

Representación Gráfica

Comando básico de Scilab para graficar: *plot2d* / *plot* (*)

Dados dos vectores $x=[x_1, x_2, \dots, x_n]$ e $y=[y_1, y_2, \dots, y_n]$

-->*plot2d*(x,y)

dibuja la curva que pasa por los puntos $(x_1, y_1) \dots x_n, y_n)$

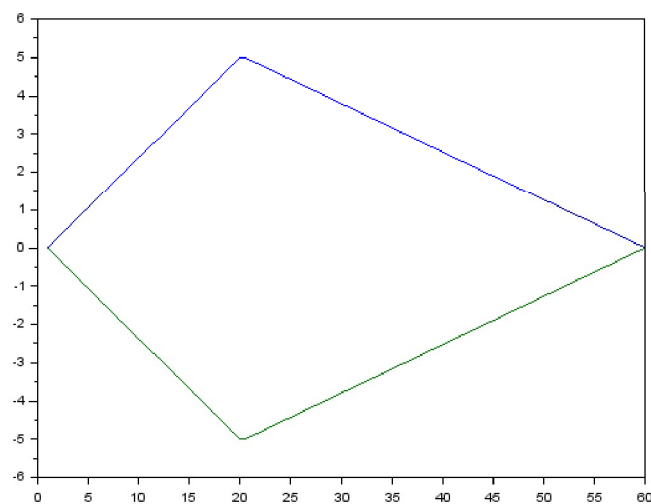
Ejemplo:

```
clf()
x=[1 20 20.5 60]';
y1=[0 5 4.9999 0]';
y2=[0 -5 -4.9999 0]';
plot(x,y1,x,y2)
```

(*): *plot* es compatible Matlab

Clase 4

Representación Gráfica



Clase 4

Representación Gráfica

Ejemplo: dibujar en el intervalo $[-5,5]$

$$y = 2 \operatorname{sen}\left(\frac{x}{2}\right) \cos(3x)$$

```
x=linspace(-5,5)';
y=2*sin(x/2).*cos(3*x);
plot2d(x,y)
```

También se podría poner, directamente:

```
x=linspace(-5,5)';
plot2d(x, 2*sin(x/2).*cos(3*x))
```

¿qué pasa si en el producto en vez de usar `(.*)` usa `(*)`?

`sin(x/2)*cos(2*x)` → explique!!!

Clase 4

plot2d() / plot()

-->plot2d(x,y)

siendo **x** un vector e **y** una matriz dibuja una curva por cada columna de **y**

Ejemplo:

```
x=linspace(-5,5)';
y=2*sin(x/2).*cos(3*x);
z=2*sin(x);
w=[y,z];
plot2d(x,w)
```

plot2d() / plot()

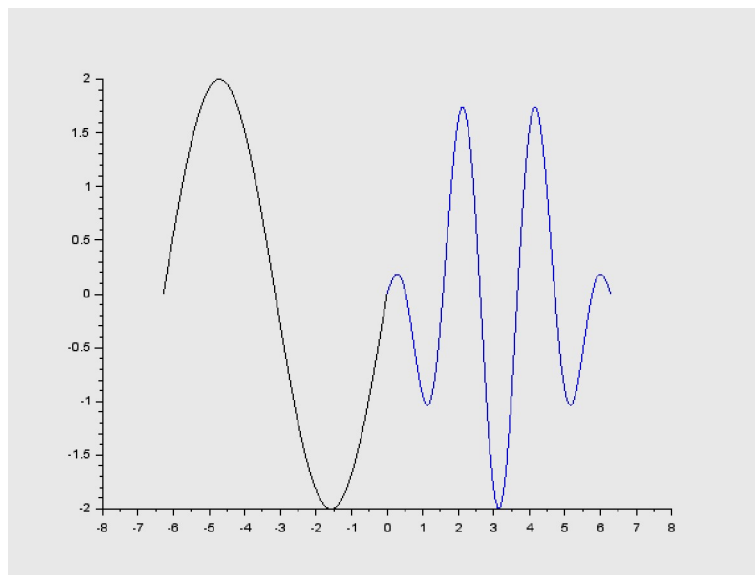
-->plot2d(x,y)

siendo **x** e **y** matrices de la misma dimensión, dibuja una curva por cada par **columna de x - columna de y**

Ejemplo: dibujar la función $y = \begin{cases} 2 \operatorname{sen}(x) & \text{si } x \leq 0 \\ 2 \operatorname{sen}(x/2) \cos(3x) & \text{si } x > 0 \end{cases}$

```
x1=linspace(-2*pi,0)';
x2=linspace(0,2*pi)';
y1=2*sin(x1);
y2=2*sin(x2/2).*cos(3*x2);
plot2d([x1,x2],[y1,y2])
```

plot2d() / plot()



plot2d() / plot()

Cuando `plot2d` dibuja varias curvas, les asigna distintos colores. El orden de los colores asignados viene determinado por la carta de colores activa. Por defecto, es la siguiente:

1	7				
2					
3					
4					
5					
6					

Tipo línea	Tipo puntos	Color
Sólida —	Cruz +	Verde <i>g</i>
A trozos --	Punto ·	Negro <i>k</i>
Puntos :	Estrellas *	Rojo <i>r</i>
Trazo y punto -.	Círculo o	Azul <i>b</i>

plot2d() / plot()

-->`plot2d(x,y)`

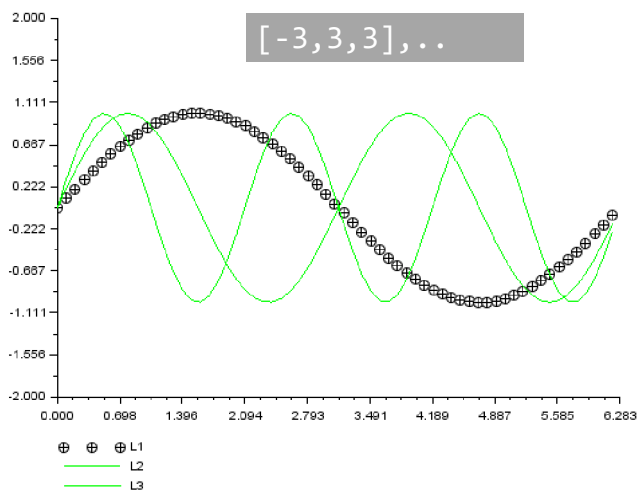
con varios argumentos o gráficos en el mismo lienzo

```
clf
u=linspace(-2*%pi,0)';
v=2*sin(u);
w=2*sin(u/2).*cos(3*u);
plot(u,[v w])
```

```
>> x = [0 : 2 * %pi/35 : 2 * %pi];
>> y = cos(x); z = cos(4 * x)
>> plot(x, y, x, z)
```

plot2d() / plot()

```
clf();
x=[0:0.1:2*pi]';
plot2d(x,[sin(x) sin(2*x) sin(3*x)], [-3,3,3],...)
```



plot2d() / plot()

-->plot2d(x,y)

control sobre la posición de los ejes

```
clf()
x=[0:0.1:2*pi]';
// eje de las ordenadas a la derecha
plot(x,sin(x))
a=gca(); // toma comando sobre los ejes
a.y_location = "right";
```

Scilab Help >> Graphics > axes_operations > subplot

subplot

virtually grids the figure and sets the plotting area to a chosen cell

Syntax

```
subplot(m,n,p)
subplot(mnp)
```

Arguments

m,n,p positive integers
mnp an integer with decimal notation `mnp`

Description

`subplot(m,n,p)` or `subplot(mnp)` breaks the graphics window into an `m`-by-`n` matrix of sub-windows and selects the `p`-th sub-window for drawing the current plot. The number of a sub-window into the matrices is counted row by row ie the sub-window corresponding to element `(i,j)` of the matrix has number `(i-1)*n + j`.

```
subplot(221)
plot2d()
subplot(222)
plot3d()
subplot(2,2,3)
param3d()
subplot(2,2,4)
hist3d()
```

Clase 4

Contour plot

```
function f = myquadratic1arg ( x )
    f = x(1)**2 + x(2)**2;
endfunction
xdata = linspace ( -1 , 1 , 100 );
ydata = linspace ( -1 , 1 , 100 );
// cuidado! dos Loops anidados...
for i = 1:length(xdata)
    for j = 1:length(ydata)
        x = [xdata(i) ydata(j)].';
        zdata ( i , j ) = cuadratica ( x );
    end
end
contour ( xdata , ydata , zdata , [0.1 0.3 0.5 0.7])
```