

# Tutorial básico de Maple

## ▼ Introducción

Maple es una herramienta que permite combinar texto con expresiones matemáticas y operar tales expresiones matemáticas.

Podemos encontrar información sobre este programa en la URL <http://www.maplesoft.com>.

En nuestra Universidad, el centro de cálculo ha confeccionado un manual (en castellano) que está disponible en el URL <http://www.sisofit.ucm.es/Manuales/MAPLE.PDF>. A este manual solo se puede acceder conectándonos desde la red de la UCM.

Una primera visión de Maple la podemos obtener con la barra de herramientas superior, ejecutando:

- Ayuda > Ayuda de Maple
- Ayuda > Tour por Maple

En la opción de Herramientas también son útiles para aprender el manejo:

- Herramientas > Asistentes
- Herramientas > Tutoriales

Este tutorial está pensado con dos objetivos:

- 1.- Introducir al alumno en las funciones básicas de Maple, de forma que pueda servir en cualquier momento como repaso antes de abordar un informe.
- 2.- Mostrar algunas herramientas especialmente útiles en el manejo y solución de las Ecuaciones Diferenciales

## ▼ Maple como tratamiento de textos

### Textos

Como tratamiento de textos, Maple tiene las funciones básicas: cambiar tipo de letra, tamaño, negrita, cursiva, subrayado, alineación, enumeración, color de la letra, etc. También permite la corrección ortográfica del texto mediante el botón F7 o en Herramientas > Corrector Ortográfico.

Lamentablemente para los castellanoparlantes, el diccionario que usa Maple es el inglés. Sin embargo, lo que sí permite Maple es utilizar un diccionario de usuario al cual podemos ir añadiendo palabras conforme nos aparezcan en el texto. Estos diccionarios deben estar en formato txt. Existen muchos lematarios (listas de palabras) muy completos en la red. Una última opción, más fácil, aunque larga, es simplemente añadir las palabras que aparezcan en nuestro documento, en relativamente poco tiempo podemos tener un diccionario razonablemente efectivo.

## Expresiones matemáticas

Maple también permite escribir las expresiones matemáticas como:

$$\int \frac{dx}{\sqrt{1 - \frac{v^2}{c^2}}} = \int \frac{v}{\sqrt{1 - \frac{v^2}{c^2}}} dv$$

Para escribir estas expresiones utilizamos, en la columna de la izquierda, la pestaña *Expression*

En la fila superior izquierda hay dos pestañas: *Texto* y *Matemáticas*. Son las dos formas básicas en las que se pueden introducir caracteres en Maple.

Hasta aquí hemos introducido el texto y las expresiones matemáticas en modo *texto*.

Si cuando introducimos una expresión matemática, al empezar, pinchamos la pestaña Matemática, después de escribirla podemos hacer algunas evaluaciones con ella.

Para evaluarla solo tenemos que utilizar el botón derecho del ratón sobre la expresión. Por ejemplo, si sobre la siguiente expresión:

$$x^2 + 2 \cdot x - 1 = 0$$

pinchamos con el botón derecho y seleccionamos las pestañas *solve>solve*, encontraremos los valores de  $x$  que verifican la igualdad.

Al realizar esta acción habremos encontrado la forma en la que Maple adquiere su máxima potencia, que es en el modo de comandos.

## Modo comando y modo texto

Para abrir el modo de comandos podemos pinchar el icono: [ $\rightarrow$ ] de la barra de herramientas superior. Junto a este icono hay otro con una T que permite pasar a modo texto.

## Imprimiendo

Para ver como va a quedar un documento conviene ejecutar: Archivo > Vista preliminar

Para imprimir: Archivo > Imprimir

Prácticamente el resto de este tutorial está dedicado al modo línea de comando

## ▼ Línea de comando

Los comandos en Maple se escriben en una *línea de comando*. A continuación se pone un ; (al que se llama *terminador*) y al pulsar el botón *Intro* se ejecutan. Se pueden poner varias operaciones en una misma línea, separadas por punto y coma y al ejecutarse aparecerán en una columna:

```
> 1+1;3-5;26*88;3489*23256/51;
      2
      -2
      2288
      1590984
```

En algún caso puede ser interesante que el resultado de alguna operación no aparezca. Para ello, en vez de terminar el comando con ; lo terminamos con : Esto se utilizará más adelante.

Conviene insertar las líneas de comando entre medias del texto para poder volver luego al texto en el mismo punto de indexación. Si pulsamos iconos [ > y T alternativamente el resultado es el siguiente:

```
>
```

```
>
```

pues Maple entiende que cada comando y párrafo están al mismo nivel.

La línea de comandos permite introducir los operadores matemáticos en forma de texto o en forma matemática para ello, cuando ejecutamos el icono [ > tenemos que asegurarnos de que está en *Texto* o *Matemática*. En modo texto los comandos se expresan con letras y en modo matemática se utilizan los símbolos matemáticos. Las dos siguientes líneas de comando ejecutan la misma operación (por cierto: observar cómo racionaliza Maple)

```
> sqrt(23/47);
      1
      47 sqrt(1081)
> sqrt(23/47);
      1
      47 sqrt(1081)
```

En modo Matemática no es necesario poner el ; al final.

Distintos operadores, en ambos modos, se expresan así: (el exponente se pone con el " ^ " o con " \*\* " y la raíz con sqrt. Los operadores básicos se pueden encontrar en la columna izquierda, en la pestaña *Expression*) :

```
> 2^10 + ln(23)/sqrt(33)
      1024 + 1/33 ln(23) sqrt(33)
> 2^10+ln(23)/sqrt(33);
      1024 + 1/33 ln(23) sqrt(33)
```

Nosotros utilizaremos una u otra forma de notación según sea más claro, conveniente o simple en cada momento.

Se puede forzar que Maple dé un resultado numérico usando el comando *evalf*

```
> evalf(2^10 + ln(23)/sqrt(33))
```

El símbolo % dentro del *evalf* da el resultado matemático de la última expresión evaluada. ¡¡OJO!! no es de la expresión que está en la línea inmediatamente superior. Si ejecutamos otra cosa después y luego volvemos a esta línea y la ejecutamos, el resultado cambiará.

```
> evalf(%)
      1024.545819
```

## El número e

Las letras especiales se deben utilizar con cuidado. El número e = 2.718281... no se puede expresar simplemente con la letra "e" de nuestro teclado

```
> evalf(e)
      e
```

Sin embargo, si en la columna izquierda, en la pestaña *expression* ejecutamos el símbolo adecuado, podemos obtener:

```
> evalf(e)
      2.718281828
```

El problema de esta "ambigüedad" es poder diferenciar entre estos dos comandos:

```
> e^2
      e^2
> e^2
      e^2
```

Conviene observar que, al ejecutarlos, el primero muestra la letra e en cursiva (aunque nos hemos asegurado de escribirla en vertical), mientras que en el segundo, al ejecutarlo, la letra e se muestra en vertical. Como norma general, cuando Maple entiende que algo es una variable lo pone en cursiva, mientras que cuando entiende que es un valor o constante lo pone en vertical. Si hacemos un *evalf* sobre los dos anteriores comandos obtenemos:

```
> evalf(e^2)
      e^2
> evalf(e^2)
      7.389056099
```

¡Ojo a esta ambigüedad!, ¡¡puede ser responsable de muchos dolores de cabeza!!

Esta ambigüedad desaparece en modo texto:

```

> e^2;
                                     e2
> evalf(%);
                                     e2
> exp(2);
                                     e2
> evalf(%);
3.789056099

```

(Por cierto, si vuelves ahora al primero de estos dos *evalf*(%) podrás observar el "peligro" del parámetro % en este comando)

Esta desambiguación hace que muchos prefieran el modo texto en los comando, pese a ser más farragoso de interpretar.

## Número $\pi$

El número  $\pi$ , sin embargo, es más fácil de evaluar. En modo texto "pi" todo en minúsculas, es la letra griega  $\pi$ , susceptible de tomar un valor. Sin embargo, "Pi" con la primera letra en mayúsculas corresponde al número  $\pi$ .

Para el modo matemática la letra griega  $\pi$  es siempre el número 3.1415926...

```

> 4*(3+Pi); evalf(4*(3+Pi));
                                     12 + 4  $\pi$ 
                                     24.56637062
> 2. $\pi$ .52
                                     50  $\pi$ 
> evalf (%)
157.0796327

```

## Operadores y precisión

En Maple están definidas la mayoría de las funciones básicas del análisis matemático, existiendo definiciones para valores especiales:

```

> sin(5*Pi/3);
                                     -  $\frac{1}{2} \sqrt{3}$ 
> cosh(0);
                                     1
> sec(Pi/4);
                                      $\sqrt{2}$ 
> arcsin(-1);
                                     -  $\frac{1}{2} \pi$ 
> 200!;
78865786736479050355236321393218506229513597768717326329474253324435944996340\
33429203042840119846239041772121389196388302576427902426371050619266249528\
29931113462857270763317237396988943922445621451664240254033291864131227428\

```

```

29485327752424240757390324032125740557956866022603190417032406235170085879\
6178922227896237038973747200000000000000000000000000000000000000000000\
00

```

Podemos especificar la precisión del *evalf*:

```

> evalf(Pi,1000);
3.1415926535897932384626433832795028841971693993751058209749445923078164062862\
08998628034825342117067982148086513282306647093844609550582231725359408128\
48111745028410270193852110555964462294895493038196442881097566593344612847\
56482337867831652712019091456485669234603486104543266482133936072602491412\
73724587006606315588174881520920962829254091715364367892590360011330530548\
82046652138414695194151160943305727036575959195309218611738193261179310511\
85480744623799627495673518857527248912279381830119491298336733624406566430\
86021394946395224737190702179860943702770539217176293176752384674818467669\
40513200056812714526356082778577134275778960917363717872146844090122495343\
01465495853710507922796892589235420199561121290219608640344181598136297747\
71309960518707211349999998372978049951059731732816096318595024459455346908\
30264252230825334468503526193118817101000313783875288658753320838142061717\
76691473035982534904287554687311595628638823537875937519577818577805321712\
26806613001927876611195909216420199

```

## Variables

Maple remite operar con variables sin un valor asignado, de forma "simbólica"

```

> a;
a

```

Para dar un valor a una variable se utiliza el operador de asignación (:=).

```

> a := 4;
a := 4
> a;
4

```

El comando **unassign** elimina asignaciones previas:

```

> unassign('a'); a;
a

```

El comando **restart** borra todas las definiciones y asignaciones realizadas.

```

> restart;

```

## Funciones definidas

Maple tiene definidas las operaciones matemáticas estándar, algunas de las cuales las podemos ver en la columna izquierda, en la pestaña *Expression*.

Una lista de las funciones definidas la podemos encontrar en [initialfunctions](#)

Además de estas funciones están definidos los operadores  $a + b$ ,  $a - b$ ,  $a \cdot b$ ,  $\sqrt{a}$ ,  $\frac{a}{b}$ ,  $a^b$ ,  $\ln(a)$ , etc con ellos podemos expresar las funciones complejas y operar con ellas tal como veremos más adelante.

En RE la funcion gamma de Lorentz es ( $c$  es la velocidad de la luz):

$$> \frac{1}{\sqrt{1-\frac{v^2}{c^2}}} :$$

Esta misma función evaluada para una velocidad igual a la mitad de la velocidad de la luz:

$$\left. > \frac{1}{\sqrt{1-\frac{v^2}{c^2}}} \right|_{v=\frac{c}{2}} \qquad \frac{1}{3} \sqrt{3} \sqrt{4}$$

$$> \textit{evalf}(\%) \qquad 1.154700539$$

En los sumetorios se asignan consecutivamente los valores de las variables, tal como se hace habitualmente

$$> \sum_{n=0}^{\infty} \frac{x^n}{(2\ n) !} \qquad \cosh(\sqrt{x})$$

Existen operador para trabajar con las funciones. Una lista de todos los operadores la tenemos en [index\[function\]](#).

Por ejemplo, el operador *series* hace un desarrollo en serie de potencias para una función

$$\begin{aligned} &> \textit{restart} \\ &> \textit{series}\left(\frac{1}{\sqrt{1-\frac{v^2}{c^2}}}, v=0, 10\right) \\ &\qquad 1 + \frac{1}{2\ c^2}\ v^2 + \frac{3}{8\ c^4}\ v^4 + \frac{5}{16\ c^6}\ v^6 + \frac{35}{128\ c^8}\ v^8 + O(v^{10}) \end{aligned}$$

O el operador *diff* deriva una función

$$\begin{aligned} &> \textit{diff}\left(\frac{1}{\sqrt{1-\frac{v^2}{c^2}}}, v\right) \\ &\qquad \frac{v}{\left(1-\frac{v^2}{c^2}\right)^{3/2}}\ c^2 \end{aligned}$$

### Números complejos

La i, j o i mayúscula: "I" de la columna izquierda, pestaña *Common Symbols*, corresponde a la constante imaginaria  $\sqrt{-1}$ .

De esta forma, podemos hacer operaciones con números imaginarios.

Nuevamente tenemos que tener cuidado con la letras i y j del teclado. La I mayúscula del teclado sí es la constante imaginaria.

$$\begin{aligned} &> (3+4i) \cdot (3-2i) \qquad 17+6\ I \\ &\left[ \begin{aligned} &> \textit{conjugate}(a); \qquad \overline{a} \\ &> a := (1+2i) \qquad a := 1+2\ I \\ &> \textit{conjugate}(a); \qquad 1-2\ I \\ &> \sqrt{3+4\ I} \qquad 2+I \end{aligned} \right. \end{aligned}$$

Cuando asignamos a una variable un valor imaginaro, se está asumiendo que es imaginaria:

$$\begin{aligned} &> a := \sqrt{-4} \qquad a := 2\ I \\ &\left[ \begin{aligned} &> a^3 \qquad -8\ I \end{aligned} \right. \end{aligned}$$

Podemos hacer cálculos con estas variables complejas, aunque solo obtendremos un resultado, para encontrar los demás debemos ser nosotros los que "lo trabajemos"

$$\begin{aligned} &> \ln(a); \textit{evalf}(\ln(a)) \\ &\qquad \ln(2\ I) \\ &\qquad 0.6931471806 + 1.570796327\ I \end{aligned}$$

Números imaginarios, además de en las raices, también aparecen en otras expresiones con números reales:

$$\begin{aligned} &> \ln(-5) \\ &\qquad \ln(5) + I\ \pi \end{aligned}$$

Se puede decidir si una variable es real o compleja (o entera, racional, etc) con el comando *assume*

[>

### Límites y sumatorios

[> **restart;**

[> **limit**( $\frac{1}{x}, x=a$ )

$$\frac{1}{a}$$

O usando los simbolos de la columna izquierda:

[> **lim**  $\ln(x)$   $x \rightarrow 0$

$$-\infty$$

[> **lim**  $e^{x+a}$   $x \rightarrow -\infty$

$$0$$

También resuelve las indeterminaciones:

[> **lim**  $\frac{(x+\epsilon)^2 - x^2}{\epsilon}$   $\epsilon \rightarrow 0$

$$2x$$

[> **lim**  $\frac{\tan(x+\epsilon) - \tan(x)}{\epsilon}$   $\epsilon \rightarrow 0$

$$1 + \tan(x)^2$$

Algunos de estos limites los identifica como funciones de numeros trascendentes como  $e$ ,  $\pi$ , etc O funciones trascnentes como  $\Psi$

[> **lim**  $\left(\frac{3n+2}{3n-2}\right)^{n+3}$   $n \rightarrow \infty$

$$e^{\frac{4}{3}}$$

[> **sum**  $\frac{1}{i^4}$   $i=1$   $\infty$

$$\frac{1}{90} \pi^4$$

[> **sum**  $(-1)^{i+1} \frac{1}{i}$   $i=1$   $\infty$

L

ln(2)

### Operando funciones y ecuaciones

Existen funciones generales de manipulación de expresiones matemáticas. La función **expand** tiene un efecto determinado sobre funciones trigonométricas e hiperbólicas:

[> **expand**(**sin**(2\*x));

$$2 \sin(x) \cos(x)$$

[> **expand**(**cosh**(4\*x));

$$8 \cosh(x)^4 - 8 \cosh(x)^2 + 1$$

La función **simplify** intenta simplificar expresiones, aunque no siempre produce una respuesta unívoca:

[> **simplify**(%);  
**simplify**(**cosh**(4\*x));

$$\frac{8 \cosh(x)^4 - 8 \cosh(x)^2 + 1}{\cosh(4x)}$$

### Definiendo nuevas funciones

Se pueden definir funciones adicionales, con una notación especial:

[> **distancia** := (x,y) -> **abs**(x-y);

$$distancia := (x,y) \rightarrow |x-y|$$

Ahora disponemos de una función nueva, **distancia(x,y)**.

[> **distancia**(3,5); **distancia**(-4,3);

$$\frac{2}{7}$$

Las operaciones básicas las podemos extraer de la tabla de expresiones que tenemos en la columna de la izquierda, en la pestaña *Expression*. Recordar que estas expresiones dependen de si estamos en modo *Texto* o *Matemáticas*. Los dos siguientes ejemplos definen las mismas funciones, pero observa que uno tiene activa la pestaña *texto* y el otro la pestaña *Matemática*.

[> **raiz** := (x) ->  $\sqrt{x}$ ;

$$raiz := x \rightarrow \sqrt{x}$$

[> **raiz** :=(x) -> **sqrt**(x);

$$raiz := x \rightarrow \sqrt{x}$$

En cada modo, las expresiones de la pestaña *expression* se escriben en la forma correcta

### Resolver ecuaciones y sistemas de ecuaciones

Podemos resolver ecuaciones mediante el comando *solve*:

```
> solve(x^2 - 5 x + 6 = 0);
3, 2
> solve(2 y - (x - 1)^2 = 2, y);
1/2 x^2 - x + 3/2
```

Asignarlas a una variable:

```
> solution := solve(x^2 - 5 x + 6 = 0, x);
solution := 3, 2
```

Si tenemos varias soluciones, estas aparecen en forma de array o vector:

```
> solution[1]; solution[2]; solution[3];
3
2
Error, invalid subscript selector
```

Para los sistemas de varias ecuaciones con varias incógnitas, introducimos las ecuaciones en el primer argumento entre corchetes:

```
> solve({3 x + 2 y + 4 z = 5, x + y + z = 4, x - y + z = 1});
{x = 8, y = 3/2, z = -11/2}
```

Si no son lineales, pueden dejar el resultado indicado en función de una ecuación:

```
> solve({3 x + 2 y + 4 z = 5, x + y + z = 4, x^2 - y + z = 1});
{x = -RootOf(_Z^2 + 6 _Z + 1, label = _L2) - 3, y = 1/2 RootOf(_Z^2 + 6 _Z + 1, label = _L2) + 7, z = 1/2 RootOf(_Z^2 + 6 _Z + 1, label = _L2)}
```

Y si lo necesitamos, podemos forzar el resultado numérico:

```
> evalf(solve({3 x + 2 y + 4 z = 5, x + y + z = 4, x^2 - y + z = 1}));
{x = -2.828427125, y = 6.914213562, z = -0.08578643765}
```

Si el sistema de ecuaciones no tiene solución única, parametriza en una de las variables:

```
> solve({3 x + 2 y + 4 z = 5, 3 x + 2 y + 4 z = 5, x - y + z = 1});
{x = -1 + 6 y, y = y, z = 2 - 5 y}
```

y podemos decirle en cual de las variables parametrizar (si le decimos cuáles son las variables en las que queremos resolver):

```
> solve({3 x + 2 y + 4 z = 5, 3 x + 2 y + 4 z = 5, x - y + z = 1}, {x, y});
```

$$\left\{ x = \frac{7}{5} - \frac{6}{5} z, y = \frac{2}{5} - \frac{1}{5} z \right\}$$

Podemos resolver inecuaciones:

```
> solve({x + y < 10, x^2 = 9}, {x, y});
{x = 3, y < 7}, {x = -3, y < 13}
```

o asumir una condición, mediante el comando *assuming*

```
> solve({x + y < 10, x^2 = 9}, {x, y}) assuming 10 < y;
{x = -3, y < 13}
```

Las incógnitas pueden a su vez depender de una variable sujeta a una condición

```
> solve(x^2 = a, x) assuming a::negative
1 sqrt(-a), -1 sqrt(-a)
```

## Matrices

Para trabajar con matrices primero hay que definir las. La forma, en modo texto es, ir a la columna de la izquierda y definirla en la pestaña Matriz (inténtalo aquí)

Una vez definida una matriz, con el botón derecho del ratón puede hacer operaciones sobre ella, calculando cosas como su determinante, autovalores, autofunciones, puntos singulares, forma canónica de Jordan, etc. Incluso aunque tenga parámetros. Prueba con la siguiente matriz:

$$\begin{bmatrix} 1 & 1 \\ \alpha & 0 \end{bmatrix}$$

En modo comando se definen, en modo texto de la siguiente forma:

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

O en modo matemáticas, de esta forma más amigable:

```
> A := [ 1 2
        2 -1 ]
A:= [ 1 2
      2 -1 ]
> B := [ 3 6
        -1 2 ]
```

$$B := \begin{bmatrix} 3 & 6 \\ -1 & 2 \end{bmatrix}$$

También podemos definir una matriz mediante una función de la forma:

$$\begin{aligned} &> f := (i, j) \rightarrow x^{i+j-2}; \\ &> C := (x) \rightarrow \text{Matrix}(2, f) \end{aligned} \quad \begin{aligned} f &:= (i, j) \rightarrow x^{i+j-2} \\ C &:= x \rightarrow \text{Matrix}(2, f) \end{aligned}$$

Ahora bien, debemos tener cuidado al utilizar las matrices definidas con funciones. Recordar que son funciones:

$$\begin{aligned} &> C; \\ &> C(x); \end{aligned} \quad \begin{aligned} &C \\ &\begin{bmatrix} 1 & x \\ x & x^2 \end{bmatrix} \end{aligned}$$

Podemos acceder a cada elemento de la matriz de la forma:

$$\begin{aligned} &> A[1,1]; A[1,2]; A[2,1]; A[2,2]; \\ &> B_{1,1}; B_{1,2}; B_{2,1}; B_{2,2}; \end{aligned} \quad \begin{aligned} &1 \\ &2 \\ &2 \\ &-1 \\ &3 \\ &6 \\ &-1 \\ &2 \end{aligned}$$

Una vez definidas las matrices puedes operar. Por ejemplo:

$$\begin{aligned} &> A, B, A.B, A.B - B.A, A^{-1}, \frac{A}{B}, A.B^{-1}; \\ &\begin{bmatrix} 1 & 2 \\ 2 & -1 \end{bmatrix} \\ &\begin{bmatrix} 3 & 6 \\ -1 & 2 \end{bmatrix} \\ &\begin{bmatrix} 1 & 10 \\ 7 & 10 \end{bmatrix} \\ &\begin{bmatrix} -14 & 10 \\ 4 & 14 \end{bmatrix} \end{aligned}$$

$$\begin{bmatrix} \frac{1}{5} & \frac{2}{5} \\ \frac{2}{5} & -\frac{1}{5} \end{bmatrix}$$

Error, (in rtable/Product) invalid arguments

$$\begin{bmatrix} \frac{1}{3} & 0 \\ \frac{1}{4} & -\frac{5}{4} \end{bmatrix}$$

Observa que el producto se hace con el punto "." no con el asterisco "\*\*"

Los operadores más frecuentes para usar con matrices empiezan con LinearAlgebra. Por ejemplo : *LinearAlgebra:-Determinant*, *LinearAlgebra:-MatrixInverse*, *LinearAlgebra:-Transpose*, ...

$$\begin{aligned} &> \text{LinearAlgebra:-CharacteristicPolynomial}(A, \lambda); \\ &> \text{solve}(\text{LinearAlgebra:-CharacteristicPolynomial}(B, \lambda) = 0); \\ &> \text{LinearAlgebra:-Eigenvalues}(A.B) \end{aligned} \quad \begin{aligned} &-5 + \lambda^2 \\ &\frac{5}{2} + \frac{1}{2} i \sqrt{23}, \frac{5}{2} - \frac{1}{2} i \sqrt{23} \\ &\begin{bmatrix} 15 \\ -4 \end{bmatrix} \end{aligned}$$

## ▼ Cálculo infinitesimal

[> **restart;**

Se pueden calcular derivadas e integrales de modo simbólico, tomándose las variables no asignadas como constantes.

$$\begin{aligned} &> f := x \rightarrow a*x^2 + b*x + c; \\ &> \text{diff}(f(x), x); \\ &> \int f(x) dx \end{aligned}$$

Se pueden calcular integrales definidas.

[> **int(f(x), x=-2..2);**

Incluso operar para obtener un valor concreto, como en valor de  $x$  que hace extremo (máximo o mínimo) a  $f(x)$

$$\begin{aligned} &> x_{ext} := \text{solve}\left(\frac{d}{dx} f(x) = 0, x\right); y_{ext} := f(x_{ext}) \end{aligned}$$

## Ecuaciones diferenciales

Se pueden introducir ecuaciones diferenciales directamente en modo texto si cambiamos a modo matemáticas y luego resolverlas con el botón derecho y la opción Solve DE

$$y'' + y' + y = 0$$

También se pueden introducir en modo comando, mediante el uso de [diff](#) en modo texto o

$\frac{d}{dx}$  en modo matemático

[> *restart*,

El siguiente ejemplo está extraído de los apuntes de (apuntes Ecuaciones Diferenciales pag 5)

$$\left[ \begin{array}{l} > \text{dsolve} \left( \frac{d}{dx} y(x) = x \cdot y(x)^3 \right) \\ y(x) = \frac{1}{\sqrt{-x^2 + C1}}, y'(x) = -\frac{1}{\sqrt{-x^2 + C1}} \end{array} \right]$$

O incluso asignarlas a una variable:

$$\left[ \begin{array}{l} > \text{edo} := \text{diff}(y(x), x) = \exp(y(x) - x^2); \\ \text{edo} := \frac{d}{dx} y(x) = e^{y(x) - x^2} \end{array} \right]$$

Para resolver la ecuación diferencial utilizamos el comando [dsolve](#) (apuntes Ecuaciones Diferenciales pag 5)

$$\left[ \begin{array}{l} > \text{dsolve}(\text{edo}, y(x)); \\ y(x) = \ln \left( -\frac{2}{\sqrt{\pi} \operatorname{erf}(x) + 2C1} \right) \end{array} \right]$$

La función erf(x) es la función error que se define como  $\operatorname{erf}(x) = \int e^{-x^2} dx$

Si se trata de un problema con condiciones iniciales, estas podemos introducir como parte del problema poniéndolas, junto con la ecuación, en una llave {}. (apuntes Ecuaciones Diferenciales pag 7):

$$\left[ \begin{array}{l} > \text{dsolve} \left( \left\{ \frac{d}{dt} y(t) = -\frac{y(t)}{t} + e^t, y(1) = 1 \right\} \right); \\ y(t) = \frac{(-1 + t) e^t + 1}{t} \end{array} \right]$$

De la misma forma, si se trata de un sistema de ecuaciones, podemos poner todos los sistemas entre llaves y resolver (apuntes Ecuaciones Diferenciales pag 35):

$$\left[ \begin{array}{l} > \text{dsolve} \left( \left\{ \frac{d}{dt} x(t) = x(t) - y(t) + 2, \frac{d}{dt} y(t) = x(t) + y(t), x(0) = 0, y(0) = 1 \right\} \right); \\ \{x(t) = -1 + e^t \cos(t), y(t) = 1 + e^t \sin(t)\} \end{array} \right]$$

El comando dsolve admite además un parámetro en el cual decirle el método que queremos utilizar para resolver el sistema

**Cálculo de soluciones por series:** se añade el parámetro

*series*. Para más información consulta [dsolve/formal\\_series](#) (apuntes Ecuaciones Diferenciales pag 65)

$$\left[ \begin{array}{l} > \text{dsolve} \left( \left\{ 2 \cdot t \cdot \frac{d}{dt} x(t) + (t-2) \cdot \frac{d}{dt} x(t) + 3 \cdot x(t) = 0, x(0) = 1 \right\}, x(t), \text{series} \right); \\ x(t) = 1 + \frac{3}{2} t + \frac{9}{4} t^2 + \frac{27}{8} t^3 + \frac{81}{16} t^4 + \frac{243}{32} t^5 + O(t^6) \end{array} \right]$$

**Cálculo de soluciones por Transformada de Laplace:** se añade el parámetro

*method=laplace*. Compara estas dos soluciones (apuntes Ecuaciones Diferenciales pag 50) :

$$\left[ \begin{array}{l} > \text{dsolve} \left( \left\{ \frac{d^2}{dt^2} x(t) - 2 \cdot \frac{d}{dt} x(t) + x(t) = 6 \cdot t \cdot e^t, x(1) = 0, D(x)(1) = 0 \right\}, x(t) \right); \\ x(t) = 2 e^t - 3 e^t t + t^3 e^t \\ > \text{dsolve} \left( \left\{ \frac{d^2}{dt^2} x(t) - 2 \cdot \frac{d}{dt} x(t) + x(t) = 6 \cdot t \cdot e^t, x(1) = 0, D(x)(1) = 0 \right\}, x(t), \right. \\ \quad \left. \text{method} = \text{laplace} \right); \\ x(t) = (-1 + t)^2 (2 + t) e^t \end{array} \right]$$

Una forma cómoda de expresar ecuaciones diferenciales es con la notación:  $\frac{d}{dt} = '$

**Por ejemplo** (Problemas para entregar del tema 2. Problema 5a)

Para  $a = 0$

$$\left[ \begin{array}{l} > \text{dsolve}(\{x'''(t) + 2 \cdot x''(t) + 2 \cdot x'(t) = 2, x(0) = 1, x'(0) = 1, x''(0) = -2\}) \\ x(t) = e^{-t} \cos(t) + e^{-t} \sin(t) + t \end{array} \right]$$

Para  $a = 4$

$$\left[ \begin{array}{l} > \text{dsolve}(\{x'''(t) + 2 \cdot x''(t) + 2 \cdot x'(t) + 4 \cdot x(t) = 2, x(0) = 1, x'(0) = 1, x''(0) = -2\}) \\ x(t) = \frac{1}{2} - \frac{1}{6} e^{-2t} + \frac{2}{3} \cos(\sqrt{2} t) + \frac{1}{3} \sin(\sqrt{2} t) \sqrt{2} \end{array} \right]$$



## Gráficos

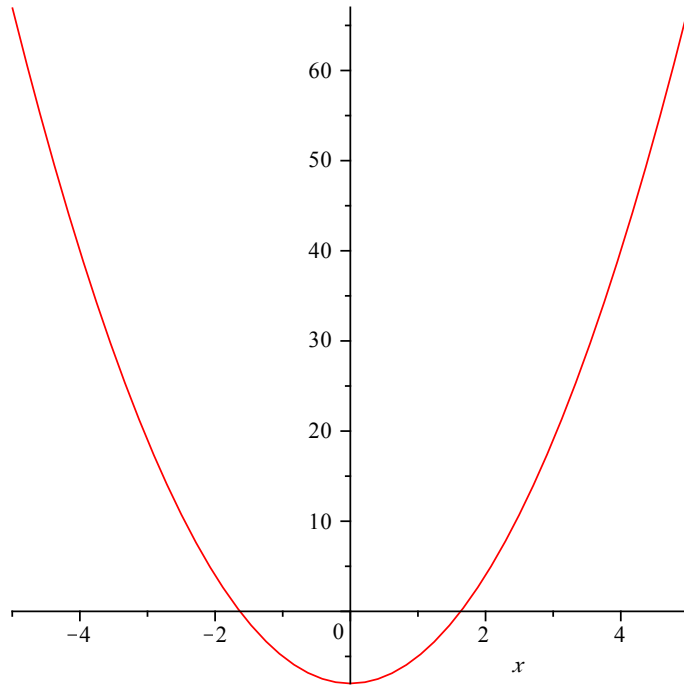
### Gráficos bidimensionales.

Como siempre, conviene hacer primero un *restart*

```
[> restart;
```

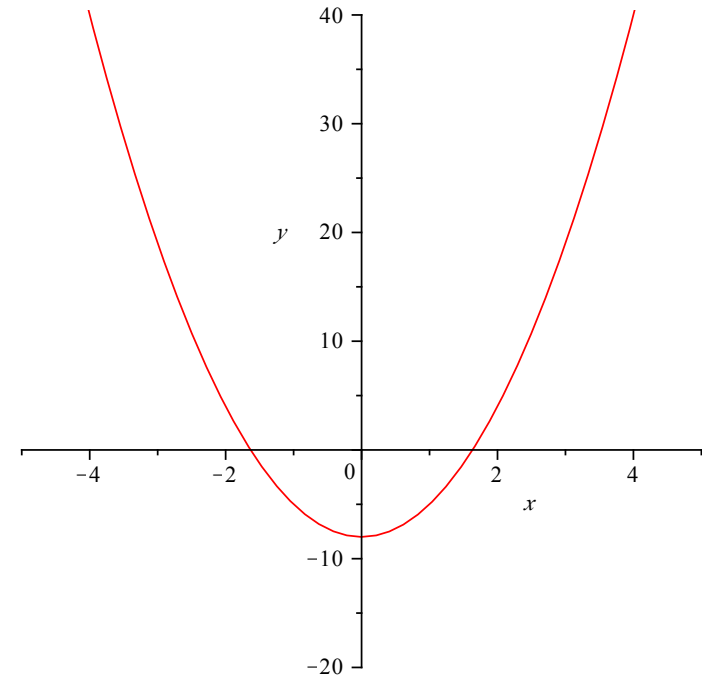
Para representar gráficamente una función dada por una expresión podemos usar el comando plot. Por ejemplo, la gráfica de  $3x^2 - 8$  para  $x$  de -5 a 5.

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



Podemos elegir el rango de los ejes:

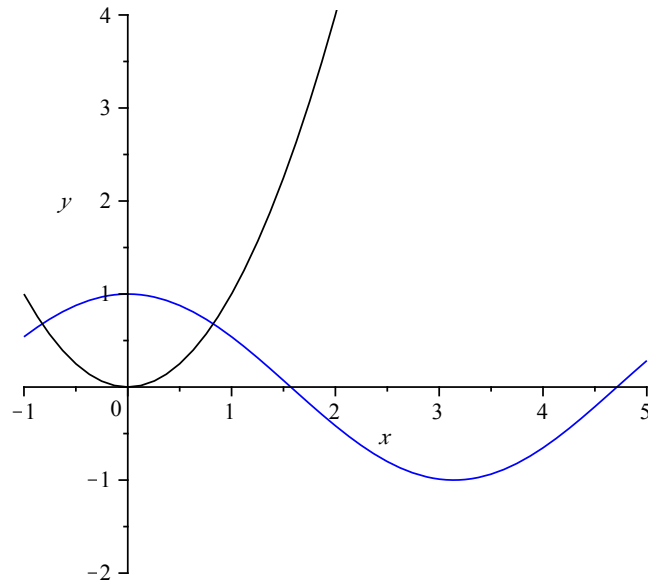
```
> f := x -> 3*x^2-8;  
plot(f(x),x=-5..5,y=-20..40);
```



( obsérvese que hemos restringido el rango de valores de  $y$  especificando  $y=-20..40$  )

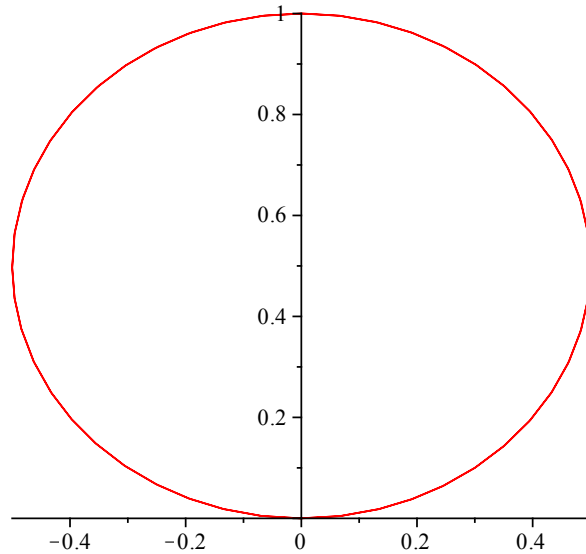
Para mostrar varias gráficas en una sola imagen, se utiliza una notación de lista entre corchetes:

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



Se pueden representar funciones dadas en coordenadas polares usando la siguiente opción:

```
> plot(sin(x),x=0..2*Pi, coords=polar);
```

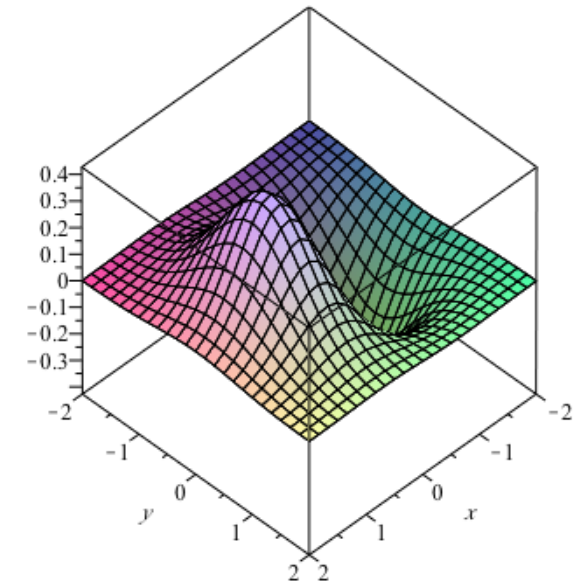


## Gráficos tridimensionales

Se pueden crear gráficos tridimensionales (obsérvese el uso de algunas opciones nuevas)

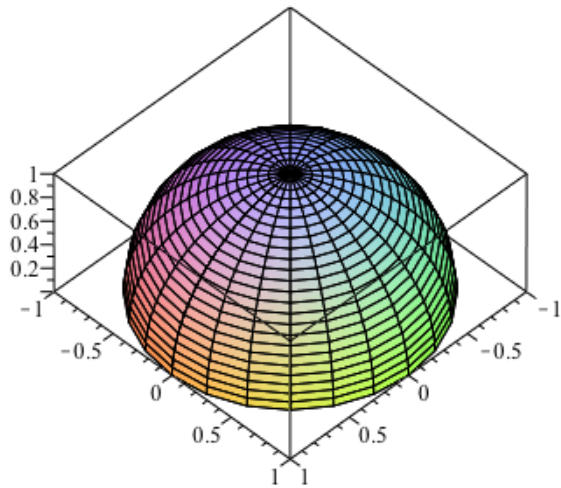
```
> plot3d( x*exp(-x^2-y^2), x=-2..2, y=-2..2, axes=BOXED,title=
"Gráfico de una superficie" );
```

Gráfico de una superficie



Para representar funciones en coordenadas esféricas podemos utilizar la siguiente notación:

```
> plot3d(1,t=0..2*Pi,p=0..Pi/2,coords=spherical,scaling=
constrained,axes=BOXED);
```



## Gráficas de datos con error

Para representar datos con sus correspondientes errores, primero debemos llamar a las librerías correspondientes. Después de poner el nombre de la librería ponemos como separador :

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

¡Ojo! ¡¡el *restart* cancela también las llamadas a las librerías!!

Introducimos los datos y ponemos los separadores : para no ver sus ejecuciones

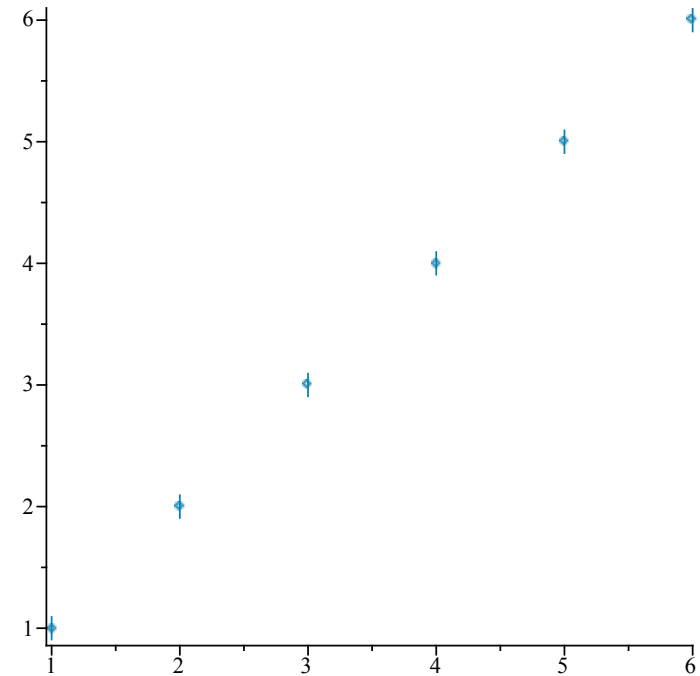
```
[> x:=[1,2,3,4,5,6]:y:=[1,2,3,4,5,6]:error:=[0.1,0.1,0.1,0.1,0.1,0.1]:
```

Asignamos a la variable A el gráfico con los puntos y sus errores

```
[> A:=ErrorPlot(y, yerrors =error, xcoords =x):
```

Visualizamos A

```
[> display(A);
```



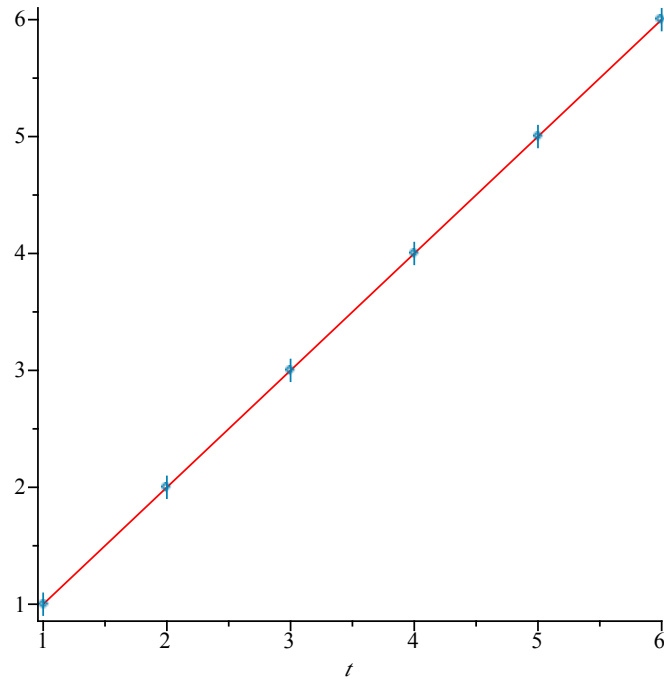
Ahora queremos visualizaz dos gráficas a la vez, una la que acabamos de hacer y otra línea de tendencia. Definiremos la gráfica de la línea de tendencia dentro de otra variable y luego ejecutaremos ambas variables dentro del mismo *display*

Asignamos a B el gráfico de la recta que siguen los puntos (x,y)

```
[> B:=plot(t,t=1..6):
```

Por último, representamos las dos gráficas juntas

```
[> display({A,B});
```



## Gráficos para Ecuaciones Diferenciales

### Representación de soluciones de ecuaciones diferenciales

Lo primero es llamar a la librería correspondiente, pero iniciamos por el restart y luego llamamos a las librerías:

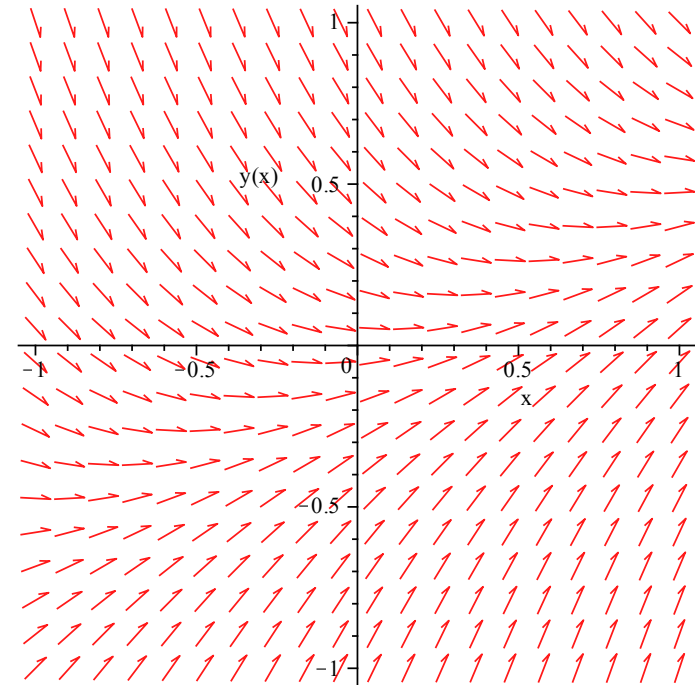
```
[> restart,
```

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

Toda ecuación diferencial de una variable representa en plano  $(x, y(x))$  un campo de direcciones. En Maple se puede representar mediante el comando *dfieldplot*

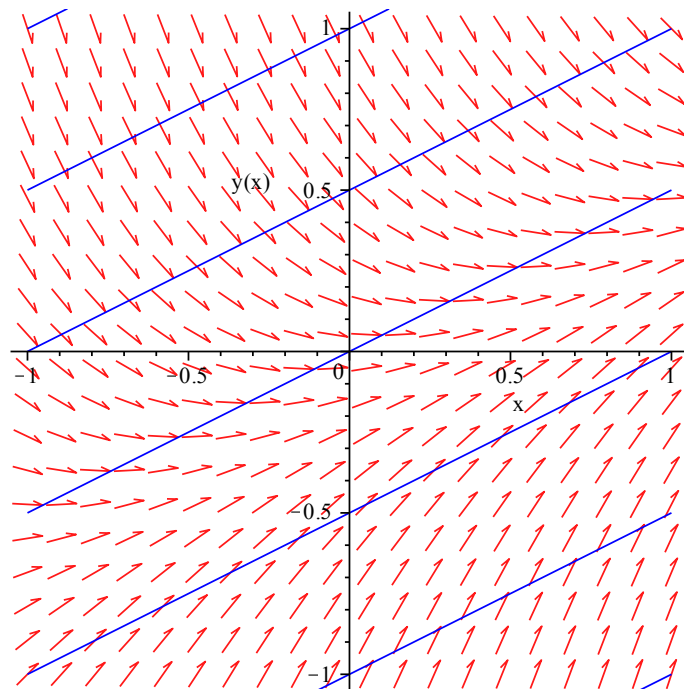
(Apuntes Ecuaciones Diferenciales pag 10)

```
[> dfieldplot(diff(y(x), x) = x-2*y(x), y(x), x=-1..1, y=-1..1);
```



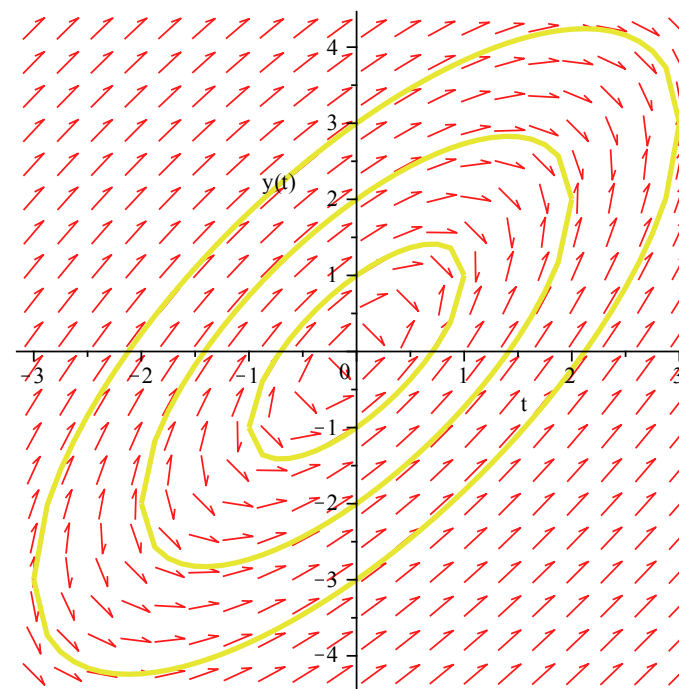
Podemos incluir las isoclinas, haciendo dos gráficos y mostrándolas a la vez con el display, de la forma (la ecuación de las isoclinas tenemos que proporcionarla nosotros o bien usando un comando solve):

```
[> A:= dfieldplot(diff(y(x), x) = x-2*y(x), y(x), x=-1..1, y=-1..1);
> B:= plot( { (x-3)/2, (x-2)/2, (x-1)/2, (x-0)/2, (x+1)/2, (x+2)/2, (x+3)/2, (x+4)/2 }, x=
-1..1, y=-1..1, color= blue ) :
> display ( { A, B } )
```



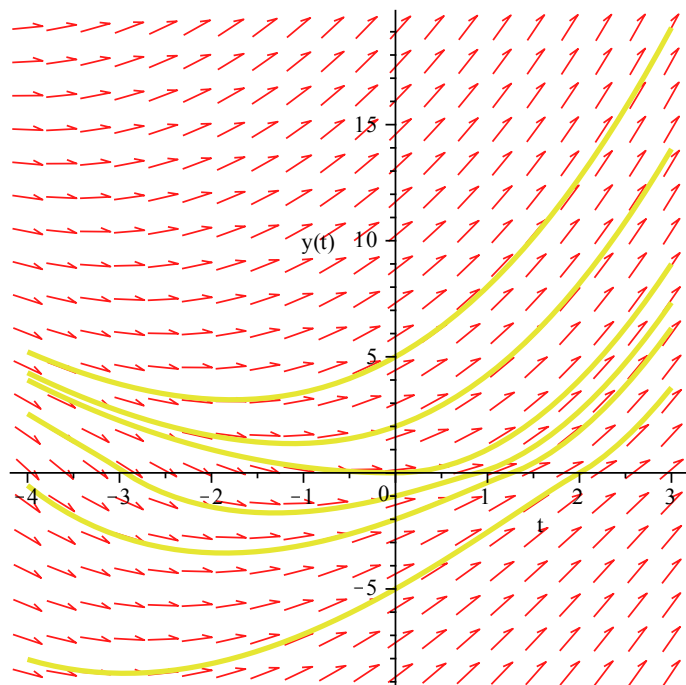
Dándos unos valores iniciales podemos representar soluciones particulares dentro del campo de direcciones (Apuntes Ecuaciones Diferenciales pag 11) (Observar que da errores para algunos valores, aunque el resultado final sale correcto)

```
> ic := [y(0) = 1, y(0) = 2, y(0) = 3, y(0) = -1, y(0) = -2, y(0) = -3]:
> DEplot(diff(y(t), t) = (2*t - y(t))/(t - y(t)), y(t), t = -3..3, ic)
Warning, plot may be incomplete, the following errors(s) were issued:
cannot evaluate the solution further right of 1.0000001, probably a singularity
cannot evaluate the solution further left of -1.0000000, probably a singularity
Warning, plot may be incomplete, the following errors(s) were issued:
cannot evaluate the solution further right of 2.0000003, probably a singularity
cannot evaluate the solution further left of -2.0000001, probably a singularity
Warning, plot may be incomplete, the following errors(s) were issued:
cannot evaluate the solution further right of 1.0000000, probably a singularity
cannot evaluate the solution further left of -1.0000001, probably a singularity
Warning, plot may be incomplete, the following errors(s) were issued:
cannot evaluate the solution further right of 2.0000001, probably a singularity
cannot evaluate the solution further left of -2.0000003, probably a singularity
```



Otro ejemplo (Apuntes Ecuaciones Diferenciales pag 12)

```
> ic := [y(0) = 0, y(0) = 2, y(0) = 5, y(0) = -1, y(0) = -2, y(0) = -5]:
> DEplot(diff(y(t), t) = sqrt(y(t)) + t, y(t), t = -4..3, ic)
```

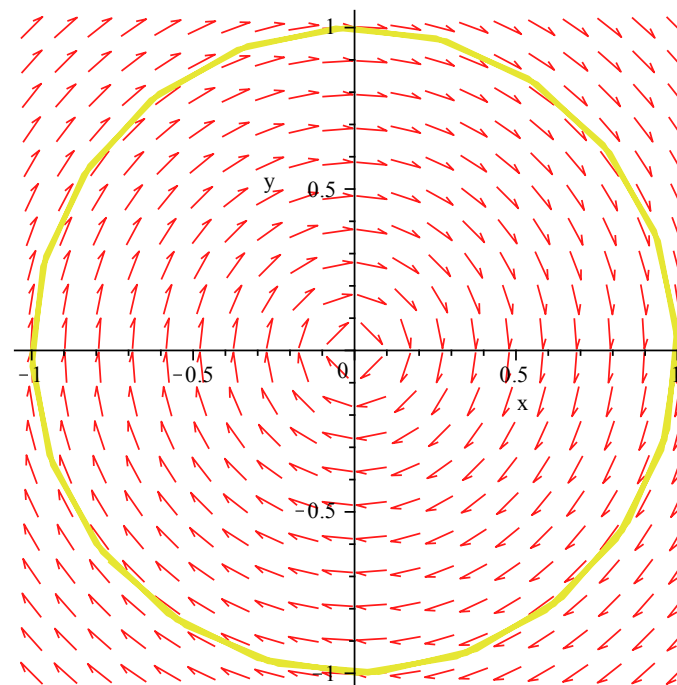


## Mapas de Fase

Consiste en dibujar las proyecciones de las soluciones de un sistema autónomo sobre el plano XY. A estas proyecciones se les llama órbitas.

El siguiente ejemplo está sacado de la pagina 72 de los apuntes de Ecuaciones Diferenciales.

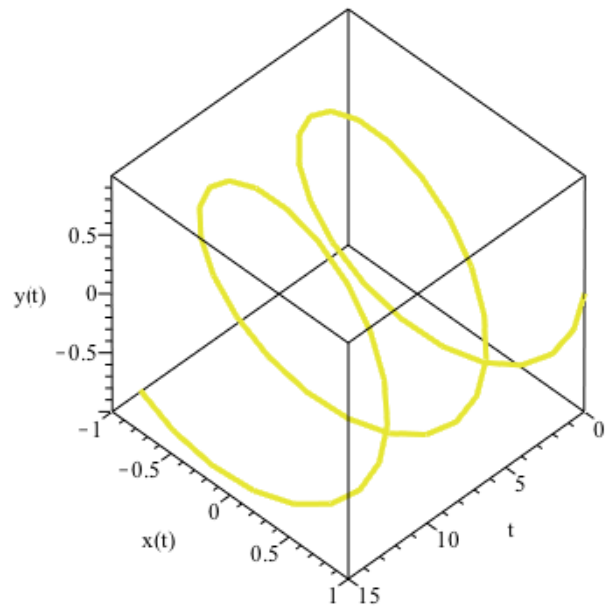
```
> DEplot( { d/dt x(t) = y(t), d/dt y(t) = -x(t) }, [x(t), y(t)], t = 0..15, [[x(0) = 1,
    y(0) = 0]] )
```



Lo que hemos representado es la órbita en el plano XY.

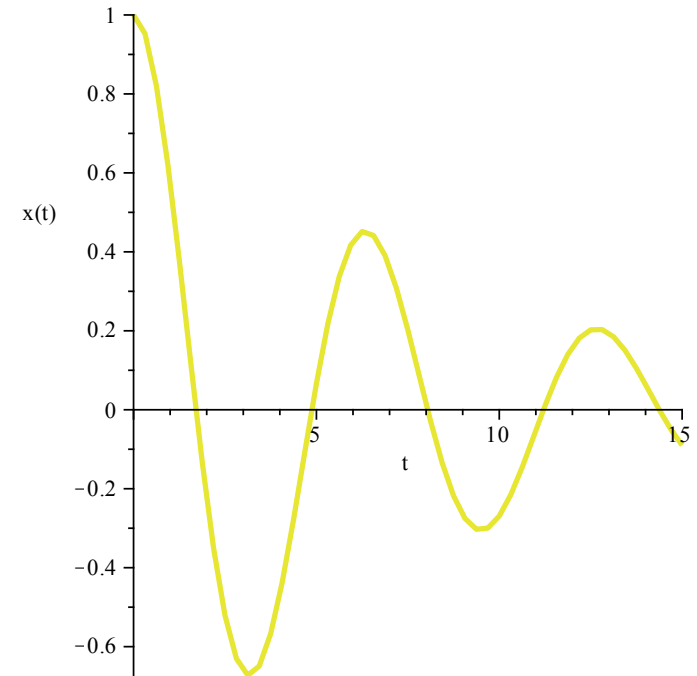
Podemos representar también la solución si al mapa de fase incluimos la variable t. De esta forma se aprecia mejor el significado de la órbita como proyección.

```
> DEplot3d( { d/dt x(t) = y(t), d/dt y(t) = -x(t) }, [x(t), y(t)], t = 0..15, [[x(0)
    = 1, y(0) = 0]], scene = [t, x(t), y(t)] )
```



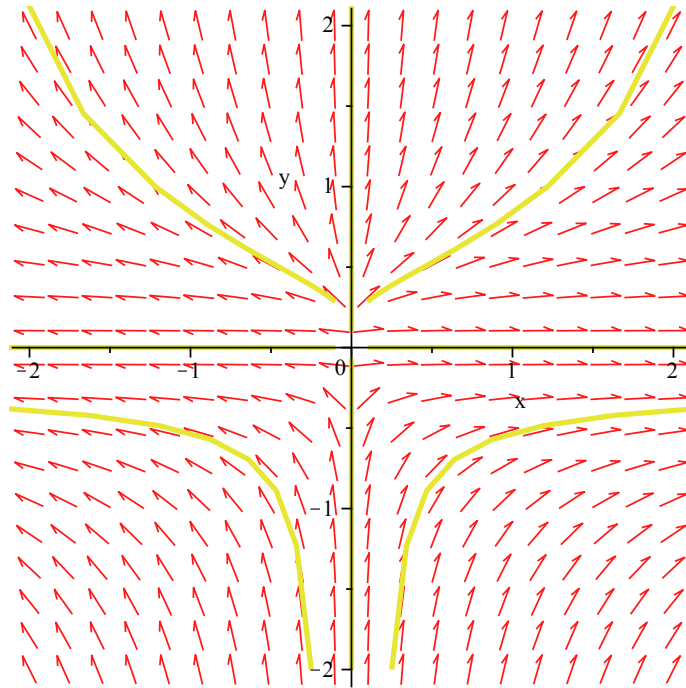
El mismo ejemplo anterior pero con una componente de amortiguación, y a continuación la solución para  $x(t)$

```
> DEplot( [ { d/dt x(t) = y(t), d/dt y(t) = -x(t) - y(t)/4 }, [x(t), y(t)], t=0..15,
  [[x(0)=1, y(0)=0]], scene=[t, x(t)] )
```



De especial utilidad son los mapas de fase para estudiar la estabilidad de las soluciones (apuntes pag 73):

```
> DEplot( [ { d/dt x(t) = x(t), d/dt y(t) = y^2(t) }, [x(t), y(t)], t=0..15, x=-2..2, y=
  -2..2, [[x(0)=0, y(0)=-2], [x(0)=1/4, y(0)=-2], [x(0)=-1/4, y(0)=
  -2], [x(0)=1/10, y(0)=0], [x(0)=-1/10, y(0)=0], [x(0)=0, y(0)=
  1/10], [x(0)=1/10, y(0)=2/7], [x(0)=-1/10, y(0)=2/7]] )
```



Algunos ejemplos útiles de mapas de fase son los siguientes. Utilizan los argumentos:

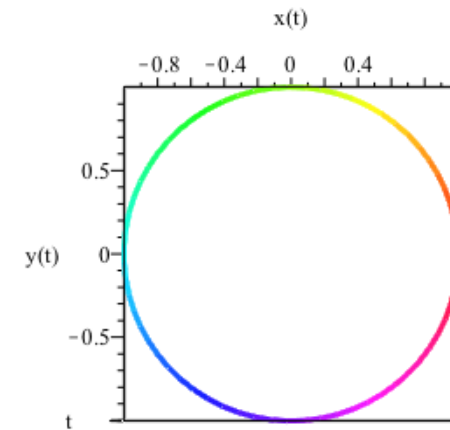
- *orientation* = [0, 90] para orientar la imagen con el eje t paralelo a nuestra visión
- *linecolor* = ...(*t*) permite cambiar el color de la línea en función del valor de *t*, esto nos da una idea del valor relativo de la variable independiente.

Y algunos argumentos para dar formato como *title*, *axes*, etc

....

(Mover las figuras para ver las formas de las soluciones)

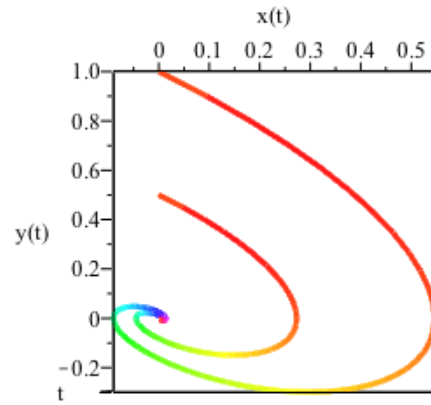
```
> DEplot3d( { d/dt x(t) = -sin(t), d/dt y(t) = cos(t) }, { y(t), x(t) }, t = -2 .. 0,
  [[ y(0) = 0, x(0) = 1 ]], scene = [ t, x(t), y(t) ], stepsize = 0.1, orientation = [ 0,
  90 ], linecolor = t)
```



```
> DEplot3d( { D(x)(t) = y(t), D(y)(t) = -x(t) - y(t) }, [ x(t), y(t) ], t = 0 .. 10,
  [[ x(0) = 0, y(0) = 1 ], [ x(0) = 0, y(0) = 0.5 ]], scene = [ t, x(t), y(t) ], stepsize
  = 0.1, title = 'Oscilaciones amortiguadas', orientation = [ 0, 90 ], linecolor = t
  - sqrt(t))
```

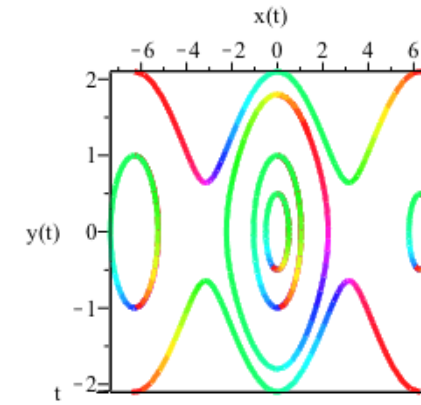


### Oscilaciones amortiguadas

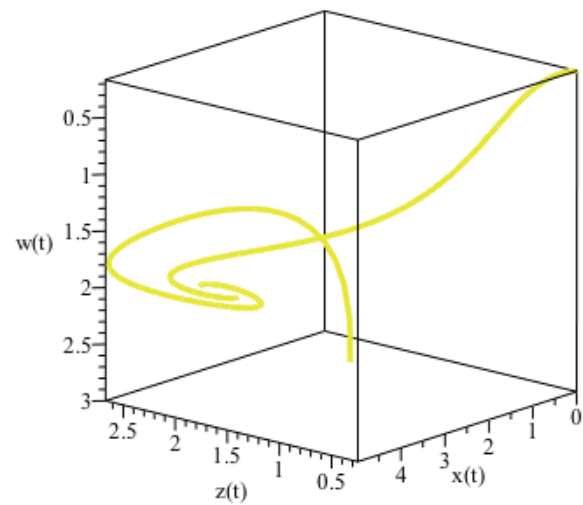


```
> DEplot3d ( { d/dt x(t) = y(t), d/dt y(t) = -sin(x(t)) }, [x(t), y(t)], t=0..10,
  [[x(0) = 0, y(0) = 0.5], [x(0) = 0, y(0) = 1], [x(0) = 0, y(0) = 1.8], [x(0) =
  -2 pi, y(0) = 1], [x(0) = 2 pi, y(0) = 0.5], [x(0) = -2 pi, y(0) = 2.1], [x(0)
  = 2 pi, y(0) = -2.1]], stepsize=0.2, title= P ndulo, orientation= [0, 90], linecolor
  = sin(t) - t)
```

### P ndulo



```
> DEplot3d ( { D(x) (t) = y(t) + z(t), D(y) (t) = -x(t) - y(t) + w(t), D(z) (t)
  = x(t) + y(t) - z(t), D(w) (t) = x(t) - w(t) z(t) }, {x(t), y(t), z(t),
  w(t)}, t=0..10, [[x(0) = 1, y(0) = 0, z(0) = 2, w(0) = 3], [x(0) = 0, z(0)
  = 0.25, y(0) = 0.5, w(0) = 0.165]], scene= [x(t), z(t), w(t)], stepsize=0.1,
  orientation= [139, -106])
```



```
> DEplot3d( { { d/dt x(t) = y(t), d/dt y(t) = -sin(x(t)) - y(t)/10 }, { x(t), y(t) }, t =
-9..9, stepsize = 0.1, [[x(0) = 1, y(0) = 1]], linecolor = t, axes = NONE )
```

