Tema 5: Introducción a los gráficos en 2 dimensiones

1. Representación gráfica en Matlab

Matlab ofrece gran número de posibilidades a la hora de realizar representaciones gráficas. Dibuja curvas planas y superficies. Permite agrupar y superponer representaciones. Todo ello con variaciones de estilo y de coordenadas. Permite a su vez realizar gráficos de tipo estadístico: de barra, histogramas, etc.

Por las características propias del programa, los gráficos, en concreto los 2D, están orientados a la representación gráfica de vectores. Se utiliza una ventana especial para la creación de los gráficos: la ventana gráfica o de dibujo y, dichos gráficos se guardan en ficheros de extensión .fig. Ciertos comandos ejecutados sobre la línea de comandos son los que abren esta ventana, otros dibujan sobre la ventana activa, bien sustituyendo lo que había en ella, bien añadiendo nuevos elementos gráficos a los que había. Los iremos estudiando con más detalle

2. Funciones básicas para las gráficas 2D

El comando básico para la representación de gráficos 2D es el comando plot. Su sintaxis puede ser:

plot(x,y): dibuja el conjunto de puntos (x,y) donde las abscisas de los puntos se encuentran en el vector x y las ordenadas en el y.

Para representar una función f(x) es necesario conocer los valores de puntos de la forma (x,f(x)). Para ello puede seguirse alguno de estos caminos:

- Definir un vector x con el rango de variación donde se desea pintar la función. Para ello puede ser muy útil el comando linspace(xmin,xmax,n). Crear el vector y evaluando f en x. Por ejemplo:

```
>> x=linspace(0,10,100);
>> y=sin(x);
>> plot(x,y)
```

Por defecto, MATLAB dibuja uniendo los puntos con línea continua de color azul y un grosor determinado, opciones todas que se podrán alterar como veremos.

- También es posible dibujar una función con el comando fplot cuya sintaxis es la siguiente: **fplot('f(x)',[xmin,xmax])**. Así, este comando admite como argumento un nombre de función o de un fichero .m en el que está definida la función a representar. Por ejemplo:

  ```
  >> fplot('sin(x)',[-3*pi,3*pi,-1,1])
  ```

En general, si no se cierra la ventana de dibujo generada al evaluar un comando como los anteriotes, si se vuelve a ejecutar uno de ellos, se dibuja sobre dicha ventana perdiéndose el primer dibujo. Si se desea representar varias funciones a la vez las opciones son:

- **plot(x,y,x,z)** donde x el vector de las abscisas, común para las dos representaciones, y es el de las ordenadas de la primera representación y z las de la segunda.
- **fplot('[f1(x),f2(x),...]',[xmin,xmax])** donde f1, f2, … son las funciones a representar en el intervalo de variación marcado por xmin y xmax.
- Mediante el comando: **hold on**, **hold off**. Todos los gráficos que se ordene dibujar entre los comandos hold on y hold off se representan en la misma figura. Si hay una figura abierta se dibujan en ésta.

  Ejemplo:

  ```
  >> hold on
  >> x=[-3*pi:1:3*pi];
  >> plot(x,sin(x))
  >> plot(x,tan(x),'r')
  >> hold off
  ```

- El comando **subplot**. Una ventana gráfica se puede dividir en m particiones horizontales y n verticales para representar mxn figuras. Cada una de las particiones tendrá sus ejes aunque las propiedades serán comunes a todas ellas. La sintaxis es: **subplot(m,n,i)**, donde m y n son el número de subdivisiones e i la subdivisión activa. Por ejemplo:

  ```
  >> x=0:0.1:2*pi;
  >> y=sin(x);z=cos(x);t=exp(-x);v=x^2;
  >> subplot(2,2,1), plot(x,y)
  ```

>> subplot(2,2,2), plot(x,z)

>> subplot(2,2,3), plot(x,t)

>> subplot(2,2,4), plot(x,v)

La ventana gráfica sería la de la figura 18

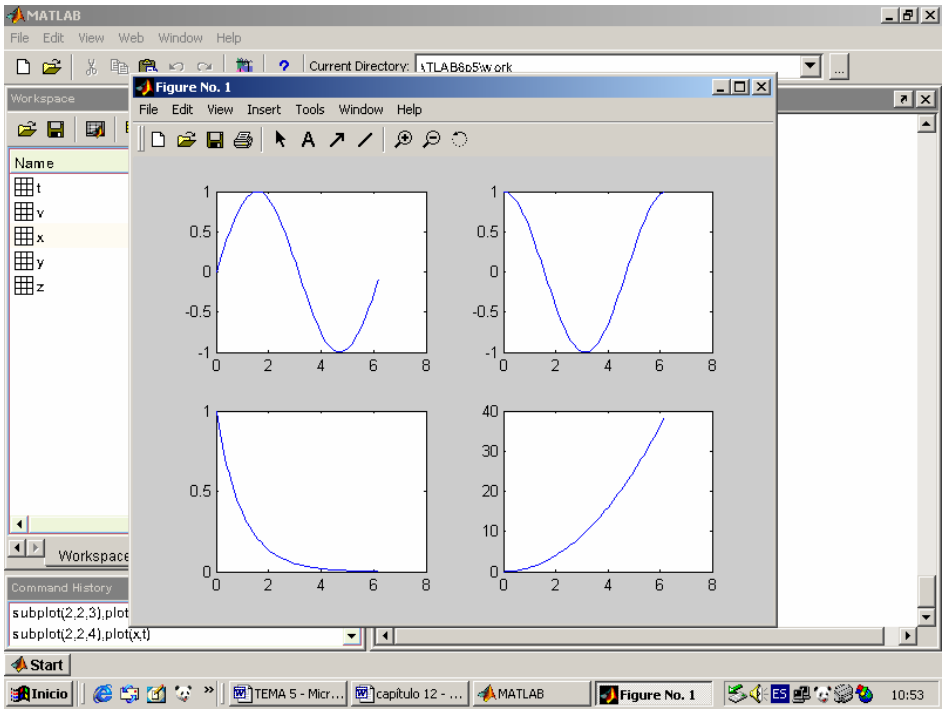

Figura 18

2.1. Opciones de dibujo

Hemos visto que el comando plot(x,y) dibuja los gráficos con unas características predefinidas en el programa, es posible alterarlas a partir de **plot(x,y,s)** donde s se compone de dos dígitos entre comillas. Uno fija el color de la línea y otro el carácter a usar en el gráfico. Por ejemplo:

>> plot(x,y,'-*g')

dibuja los puntos unidos con una línea continua, marcando los puntos con *, y en verde.

Otros tipos de marcadores son: . * x o + (marcan los puntos en el gráfico).

Otros tipos de línea: los puntos se unen con una línea con las siguientes posibilidades de apariencia:

- (línea continua)

- - (línea formada por trazos discontinuos)

-. (línea formada por puntos y trazos)

: (línea formada por puntos)

Los colores vienen dados por: **y**: amarillo, **g**: verde, **m**: magenta, **b**: azul, **c**: cían, **w**: blanco, **r**: rojo, **k**: negro.

Se puede modificar el grosor de línea incluyendo la cadena: 'Linewidth', número_indicativo_del_grosor. Por ejemplo: plot(x,y,'linewidth',2)

En general, se puede obtener una excelente descripción del comando plot y otros relacionados con el mediante la ejecución de help plot:

```
PLOT   Linear plot.
   PLOT(X,Y) plots vector Y versus vector X. If X or Y is a matrix,
   then the vector is plotted versus the rows or columns of the matrix,
   whichever line up.  If X is a scalar and Y is a vector, length(Y)
   disconnected points are plotted.
   PLOT(Y) plots the columns of Y versus their index.
   If Y is complex, PLOT(Y) is equivalent to PLOT(real(Y),imag(Y)).
   In all other uses of PLOT, the imaginary part is ignored.
   Various line types, plot symbols and colors may be obtained with
   PLOT(X,Y,S) where S is a character string made from one element
   from any or all the following 3 columns:
        b     blue          .     point              -     solid
        g     green         o     circle             :     dotted
        r     red           x     x-mark             -.    dashdot
        c     cyan          +     plus               --    dashed
        m     magenta       *     star
        y     yellow        s     square
        k     black         d     diamond
                            v     triangle (down)
                            ^     triangle (up)
                            <     triangle (left)
                            >     triangle (right)
                            p     pentagram
                            h     hexagram

   For example, PLOT(X,Y,'c+:') plots a cyan dotted line with a plus
   at each data point; PLOT(X,Y,'bd') plots blue diamond at each data
```

point but does not draw any line.
PLOT(X1,Y1,S1,X2,Y2,S2,X3,Y3,S3,...) combines the plots defined by
the (X,Y,S) triples, where the X's and Y's are vectors or matrices
and the S's are strings.
For example, PLOT(X,Y,'y-',X,Y,'go') plots the data twice, with a
solid yellow line interpolating green circles at the data points.
The PLOT command, if no color is specified, makes automatic use of
the colors specified by the axes ColorOrder property. The default
ColorOrder is listed in the table above for color systems where the
default is blue for one line, and for multiple lines, to cycle
through the first six colors in the table. For monochrome systems,
PLOT cycles over the axes LineStyleOrder property.
PLOT returns a column vector of handles to LINE objects, one
handle per line.
The X,Y pairs, or X,Y,S triples, can be followed by
parameter/value pairs to specify additional properties
of the lines.
See also SEMILOGX, SEMILOGY, LOGLOG, PLOTYY, GRID, CLF, CLC, TITLE,
XLABEL, YLABEL, AXIS, AXES, HOLD, COLORDEF, LEGEND, SUBPLOT, STEM.

- Títulos y etiquetas

Matlab permite manejar correctamente anotaciones sobre los gráficos y los ejes mediante la colocación adecuada de títulos y etiquetas, rejillas o leyendas. Los comandos más usados son:

title (‘texto’): añade el texto entre comillas como título del gráfico.

xlabel(‘texto’): añade el texto entre comillas como texto al lado del eje x.

ylabel(‘texto’): añade el texto entre comillas como texto al lado del eje y.

legend(‘texto’): sitúa la leyenda especificada en el texto.

grid: crea rejillas en los ejes

gtext(‘texto’): permite situar el texto especificado en el punto que señalemos con el ratón dentro de la ventana de trabajo.

- Control de ejes

También aquí Matlab tiene sus opciones por defecto. En muchas ocasiones es interesante alterarlas. Ya hemos visto que la variación en el eje x se elige al fijar las coordenadas x de los puntos. Por defecto el programa ajusta la escala de cada uno de los ejes de modo que varíe entre un máximo y el mínimo valor a representar (es el

modo auto). Para definir otros se utiliza el comando axis cuya sintaxis es: **axis ([xmin,xmax,ymin,ymax])**. axis('auto') devuelve la escala al valor por defecto.

Destacar que:

axis off elimina los ejes del dibujo y **axis on** los incorpora.

- Entrada de puntos con el ratón

Matlab permite introducir las coordenadas de los puntos sobre los que se encuentra el cursor, al pinchar o al pulsar alguna tecla. El comando que lo realiza es ginput. Algunas formas de utilizarlo son:

[x,y]=ginput: lee los puntos cada vez que se pincha o se pulsa alguna tecla. Finaliza al pulsar intro.

[x,y]=ginput(n): lee las coordenadas de n puntos

3. La ventana gráfica de Matlab.

La mayoría de las opciones de cambio en una gráfica presentadas con anterioridad, pueden realizarse desde la pantalla de dibujo. Posibilidades que se han mejorado en las versiones 6 del programa.

Resulta muy útil inspeccionar cada uno de los menús de esta ventana ejecutar las opciones directamente en ella. Así, por ejemplo, el menú insert permite directamente insertar en la figura etiquetas en los ejes, leyendas o título entre otras cosas (figura 19).

En el menú edit aparecen entre otros los comandos Figure Properties, axes Properties y Current Object Properties, con ellos se abre paso a los editores correspondientes donde se nos da la posibilidad de cambiar las opciones de dibujo (figura 20).

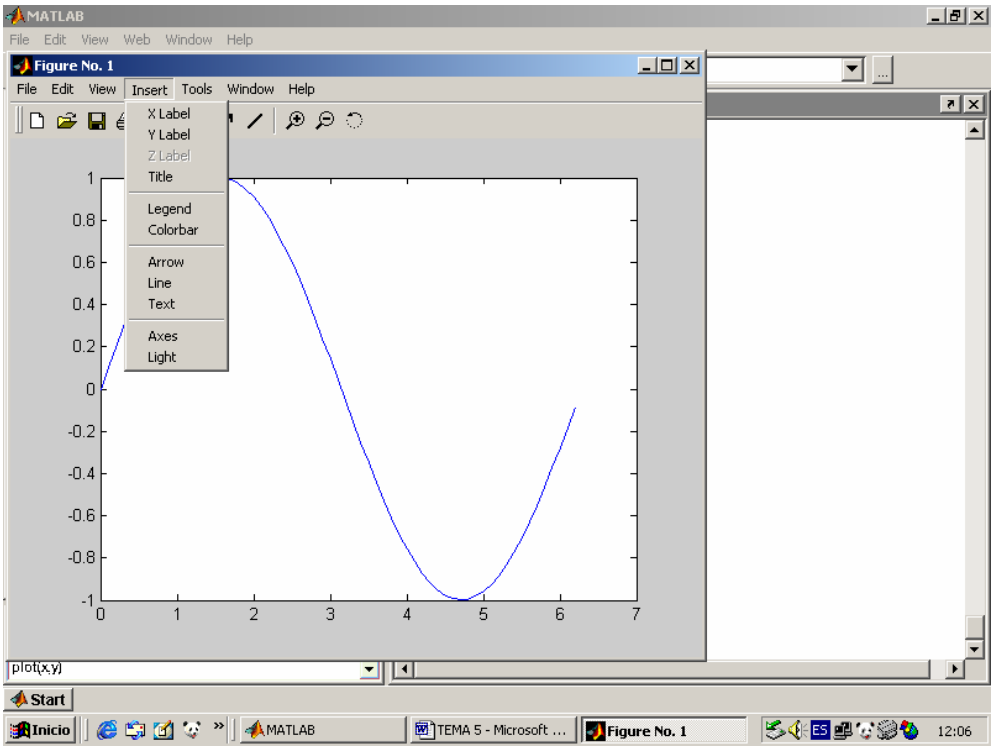

Figura 19

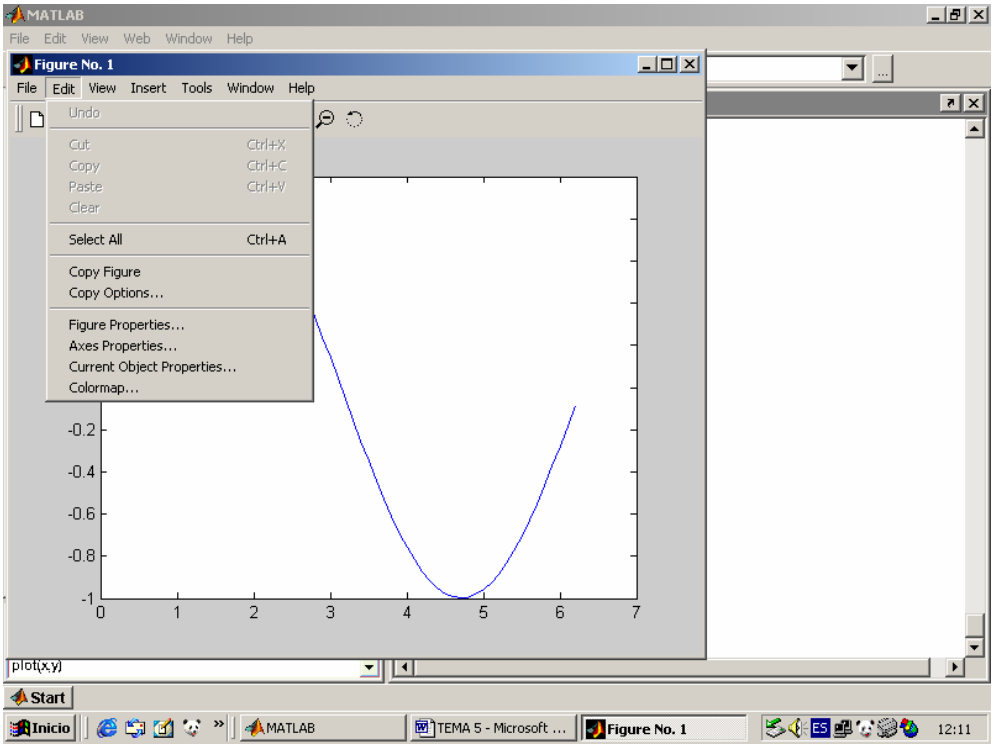

Figura 20

Veamos el editor de ejes, figura 21:

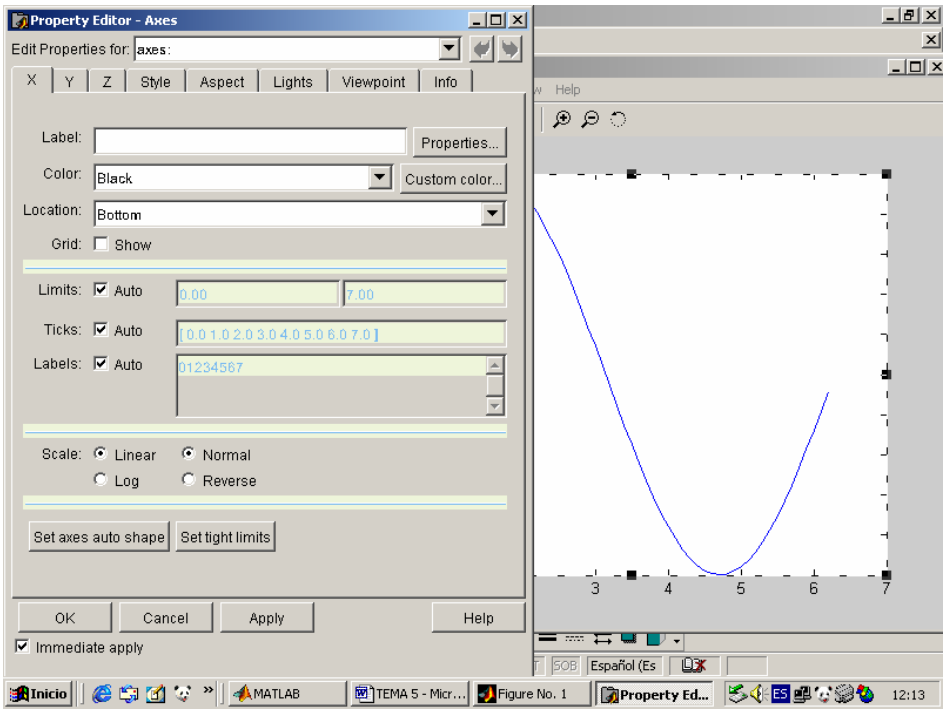

Figura 21

Las pestañas x,y,z nos permiten actuar sobre cada uno de los ejes, poniendo etiqutas (Label), cambiando el color (Color), cambiando los límites (Limit),...

Desde la pestaña superior podemos acceder a otros editores como los de linea (line). Desde los que se pueden cambiar las opciones correspondientes (figura 22).

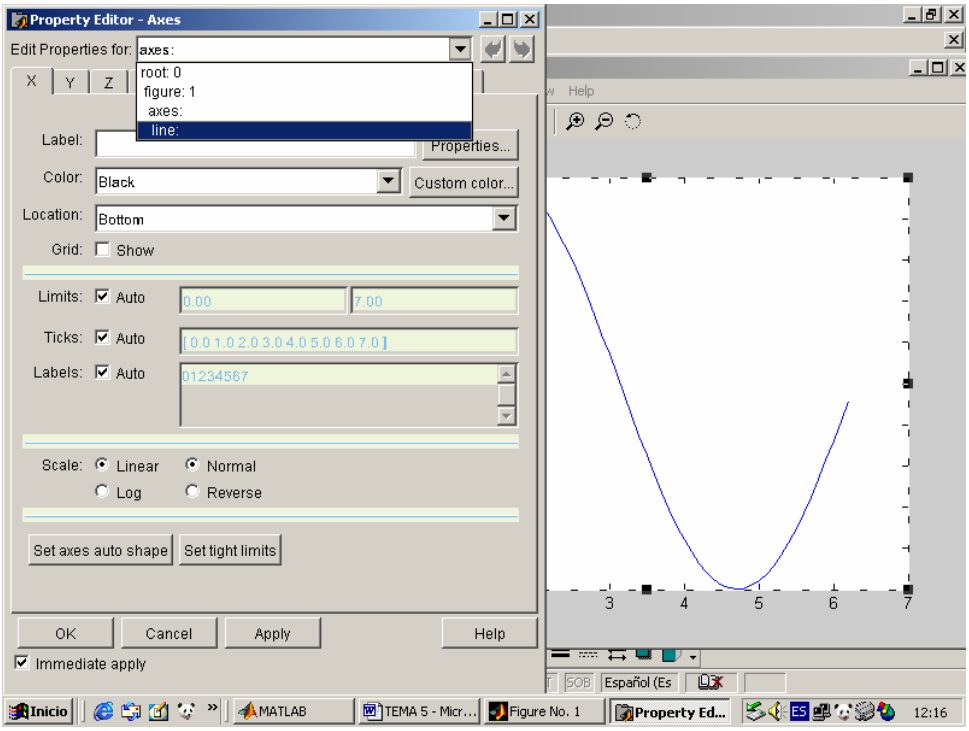

Figura 22

A los mismos editores se llega desde edidt, Current Object Properties, si tenemos seleccionado, pinchando primeramente el botón de la flecha de la ventana de dibujo, el elemento que se quiere ajustar, bien los ejes o bien una línea de las que componen la gráfica.

Práctica 5: Gráficos 2D

1. Representar gráficamente los puntos p_1=(1,1), p_2=(3,2), p_3=(0,4), p_4=(-3,6) primeramente conectados y luego aislados.
2. Representar gráficamente las siguientes funciones en ventanas diferentes, f(x)=sen(x), g(x)=x^2+3x en el intervalo [0,2π].
3. Representar la gráfica de la función f(x)=xsen(x) en el intervalo [0,2π], con rejilla.
4. Representar f(x)=sen(x)cos(x) en [0,2π], con etiquetas en los ejes, título y en color rojo.
5. Dibujar el polinomio x^2+5x-3 con 200 puntos, en color rojo, con trazo discontinuo, con título y con rejilla.
6. Representar en [0,6] y en la misma gráfica las funciones:
   a. f(x)=3xe^x en azul.
   b. g(x)=sen(x+3) en rojo y con trazo discontinuo.

   Poner leyendas.
7. Representar la función f(x)=3sen(x)-sen(3x) en el intervalo [0,2π] con 200 puntos, con título y etiquetas en los ejes.
   a. Eliminar los ejes con sus etiquetas y volver a activarlos.
   b. Hacer que el eje de abscisas sea el intervalo [0,3] y el rango de las imágenes el intervalo [0,1].
   c. Volver a dibujarla como estaba.
   d. Representarla con la misma escala en ambos ejes.
8. Representar en la misma gráfica pero en distintas ventanas las siguientes funciones: x, x^2, x^3, x^4, x^5, x^6. Las pares en color rojo y con rejilla, las impares en azul y sin rejilla.
9. Resolver gráficamente la ecuación $\frac{x - sen(x)}{2} = 0.02$ en el intervalo $\left[0, \frac{\pi}{4}\right]$. Para ello se aconseja seguir los siguientes pasos:
   a. Dibujar la gráfica de la función f(x)=$\frac{x - sen(x)}{2}$ en el intervalo dado.
   b. Dibujar la recta y=0.02 en color rojo en el mismo intervalo y en la misma ventana gráfica.
   c. Poner nombre (x_0,y_0) al punto de corte con el comando gtext.
   d. Determinar gráficamente el punto de corte (x_0,y_0) de ambas curvas.

e. Poner leyendas, etiquetas a los ejes y un título que indique cuál es el punto de corte solución de la ecuación.

10. Representar la función $x\text{sen}\frac{1}{x}$ en el intervalo [-1,1] y utilizar el zoom para observar lo que sucede en las cercanías de (0,0).

11. Representar la curva de ecuaciones paramétricas $\begin{cases} x = sen(3t) \\ y = sen(2t) \end{cases}$