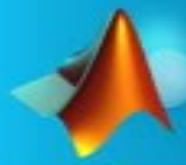
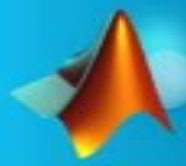


Herramientas computacionales para la matemática

MATLAB: Gráficas 2D

Verónica Borja Macías

Marzo 2013



Matlab

Gráficas bidimensionales

- Las tablas de datos muy grandes son difíciles de interpretar por lo que es necesario graficar la información para hacer que se entienda fácilmente.
- Con una gráfica es fácil identificar tendencias, elegir altos y bajos y aislar puntos de datos que pueden ser mediciones o cálculos de errores.
- Las gráficas también se pueden usar como una rápida verificación para determinar si una solución de computadora produce los resultados esperados.
- Las gráficas 2D de MATLAB están fundamentalmente orientados a la representación gráfica de vectores (y matrices).

- En el caso más sencillo los argumentos básicos de la función plot van a ser vectores. Cuando una matriz aparezca como argumento, se considerará como un conjunto de vectores columna (en algunos casos también de vectores fila).
- MATLAB utiliza un tipo especial de ventanas para realizar las operaciones gráficas.
- Ciertos comandos abren una ventana nueva y otros dibujan sobre la ventana activa, bien sustituyendo lo que hubiera en ella, bien añadiendo nuevos elementos gráficos a un dibujo anterior.

- plot(x) Dibuja los pares de puntos (i, x_i) unidos por segmentos (x_i representa las componentes del vector x)

Ejemplo

```
>> x=[-4 -2 0 1 3 5];
>> plot(x);
```

- plot(x,y) Dibuja los pares de puntos (x_i, y_i) unidos por segmentos (x_i e y_i representan las componentes de los vectores x e y respectivamente).

Ejemplo

```
>> x=[-4 -2 0 1 3 5];
>> y=[16 4 0 1 9 25];
>> plot(x,y);
```

- plot(z) Dibuja en el plano complejo los pares de puntos $(Re(z_i), Im(z_i))$ unidos por segmentos (z_i representa las componentes del vector z).

Ejemplo

```
>> z=[1 2+i 3 2-i 3-2*i];
>> plot(z);
```

- plot(A) Para cada j dibuja los pares de puntos (i, a_{ij}) unidos por segmentos. En la misma gráfica, cada poligonal se dibuja con color y tipo de línea diferente.

Ejemplo

```
>> A=[1 1 0.5; 2 4 -0.5; 3 9 0.5; 4 16 -0.5; 5 25 0.5];
>> plot(A);
```

- plot(x,A) Independientemente de que x sea un vector fila o columna, ejecuta plot(x,y) donde y es una fila o columna de A. Para la elección de la fila y columna se tiene en cuenta la coincidencia de dimensiones. En la misma gráfica, cada línea se dibuja con un color y con un tipo de línea diferente.

Ejemplo

```
>> x=[0 0.1 0.2 0.3 0.4];
>> A=[1 1 0.5; 2 4 -0.5; 3 9 0.5; 4 16 -0.5; 5 25 0.5];
>> plot(x,A);
```

- plot(A,x) En las mismas condiciones que el caso anterior dibuja los pares ordenados siendo x el valor de las ordenadas.

Ejemplo

```
>> x=0:0.1:2;
>> A=[sin(pi*x); 0.5+0.5*x];
>> plot(A,x);
```

- plot(A,B) Ejecuta plot(x,y), donde x es una columna de A e y es una columna de B. En la misma gráfica, cada poligonal se dibuja con un color y con un tipo de línea diferente.

- fplot('fcn', lim) Dibuja la gráfica de la función especificada en la cadena fcn en los intervalos de abcisas y ordenadas determinados por lim.

Ejemplo

```
>> fplot('sin(x^2)', [0 10])
>> fplot('sin(x^2)', [0 10 -1.5 0.5])
```

Ejercicio

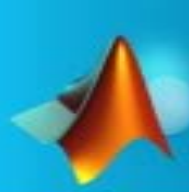
1. $f(x) = \sqrt{x^2 - 1}$ con x en $[-3.3, -1.3]$
2. $f(x) = x^2 \sin\left(\frac{1}{x}\right)$ con x en $[-2, 2]$
3. $f(x) = x^2 \sin\left(\frac{1}{x}\right)$ con x en $[-0.1, 0.1]$

Opciones de los comandos plot y fplot

plot(x,y,'esp','prop',val) Dibuja la gráfica de abcisas x y ordenadas y con las opciones de estilo dadas por esp con la propiedad prop con valor val (puede haber mas de una propiedad).

fplot('fcn', lim,'esp') Dibuja la gráfica de la función fcn con el estilo de líneas esp.

plot(x1,y1,'esp1', x2,y2,'esp2',...) Dibuja la gráfica de abcisas x1 y ordenadas y1 con las opciones de estilo dadas por esp1, la gráfica de abcisas x2 y ordenadas y2 con las opciones de estilo dadas por esp2 y así con el resto de ternas. Si se omiten las opciones de estilo, MATLAB escoge el color y estilo para cada gráfica.



Matlab

Gráficas bidimensionales: Especificadores de línea

Marcadores	Líneas	Colores
. punto	- Sólida	y yellow
o círculo	-- Discontínua	m magenta
x equis	-. Punto raya	c cyan
+ mas	: Punteada	r red
* asterisco		g green
s cuadrado		b blue
d diamante		w white
v triángulo (abajo)		k black
^ triángulo (arriba)		
< triángulo (izquierda)		
> triángulo (derecha)		
p pentágono		
h hexágono		

Propiedad	Descripción	valor
LineWidth linewidth	Especifica el color de línea.	El valor es en puntos (por default 0.5)
MarkerSize markersize	Especifica el tamaño de las marcas	Tamaño dado en puntos
MarkerEdgeColor markeredgecolor	Especifica el color de la marca o el contorno de la marca	Un color valido como en especificadores de línea
MarkerFaceColor markerfacecolor	Especifica el color de relleno de la marca	Un color valido como en especificadores de línea

Ejemplos

```
>> plot(x,sin(x),'o--g','LineWidth', 2, 'MarkerSize', 8,
   'MarkerEdgeColor', 'b', 'markerfacecolor', 'y')

>> fplot('sen(x)', [-2 2], 'o--g')

>> x=linspace(0,3,50); % Construimos el vector x
>> e1=exp(-x.^2); % Construimos los vectores de abcisas
>> e2=(x.^2).*exp(-x.^2);
>> e3=x.*exp(-x.^2);
>> e4=exp(-x);
>> plot(x,e1, '+-g',x,e2, '*:k',x,e3, 'o-.y',x,e4, 'x'); % Graficamos
```

- El comando subplot le permite subdividir la ventana de graficación en una retícula de m filas y n columnas. La función subplot(m,n,p) separa la figura en una matriz mxn. La variable p identifica la porción de la ventana donde se dibujará la siguiente gráfica.
- Por ejemplo, si se usa el comando subplot (2,2,1) la ventana se divide en dos filas y dos columnas, y la gráfica se dibuja en la ventana superior izquierda. Las ventanas se numeran de izquierda a derecha, de arriba abajo.

Ejemplo

```
>> x=0: pi/20:2*pi ;
>> subplot(2,2,1);
>> plot (x,sin(x));
>> subplot(2,2,2);
>> plot (x,cos(x));
>> subplot(2,2,3);
>> plot(x,sin(2 *x));
>> subplot(2,2,4);
>> plot(x,cos(2 *x));
```

- MATLAB tiene sus opciones por defecto para controlar los ejes, que en algunas ocasiones puede interesar cambiar. El comando básico es el comando axis. Por defecto, MATLAB ajusta la escala de cada uno de los ejes de modo que varíe entre el mínimo y el máximo valor de los vectores a representar.
- axis Devuelve los límites del dibujo actual en un vector fila. Para gráficos de dos dimensiones tiene los elementos $[x_{min}\ x_{max}\ y_{min}\ y_{max}]$.
- axis(v) Establece la escala de los ejes conforme al vector v que será $[x_{min}\ x_{max}\ y_{min}\ y_{max}]$.
- axis(axis) Bloquea la escala actual cuando se añaden dibujos posteriores a uno que se ha mantenido con el comando hold.

- axis(cad) Establece la escala de los ejes con diferentes resultados dependiendo de la cadena cad utilizada:
  - 'auto' Vuelve a la escala automática.
  - 'equal' Da la misma escala en ambos ejes.
  - 'ij' Intercambia la parte positiva y negativa del eje y.
  - 'xy' Deshace el anterior.
  - 'image' Igual que equal pero se ajusta al dibujo.
  - 'square' Modifica la ventana gráfica para hacer que la caja del dibujo sea cuadrada.
  - 'normal' Modifica la ventana gráfica para que la caja del dibujo vuelva al tamaño habitual.
  - 'off' No se muestran los ejes.
  - 'on' Deshace el comando anterior.

- Existen además otras funciones orientadas a añadir títulos al gráfico, a cada uno de los ejes, a dibujar una cuadrícula auxiliar, a introducir texto, etc.
- grid on Dibuja una red en la ventana gráfica.
- grid off Borra la red de la ventana gráfica.

Ejemplo

```
>> t=0:0.2:2*pi+0.2; x=sin(t); y=cos(t);
>> subplot(3,1,1); plot(x,y,'-');
>> subplot(3,1,2); plot(x,y,'-'); axis square;
>> subplot(3,1,3); plot(x,y,'-'); axis normal; grid;
>> axis([-2 2 -3 3]);
```

- title(txt) Escribe la variable cadena txt en la cabecera del gráfico.
- Puede usar letras griegas en sus etiquetas al poner una diagonal inversa (\) antes del nombre de la letra. Por ejemplo title('\alpha \beta \gamma') crea el título de la gráfica $\alpha\beta\gamma$, para crear un subíndice, use _ y llaves, title('x_{2a}') produce x_{2a}.Para cambiar el tamaño o tipo de letra usamos los comandos \fontsize{} y \fontname{}.
- MATLAB tiene la habilidad de crear expresiones matemáticas más complicadas para usar como títulos, etiquetas de ejes y otras cadenas de texto, al usar el lenguaje TEX.

- xlabel(txt), ylabel(txt) Escribe la variable cadena txt como una etiqueta junto al eje x e y respectivamente.
- text(x,y,txt) Escribe la variable cadena txt en la posición (x,y) de la ventana gráfica. Las coordenadas x e y están proporcionadas en las mismas unidades en las que está dibujado el gráfico. Si x e y son vectores, la variable cadena se escribe en todos los pares de puntos (x_i, y_i). Si txt es un vector con varias cadenas con el mismo número de filas que x e y, se escribe una de ellas en cada una de las posiciones. Se puede elegir el tamaño y tipo de letra empleado.

- El comando text puede tener varios argumentos del tipo text(x,y,txt,prop,val) donde prop y val pueden ser las siguientes propiedades con sus respectivos valores.

Propiedad	Descripción	valor
Rotation	Especifica orientación del texto	Grados (0 por default)
FontAngle	Cambia entre cursivas y normal	normal/italic
FontName	Especifica la fuente de letra	Fuentes disponibles
FontSize	Especifica el tamaño de letra	Puntos (10 por default)
FontWeight	Especifica grosor de la letra	light /normal/bold
Color	Especifica color de texto	Especificadores de color
BackgroundColor	Especifica color de fondo	Especificadores de color
EdgeColor	Especifica color del borde	Especificadores de color
LineWidth	Especifica grosos del borde	Puntos (0.5 por default)

- Borrar texto (u otros elementos gráficos) es un poco más complicado; de hecho, hay que preverlo de antemano. Para poder hacerlo hay que recuperar previamente el valor de retorno del comando con el cual se ha creado. Después hay que llamar a la función delete con ese valor como argumento.

Ejemplo

```
>> v = text(1,.0,'seno')
v =
76.0001
>> delete(v)
```

- gtext(txt) Escribe la variable cadena txt en la posición de la ventana gráfica que elija el usuario mediante el ratón.
- legend(st1,st2, ...) Escribe en un pequeño recuadro, las variables cadenas st1, st2, etc, al lado de los estilos de linea utilizados en cada una de las gráficas. Este pequeño recuadro puede moverse mediante el ratón.
- legend(L1,st1, L2,st2, ...) Escribe las leyendas L1, L2, … especificando los estilos de línea st1, st2,...
- legend off Elimina la leyenda del dibujo actual.

- line() permite dibujar una o más líneas que unen los puntos cuyas coordenadas se pasan como argumentos. Permite además especificar el color, grosor, tipo de trazo, marcador, etc.
- Es una función de más bajo nivel que la función plot(), pero ofrece una mayor flexibilidad. En su versión más básica, para dibujar un segmento de color verde entre dos puntos, esta función se llamaría de la siguiente manera:

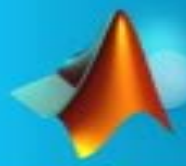
Ejemplo

```
>> line([xini, xend]', [yini, yend]', 'color', 'g')
>> line([xini1 xini2; xend1 xend2], ([yini1 yini2; yend1 yend2]);
```

- Finalmente, si cada columna de la matriz X contiene la coordenada x inicial y final de un punto, y lo mismo las columnas de la matriz Y con las coordenadas y, la siguiente sentencia dibuja tantas líneas como columnas tengan las matrices X e Y: line([X], [Y]);
- Se pueden controlar las características de la línea por medio de pares parámetro/valor, como por ejemplo:

Ejemplo

```
>> line(x,y,'Color','r','LineWidth',4,'MarkerSize',12,'LineStyle',...
   '—','Marker','*')
```



Matlab

Control de gráficos bidimensionales

- figure(n) Permite mostrar la ventana gráfica actual y crear nuevas ventanas gráficas.
- figure(gcf) (get current figure) permite hacer visible la ventana de gráficos desde la ventana de comandos
- clf Borra el contenido la última ventana gráfica utilizada.
- close(n) Cierra la ventana gráfica n.
- hold on Permite superponer gráficos en una misma ventana hasta que se desactiva la opción.
- hold off Desactiva la opción hold on.
- ishold Es un comando lógico que devuelve 1 si para el actual gráfico está activada la opción hold on o 0 en caso contrario.

Ejemplos

```
>> x=[-4*pi:pi/20:4*pi];
>> plot(x,sin(x),'r',x,cos(x),'g')
>> title('Función seno(x) -en rojo- y función coseno(x) -en verde-')
>> xlabel('ángulo en radianes'), figure(gcf)
>> ylabel('valor de la función trigonométrica'), figure(gcf)
>> axis([-12,12,-1.5,1.5]), figure(gcf)
>> axis('equal'), figure(gcf)
>> axis('normal'), figure(gcf)
>> axis('square'), figure(gcf)
>> axis('off'), figure(gcf)
>> axis('on'), figure(gcf)
>> axis('on'), grid, figure(gcf)
```

Ejemplos

```
>> x=0:pi/100:2*pi; y1=cos (x*4);
>> plot(x,y1), figure(gcf);
>> plot(x,y2), figure(gcf);
>> figure(2)
>> plot(x,y1), figure(gcf);
>> y2=sin(x);
>> hold on;
>> plot(x,y2), figure(gcf);
>> legend('coseno','seno'), figure(gcf);
>> hold off, figure(gcf);
>> legend off, figure(gcf);
>> close(2);
```

- zoom on Permite ampliar la ventana con el botón izquierdo del ratón y reducirla con el botón derecho. También permite seleccionar el área que queremos ampliar.
- zoom off Desactiva el comando anterior.
- zoom out Recupera el tamaño inicial.