

INGENIERÍA TÉCNICA INDUSTRIAL - Esp. Mecánica

FUNDAMENTOS MATEMÁTICOS DE LA INGENIERÍA**Práctica nº 4: Representación de funciones de una variable**

RESUMEN: Abordamos en esta práctica el tratamiento con MAPLE de la representación de funciones en una variable. Más concretamente estudiaremos nuevas opciones, centrándonos en el estudio de la representación de funciones en paramétricas y en implícitas.

COMANDOS MÁS IMPORTANTES:**implicitplot****- Introducción y recordatorio**

Recordamos que se pueden representar en unos mismos ejes varias gráficas a la vez. Por ejemplo dos funciones, o bien, una función y una serie de puntos. El primer caso se ha estudiado bastante y lo realizamos por medio de la orden es:

```
plot({función1,función2}, x=x0..x1, y=y0..y1, title=`título de la
gráfica`, color=[color 1,color 2], thickness=[grosor 1,grosor
2], linestyle=[tipo1,tipo 2], discont=true)
```

En el segundo caso estaríamos haciendo dos gráficas con diferentes formatos por lo que hay que recurrir a la opción del display que se encuentra en el paquete plots para visualizarlas en unos mismos ejes coordenados. Para hacer una gráfica de este tipo, primero se carga el paquete utilizando la orden

with(plots)

A continuación se definen las gráficas que se quieren dibujar

```
Grafica1:= plot(f1(x), x=a..b, opciones);
```

```
Grafica2:=plot(f2(x), x=a..b, opciones);
```

Es aconsejable terminar estas órdenes con dos puntos. Si se termina con punto y coma, el resultado será un listado de puntos. Finalmente se utiliza la orden **display** para superponer los dos gráficos.

```
display(Grafica1,Grafica2);
```

En la ayuda del plot se pueden ver diversas opciones. A continuación citaremos dos que no se han visto:

axes=framed/boxed/normal/none Especifica el tipo de ejes.

labels=[x,y] Page 1 Especifica etiquetas para los ejes

Ejemplo: Dibujar en unos mismos ejes las funciones $f(x) = \sin(x)$ y $g(x) = x$.

Solución 1:

```
[ > plot({sin(x), x}, x=-20..20, labels=[etiqueta1, etiqueta2]);
```

Solución 2:

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

```
[ > grafica1:=plot(sin(x), x=-6*Pi..6*Pi) :
```

```
[ > grafica2:=plot(x,x=-20..20) :
```

```
[ > display({grafica1,grafica2}, style=point) ;
```

Si no se va a usar ninguna opción se puede hacer de la siguiente forma

```
[ > display(grafica1,grafica2) ;
```

- Dibujo de un conjunto de puntos y poligonales

Una lista en Maple es una secuencia de números o expresiones separadas por comas y encerradas entre corchetes. Los puntos en Maple son listas de elementos (las coordenadas del punto). Así por ejemplo, el punto de coordenadas $x=1, y=1$, en Maple se escribe $[1,1]$.

Para la representación gráfica de puntos, los argumentos obligatorios en la orden plot son: la lista de puntos especificando sus coordenadas (entre corchetes) y la expresión `style=point`.

```
plot([[a1,b1],[a2,b2],[a3,b3]], x=a..b, y=c..d,  
      style=point)
```

En este caso no es necesario poner la opción del rango de los ejes.

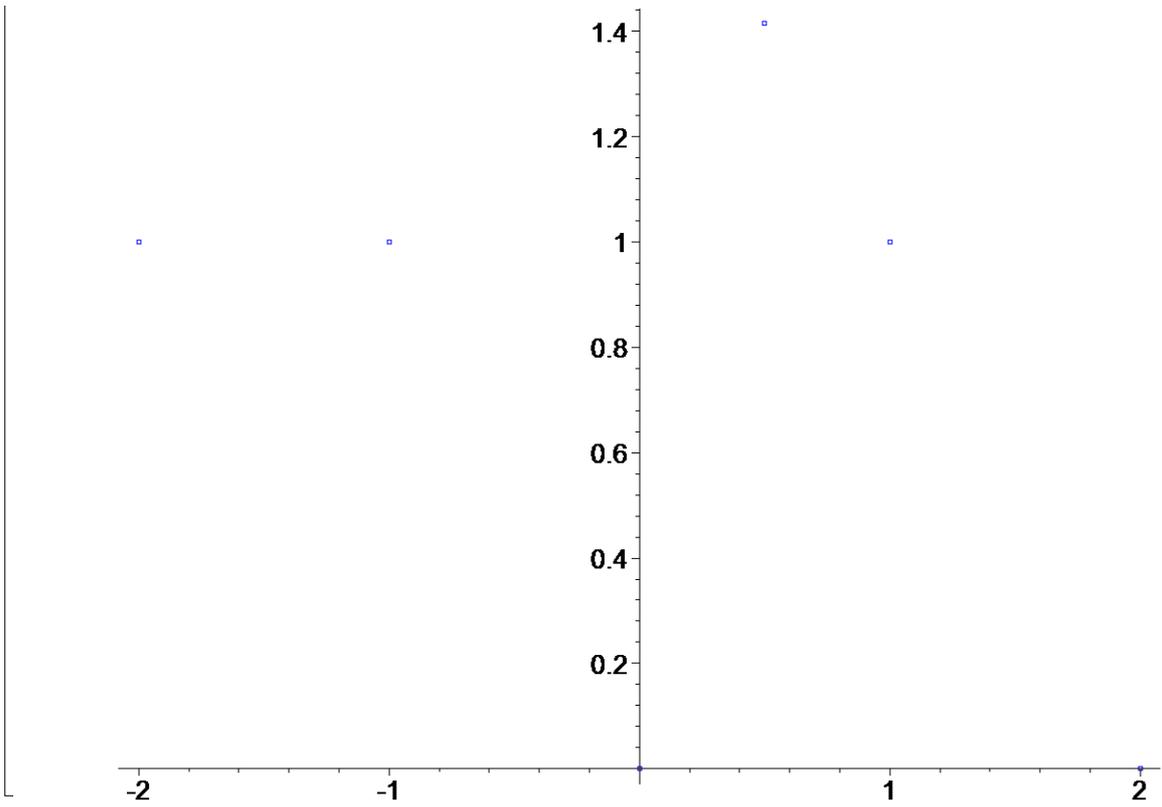
Se puede elegir el símbolo a utilizar para la representación de los puntos mediante la opción `symbol=tipo de símbolo`. Entre los tipos de símbolos están los siguientes: `box`, `cross`, `circle` y `diamond`.

El omitir el estilo en la orden anterior produce la poligonal que une dichos puntos, esto es la unión de segmentos que une los puntos.

Ejemplo: Representar los puntos $[0,0], [1,1],[2,0],[-2,1],[-1,1], [0.5,\sqrt{2}]$.

Solución:

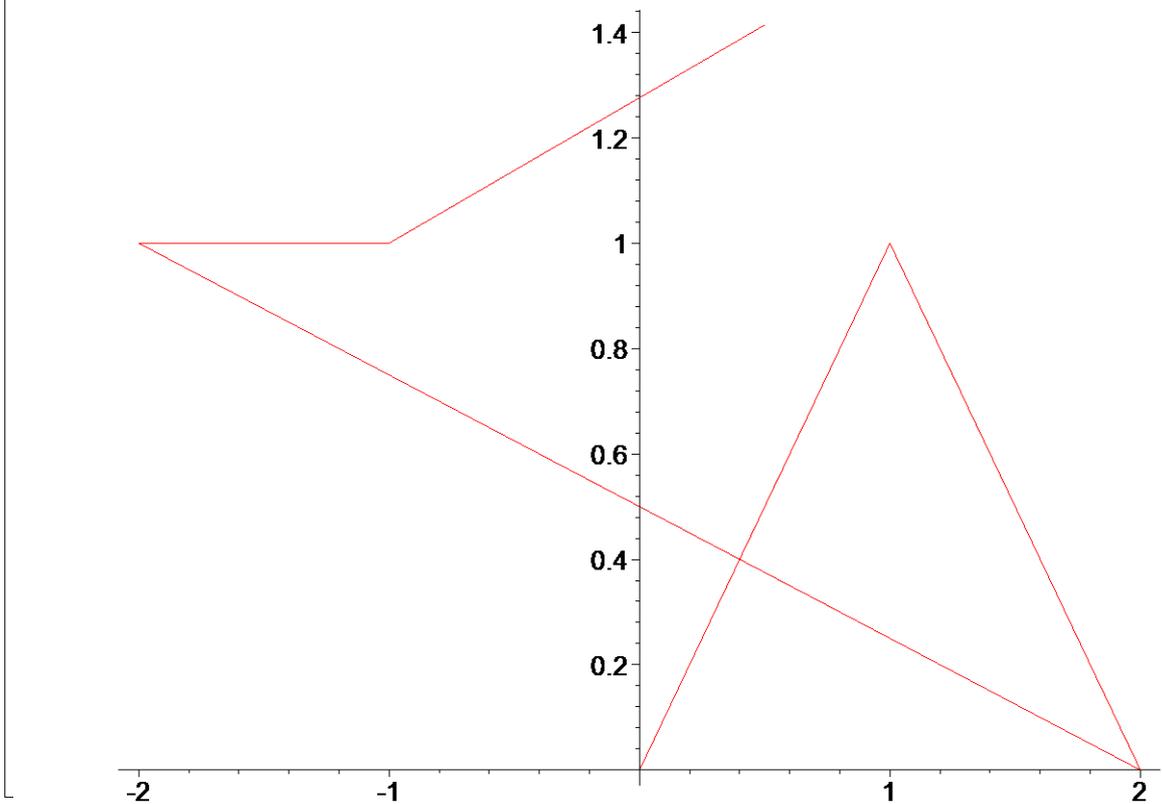
```
[ > plot([[0,0],[1,1],[2,0],[-2,1],[-1,1],[0.5,sqrt(2)]], style=point,  
      symbol=box,color=blue) ;
```



Ejemplo: Representar la poligonal que une los puntos $[0,0]$, $[1,1]$, $[2,0]$, $[-2,1]$, $[-1,1]$, $[0.5, \sqrt{2}]$.

Solución:

```
> plot([[0,0],[1,1],[2,0],[-2,1],[-1,1],[0.5,sqrt(2)]],color=red);
```



- Curvas en paramétricas

Con el fin de motivar la representación en paramétricas, veamos un ejemplo. Consideremos el camino que sigue un objeto cuando lo lanzamos hacia arriba con un determinado ángulo y que supongamos que viene dado por el camino parabólico dado por

$$(1) \quad y = -\frac{x^2}{4} + x$$

Si lo dibujamos tenemos

```
[ > plot(-x^2/4+x, x=0..4, y=0..1.5);
```

Sin embargo esta ecuación no nos cuenta toda la historia. Es decir nos dice donde está el objeto pero no dice cuando está el objeto en un punto (x,y) en un tiempo t . Para ello se introduce una variable llamada parámetro. Si escribimos los puntos (x,y) en función de un parámetro t por ejemplo de la forma

$$x = 2t$$

$$y = -t^2 + 2t$$

obtenemos las ecuaciones paramétricas de la función dada en (1).

En general, se dice que una curva viene dada en forma paramétrica si está dada en la forma

$$x = x(t)$$

$$y = y(t)$$

donde t es un parámetro que varía en algún intervalo, por ejemplo $[a,b]$, o incluso en todo los números reales.

Para representarla usamos el comando

plot([x(t), y(t), t=a..b], opciones)

El hecho de que la lista conste de tres elementos y el tercero de ellos sea el rango, hace que el programa reconozca una gráfica en paramétricas en lugar de un gráfico múltiple. Una opción útil es la de

scaling= constrained

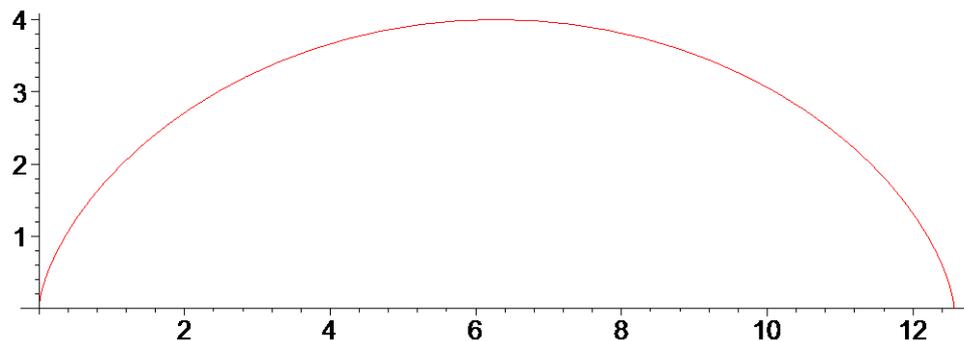
Esta opción lo que nos permite es que el dibujo no lo deforme, es decir que la escala sea la misma 1:1. Maple usa por defecto el unconstrained.

Ejemplo: Representar la curva en paramétricas:

$$x = 2(t - \sin t)$$

$$y = 2(1 - \cos t)$$

```
[ > plot([2*(t-sin(t)), 2*(1-cos(t)), t=0..2*Pi], scaling=constrained);
```



Pueden combinarse en un gráfico múltiple curvas en cartesianas y en paramétricas.

Ejemplo: Representar en unos mismos ejes las funciones $x(t) = 3 \cos(t)$, $y(t) = \sin(t)$, $f1(x)=x$ y $f2(x)=-x$.

Solución:

```
> plot([3*cos(t), sin(t),
        t=0..2*Pi], x, -x), x=-4..4, scaling=constrained, color=[red,black
        ,blue]);
```

A veces la representación en paramétricas no permite hacer el dibujo en todo los números reales, por ejemplo una recta, para evitarlo hay que limitar el rango del parámetro.

Veamos algunos ejemplos importantes de funciones en paramétricas.

Ejemplo: Representar la **recta** que pasa por el punto $P=(8,3)$ y tiene como vector director $v=(1,-1)$.

Solución: La ecuación de la recta es $x=8+t$, $y=3-t$.

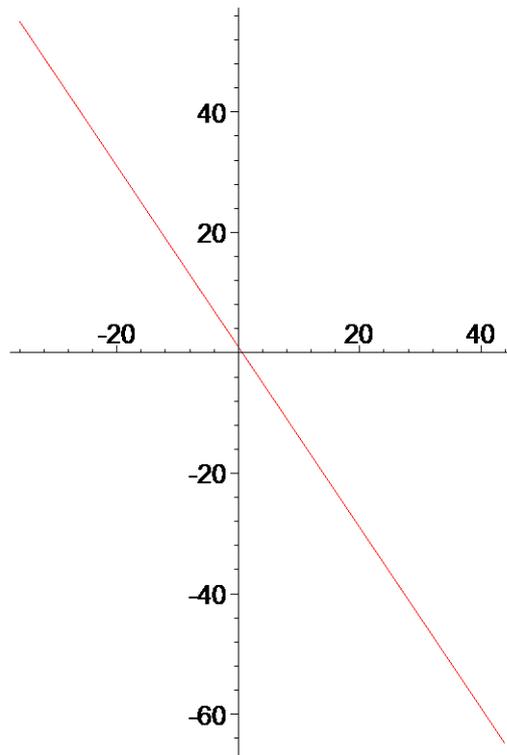
```
> plot([8+t,3-t,t=-infinity..infinity], title=`Gráfica
        utilizando todo los números reales`);
> plot([8+t,3-t,t=-20..20], title=`Gráfica de la recta que pasa
        por P=(8,3) con dirección v=(1,-1)`, scaling=constrained);
```

Ejemplo: Representar la **recta** que pasa por el punto $P=(4,-5)$ y tiene como vector director $v=(2,-3)$.

Solución: La ecuación de la recta es $x=4+2t$, $y=-5-3t$.

```
> plot([4+2*t,-5-3*t,t=-20..20], title=`Gráfica de la recta que
        pasa por P=(4,-5) con dirección
        v=(2,-3)`, scaling=constrained);
```

Gráfica de la recta que pasa por P=(4,-5) con dirección v=(2,-3)



Ejemplo: Representa en paramétricas la ecuación de la **circunferencia** de centro (1,-1) y radio $\sqrt{2}$

Solución: La ecuación de la circunferencia es

$$x = 1 + \sqrt{2} \cos(t)$$

$$y = -1 + \sqrt{2} \sin(t)$$

donde t es un parámetro en un intervalo de longitud 2π , por ejemplo $[0, 2\pi]$, $[-\pi, \pi]$ etc...

Si hacemos el dibujo sin la opción `scaling=constrained`, nos sale bastante deformado, de hecho aparenta una elipse.

```
[ > plot([1+sqrt(2)*cos(t), -1+sqrt(2)*sin(t), t=-Pi..Pi],
  title=`Representación de una circunferencia de centro (1,-1)
  y radio sqrt(2), sin la opción scaling` );
[ > plot([1+sqrt(2)*cos(t), -1+sqrt(2)*sin(t), t=-Pi..Pi],
  scaling=constrained, title=`Representación de una
  circunferencia de centro (1,-1) y radio sqrt(2)` );
```

¿Qué ocurre si cambiamos el rango del parámetro t, por ejemplo $[0, \pi]$, $\left[0, \frac{\pi}{4}\right]$?

```
[ > plot([1+sqrt(2)*cos(t), -1+sqrt(2)*sin(t), t=0..Pi],
  scaling=constrained);
[ > plot([1+sqrt(2)*cos(t), -1+sqrt(2)*sin(t), t=0..Pi/4],
  scaling=constrained);
```

Ejemplo: Dibujar la **elipse** de centro (0,0) con semieje X igual a 4 y semieje Y igual a 2.

Solución: La ecuación de la elipse es:

$$x = 4 \cos(t)$$

$$y = 2 \sin(t)$$

donde t se mueve en cualquier intervalo de longitud 2π . Igual que en la circunferencia, para que no salga deformada se usa la opción `constrained`:

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

Ejemplo: El **cardioides** es una curva de una forma especial, que en general tiene por ecuaciones

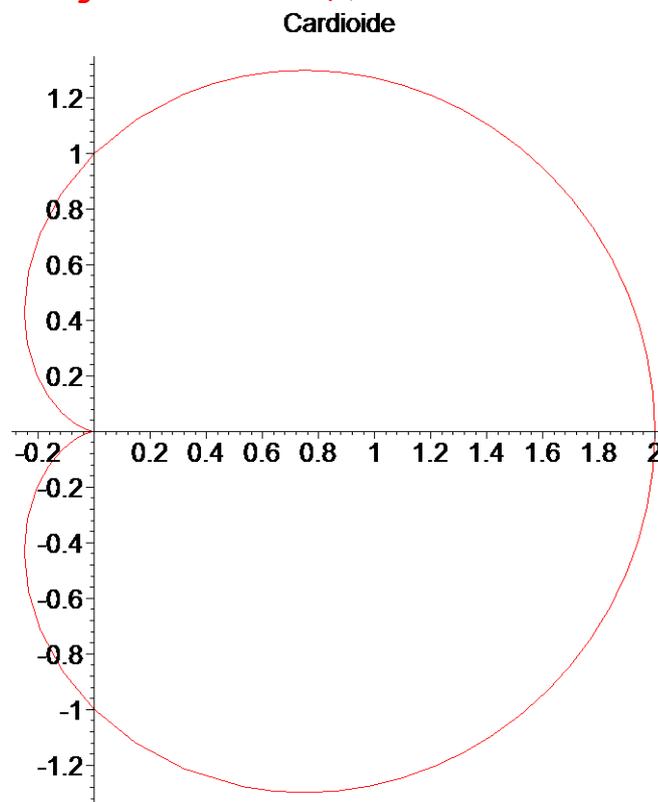
$$x(t) = (a + b \cos(t)) \cos(t)$$

$$y(t) = (a + b \cos(t)) \sin(t)$$

donde a y b son valores reales fijos. ¿En qué intervalo de la variable t queda definida completamente esta curva?

Dibujar el cardioides con $a=b=1$. Modificar los valores de a y b y representarlo

```
[ > plot([(1+cos(t))*cos(t), (1+cos(t))*sin(t), t=0..2*Pi], title='Cardioides', scaling=constrained);
```



```
[ > plot([(cos(t))*cos(t), (2+cos(t))*sin(t), t=0..2*Pi], title='Cardioides', scaling=constrained);
```

```
[ > plot([(2+cos(t))*cos(t), (cos(t))*sin(t), t=0..2*Pi], title='Cardioides', scaling=constrained);
```

Ejemplo: Se llama **cicloides** a toda curva plana que se obtiene físicamente por la trayectoria de un punto de una circunferencia que, sin deslizarse rueda por una recta horizontal. La ecuación en paramétricas de una cicloides es:

$$x(t) = a(t - \sin(t))$$

$$y(t) = a(1 - \cos(t))$$

Representar la cicloides con parámetro $a=1$ y 3 .

```
[ > plot([t-sin(t), 1-cos(t), t=0..6*Pi], scaling=constrained);
```

```
> plot([3*(t-sin(t)), 3*(1-cos(t)), t=0..6*Pi], scaling=constrained);
```

Curvas en implícitas

Otra forma en que puede venir dada una curva es en implícitas. En este caso las variables cartesianas x e y vienen relacionadas por una ecuación donde no podemos expresar o despejar una en función de la otra, esto es, tenemos una ecuación de la forma $f(x, y) = 0$.

Para representarla necesitamos primero cargar el paquete de gráficos con el comando

with(plots)

y la gráfica la obtenemos con la orden

implicitplot ($f(x, y) = 0$, $x = a..b$, $y = c..d$, *opciones*)

donde $[a, b]$ y $[c, d]$ son el rango de las variables x e y , respectivamente. En este tipo de representaciones es necesario poner el rango de las dos variables x e y .

Suele ser útil la opción de

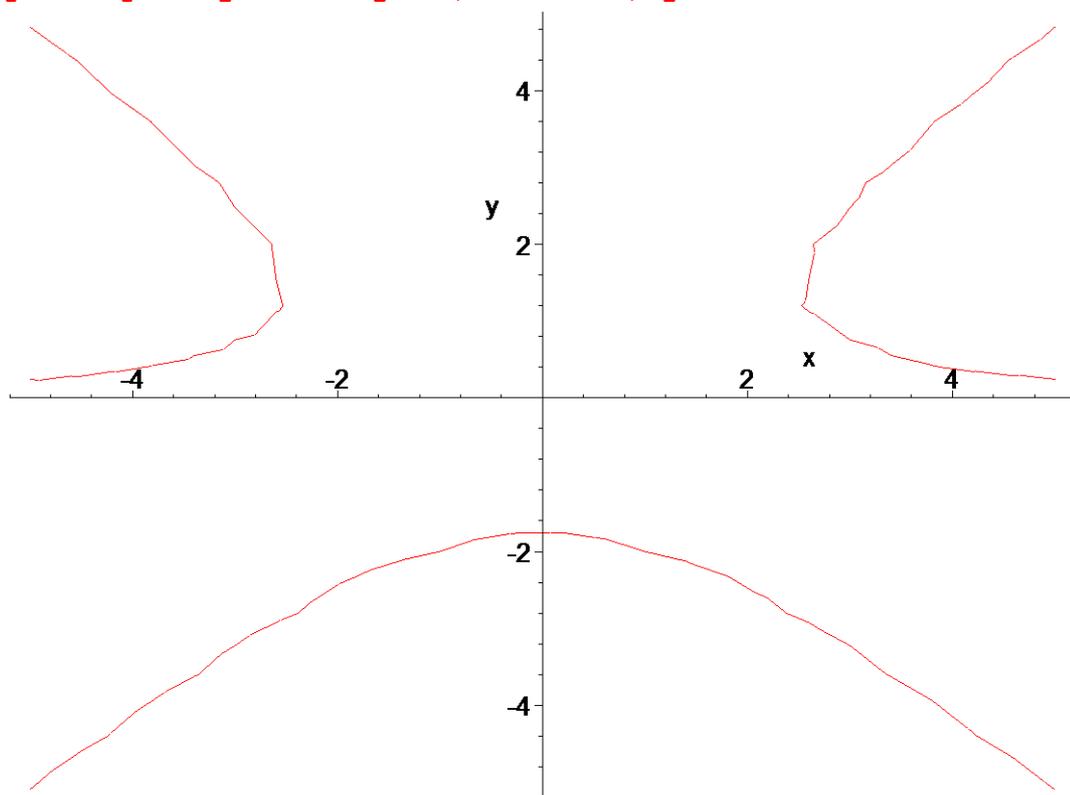
numpoints = *números de puntos a usar*

cuando la salida o gráfica mostrada por implicitplot no es satisfactoria.

Ejemplo: Dibujar la curva de ecuación $y^3 - x^2 y = -6$

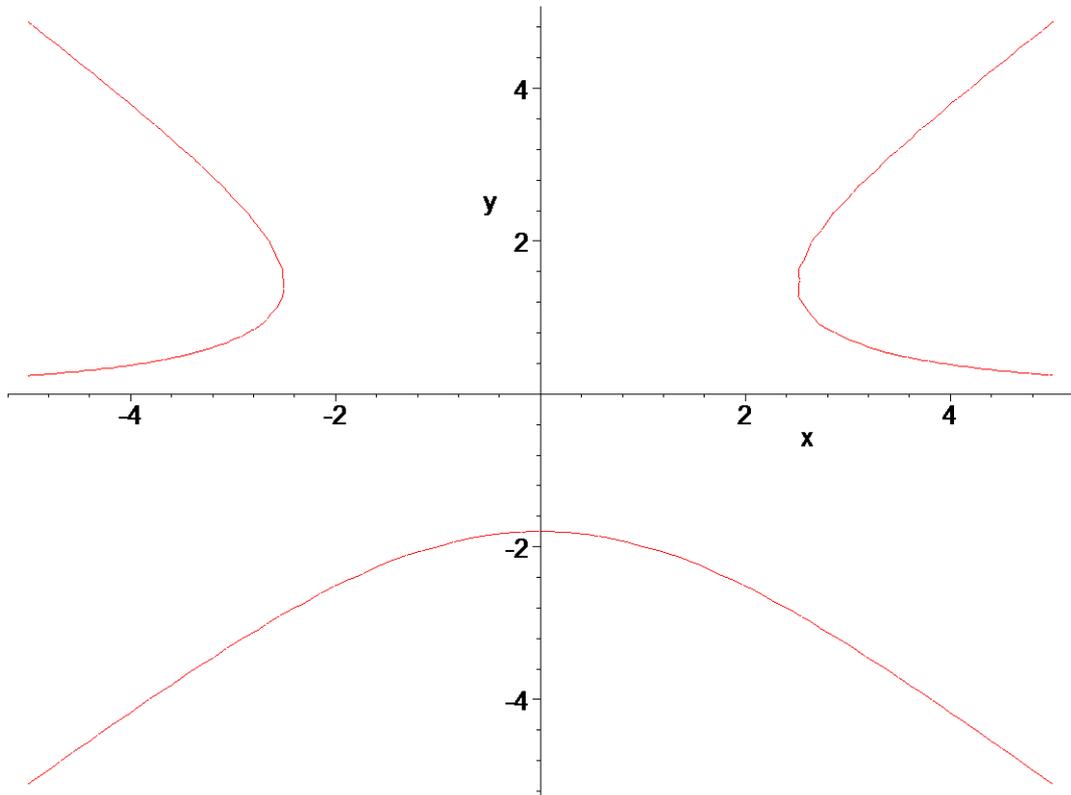
```
> with(plots):
```

```
> implicitplot(y^3-x^2*y=-6, x=-5..5, y=-10..10);
```



En este caso utilizaremos el implicitplot con la opción de numpoints

```
> implicitplot(y^3-x^2*y=-6, x=-5..5,
y=-10..10,numpoints=3000);
```



Las curvas implícitas más conocidas son las cónicas: **la circunferencia, la elipse y la hipérbola**.

Ejemplo: Representa en implícitas la ecuación de la **circunferencia** de centro (1,-1) y radio $\sqrt{2}$. La ecuación implícita de esta circunferencia es:

$$(x - 1)^2 + (y + 1)^2 = 2$$

```
> implicitplot((x-1)^2+(y+1)^2=2,x=-5..5,y=-5..5,scaling=constrained,numpoints=5000);
```

Ejemplo: Dibujar la **elipse** de centro (0,0) con semieje X igual a 4 y semieje Y igual a 2. La ecuación implícita de esta elipse es

$$\frac{x^2}{4^2} + \frac{y^2}{2^2} = 1$$

```
> implicitplot(x^2/16+y^2/4=1,x=-5..5,y=-5..5,scaling=constrained,numpoints=5000);
```

Ejemplo: Dibujar la **hipérbola** de centro (1,1) y semiejes X e Y 4 y 5 respectivamente. La ecuación implícita resulta

$$\frac{(x - 1)^2}{4^2} - \frac{(y - 1)^2}{5^2} = 1$$

```
[ > implicitplot((x-1)^2/16-(y-1)^2/25=1,x=-10..10,y=-10..10,scaling=constrained,numpoints=5000);
```

Ejemplos: Dibujar en un mismo gráfico: (a) la circunferencia de centro (2,0) y radio 3 en paramétricas; (b) la elipse de centro (0,2) y semiejes X e Y dados por 6 y 3 respectivamente y (c) la hipérbola de centro (0,2) y semiejes X e Y 6 y 3 respectivamente, siendo estas dos últimas en implícitas.

Solución:

```
[ > a:=plot([2+3*cos(t),3*sin(t), t=0..2*Pi],
  scaling=constrained, color=blue):
[ > b:=implicitplot((x-2)^2/6^2+y^2/3^2=1, x=-20..20,y=-10..10,
  numpoints=5000,color=red):
[ > c:=implicitplot((x-2)^2/6^2-y^2/3^2=1, x=-20..20,y=-10..10,
  numpoints=5000,color=green):
[ > display(a,b,c, title=`Circunferencia en azul, elipse en rojo
  y hipérbola en verde`);
```

Ejemplo: Dibujar la siguiente curva $y^2(y^2 - b^2) - x^2(x^2 - a^2) = 0$, con los valores de (1) $a=1$ y $b=0.85$, (2) $a=0.2$ y $b=0.3$ Llamada curva del diablo.

Solución:

```
[ > implicitplot(y^2*(y^2-0.85^2)-x^2*(x^2-1)=0,x=-1.5..1.5,
  y=-1.5..1.5, numpoints=6000);
[ > implicitplot(y^2*(y^2-0.3^2)-x^2*(x^2-0.2^2)=0,x=-1..1,
  y=-1..1, numpoints=6000);
```