

Curs pràctic de Maple

Pràctica 10

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 $\{x(t) = t \cos(2\pi t), y(t) = t \sin(2\pi t)\}$

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, $\theta =$ 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 a 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 dona 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:

<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>

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 apareix 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 dona 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 en el paquet **plots** de la comanda `contourplot3d` 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	

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 `display`, `polarplot`, `implicitplot`). El llistat complet de les funcions que defineix aquest paquet és:

<code>animate</code>	<code>contourplot3d</code>	<code>gradplot</code>	<code>listplot3d</code>
<code>animate3d</code>	<code>coordplot</code>	<code>gradplot3d</code>	<code>loglogplot</code>
<code>animatecurve</code>	<code>coordplot3d</code>	<code>implicitplot</code>	<code>logplot</code>
<code>arrow</code>	<code>cylinderplot</code>	<code>implicitplot3d</code>	<code>matrixplot</code>
<code>changecoords</code>	<code>densityplot</code>	<code>inequal</code>	<code>odeplot</code>
<code>complexplot</code>	<code>display</code>	<code>listcontplot</code>	<code>pareto</code>
<code>complexplot3d</code>	<code>display3d</code>	<code>listcontplot3d</code>	<code>pointplot</code>
<code>conformal</code>	<code>fieldplot</code>	<code>listdensityplot</code>	<code>pointplot3d</code>
<code>contourplot</code>	<code>fieldplot3d</code>	<code>listplot</code>	<code>polarplot</code>

<code>polygonplot</code>	<code>replot</code>	<code>setoptions3d</code>	<code>surfdata</code>
<code>polygonplot3d</code>	<code>rootlocus</code>	<code>spacecurve</code>	<code>textplot</code>
<code>polyhedraplot</code>	<code>semilogplot</code>	<code>sparsematrixplot</code>	<code>textplot3d</code>
<code>polyhedra_supported</code>	<code>setoptions</code>	<code>sphereplot</code>	<code>tubeplot</code>

En l'ajuda de Maple podreu trobar què fa i com s'utilitza cada una d'elles (`?plots`).