la matriu corresponent. És a dir, si $p(x)$ és el polinomi característic de la matriu A es compleix $p(A) = 0$. En l'exercici següent es posa de manifest aquest fet en un parell d'exemples.

Exercici 12.3
Considereu les matrius A donades per

$$\begin{pmatrix} 1 & 3 & 4 & -1 \\ 0 & 3 & 2 & 1 \\ -1 & -1 & 2 & 5 \\ 2 & 3 & 1 & -1 \end{pmatrix} \quad \begin{pmatrix} -2 & 3 & 1 & -1 \\ 2 & -1 & 0 & 1 \\ -1 & -2 & 2 & 3 \\ 3 & 3 & 1 & -1 \end{pmatrix}$$

calculeu els seus polinomis característics i comproveu que les anul·len.

D'entre tots els polinomis que anul·len un endomorfisme s'en pot determinar el que té el grau més petit (polinomi mínim). La comanda de Maple `MinimalPolynomial( )` calcula el polinomi mínim d'una matriu.

Exemple 12.2
Es pot calcular el polinomi mínim de

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

amb

```
> A:= < <0,3,2>|<-1,-4,-2>|<2,6,3> >;
> pm:=MinimalPolynomial(A,x);
> Pm:=unapply(pm,x);
> Pm(A);
```

Observeu que la comanda `Pm` és la funció que s'obté *prenent com a variable* la `x` en l'expressió `pm` (com quan utilitzem `D( )` en comptes de `diff( )` per a obtenir una derivada en l'exemple 9.6), que s'aplica correctament sobre una matriu i sobre A dóna com a resultat la matriu nul·la.

Exercici 12.4
Calculeu el polinomi característic de la matriu A anterior i comproveu que té els mateixos factors irreductibles que el polinomi mínim (només que amb multiplicitats menors).

12.3 Formes de Jordan

Com ja haureu vist, un endomorfisme d'un espai vectorial (matriu quadrada) no sempre diagonalitza. Per a aquests casos és possible intentar obtenir altres *formes canòniques.* En l'exemple següent veureu una matriu 3×3 que té un únic valor propi i que, a més, l'espai de vectors propis només té dimensió 1.

Exercici 12.5
Sigui

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

a) Calculeu el polinomi característic de A, comproveu que A té un únic valor propi λ (triple) i que l'espai de vectors propis de A té dimensió 1.

b) Sigui $\vec{v}_3$ un vector propi no nul de A. Determineu un vector $\vec{v}_2$ tal que $(A-\lambda\cdot\mathrm{Id})\vec{v}_2=\vec{v}_3$ i un vector $\vec{v}_1$ tal que $(A-\lambda\cdot\mathrm{Id})\vec{v}_1=\vec{v}_2$.

c) Comproveu que $\vec{v}_1$, $\vec{v}_2$, $\vec{v}_3$ són independents.

d) Determineu la matriu $P^{-1}\cdot A\cdot P$ si P és la matriu que té per columnes les components del vectors $\vec{v}_i$.

La matriu que haureu obtingut en l'últim apartat de l'exercici anterior és el que normalment s'anomena un *bloc de Jordan* (un mateix valor en tots els elements de la diagonal i 1 just en els llocs que estan per sobre, o per sota, d'aquesta diagonal). En general, es poden buscar formes canòniques de la forma $\left(\begin{array}{c|c|c} J_1 & 0 & \cdots \\ \hline 0 & J_2 & \cdots \\ \hline \vdots & \vdots & \ddots \end{array}\right)$, on cada un dels J_i és un bloc de Jordan.

De la mateixa forma que per als valors i vectors propis, dins del paquet **LinearAlgebra** hi ha comandes específiques per a determinar la forma de Jordan d'un endomorfisme i per a obtenir el canvi de base que permet obtenir aquesta forma a partir de qualsevol altra forma equivalent. La comanda que fa les dues feines és `JordanForm( )`. Per a obtenir la forma de Jordan equivalent a una matriu A donada basta fer `JordanForm(A);`, si es vol la matriu de canvi es pot modificar la comanda anterior i posar `JordanForm(A, output='Q');` i si es volen les dues coses `JordanForm(A, output=['J','Q']);`. Veieu a continuació uns exemples

Exemple 12.3

```
> A:=<<2, 2, 0, 0>|<1/2, 3/2, -1/2, 1/2>|<0, 0, 2, 0>|<-1/2, -3/2, 1/2, 3/2>>;
> JordanForm(A);
> B:= JordanForm(A,output='Q');
> C:= JordanForm(A,output=['J','Q']);
> simplify(B^(-1).A.B);
> simplify(C[2]^(-1).A.C[2]);
```

No ha de ser difícil fer l'exercici següent:

Exercici 12.6
Considereu la matriu A donada per

$$\begin{pmatrix} 3 & 0 & 0 & 1 & 4 & -3 & 2 \\ -1 & 3 & 0 & -2 & -5 & 4 & -2 \\ 0 & 2 & 3 & -3 & -1 & 2 & 1 \\ 0 & 1 & 0 & 1 & -1 & 1 & 0 \\ -2 & -2 & -1 & 0 & -8 & 6 & -7 \\ -1 & -2 & -1 & 1 & -6 & 5 & -5 \\ 2 & 0 & 1 & 1 & 9 & -6 & 7 \end{pmatrix}$$

i la matriu B

$$\begin{pmatrix} 2 & 1 & -1 & 1 & 4 & -3 & 3 \\ 0 & 3 & 0 & -2 & -4 & 3 & -2 \\ 0 & 3 & 2 & -3 & 0 & 1 & 2 \\ 0 & 1 & 0 & 1 & -1 & 1 & 0 \\ 0 & -4 & 1 & 0 & -8 & 6 & -9 \\ 0 & -4 & 1 & 1 & -7 & 6 & -7 \\ 0 & 1 & 0 & 1 & 8 & -5 & 8 \end{pmatrix}$$

Existeix alguna matriu invertible P tal que $P^{-1} \cdot A \cdot P = B$? (Determineu les formes de Jordan de A i B).

Exercici 12.7
Comproveu que existeix una matriu Q (com a mínim) tal que $Q^{-1} \cdot A \cdot Q = B$ per a

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

i calculeu una d'aquestes matrius Q.

13 Interpolació i ajust de dades

13.1 Aproximació de funcions, polinomi interpolador

Recordem que Maple pot representar gràficament un núvol de punts $(x_1, y_1), \ldots, (x_n, y_n)$ amb la funció `plot( )`.

Exercici 13.1
Definiu una llista `l` com els següents punts de $\mathbb{R}^2$: $(0,2)$, $(1,1)$, $(2,2)$, $(3,1)$, $(4,2)$, $(5,1)$, $(6,2)$, $(7,1)$, $(8,2)$, $(9,1)$ i $(10,2)$. Dibuixeu en una gràfica el núvol de punts corresponent.

Donats n punts $(x_1, y_1), \ldots, (x_n, y_n)$ amb $x_i \neq x_j$ si $i \neq j$ existeix un polinomi p de grau menor o igual a $n-1$ complint que $p(x_i) = y_i$ per a tot $i = 1..n$, que anomenem *polinomi interpolador*. La funció que calcula polinomis interpoladors està al paquet **CurveFitting** i s'anomena `PolynomialInterpolation( )`.

Exemple 13.1
Si volem calcular el polinomi que interpola els punts $(0,0), (1,1)$ i $(2,4)$, que ha de ser de grau menor o igual a 2, ho podem fer amb Maple:

```
> with(CurveFitting); ll:=[[0,0],[1,1],[2,4]]:
> PolynomialInterpolation(ll,x);
```

Exercici 13.2
Calculeu el polinomi interpolador p als punts de l'exercici anterior. Quan val el polinomi avaluat al punt 5? I al punt 9.5? Feu la representació gràfica dels punts i el polinomi interpolador.

13.2 Spline

Una altra opció per a trobar funcions que passin per punts fixats és "*agrupar*" els punts en subconjunts més petits i calcular el polinomi de grau més petit per a cada un dels subconjunts. Per exemple, podem considerar que la unió dels punts mitjançant segments es correspon a considerar els punts de dos en dos i considerar el polinomi de grau 1 (recta) que els uneix.

Dins el paquet **CurveFitting** tenim la funció `Spline( )` (la `S` en majúscules) per a definir aquestes noves aproximacions.

Exemple 13.2
Si volem unir els punts $(0,0), (1,1)$ i $(2,4)$ amb rectes escrivim:

```
> ll:=[[0,0],[1,1],[2,4]]:
> Spline(ll,x,degree=1);
```

On l'opció `degree=1` és per a definir el grau dels polinomis que estem calculant.

Exercici 13.3
Dibuixeu tres gràfiques amb els "*splines*" de graus 1, 2 i 3 de la llista de punts de l'exercici 13.1.

13.3 Interpolació racional

Hi ha cops en els que pot interessar obtenir no un polinomi si no, més generalment, funció racional (quocient de polinomis) per a interpolar una llista de valors coneguts. Dins el paquet **CurveFitting** hi ha la funció `RationalInterpolation( )` que pot fer aquesta feina.

Exemple 13.3
Si teniu la llista de punts

```
> a := [[-3, -55/14],[-2, -19/9],[-1, -1/6],[0, 1],[1, 5/6],[2, 5/9],[3, 11/14]];
> plot(a,style=point);
```

podeu buscar una expressió racional que interpoli aquests valors amb

```
> with(CurveFitting):
> RationalInterpolation(a,x);
```

(Observeu que, apart de la llista de valors per interpolar, s'ha d'especificar un nom de variable per a poder escriure l'expressió resultant).

Aquesta funció també es pot cridar posant com arguments dues llistes separades per als valors de les x's i de les y's com en el següent

```
> valx := [-3, -2, -1, 0, 1, 2, 3];
> valy := [-55/14, -19/9, -1/6, 1, 5/6, 5/9, 11/14];
> RationalInterpolation(valx,valy,x);
```

Es poden consultar altres opcions amb `?RationalInterpolation`.

13.4 Exercicis d'interpolació

Exercici 13.4
Feu una funció (o procediment) `dibuix` que fixada una llista de punts dibuixi de colors diferents el núvol de punts, els segments que uneixen els punts, el "*spline*" de grau 2, el "*spline*" de grau 3 i el polinomi interpolador en una sola gràfica.

Exercici 13.5
Considereu la llista de punts:
`lli1:=[[-9,1], [-8,1], [-7,1], [-6,1], [-5,2], [-4,3], [-3,5], [-2,8], [-1,12], [0,16], [1,12], [2,8], [3,5], [4,3], [5,2], [6,1], [7,1], [8,1], [9,1]].`
Dibuixeu el núvol de punts en una gràfica. Apliqueu la funció `dibuix` de l'exercici anterior a aquest núvol de punts. Que observeu?

Exercici 13.6
Definiu una funció `aproxf` que depengui de 4 paràmetres `f, a, b, n` que dibuixi la funció `f` entre `a` i `b` i el polinomi de grau menor o igual a n que interpoli a `n+1` punts equiespaiats entre `a` i `b` (inclosos).

Exercici 13.7
Apliqueu la funció `aproxf` a la funció $f(x) = \dfrac{1}{1+x^2}$ entre -5 i 5 i per als valors de `n` 1, 3, 6, 10 i 15. Què observeu?

Exercici 13.8
Feu un gràfic on es puguin veure al mateix temps les interpolacions polinòmica i racional del núvol de punts `a:= [[-3, -55/14], [-2, -19/9], [-1, -1/6], [0, 1], [1, 5/6], [2, 5/9], [3, 11/14]];` (reflexioneu sobre les diferències que s'observen entre els dos resultats).

13.5 Ajust a les dades

Una qüestió que es planteja sovint amb dades experimentals és la següent: es realitza un experiment i es voldria saber si hi ha una relació entre dues de les seves propietats (una relació de proporcionalitat per exemple). És a dir, pot tenir interès descriure a través d'una funció la relació entre dues variables quantitatives X i Y. Posem $Y = f(X)$ on f és una funció que depèn d'uns paràmetres $a_1, ..., a_d$ que haurem de determinar per garantir un *bon* ajust. Naturalment, a més de disposar d'observacions, $(x_1, y_1), ..., (x_n, y_n)$, dades que suposem bivariants (x, y) $x, y \in \mathbb{R}$, hem de recorre a un criteri que valori la bondat de l'ajust. El criteri que estudiarem és l'anomenat de *mínims quadrats*.

13.5.1 Llei de Hooke

Començarem plantejant un experiment concret. La llei de Hooke a la física ens diu que hi ha una relació de proporcionalitat entre el pes que es penja d'una molla i la distància que aquesta s'estira. És a dir, si x és un pes i y representa la distància que una molla s'estira al penjar-li aquell pes, existeix una constant k tal que $y = kx$. k s'anomena la constant d'elasticitat de la molla.

L'experiment que realitzarem consisteix en determinar aquesta constant k per a una molla concreta. Per això, es realitza l'experiment que consisteix en mesurar les distàncies que la molla s'estira per a diferents pesos i obtenim la següent taula,

Pes	2	2.5	3	3.5	4	5
cm	3.2	4	5	5.07	6.5	8

Si l'experiment es realitzés en condicions perfectes (mesures de precisió perfecte, sense fregament amb l'aire,...) les dades obtingudes serien de la forma $y_i = kx_i$ i per tant, seria fàcil determinar la constant k. Però com que no podem suposar condicions perfectes, obtenim unes dades que es troben lleugerament distorsionades.

Exercici 13.9
En uns eixos de coordenades representeu la taula de punts $(x, y) = (\text{pes}, \text{cm})$ de l'experiment.

L'objectiu doncs es trobar la recta de la forma $y = kx$ que millor s'ajusta a les dades obtingudes experimentalment. Per això hem de fixar un criteri. És a dir, què vol dir que una recta s'ajusta millor que una altra al conjunt de dades?

El criteri a fer servir és el següent: buscarem k tal que la suma de les distàncies (al quadrat) dels punts obtinguts experimentalment a la recta sigui mínima. És a dir, el valor de k tal que

$$\sum_{i=1}^{6}(y_i - kx_i)^2$$

pren el valor mínim. Observeu que aquesta funció mesura l'error comés en l'experiment respecte les condicions perfectes.

Exercici 13.10
Sigui $d(k) = \sum_{i=1}^{6}(y_i - kx_i)^2$, busqueu el valor de k pel qual d pren el valor mínim. En una mateixa gràfica representeu els punts obtinguts a l'experiment i la gràfica de la recta $y = kx$ pel valor de k obtingut.

Aquest mètode que acabem de descriure per aproximar dades obtingudes experimentalment per funcions és el que es coneix com el mètode dels mínims quadrats.

13.5.2 El mètode dels mínims quadrats

En general, aquest mètode s'utilitza quan es vol analitzar la relació funcional entre un conjunt de dades (x_i, y_i) obtingudes experimentalment. Es suposa conegut el tipus de funció f que les relaciona però no els paràmetres $a_1, \ldots, a_d$ de la funció.

El criteri de mínims quadrats proposa calcular els paràmetres $a_1, \ldots, a_d$ tals que l'expressió dels errors

$$\sum_{i=1}^{n}\left(y_i - f(x_i)\right)^2,$$

és mínima, és a dir, la suma dels desajustos quadràtics entre els valors que s'han observat de les y_i i els valors $f(x_i, ; a_1, \ldots, a_d)$ proposats per f.

El primer pas sempre consisteix en representar gràficament en uns eixos de coordenades els punts obtinguts experimentalment com ja heu fet en la secció anterior. D'aquest tipus de diagrama se'n diu un diagrama de dispersió. Dins el paquet **plots** hi ha la comanda `pointplot( )` que permet fer aquests tipus de gràfics fàcilment.

Exemple 13.4
Donades les observacions següents: $(0.70, 0.035)$, $(0.76, 0.025)$, $(0.37, -0.18)$, $(0.82, 0.045)$, $(0.29, -0.16)$, $(0.56, -0.058)$, $(0.42, -0.11)$, $(0.47, -0.085)$, el següent grup de comandes realitza la representació gràfica de les dades anteriors en un gràfic de dispersió:

```
> with(plots):
> dades:=[ [0.70,0.035],[0.76,0.025],[0.37,-0.18],[0.82,0.045],
> [0.29,-0.16],[0.56,-0.058],[0.42,-0.11],[0.47,-0.085]];
> pointplot(dades);
```

(Per a obtenir aquest gràfic també es pot utilitzar un `plot( )` com haureu fet abans. L'avantatge de `pointplot( )` és que no cal especificar ni `style=point` ni `symbol=circle`).

Un cop fet el diagrama, el següent pas consisteix en analitzar la forma del núvol de punts obtinguts per determinar quin tipus de funció s'hi ajustaria millor.

Exercici 13.11
Quin tipus de funció creieu que podríem triar per aconseguir un bon ajust a les dades de l'exemple anterior? (Proveu *a ull* una funció lineal del tipus $f(x) = a\,x + b$ i feu un gràfic de la situació).

Exercici 13.12
Donades les observacions: $(-1, 5)$, $(-0.4, 2.5)$, $(0.1, 2.1)$, $(0.8, 3.98)$, $(1.2, 6.5)$, $(1.5, 8.8)$, $(2.1, 15.2)$,

Representeu les dades en un gràfic de dispersió. En aquest cas, quina mena de funció creieu que podríem triar per ajustar a les dades? (sembla el més adequat triar $y = f(x) = a + bx + cx^2$, intenteu donar valors als paràmetres a, b i c per ajustar aquestes dades).

Un cop determinat el tipus de funció, podem calcular els paràmetres pels quals la suma dels quadrats dels errors de mesura és mínim. La comanda que calcula els paràmetres que minimitzen aquests errors és `LeastSquares( )` del paquet **CurveFitting**. Com a paràmetres necessita el nom de les variables, el tipus de funció i les llistes de dades experimentals.

En l'exemple següent veureu com utilitzar aquesta comanda pel cas d'ajust per una recta. Aplicarem `LeastSquares( )` a les dades obtingudes per l'experiment de la molla. (Anomeneu `Pes` i `Cm` les dades obtingudes en l'experiment de la molla).

```
> with(CurveFitting):
> Pes:=[2,2.5,3,3.5,4,5];
> Cm:=[3.2,4,5,5.07,6.5,8];
> PesCm:=[seq([Pes[i],Cm[i]],i=1..nops(Pes))];
> minq:=LeastSquares(Pes,Cm,pes);
> minq2:=LeastSquares(PesCm,pes);
```

(Noteu que la comanda `LeastSquares( )` es pot aplicar o bé al parell de llistes `Pes`, `Cm` o a la llista individual de les mesures de pes i estirament combinades. Aplicar una versió o l'altre només dependrà de la manera en la que es tinguin guardades les dades).

Exercici 13.13
Compareu aquest resultat amb l'obtingut anteriorment i feu un gràfic amb el diagrama de dispersió juntament amb la recta de mínims quadrats que acabeu de calcular.

Exercici 13.14
Calculeu els paràmetres de la recta que millor s'ajusta a les dades de l'exemple 13.4. En una mateixa gràfica representeu el diagrama de dispersió i la recta obtinguda.

En el cas d'ajustar altra tipus de corbes, com és el cas de l'exercici 13.12 on un polinomi de grau 2 semblava més adequat, la sintaxi és

```
> dades:=[[-1,5],[-0.4,2.5],[0.1,2.1],[0.8,3.98],[1.2,6.5],[1.5,8.8],[2.1,15.2]];
> curva:=LeastSquares(dades,x,curve=a*x^2+b*x+c);
```

Exercici 13.15
Representeu en una mateixa gràfica el diagrama de dispersió i la funció quadràtica obtinguda.

Exercici 13.16
El punt d'ebullició d'una mescla d'etanol i aigua depèn de la proporció d'etanol a la mescla. S'ha realitzat un experiment en el que s'han obtingut els resultats següents

Proporció	.0	.0190	.0721	.0966	.1238	.1661	.2337	.2608
Temperatura	100.	95.5	89.0	86.7	85.3	84.1	82.7	82.3

Proporció	.3273	.3965	.5079	.5198	.5732	.6763	.7472	.8943
Temperatura	81.5	80.7	79.8	79.7	79.3	78.74	78.41	78.15

(no cal dir que les temperatures venen donades en graus centígrads).

Si sabem que la funció que relaciona el punt d'ebullició t i la proporció d'etanol p en la mescla és de la forma $t = a\,e^{-14p} + b\,p + c$, calculeu les constants a, b, c per a l'etanol. Feu un gràfic on es mostrin els resultats de l'experiment (punts d'ebullició respecte la proporció d'etanol) i la corba que ajusta aquests resultats.

13.5.3 Relacions no polinòmiques

No tots els tipus de relacions que voldrem analitzar entre variables serà de tipus polinomial. Hi ha experiments que es modelen amb d'altres tipus de funcions matemàtiques com per exemple, la desintegració de substàncies radioactives.

Un punt clau en la determinació de l'edat d'una pintura o un fòssil és el fenomen de la radioactivitat. El físic Rutherford va determinar que els àtoms de certs elements radioactius són inestables i que, en un interval de temps donat, uns fracció fixa dels àtoms presents en aquell instant de temps es desintegra espontàniament per formar altres substàncies. Aquest principi té com a conseqüència que el nombre d'àtoms presents a cada instant de temps segueix una funció exponencial

$$N(t) = N_0 e^{-\lambda(t-t_0)},$$

on t_0 és l'instant on es van començar a mesurar el nombre d'àtoms. Es diu que λ és el coeficient de desintegració i depèn de cada substància.

L'estratègia és la mateixa que hem seguit fins ara: donades unes dades experimentals preses sobre un material concret (nombre d'àtoms a certs intervals de temps) volem obtenir el coeficient de desintegració λ. Aquesta dada és molt útil si volem fer prediccions sobre el nombre d'àtoms en temps futurs o l'edat del material.

El primer que s'ha de fer és transformar aquest problema d'ajust de dades per una exponencial en un problema d'ajust lineal. Aquesta tècnica que ara descriurem és molt utilitzada i s'anomena la transformació logarítmica.

Sigui $y = ke^{at}$, si apliquem el logaritme als dos costats de l'equació obtenim $\ln(y) = \ln(k) + at$ que és lineal en $\ln(k)$ i a (anomenarem $K = \ln(k)$).

Exercici 13.17

S'han pres unes dades experimentals sobre dues mostres de material radioactiu. Ens asseguren que totes dues són de plom 210. Per verificar-ho prenem diferents mesures dels àtoms de plom 210 en ambdues mostres al llarg de dos anys i obtenim les dades següents, on t es mesura en mesos i n_i són grams de plom a la mostra i.

t	n_1	n_2
1	22.23	32
2	1.2	1.3
4	0.04	0.002
6	0.0001	$0.36 \cdot 10^{-5}$
8	$0.36 \cdot 10^{-7}$	$0.6 \cdot 10^{-8}$
14	$0.1 \cdot 10^{-14}$	$0.3 \cdot 10^{-16}$
16	$0.3 \cdot 10^{-17}$	$0.5 \cdot 10^{-19}$
24	$0.3 \cdot 10^{-27}$	$0.35 \cdot 10^{-30}$

Representeu les dades gràficament en uns eixos de coordenades i determineu la funció que determina la seva desintegració (una per a cada una de les mostres). Ens han enganyat?

14 Estadística descriptiva i representació gràfica de dades

Maple disposa del paquet anomenat **stats** que recull un seguit de subpaquets cadascun adequat per a diferents tipus d'anàlisis estadístics de dades. En aquesta pràctica farem un cop d'ull als subpaquets **describe**, **statsplots** i **transform**.

Les dades que es vulguin analitzar estaran guardades en una llista sobre la que aplicarem les diferents funcions del paquet. Si volem estudiar estadísticament els resultats del llançament d'un dau de sis cares cinc cops i els resultats obtinguts són $1, 3, 4, 3, 2$ posarem

```
> with(stats):
> data:=[1,3,4,3, 2];
```

i podrem aplicar a la llista `data` les diferents funcions.

En cas que l'ordre de les dades sigui irrellevant podem incorporar a la llista comandes del tipus `Weight(valor, pes)` on el paràmetre `valor` indica un dels valors de la llista i `pes` és el nombre de vegades que el valor apareix a la llista (com si poséssim `valor$pes`).

```
> [1,46, Weight(3,2), 90];
```

Maple ens permet importar un fitxer de dades ja existent mitjançant la comanda `importdata("fitxer",n)` on `fitxer` és l'adreça del fitxer de dades (que serà un fitxer text on les dades estan organitzades per columnes) i `n` és el número de columnes del fitxer. Maple importa el fitxer com una seqüència de n llistes de dades, una per cada columna. En l'exemple següent importem un fitxer on hi ha guardat el temps de vida (en hores) d'una mostra de llums fluorescents que està en el directori `Mis documentos` del disc `c:` del nostre sistema (aquest fitxer el podeu trobar a `http://mat.uab.es/gguasp/PractInt/Lamp/tempsvida.dat` i s'ha de guardar en algun lloc accessible pel sistema operatiu).

Si feu

```
> llums:=importdata("c:/Mis documentos/tempsvida.dat",1);
```

(o el que correspongui pel lloc on heu guardat el fitxer) tindreu en `llums` les dades recollides del que han durat aquests llums.

14.1 Les comandes de describe

Habitualment iniciem l'anàlisi de les dades amb un estudi descriptiu. En aquest punt és el subpaquet **describe**, el que ens proporciona les funcions necessàries com són, per exemple, `count` (que compta el nombre de valors que apareixen en la llista de dades), `mean` (que calcula la mitjana aritmètica de les dades introduïdes), `mode` (que dóna com a resultat el valor que apareix més cops dins del conjunt de dades), `variance` (que dóna una mesura de la dispersió dels valors de les dades calculant la mitjana aritmètica dels quadrats de les diferències de cada valor amb la mitjana de les dades) i moltes altres que podreu consultar a l'ajuda de Maple a partir de `help(describe);`

Exemple 14.1

```
> with(stats):
> data:=[1,1,3,Weight(2,4)];
> describe[count](data);
> describe[mean](data);
> describe[mode](data);
> describe[variance](data);
```

Noteu que, de fet, **describe** és una `table` de Maple en la que cada un dels seus elements és una funció diferent que es pot aplicar al conjunt de dades `data`.

Podem agrupar un seguit d'aquestes comandes per formar una llista d'operacions que s'apliquen conjuntament sobre la mateixa llista de dades i així obtenir el conjunt dels valors significatius en un sol resultat.

Exemple 14.2

```
> descriptiva:=[describe[mean],describe[median],describe[variance]];
> descriptiva(data);
```

Exercici 14.1
Apliqueu la comanda `descriptiva` a les dades del temps de vida dels llums fluorescents guardats en `llums`.

Exercici 14.2
Construïu una llista d'operacions que calculi els quartils de les dades de la llista `llums`. Els quartils d'una llista de dades són els tres valors que divideixen el conjunt total de dades, un cop s'han ordenat de menor a major, en quatre parts amb el mateix nombre de dades en cada un d'ells (comptant amb les repeticions que hi pot haver). Busqueu informació sobre `quartile( )` en l'ajuda de Maple.

14.2 Representació gràfica de les dades

Dins el paquet **stats** l'apartat que conté les funcions per a fer representacions gràfiques és **statplots**

14.2.1 Gràfics `boxplot`

Les representacions gràfiques resumeixen alguns aspectes de les dades. Un tipus de gràfic adequat per la representació de dades quantitatives és l'anomenat gràfic BoxPlot (diagrama de caixes) del que en teniu un exemple a continuació.

Exemple 14.3

```
> Xdata := [4.535, 4.029, 5.407, 1.605, 5.757, 3.527,
> 7.890,8.159, 6.092, 13.442, 2.845, 5.172, 3.277, 8.810, 3.657, 7.226,3.851,
> 2.162, 2.668, 4.692];
> statplots[boxplot](Xdata);
```

la línia central de la caixa representa la mediana (el valor central de les dades), les línies superior i inferior representen el primer i el tercer quartil i les línies que surten cap dalt i cap baix s'estenen fins a 3/2 del rang de valors entre quartils.

Noteu que Maple situa el diagrama de caixa sobre la posició zero de l'eix horitzontal. Si volem que el desplaci podem utilitzar l'opció `shift=n`, on `n` és el número d'unitats que volem que es desplaci. A més podem indicar l'amplada de la caixa mitjançant l'opció `width=n`. Si a la comanda `boxplot` introduïu més d'un fitxer de dades

```
> statplots[boxplot](Xdata1,Xdata2);
```

obtindreu els diagrames de caixa corresponents en un sol gràfic. Aquesta opció és útil quan es vol comparar el comportament de més d'un conjunt de dades.

Exercici 14.3
Feu un diagrama de caixa del conjunt de dades del fitxer `tempsvida.dat`

14.2.2 Histogrames

La distribució de les freqüències dels valors observats podem representar-la amb un histograma de freqüències com en l'exemple següent:

Exemple 14.4

```
> statplots[histogram](Xdata, numbars=5, area= 100);
> statplots[histogram](Xdata, numbars=5, area= count, color=cyan);
```

Observeu que l'opció `numbars=5` fa que es distribueixen les dades en cinc grups, que cada una de les barres té la mateixa amplada i una altura proporcional al nombre de dades que cauen dins el grup corresponent i que el valor numèric d'aquesta altura ens dóna els percentatge de dades en cada grup quan posem l'opció `area=100` i el nombre total de dades quan l'opció és `area=count`. Noteu també que s'hi poden afegir opcions gràfiques com per exemple `color=cyan` que farà que el color del dibuix sigui `cyan`.

Exercici 14.4
Feu ara un histograma en el que es representin els temps de vida dels fluorescents dividits en 10 grups i de tal forma que l'altura de les barres representi el nombre de mesures en cada un dels intervals.

Exercici 14.5
Importeu el fitxer `data3.dat` que podreu trobar a

`http://mat.uab.es/gguasp/PractInt/Lamp/data3.dat`.

Aquest fitxer conté tres columnes que corresponen al temps de vida en hores de tres tipus de fluorescents segons el seu cebador: ràpid, de pre-escalfament i instantani. Anomeneu els tres tipus de dades com `rapid`, `escalfament` i `instantani`.

a) Calculeu el número de fluorescents de cada tipus estudiats.

b) Per cada classe de fluorescent calculeu la mitjana, moda i variància del seu temps de vida.

c) Feu un histograma de 10 barres per cada tipus de fluorescent en que les columnes representin la freqüència de casos.

d) En un mateix gràfic dibuixeu els corresponents diagrames de caixa i analitzeu el resultat.

14.3 La comanda `rand`

Maple permet la generació de nombres aleatoris mitjançant la funció `rand( )`.

```
> exp1:=rand(1..10);
> exp2:=rand(6);
```

La primera línia crea una funció que genera un número aleatori enter entre 1 i 10. La segona línia crea una funció que genera un número aleatori entre 0 i 5. Si es crida `rand()` (sense arguments) el resultat és un nombre a l'atzar de 12 dígits (no negatiu). Comproveu que passa quan executeu `exp1()` i `exp2()` amb altres valors per als arguments de les dues versions de la comanda `rand( )`,

```
> exp1();
> exp2();
```

Exemple 14.5
Podem simular el llançament d'una moneda fent que el Maple generi números aleatoris enters entre 0 (=cara) i 1 (=creu). Si volem guardar 20 llançaments de la moneda, podem crear la seqüència que els conté.

```
> moneda:=rand(2);
> sim:=[seq(moneda(),i=1..20)];
```

14.4 Les comandes de transform

Finalment analitzarem algunes de les comandes del subpaquet **transform**. En particular veurem com ordenar un conjunt de dades i com comptar el número de vegades que apareix un mateix valor.

La comanda `transform[statsort](data)` serveix per ordenar un conjunt de dades (`data`) en ordre creixent.

Exemple 14.6

```
> ord:=transform[statsort](sim);
```

La comanda `transform[tally](data)` agrupa totes les dades que prenen el mateix valor mitjançant l'expressió `Weight`. D'aquesta manera podem observar el número de vegades que apareix cada valor.

Exemple 14.7

```
> agrup:=transform[tally](sim);
```

Finalment, i un cop hem agrupat les dades, la comanda `transform[frequency](data)` retorna una llista amb la freqüència de cada valor de la llista (número de vegades que es repeteix cada valor).

Exemple 14.8

```
> freq:=transform[frequency](agrup);
```

Exercici 14.6

a) Simuleu l'experiment que consisteix en llançar un dau 1000 vegades. Calculeu el número de vegades que ha sortit cada cara del dau.

b) Fixeu vos que la comanda `histogram( )` de **statplots** està pensada per agrupar les freqüències d'uns resultats en els que els valors possibles són valors reals però que en el cas d'un experiment en el que els resultats siguin només uns pocs valors (com en el cas del dau o de la moneda) no funcionarà massa bé (proveu-ho). Definiu un procediment que, partint del resultat d'un experiment amb valors discrets (dau, moneda. . .), doni un gràfic de barres amb les freqüències de cada un dels resultats possibles. Apliqueu-la als resultats de la simulació del llançament del dau. Són més o menys igual d'altes totes les columnes? El dau està trucat?

Índex alfabètic

Índex de taules

Llistat dels *paquets* disponibles

Els paquets standard als que es pot accedir amb la comanda `with( )` són els següents:

algcurves	Corbes algebraiques
codegen	Eines per a exportar en altres llenguatges de programació (C, FORTRAN. . .)
combinat	Funcions combinatòries
combstruct	Estructures de combinatòria
context	Generació de menus contextuals
CurveFitting	Interpolació
DEtools	Eines per a les equacions diferencials
diffalg	Àlgebra diferencial
difforms	Formes diferencials
Domains	Creació de dominis de computació
ExternalCalling	Enllaços a funcions externes
finance	Funcions de finances
GaussInt	Enters de Gauss
genfunc	Generar funcions racionals
geom3d	Geometria Euclidea en dimensió 3
geometry	Geometria Euclidea
Groebner	Càlculs amb bases de Groebner basis
group	Grups de permutacions i grups finitament presentats
inttrans	Transformacions integrals
liesymm	Implementació del mètode de Harrison-Estabrook
linalg	Àlgebra Lineal basada en estructures `array`
LinearAlgebra	Àlgebra Lineal basada en estructures `rtable`
LinearFunctionalSystems	Troba solucions d'equacions funcionals lineals
LinearOperators	Troba solucions d'equacions funcionals lineals
ListTools	Eines per a manipular llistes
LREtools	Eines per a manilupar relacions de recurència lineal
MathML	Importa i exporta expressions entre Maple i MathML
Matlab	Enllaç amb el programa Matlab
networks	Eines per estudiar grafs
numapprox	Eines per a aproximacions numèriques
numtheory	Eines per a teoria de nombres
Ore_algebra	Càlculs bàsics amb àlgebres d'operadors lineals
OrthogonalSeries	Eines per a treballar amb series de polinomis ortogonals
orthopoly	Polinomis ortogonals
padic	Nombres p-adics
PDEtools	Eines per a trobar solucions de equacions en derivades parcials
plots	Paquet per a gràfics
plottools	Objectes gràfics bàsics
PolynomialTools	Eines per a treballar amb polinomis
polytools	Eines per a treballar amb polinomis
powseries	Eines per a treballar amb sèries de potències formals
process	(Unix)-multiprocés

RandomTools	Eines per a treballar amb nombres aleatoris
RationalNormalForms	Formes racionals normals com a base per a construir representacions minimals i descomposicions de termes hipergeomètrics
RealDomain	Funcions definides sobre els nombres reals
simplex	Mètode del símplex
Sockets	Eines per a treballar en xarxa amb el Maple
SolveTools	Eines per a solucionar equacions
Spread	Fulls de càlcul
stats	Estadística
StringTools	Eines per a manipular cadenes
student	Paquet amb funcions bàsiques
sumtools	Eines per a manipular sumes definides i indefinides
tensor	Càlcul tensorial
Units	Conté unitats ja definides i funcions per a fer els canvis
XMLTools	Eines per a utilitzar documents XML

Podeu obtenir més informació de cada paquet utilitzant l'ajuda: `?paquet`, on `paquet` vol dir de la llista anterior.