

Matemàtiques amb ordinador.

Curs pràctic de Maple

Temes variats

Curs 2003–04

Índex

III Temes variats	89
12 Gràfics “avançats”	89
12.1 Gràfics al pla d’expressions paramètriques	89
12.1.1 Coordenades polars	89
12.2 Gràfics al pla definits implícitament	90
12.3 Opcions de la comanda <code>plot</code>	90
12.4 Gràfics a l’espai	90
12.4.1 Gràfics paramètrics	91
12.4.2 Corbes de nivell	92
12.4.3 Superfícies definides implícitament	92
12.4.4 Opcions dels gràfics a l’espai	93
12.5 Animació de gràfics	93
12.6 Comandes del paquet <code>plots</code>	95
13 Les eines bàsiques per al càlcul infinitesimal	97
13.1 Límits	97
13.2 Derivades	98
13.3 Integrals	101
13.4 Polinomis i sèries de Taylor	101
14 Equacions diferencials	105
14.1 Gràfics d’equacions diferencials	107
14.2 Solucions numèriques d’equacions diferencials	108
15 Interpolació i ajust de dades	111
15.1 Aproximació de funcions, polinomi interpolador	111
15.2 Spline	111
15.3 Interpolació racional	112
15.4 Exercicis d’interpolació	113

15.5	Ajust a les dades	114
15.5.1	Llei de Hooke	114
15.5.2	El mètode dels mínims quadrats	115
15.5.3	Relacions no polinòmiques	118

Part III

Temes variats

12 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.

12.1 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 12.1

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

12.1.1 Coordenades polars

Per a determinar corbes del pla s'utilitzen sovint les coordenades polars ($r =$ distància d'un punt a l'origen, $\theta =$ 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.

12.2 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 dona un gràfic que passi per $(0, 0)$ sense una mica més d'ajuda.

12.3 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:

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

Taula 1: 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.

12.4 Gràfics a l'espai

La comanda bàsica per 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 12.1

```
> 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.

12.4.1 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 12.2

```
> 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 12.2

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 12.3

```
> spacecurve([3*cos(v),3*sin(v),v],v=-2*Pi..2*Pi);
```

fa el gràfic d'una hèlix.

12.4.2 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 12.4

```
> 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 12.5

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

12.4.3 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 12.6

```
> 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 12.3

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

12.4.4 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:

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

Taula 2: Opcions de la comanda `plot3d()`

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

12.5 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 12.7

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 12.8

```
> 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 12.4

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 (unes quantes vegades).

No sempre que es vol fer una animació l'estructura de la comanda `animate` ens permet fer de forma satisfactòria la feina. Per a aquests casos cal tenir en compte que considerant la opció `insequence=true` en una comanda `display` (que presenta diferents dibuixos a l'hora) s'obté també una animació. En l'exemple següent podeu veure un punt que va apareixent en punts aleatoris del pla amb les dues coordenades entre 0 i 2.

Exemple 12.9

```
> randomize();
> punts:= [seq(
> plot(
> [ [ rand(100)()/50,rand(100)()/50 ] ],
> style=point,symbol=circle
> )
> ,i=1..20)];
> display(punts,insequence=true);%
```


12.6 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`).

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

Taula 3: Comandes del paquet **plots**

13 Les eines bàsiques per al càlcul infinitesimal

Encara que algunes de les funcions d'aquesta secció ja han sortit en algun moment, fem un repàs per les funcions bàsiques del càlcul infinitesimal (límits, derivades, integrals...).

13.1 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 13.1

```
> 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 13.2

Si `n` és un nombre natural i voleu calcular $\lim_{n \rightarrow \infty} \sin(n\pi)$ (que dona 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 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 13.3

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 pot 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 13.1

Utilitzeu `limit()` per a conèixer els límits de les successions:

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

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

c) $x_n = \frac{1}{n+1} + \frac{1}{n+2} + \dots + \frac{1}{2n}$.

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

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.

13.2 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 13.4

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 podem 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 13.5

```
> 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 13.6

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 13.7

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);

```

obtidrem tres funcions g_1 , g_2 , g_3 que són les derivades de f respecte x , y , b respectivament.

Exercici 13.2

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

- La funció h_1 derivada de h respecte x .
- La funció h_2 derivada de h respecte y .
- Els valors de h_1 i h_2 per a $x=1$ i $y=\pi/4$.
- La funció h_{11} 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);
```

13.3 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 13.8

```
> 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 13.9

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

13.4 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 13.10

```
> 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 13.11

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

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

Exercici 13.3

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 13.4

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 13.5

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 $\frac{f(x) - p(x)}{x^7}$ per a $x = \pm \frac{1}{2^n}$ (amb n des de 1 fins a 20 i amb 25 decimals com a mínim).
Quin sembla que serà el límit $\lim_{x \rightarrow 0} \frac{f(x) - p(x)}{x^7}$?

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 dona 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 13.12

```
> 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 13.6

Donada $f(x) = \arctan(x)$ determineu les expressions de les derivades successives $f^{(n)}(x)$ i avalueu-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 13.7

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 13.8

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 $f(x) = \frac{1}{(2x+1)^4}$ al voltant del punt $x = 2$ fins al grau 7.

Una variant d'aquest exercici seria fer una animació (recordeu `insequence=true` de la comanda `display`) d'aquests mateixos gràfics fent que en cada un dels fotogrames aparegui el gràfic de la funció i el de un dels polinomis de Taylor.

Exercici 13.9

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!

14 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 14.1

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 14.1

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

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

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

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

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

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

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

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 dona la comanda `dsolve()` són correctes.

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

Exercici 14.2

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 14.2

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 14.3

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 compondre n cops la funció f amb ella mateixa (en l'exemple `(D@@2)(y)` és sinònim de `D(D(y))`).

Exercici 14.3

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 14.4

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 14.4

- Feu un gràfic de la solució de l'equació diferencial de l'exemple anterior que passa pel punt (3, 5).
- 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 14.5

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`).

14.1 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 14.6

```
> 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 14.7

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`.

14.2 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:

Exemple 14.8

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

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 14.9

```
> 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.

15 Interpolació i ajust de dades

15.1 Aproximació de funcions, polinomi interpolador

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

Exercici 15.1

Definiu una llista `l` amb 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 $(x_1, y_1), \dots, (x_n, y_n)$ 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 15.1

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 15.2

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.

15.2 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 15.2

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

```
> l1:=[[0,0],[1,1],[2,4]]:
> Spline(l1,x,degree=1);
```

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

Exercici 15.3

Dibuixeu tres gràfiques amb els “*splines*” de graus 1, 2 i 3 de la llista de punts de l'exercici 15.1.

15.3 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 15.3

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`.

15.4 Exercicis d'interpolació

Exercici 15.4

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 15.5

Considereu la llista de punts:

```
l1i1:= [[-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 15.6

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 15.7

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

Exercici 15.8

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).

15.5 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, \dots, a_d que haurem de determinar per garantir un *bon* ajust. Naturalment, a més de disposar d'observacions, $(x_1, y_1), \dots, (x_n, y_n)$, 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*.

15.5.1 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 15.9

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 15.10

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.

15.5.2 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, \dots, a_d de la funció.

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

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

és mínima, és a dir, la suma dels desajustos quadràtics entre els valors que s'han observat de les y_i i els valors $f(x_i; a_1, \dots, a_d)$ 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 15.4

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 15.11

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 $f(x) = ax + b$ i feu un gràfic de la situació).

Exercici 15.12

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 15.13

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 15.14

Calculeu els paràmetres de la recta que millor s'ajusta a les dades de l'exemple 15.5.2. 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 15.5.2 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 15.15

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

Exercici 15.16

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 $t = a e^{-14p} + bp + c$, 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.

15.5.3 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, una 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.

Si 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 15.17

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?