

Introducción a los gráficos en Matlab

Gráficos 2D:

plot

(+info: doc plot)

Las formas más habituales para la utilización de `plot` son:

```
plot(Y [, 'opciones de línea '])
plot(X1, Y1 [, 'opciones de línea 1'], X2, Y2[, 'opciones de línea 2'], ...)
plot(..., 'Propiedad', Valor,...)
```

Esta función muestra una gráfica con los valores de Y_n con respecto a X_n . Opcionalmente, tras cada par de valores X_n e Y_n se pueden especificar distintos estilos de línea. Asimismo, se pueden asignar una serie de propiedades a todas las líneas del gráfico especificando parejas 'Propiedad' Valor en los argumentos. Si no se asigna ningún estilo entonces Matlab mostrará cada línea con un estilo distinto generado automáticamente.

Dependiendo de la naturaleza de las variables X_n e Y_n hay que tener en cuenta lo siguiente:

- Si no se especifica valores para la X_n (forma 1ra de llamar a plot) entonces se dibujan los valores de Y_n con respecto al índice.
- Si X_n e Y_n son vectores entonces basta con que sean de igual longitud. No es necesario que los dos vectores sean ambos vector columna o ambos vector fila.
- Si X_n e Y_n son matrices entonces deben tener las mismas dimensiones. Se genera una gráfica con tantas líneas como columnas tengan las matrices donde la línea k corresponde a la gráfica de $Y_n(:, k)$ con respecto a $X_n(:, k)$.
- Si X_n es un vector e Y_n es una matriz entonces una de las dimensiones de Y_n debe coincidir con la longitud de X_n . Dibuja todas las filas o columnas de Y_n con respecto a la misma x , esto es el vector X_n .

Veamos algunos ejemplos sin utilizar las opciones de línea:

Gráfico simple de un vector y con respecto a otro x

```
x = 0:0.1:4;
y = 1./x;
plot(x, y)
% Se obtiene el mismo gráfico con :
    plot(x', y)
    plot(x, y')
    plot(x', y')
```

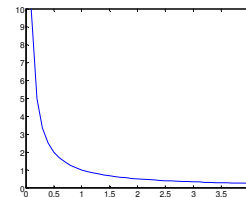


Gráfico con el seno y coseno sobre la misma x

```
x = linspace(-pi,pi,20);
Y = [sin(x);cos(x)]; % la Y es una matriz
plot(x, Y)
% Es equivalente a:
    plot(x, Y(1, :), x, Y(2, :))
```

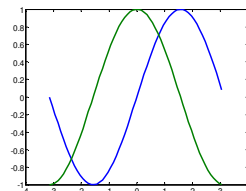
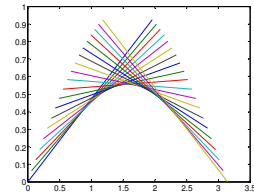
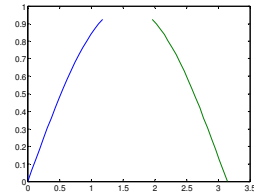


Gráfico con el seno en dos tramos $[0, 3\pi/8]$ y $[5\pi/8, \pi]$

```
X = [linspace(0,pi*3/8,20); linspace(pi*5/8, pi, 20)];
Y = sin(X); % Tanto la X como la Y son matrices
plot(X,Y) % Esto no es lo que se esperaba!!! Hay que tener
           en cuenta que se imprime por columnas
```



```
plot(X',Y') % Ahora sí
% Es equivalente a:
plot(X(1, :), Y(1, :), X(2, :), Y(2, :))
```



Opciones de línea (+info: doc LineSpec)

Opcionalmente se puede especificar cómo se desea representar la gráfica. Para ello se debe pasar una cadena de texto detrás de cada par de parámetros X_n e Y_n .

Color de línea	
y	Amarillo
m	Magenta
c	Celeste
r	Rojo
g	Verde
b	Azul
w	Blanco
k	Negro

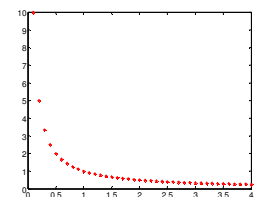
Tipo de línea	
-	Continua
--	A trazos
:	Punteada
-.	Trazo-ptó

Marcador de punto	
+	Signo más
o	Círculo
*	Asterisco
.	Punto
x	Aspa
s	Cuadrado
d	Diamante
^	Triángulo apuntando hacia arriba
v	Triángulo apuntando hacia abajo
>	Triángulo apuntando hacia dcha.
<	Triángulo apuntando hacia izq.
p	Estrella de 5 puntas
h	Estrella de 6 puntas

Veamos algunos ejemplos utilizando las opciones de línea:

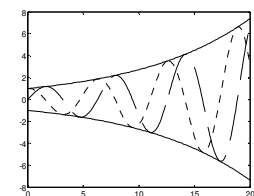
Gráfico simple con asteriscos rojos

```
x = 0:0.1:4;
y = 1./x;
plot(x, y, 'r*')
```



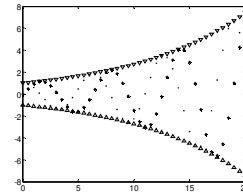
Distintos tipos de trazos

```
x = 0:0.1:20;
y = exp(0.1*x);
y1 = y.*sin(x);
y2 = y.*cos(x);
plot(x, y, 'k', x, -y, 'k', x, y1, 'k--', x, y2, 'k:')
```



Distintos tipos de marcadores

```
x = 0:0.5:20;
y = exp(0.1*x);
y1 = y.*sin(x);
y2 = y.*cos(x);
plot (x, y, 'kv', x, -y, 'k^', x, y1, 'k*', x, y2, '.k')
```



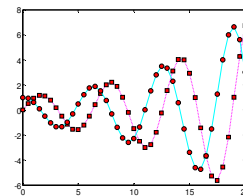
Propiedades de línea (+info: [doc line](#))

También se pueden especificar otras propiedades gráficas de la línea mediante parejas de nombre de propiedad y valor. Estas propiedades se asignarán a todas las líneas del gráfico. Algunas de las propiedades más habituales son:

- LineWidth: Especifica el grosor de la línea en píxeles.
- MarkerEdgeColor: Especifica el color del borde del marcador.
- MarkerFaceColor: Especifica el color del interior del marcador.
- MarkerSize: Especifica el tamaño del marcador.

Otras propiedades

```
x = 0:0.5:20;
y = exp(0.1*x);
y1 = y.*sin(x);
y2 = y.*cos(x);
% Las propiedades especificadas afectan a todas las líneas
plot (x, y1, ':ms', x, y2, '-co', 'LineWidth', 2,
      'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'r',
      'MarkerSize', 10)
```



pie

(+info: [doc pie](#))

Las formas más habituales para la utilización de `pie` son:

```
pie(X)
pie(X, Explode)
```

Esta función muestra un gráfico de tarta con los valores de `x`. Opcionalmente, se pueden resaltar porciones utilizando la segunda llamada. Por ejemplo:

Gráfico de tarta

```
x = [1 1 2 3 5 8];
pie(x);
```

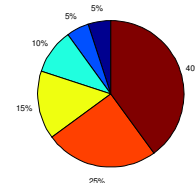
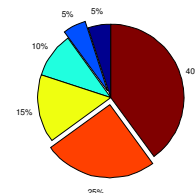


Gráfico de tarta con elementos resaltados

```
x = [1 1 2 3 5 8];
ex = [0 1 0 0 1 0];
pie(x, ex);
```



hist

(+info: doc hist)

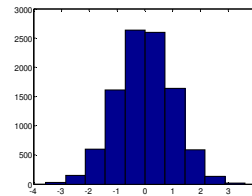
Las formas más habituales para la utilización de `hist` son:

```
hist(X)
hist(X, nbins)
```

Estas funciones realizan el histograma de los valores contenidos en `x`. Si no se especifica nada se hace una división en 10 contenedores igualmente espaciados. Si `x` es una matriz entonces se realiza un histograma para cada columna de la matriz.

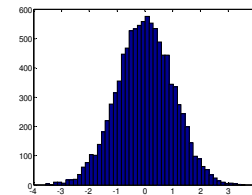
Histograma

```
x = randn(1,10000);
hist(x);
```



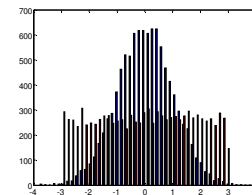
Histograma de 50 barras

```
% Generamos 10000 números aleatorios con distribución normal
x = randn(1,10000);
hist(x, 50);
```



Histograma de una matriz

```
% Generamos números aleatorios con distribución normal y
    uniforme
x = [randn(10000, 1) rand(10000, 1)*6-3]
hist(x, 50);
```



bar y barh

(+info: doc bar)

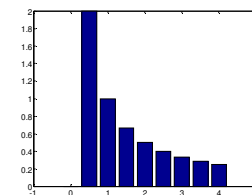
Las formas más habituales para la utilización de `bar` y `barh` son:

```
bar(Y)
bar(x,Y)
bar(..., width)
bar(..., 'style')
```

Estas funciones realizan un gráfico de barras verticales (`bar`) u horizontales (`barh`). Se puede regular la separación entre las barras con el parámetro `'width'`. Si la `Y` es una matriz entonces agrupa las barras o si se especifica `'stacked'` en el parámetro `style` entonces apila las barras.

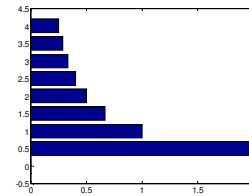
Barras verticales

```
x = 0.5:0.5:4;
y = 1./x;
bar(x, y)
```



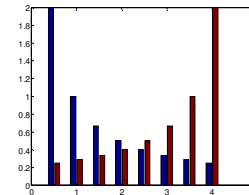
Barras horizontales

```
x = 0.5:0.5:4;  
y = 1./x;  
barh(x, y)
```



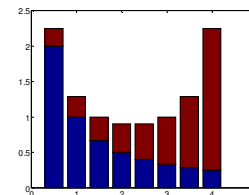
Grupos de barras

```
x = 0.5:0.5:4;  
y = 1./x;  
Y = [y' fliplr(y)'];  
bar(x, Y)
```



Barras apiladas

```
x = 0.5:0.5:4;  
y = 1./x;  
Y = [y' fliplr(y)'];  
bar(x, Y, 'stacked')
```



area

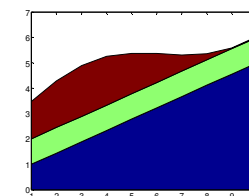
(+info: doc area)

```
area(X,Y)
```

Esta función genera un gráfico apilado, útil para mostrar las contribuciones de distintas componentes de un todo. Si x e y son vectores el resultado es igual a `plot` excepto que se rellena el área bajo la curva. Si y es una matriz, la función acumula los distintos valores de las columnas de y . x debe ser monótona.

Area

```
% Creamos una columna creciente, otra constante y otra que  
% varía con el seno  
x = 1:10;  
Y = [linspace(1,5,10)' ones(10, 1) 1+sin(0.5:0.5:5)']  
Area(x, Y)
```



contour

(+info: doc contour)

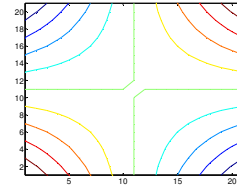
Algunas formas habituales de esta función son:

```
contour(Z)  
contour(Z, n)
```

Esta función dibuja un gráfico de isolíneas a partir de la matriz z , donde z se interpreta como alturas con respecto al plano xy . z debe ser al menos una matriz de 2×2 . El número de niveles de contorno se puede especificar utilizando la segunda forma de llamar a la función, si no se calculan automáticamente basándose en las alturas máximas y mínimas de z .

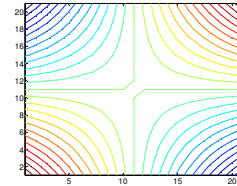
Isolíneas

```
x = -5:0.5:5;  
Z = x'*x;  
contour(Z)
```



Isolíneas con 25 niveles

```
x = -5:0.5:5;  
Z = x'*x;  
contour(Z, 25)
```



pcolor

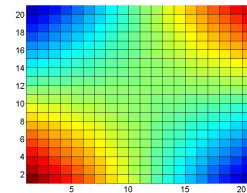
(+info: doc pcolor)

```
pcolor(C)
```

Se dibuja una matriz de celdas con colores determinados por la matriz c . Se muestran una fila y columna menos de celdas que filas y columnas de la matriz. Esta función (si no se especifican otros parámetros) obtiene el color para la celda (i, j) de la componente (i, j) de la matriz c y usando el mapa de color actual (+ info: doc colormap).

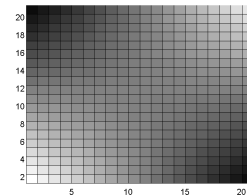
Celdas coloreadas

```
x = -5:0.5:5;  
C = x'*x;  
pcolor(C)
```



Celdas coloreadas en grises

```
x = -5:0.5:5;  
C = x'*x;  
colormap(gray);  
pcolor(C)
```



scatter

(+info: doc scatter)

Las formas más habituales de esta función son:

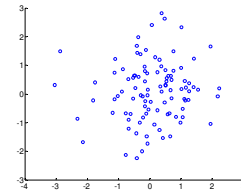
```
scatter(x, y)  
scatter(x, y, size)  
scatter(x, y, size, color)
```

Hace un gráfico de dispersión con las coordenadas de x e y . Los vectores deben ser del mismo tamaño. También se puede especificar el tamaño de los puntos y su color. Tanto el tamaño como el color se pueden especificar para todos los puntos en conjunto o para cada punto individualmente. Para lo primero basta con especificar con un escalar el tamaño y con un vector de 1x3 el color (en RGB). Para lo segundo hay que proporcionar para el tamaño de los puntos un vector (de igual longitud que x e y) y para

el color, o bien, un vector de índices (que harán referencia al colormap), o bien, una matriz de `length(x) x 3` con los colores (en RGB).

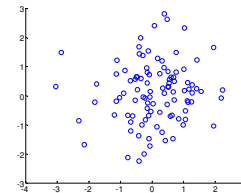
Puntos distribuidos aleatoriamente con distribución normal en 2D

```
x = randn(1,100);  
y = randn(1,100);  
scatter(x, y)
```



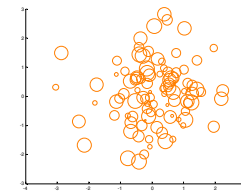
...con puntos de mayor tamaño

```
scatter(x, y, 100)
```



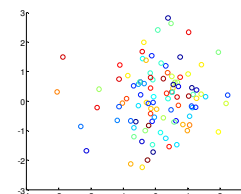
...con puntos naranjas y tamaños aleatorios

```
scatter(x, y, rand(1, 100)*1000, [1 0.5 0])
```



...con puntos de color aleatorio usando el colormap

```
scatter(x, y, 100, rand(1, 100))
```



plotyy

(+info: doc plotyy)

Algunas formas habituales de llamar a esta función son:

```
plotyy(X1,Y1,X2,Y2)  
plotyy(X1,Y1,X2,Y2,'function1','function2')
```

Realiza un gráfico utilizando dos ejes verticales distintos, uno para la serie Y_1 (eje de la izquierda) y otro para la serie Y_2 (eje de la derecha). Esta función es útil cuando se desean visualizar dos series con distinta escala y/o unidades en un mismo gráfico, por ejemplo en gráficos de temperatura y pluviometría. Se pueden especificar los nombres de función con que se quiere mostrar el gráfico. Se puede usar cualquier función que acepte llamadas del tipo `function(x, y)`.

Gráfico con dos ejes verticales con distinta escala

```
x = 1:12;  
pluv = sin(linspace(pi/2,pi+3*pi/2,12))*100+100;  
temp = sin(linspace(0, pi, 12))*40;  
plotyy(x, pluv, x, temp)  
plotyy(x, pluv, x, temp, 'bar', 'plot')
```

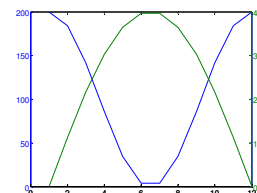


Gráfico con dos ejes verticales usando distintas funciones para dibujarlo

```
plotyy(x, pluv, x, temp, 'bar', 'plot')
```

