INFORMÁTICA Y PROGRAMACIÓN

PRÁCTICA 3. GRÁFICAS

1. INTRODUCCIÓN

En esta práctica trabajaremos las posibilidades gráficas de ***Matlab*** (también disponibles en Octave). Tanto Matlab como Octave disponen de una amplia variedad de funciones muy sencillas de implementar, y que permiten representar gráficamente funciones matemáticas o conjuntos de datos complejos tanto en 2D como en 3D.

Un gráfico es creado por Matlab en una ventana de figura, para ello se usa el comando figure(n). Escribe por ejemplo en la Command Window de Matlab:

>> **figure**(1)

En cada ventana de figura se pueden colocar uno o varios gráficos independientes.

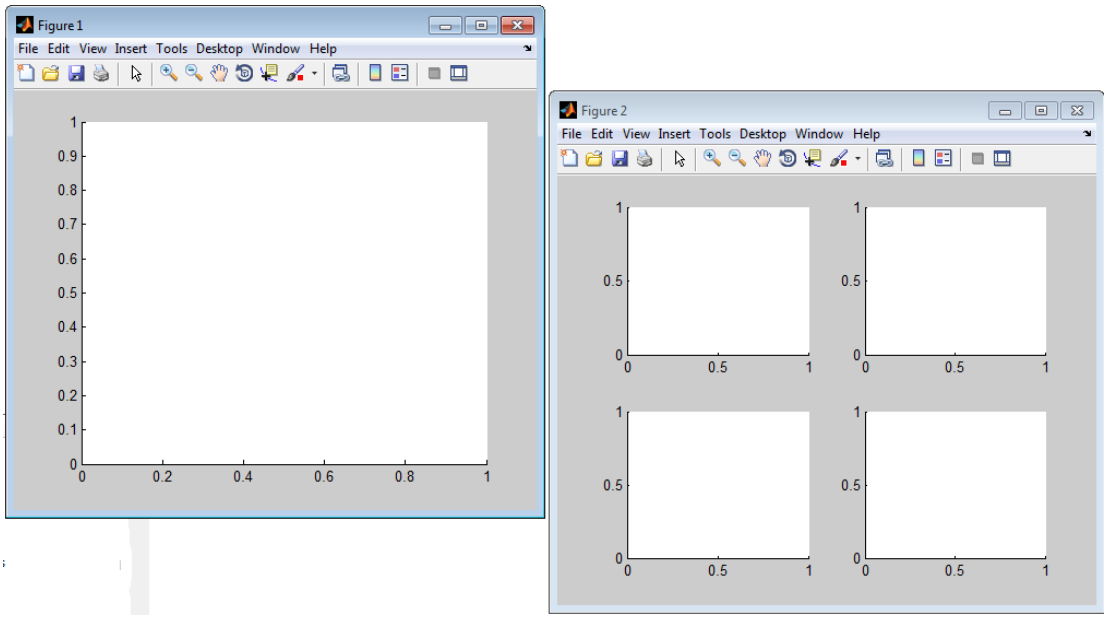

Figura 1

Tanto en Matlab como en Octave, es posible indicar una secuencia de instrucciones o comandos tanto a través de la Command Window como por medio de un fichero .m (extensión). En clases anteriores se ha utilizado la ventana de comandos (Command Window) (también modo calculadora) para indicar como ejecutar los diferentes comandos, funciones etc. En esta práctica 3 utilizaremos ficheros .m, para ello manejaremos el editor de Matlab (véase Anexo).

2. GRÁFICOS 2D

Para gráficos 2D, el comando básico de Matlab es:

plot(x,y,c1,c2…)

Donde x e y son las coordenadas de un punto o serie de puntos. Los valores c son opcionales y sirven para especificar atributos como color, tipo, tamaño etc. de la serie de puntos. Hágase >> **help plot** en la Command Window de Matlab para ver las diferentes opciones de **plot**.

Ejemplo resuelto 1

Escríbase un fichero .m que dibuje el punto x=4, y=7, tipo pentagrama y color rojo. Luego, ejecútese.

```
%ejemplo1.m
plot(4,7,'pr');
```

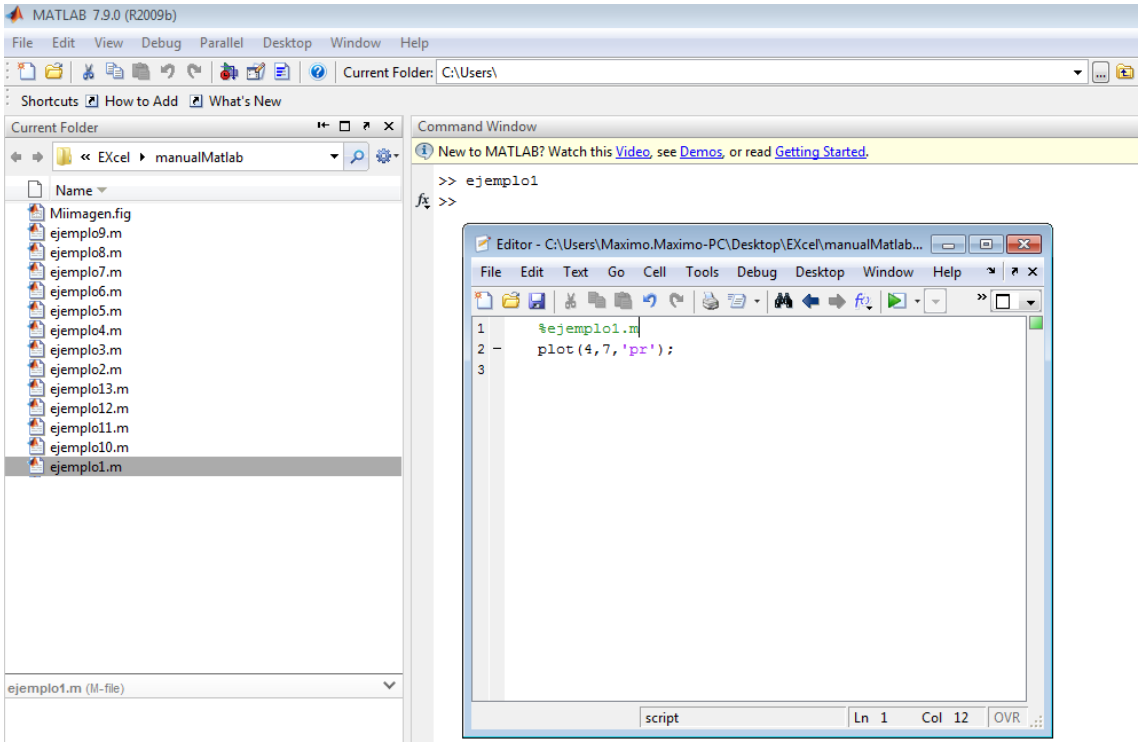

Figura2.1

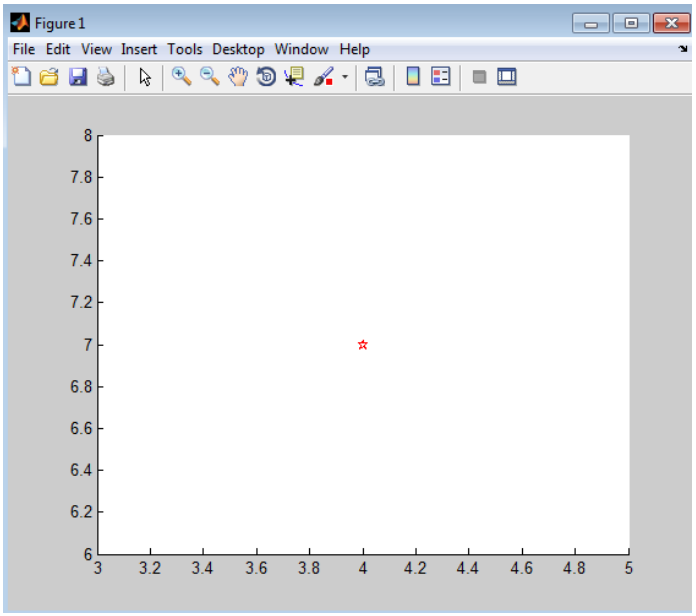

Figura2.2

Como cada vez que se ejecuta **plot**, se genera una nueva ventana de figura, si queremos dibujar más de una gráfica en la misma ventana es necesario usar el comando:

hold on

Ejemplo resuelto 2

Escríbase un fichero .m que dibuje el punto x=4, y=7 tipo pentagrama y color rojo. Luego, trácense dos líneas discontinuas vertical y horizontal que unan el punto 4,7 con los ejes x e y respectivamente. Cámbiese también la apariencia de la gráfica con el comando:

Axis([xmin xmax ymin ymax])

```
%ejemplo2.m
hold on; %permite varias gráficas en la misma figura
plot(4,7,'pr');
plot([4,4],[0,7],'k--');
plot([0,4],[7,7],'k--');
axis([0 8 0 11]);
```

El resultado del código anterior es la figura 3.

Luego, una vez ejecutado el programa elimínese la línea 2 del programa (**hold on**), ejecútese de nuevo y reflexionar que sucede.

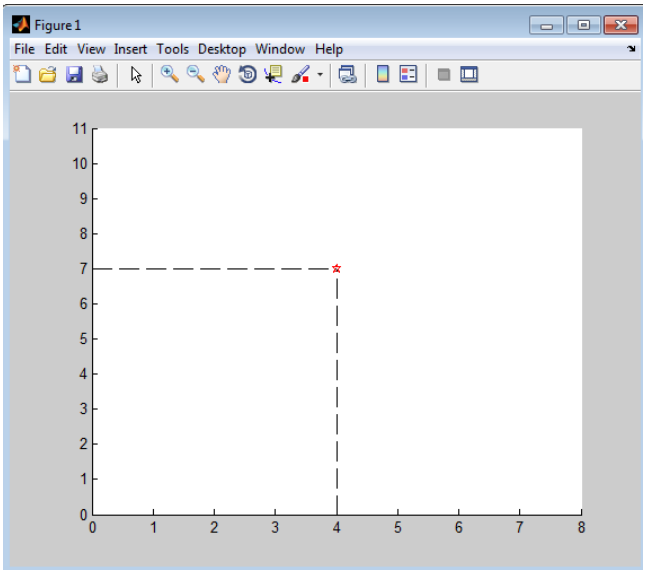

Figura3

En la práctica anterior aprendimos a definir rangos en el entorno de Matlab y Octave, como por ejemplo para rellenar los elementos de un vector. Un ejemplo podría ser el crear un vector x conteniendo **length**(x) valores equiespaciados en 0.1 unidades en el rango [0,2π]. Una posible solución sería:

```
%genera vector
x=(0:0.1:2*pi);
```

Este vector podría ahora ser utilizado por ejemplo para dibujar en el plano cartesiano la función sin(x) en el rango [0,2π].

```
%dibuja sin(x)
clc;
clear all;
x=(0:0.1:2*pi);
plot(x,sin(x));
axis([0 2*pi -1 1]);
```

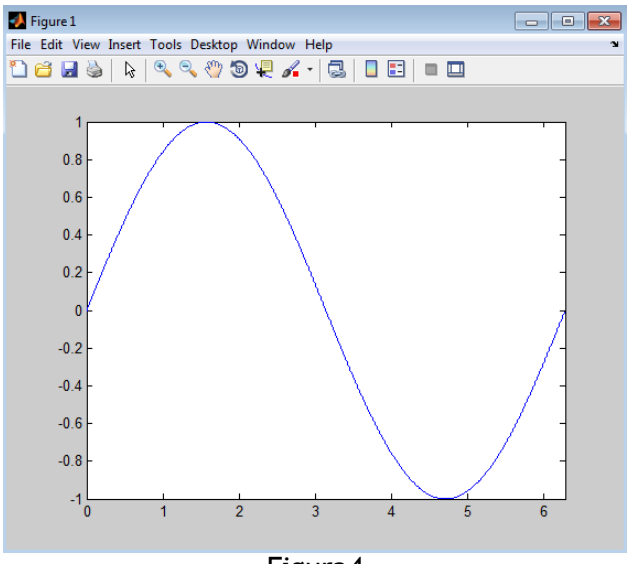

Figura4

Observa que el primer parámetro que se le proporciona al comando ***plot*** es el rango de valores para el que hemos calculado el seno, mientras que el segundo es el vector que contiene los valores resultantes.

Si redefinimos los valores del vector ***x*** y lanzamos de nuevo el ***plot***, obtendremos distintas gráficas, por ejemplo pintemos hasta *6*p*i.

```
%dibuja sin(x)
clc;
clear all;
figure(1);
x=(0:0.1:2*pi);
plot(x,sin(x));
axis([0 2*pi -1 1]);
figure(2);
x=(0:0.1:6*pi);
plot(x,sin(x));
axis([0 6*pi -1 1]);
```

En la figura 5 podemos observar que se abrieron 2 ventanas de figuras **¿porqué?**

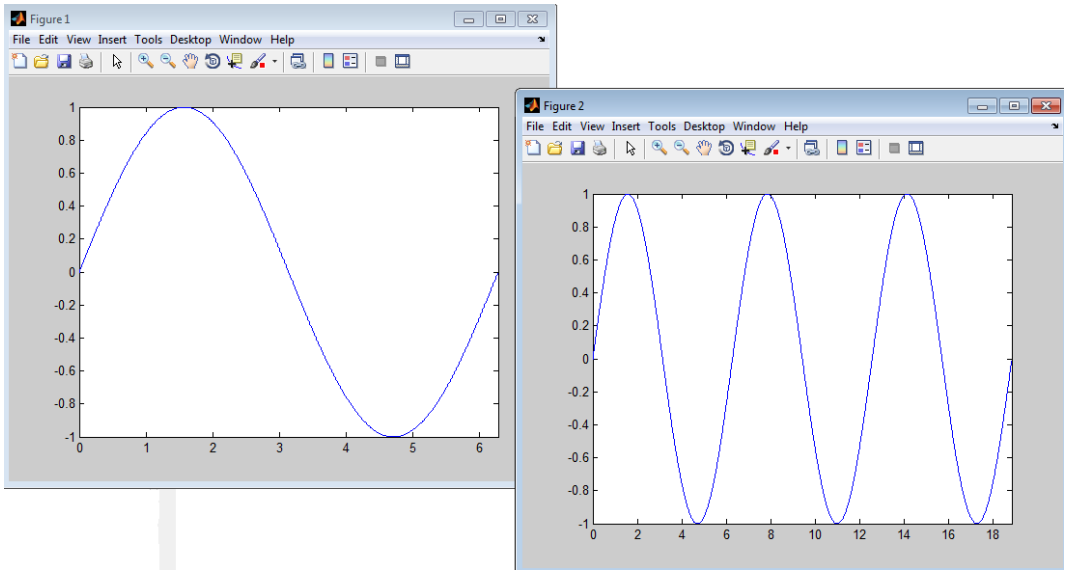

Figura5

Llegados a este punto, seguramente habrás observado en las ventanas figuras, incluso que existe una barra de menú que te permite hacer zoom, rotar la imagen, y hasta salvarla (quizás para un informe o memoria). Pero ciertamente, tenemos muy poca información en esta gráfica, ni tan siquiera al verla podemos identificar lo que se representa. Antes de añadir ese tipo de información a la gráfica, indicarte que puedes **personalizar usando parámetros** la gráfica una vez dibujada. Escribe por ejemplo el programa:

```
%dibuja sin(x)
clc;
clear all;
x=(0:0.1:6*pi);
plot(x,sin(x),'r');
axis([0 6*pi -1 1]);
```

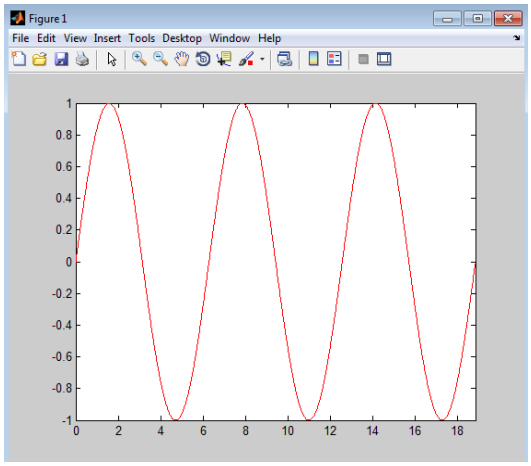

Figura6

¿Has apreciado el cambio de color?

En este caso ha sido rojo el color utilizado para la gráfica. Con ese tercer parámetro del comando ***plot*** podemos justamente indicar el color de la gráfica, muy útil si tenemos más de una gráfica. Es la inicial de su nombre en inglés. Otros colores disponibles son:

b	blue
g	green
r	red
c	cyan
m	magenta
y	yellow
k	black

Pero hay más, tras la inicial del color puedes añadir un segundo carácter en ese tercer parámetro establecer la marca que se usa para pintar cada par *(x,y)*

```
%dibuja sin(x)
clc;
clear all;
x=(0:0.1:6*pi);
plot(x,sin(x),'rs');
axis([0 6*pi -1 1]);
```

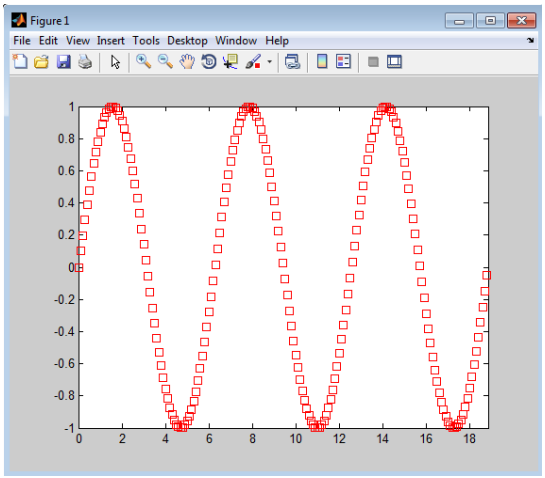

Figura 7

Otras marcas posibles son:

.	punto
o	círculo
x	equis
+	símbolo más
*	estrella
s	cuadrado
d	diamante
v	triángulo (abajo)
^	triángulo (arriba)
<	triángulo (izquierda)
>	triángulo (derecha)
p	pentagrama
h	hexagrama

Y con un tercer carácter establecer un estilo diferente para las líneas que unen los diversos pares *(x,y)*

-	línea sólida
--	
:	punteada
-.	líneas y puntos
--	a trozos
(ausente)	sin línea

Puede que te asusten porque son unas cuantas posibilidades, pero no es necesario, recuerda que para nuestro ejemplo básico pudimos hacerlo simplemente con los dos primeros parámetros. Sin embargo juega con las posibles combinaciones, por ejemplo:

```
%dibuja sin(x)
clc;
clear all;
x=(0:0.1:6*pi);
plot(x,sin(x),'gs-');
axis([0 6*pi -1 1]);
```

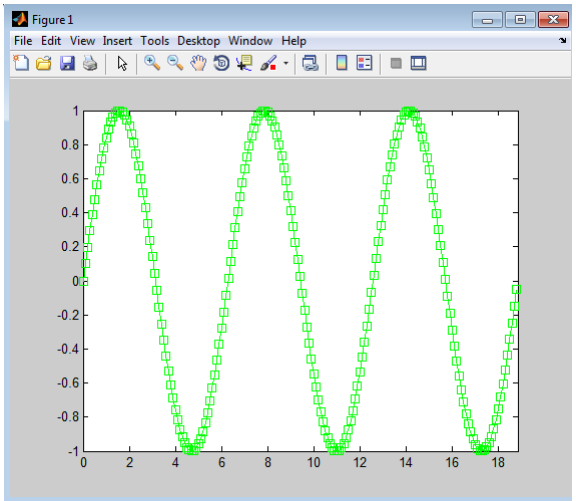

Figura 8

Recuerda que estamos ~~pintando~~ representando el valor del seno en un rango de valores dados en radianes. Es un dato que no se ve reflejado en la gráfica. Para que la gráfica aporte más información sobre su contenido, podemos añadirle etiquetas (**xlabel, ylabel**) a cada eje, un título (**title**), cambiar el fondo de la ventana figura etc. Véamos el siguiente ejemplo:

```
%dibuja sin(x)
clc;
clear all;
x=(0:0.1:6*pi);
plot(x,sin(x),'bs-');
xlabel('Radianes');
ylabel('Valor del seno');
title('Valores del seno entre 0 y 6*pi');
axis([0 6*pi -1 1]);
set(gcf,'Color','w');    %define el color del fondo
```

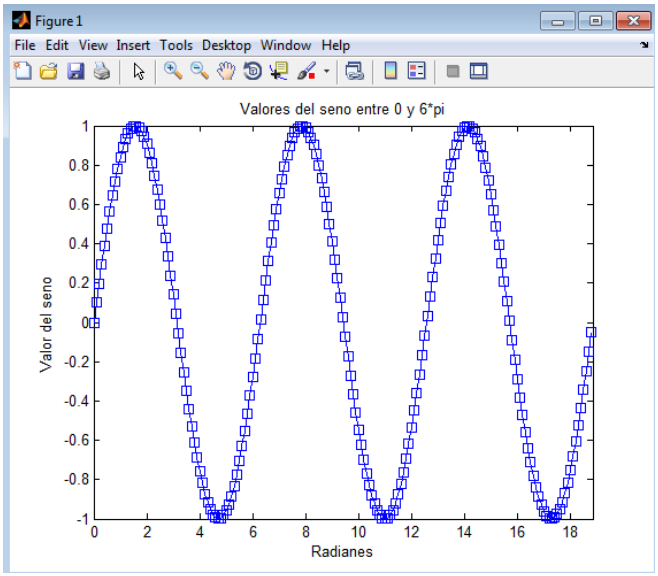

Figura 9

También puede ser interesante el comando ***grid*** que añade una rejilla que sirve como referencia a quien ve la imagen.

```
%dibuja sin(x)
clc;
clear all;
x=(0:0.1:6*pi);
plot(x,sin(x),'bs-');
xlabel('Radianes');
ylabel('Valor del seno');
title('Valores del seno entre 0 y 6*pi');
axis([0 6*pi -1 1]);
grid on;                 %inserta rejilla
set(gcf,'Color','w');    %define el color del fondo
```

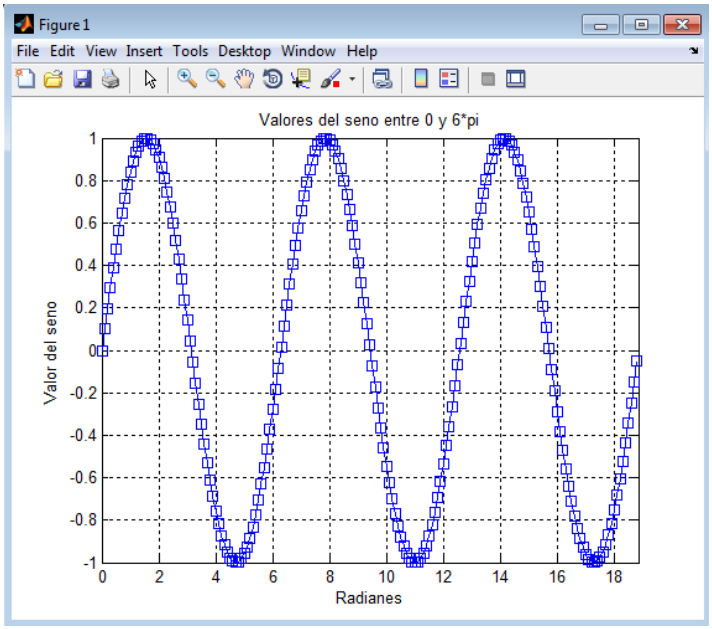

Figura 10

En cualquier momento es posible salvar la imagen con el comando ***saveas*** por ejemplo, revisa la última línea del código siguiente. Tras ejecutar, vete a tu carpeta de trabajo y abre el fichero Miimagen.png.

```
%dibuja sin(x)
clc;
clear all;
x=(0:0.1:6*pi);
plot(x,sin(x),'bs-');
xlabel('Radianes');
ylabel('Valor del seno');
title('Valores del seno entre 0 y 6*pi');
axis([0 6*pi -1 1]);
grid on;                        %inserta rejilla
set(gcf,'Color','w');           %define el color del fondo
saveas(gcf,'Miimagen','png'); %guardar imagen
```

En el comando saveas(gcf,'Miimagen','png'), el primer parámetro identifica la imagen a salvar, la que esté activa en ese momento, el segundo especifica el nombre que se le dará al fichero, y el tercero establece el formato de la imagen. En este caso **png**, que es un formato sin pérdida de calidad, pero hay otros como **jpeg** que se indica usando como parámetro 'jpg', **eps** que se indica usando como parámetro 'eps', (este formato puede contener tanto gráficos vectoriales como de mapa de bits, es por tanto muy buena opción cuando se quieren gráficos vectoriales) también, otra posibilidad es el formato del propio Matlab **fig** que se indica usando como parámetro 'fig', este formato te permite abrir la imagen guardada en cualquier momento sin tener que ejecutar el programa .m que la genera.

Si queremos representar dos gráficas en una ventana, sin querer usar el comando **hold on** que vimos más arriba en el **ejemplo resuelto 2**, basta con añadir parámetros al comando ***plot***. Realiza el siguiente programa cuyo resultado es la figura 11:

```
%dibuja sin(x) cos(x)
clc;
clear all;
x=(0:0.1:6*pi);
plot(x,sin(x),'bs-',x,cos(x),'ro-');
xlabel('Radianes');
ylabel('Valor del seno y coseno');
title('Valores del seno y coseno entre 0 y 6*pi');
axis([0 6*pi -1 1]);
set(gcf,'Color','w');   %define el color del fondo
```

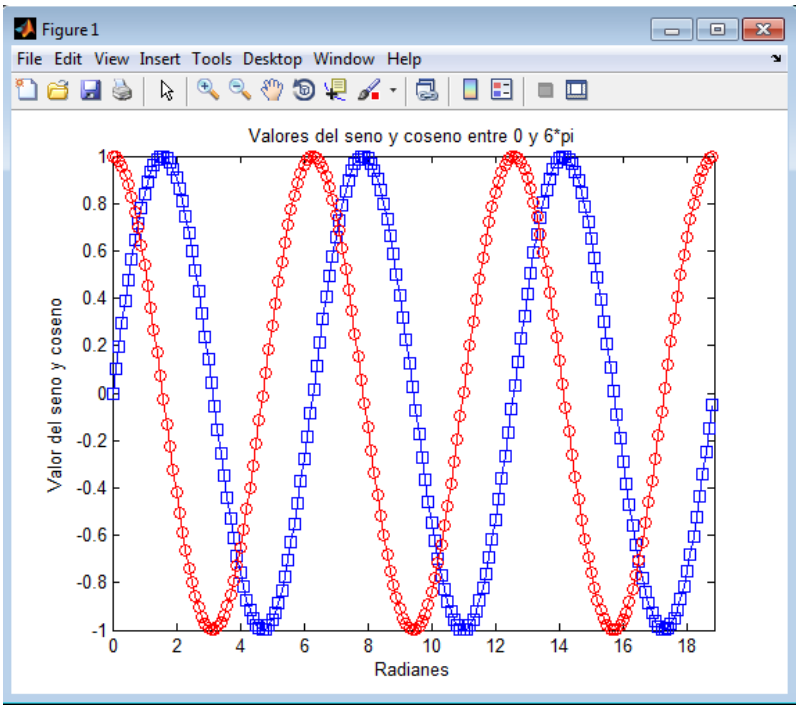

Figura 11

Puedes añadir aún más gráficas y mostrar la leyenda (**legend**) para aclarar lo que contiene la imagen resultante. Realiza el siguiente programa donde además se ha añadido el comando **box off** para eliminar la caja del gráfico y quedarnos sólo con los ejes x,y.

```
%dibuja sin(x) cos(x)
clc;
clear all;
x=(0:0.1:6*pi);
plot(x, sin(x), 'bs-', x, cos(x), 'ro-', x, sin(x)-cos(x), 'k+-');
legend('Seno', 'Coseno', 'Seno-Coseno' ,'Location', 'NorthEast');
xlabel('Radianes');
ylabel('Valor del seno y coseno');
title('Valores del seno y coseno entre 0 y 6*pi');
axis([0 6*pi -1 1]);
set(gcf,'Color','w');   %define el color del fondo
box off;
```

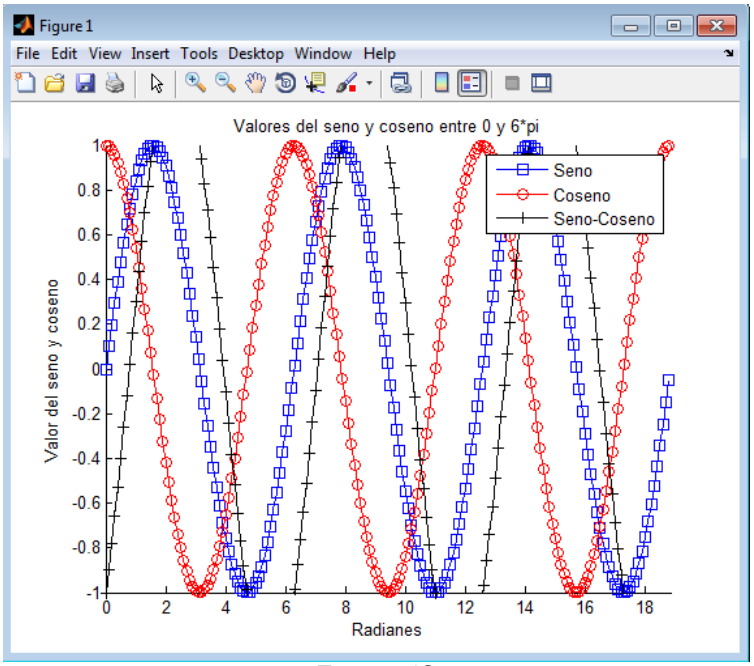

Figura 12

En algunas ocasiones nos puede interesar modificar las **etiquetas** que se muestran en cada eje. Si quisieras modificar el número de ellas tras el ***plot*** teclea por ejemplo

set(gca,'XTick',0:3:18);

para que se muestren sólo una de cada tres. El valor 'gca' designa los ejes de coordenadas de la figura activa. Si modificamos el programa anterior según:

```
%dibuja sin(x) cos(x)
clc;
clear all;
x=(0:0.1:6*pi);
plot(x, sin(x), 'bs-', x, cos(x), 'ro-', x, sin(x)-cos(x), 'k+-');
legend('Seno', 'Coseno', 'Seno-Coseno' ,'Location', 'NorthEast');
xlabel('Radianes');
ylabel('Valor del seno y coseno');
title('Valores del seno y coseno entre 0 y 6*pi');
axis([0 6*pi -1 1]);
set(gca,'XTick',0:3:18);
set(gcf,'Color','w');   %define el color del fondo
box off;
```

el resultado sería el mostrado en la figura 13, observa variaron el número de Ticks en el eje de las x. Similarmente se puede utilizar YTick para el eje de las y.

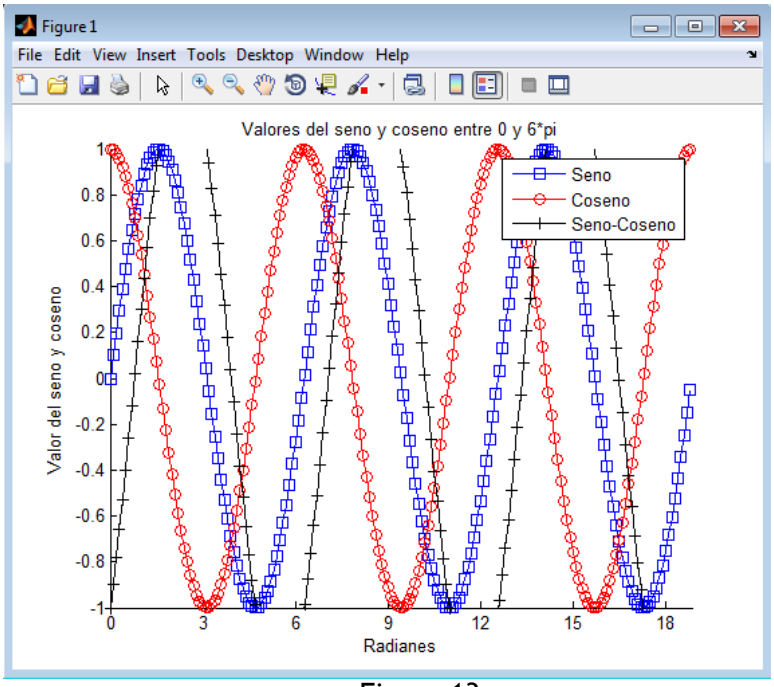

Figura 13

El comando ***subplot(m,n,p)*** como se dijo en la introducción, permite dividir el área de dibujo en una matriz de mxn rectángulos. Con el parámetro p seleccionamos el rectángulo en el que dibujamos a continuación, con el ya conocido comando ***plot***. El parámetro p = 1 designa a la primera celda de la primera fila, p = 2 la siguiente, y así consecutivamente, de manera que la primera celda de la fila s tendrá como índice p = (s-1)*n + 1.

Escribe ahora el siguiente programa y luego ejecútalo. Investiga que hace el comando **text** que se ha añadido en el código para las gráficas 3 y 4 de la figura 14. Investiga asimismo las opciones 'MarkerSize',8,'Markerfacecolor','b','MarkerEdgeColor','b' en el comando **plot** para dibujar la gráfica 3 de la figura 14.

```
%Ejemplo figura con subplot (4 gráficas en una figura)
clc;
clear all;
x=(0:0.3:6*pi);       %vector que se usará en gráficas 2 y 3
set(gcf,'Color','w'); %define el color del fondo

%%%%%%%%%%%%%%%% Figura1 %%%%%%%%%%%%%
subplot(2,2,1);
plot(x, sin(x), 'bs-');
axis([0 6*pi -1 1]);
xlabel('Radianes');
ylabel('Valor del seno');
title('Valores del seno entre 0 y 6*pi');
box off;

%%%%%%%%%%%%%%%% Figura2 %%%%%%%%%%%%%
subplot(2,2,2);
plot(x, cos(x), 'ro-');
axis([0 6*pi -1 1]);
xlabel('Radianes');
ylabel('Valor del coseno');
title('Valores del coseno entre 0 y 6*pi');
box off;
```

```
%%%%%%%%%%%%%%%% Figura3 %%%%%%%%%%%%%
subplot(2,2,3);
x=(0:0.4:10);
plot(x,(x-5).^2,'s','MarkerSize',8,'Markerfacecolor','b','MarkerEdgeColor','b');
xlabel('x');
ylabel('y=x^2');
box off;
text(5,10,'Parábola','FontSize',9,'FontName','Times');

%%%%%%%%%%%%%%%% Figura4 %%%%%%%%%%%%%
subplot(2,2,4);
x=(0:0.4:10);
plot(x,0.5.*x+2,'r-');
axis([0 10 0 8]);
xlabel('x');
ylabel('y');
box off;
text(2.5,3,'y=0.5*x+2','FontSize',9,'FontName','Times');
```

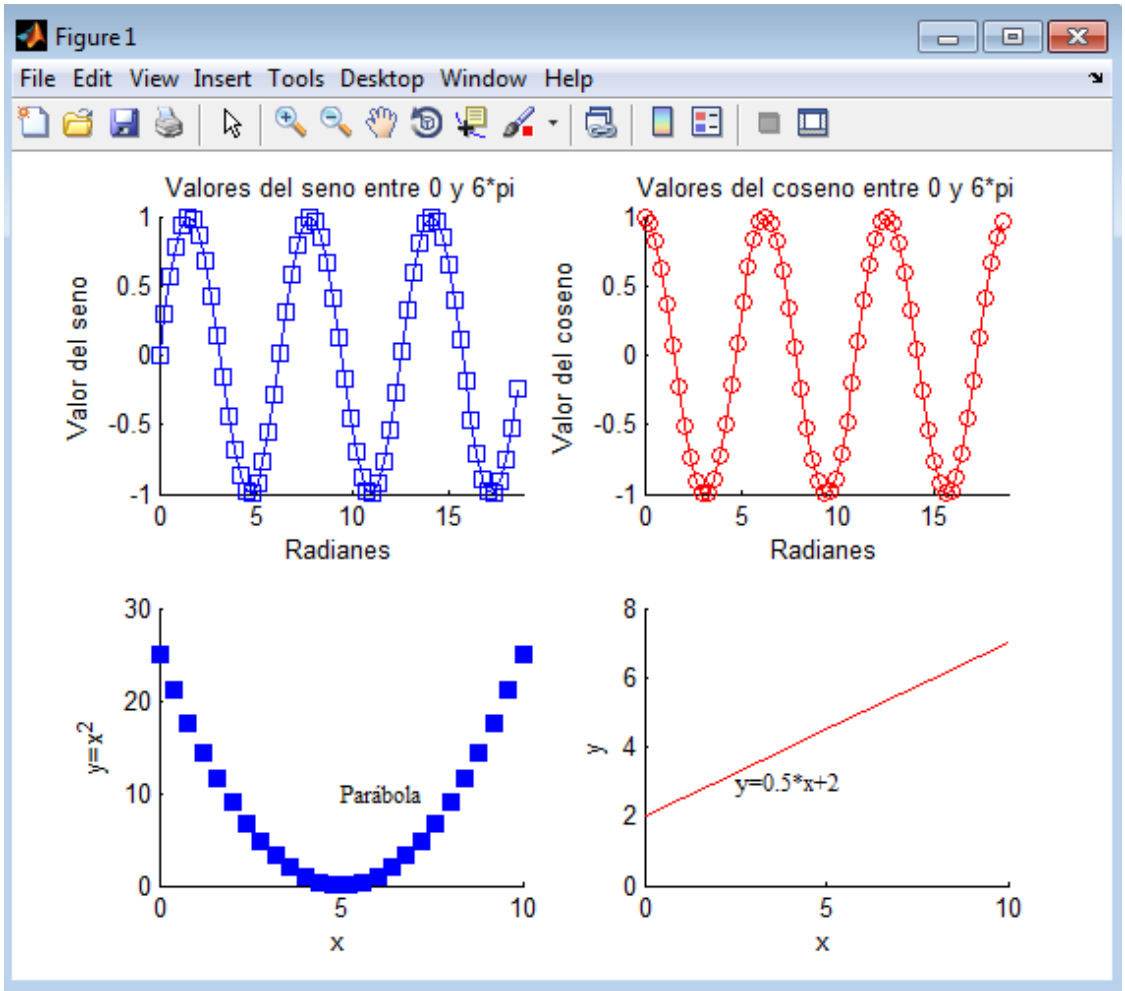

Figura 14

3. GRÁFICOS 3D

Matlab permite representar también gráficos en tres dimensiones, tanto **curvas** como **superficies**. Probemos primero utilizando la función **plot3** para dibujar una curva en espiral, primero almacenamos en el vector t valores en el rango **[0,10*π]** con un cierto incremento. Luego, a partir de los valores de *t* obtenemos los valores de nuestra curva considerando para la *x* el seno, para la *y* el coseno, y para la z el valor de t. Véase el siguiente código:

```
%Ejemplo curva 3D
clc;
clear all;
t=0:pi/50:10*pi;
plot3(sin(t),cos(t),t);
```

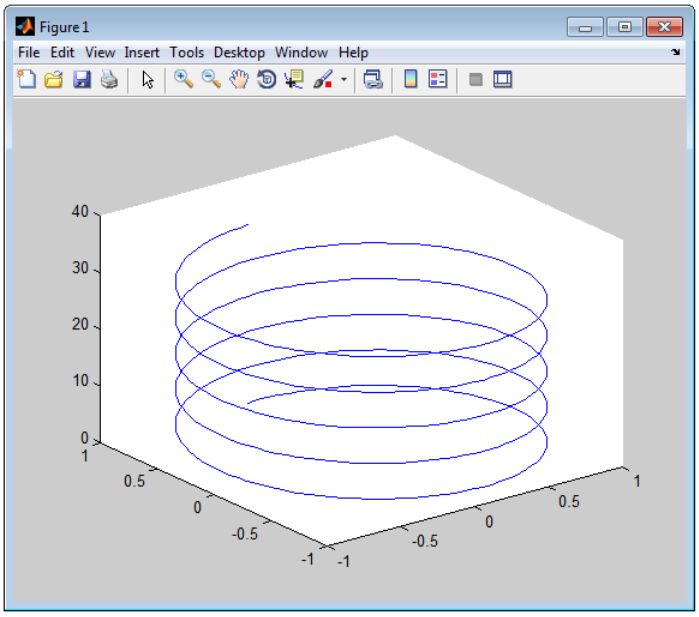

Figura 15

Otras definiciones para las coordenadas *x*, y, *z* producirán figuras diferentes.

Si disponemos de la definición analítica de una **superficie** 3D, por ejemplo $z = x^2 - y^2$, también es posible obtener su representación. Como primer paso guardemos en *x* e *y* los valores del intervalo que queremos representar, en nuestro ejemplo escogemos [-2,2] tecleando

[x,y]=**meshgrid**(-2:0.2:2,-2:0.2:2)

Los valores *x* e *y* son dos matrices en las que se almacenan repetidas veces, valores en el intervalo en dicho rango. Para evaluar dichos valores en una función usamos el comando ***surf:***

surf(x,y,x.^2-y.^2)

El siguiente código de programa genera la figura 16.1:

```
%Ejemplo superficie
clc;
clear all;
[x,y]=meshgrid(-2:0.3:2,-2:0.3:2);
surf(x,y,x.^2-y.^2),colorbar;
```

Incluyendo ***colorbar*** en la línea 4 aparece una barra indicando el color utilizado para los distintos valores de z.

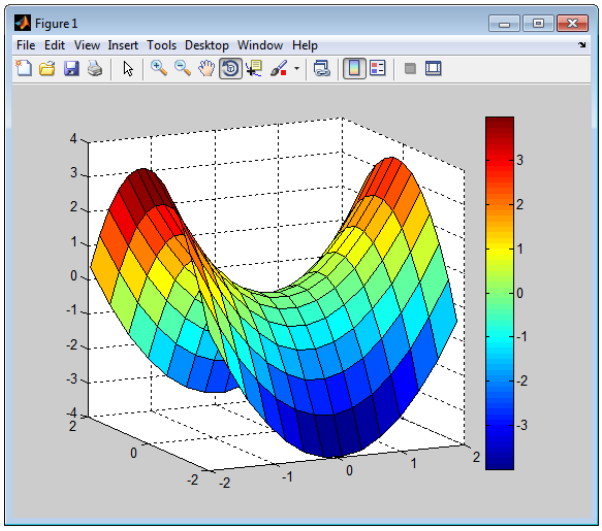

Figura 16.1

Reflexiona y estudia ahora la modificación realizada en el código anterior cuyo resultado es la gráfica 16.2. **¿Qué ha sucedido?**

```
%Ejemplo superficie
clc;
clear all;
[x,y]=meshgrid(-2:0.1:2,-2:0.1:2);
surf(x,y,x.^2-y.^2),colorbar;
```

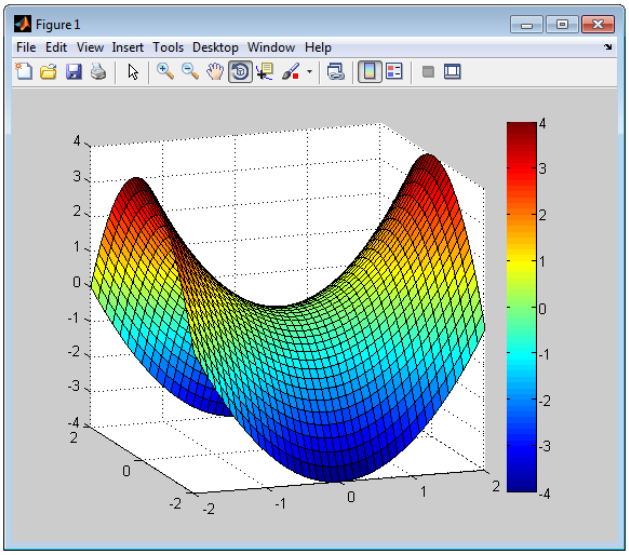

Figura 16.2

4. GRÁFICOS ESTADÍSTICOS

También es posible representar datos estadísticos mediante diagramas de barras (***bar***, ***barh***, ***bar3***, **bar3h**) o diagramas de sectores (***pie***, ***pie3***). Veamos el siguiente ejemplo de diagrama de barras:

```
%Ejemplo diagrama de barras
clc;
clear all;
set(gcf,'Color','w');    %define el color del fondo
x=round(1+(10-1).*rand(1,8));  %crea vector 8 elementos [1,10]
%gráfica1
subplot(1,2,1);
bar(x);
%gráfica2
subplot(1,2,2);
bar3(x,'r');
```

Obsérvese que en la línea 5 del código anterior, se crea un vector **x** de 8 elementos enteros al azar en el intervalo [1,10]. Tal vez sea ahora un buen momento para revisar y recordar la función **rand** de Matlab para generación de números aleatorios y la función **round** para redondear, teclea en la Command Window:

>> help rand

>> help round

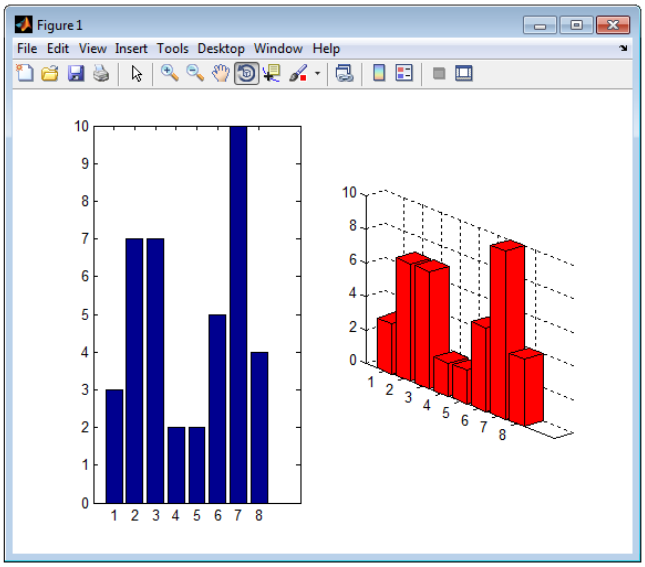

Figura 17

Ahora, el mismo ejemplo anterior pero con diagramas de sectores, el resultado es la figura 18 y el código es el siguiente:

```
%Ejemplo diagrama de sectores
clc;
clear all;
set(gcf,'Color','w');    %define el color del fondo
x=ceil(10.*rand(1,8));   %crea vector 8 elementos [1,10]
%gráfica1
subplot(1,2,1);
pie(x);
%gráfica2
subplot(1,2,2);
pie3(x);
```

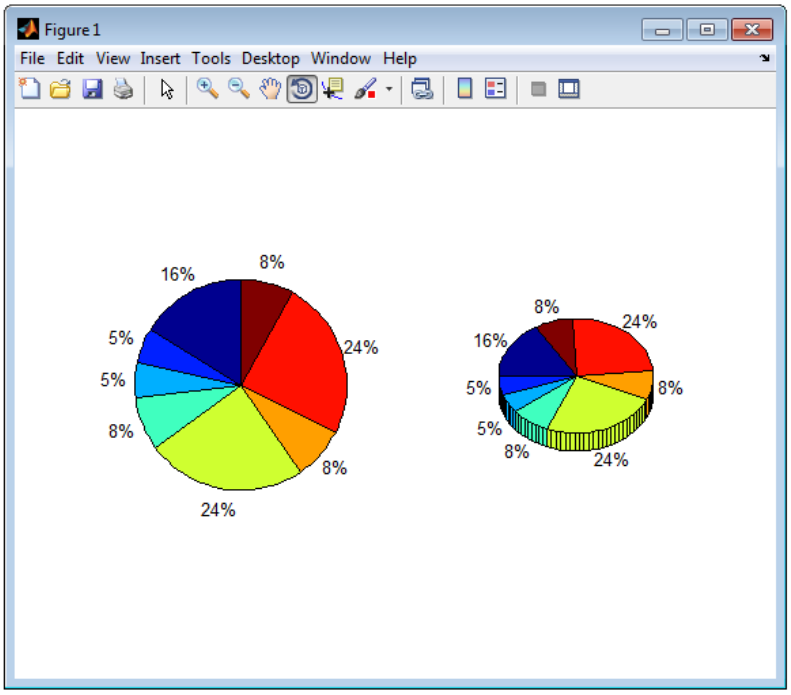

Figura 18

5. EJERCICIOS PROPUESTOS

- Escriba un fichero programa .m que dibuje las gráficas de las siguientes funciones en el rango indicado:

$$f(x) = x^3 - x + 1 \qquad -3 \le x \le 3 \qquad paso = 0.1$$

$$f(x) = \frac{1}{1+|x|^2} \qquad -5 \le x \le 5 \qquad paso = 0.05$$

$$f(x) = \frac{x^2}{x^2 - 1} \qquad -10 \le x \le 10 \qquad paso = 0.01$$

$$f(x) = sen\left(e^x\right) \qquad -\pi \le x \le \pi \qquad paso = 0.01$$

- Entra en esta página (http://clima.meteored.com) y escoge una ciudad seleccionando además un mes y año (observa que algunas localidades no tienen datos para cualquier fecha). A continuación dibuja una gráfica para las

temperaturas media, mínima y máxima del mes que escojas. Escribe el fichero .m que lo realice y grabé la imagen a disco en formato png o jpeg.

- El número de personas que aprobaron una cierta oposición, por comunidad autónoma, está reflejado en la siguiente tabla:

Comunidad autónoma	N° de aprobados
Canarias	181
País Vasco	102
Madrid	98
Cataluña	53
Asturias	49
Extremadura	23
Andalucía	8
Otras	6
Total	**520**

Escriba un fichero .m con las órdenes necesarias para crear una gráfica de sectores (en 3 dimensiones) de los datos estadísticos anteriores y que contenga las leyendas de las comunidades autónomas.

- Haciendo uso de un fichero o programa .m y utilizando el comando ***surf***, dibuje la función tridimensional $z = e^{-\left(x^2+y^2\right)} \cdot \cos\left(3\left(x^2+y^2\right)\right)$ en el intervalo -2 +2 y un paso de 0.1 (tanto para x como para y).

6. EJERCICIO ADICIONAL RESUELTO

Este ejercicio es voluntario para realizar cuando el alumno tenga un ratito libre y quiera revisar algunas de las funciones vistas en esta y anterior prácticas.

El ejercicio trata de una viga cuadrada de acero simplemente apoyada con una carga puntual concentrada entre los apoyos (figura 19). El fichero datosviga.txt (descárgalo de la página de la asignatura en el campus virtual) contiene los diseños óptimos posibles calculados para P=1000 N, L= 100 cm., módulo de Young E=2.1 x 10^4 kN/cm^2, momento de inercia $I = \frac{x^4}{12} cm^4$ y x_1 es la variable de decisión (longitud de la sección en cm). La columna 1 del fichero contiene los valores óptimos cuando se minimiza la sección de la viga (x_1^2 cm^2) mientras la segunda columna contiene los valores óptimos cuando se minimiza la deformación (flecha) en el centro de la viga ($\frac{PL^3}{48EI} cm$).

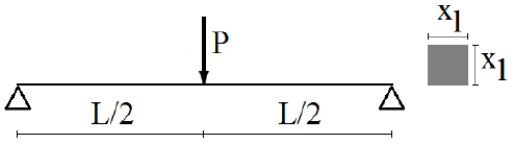

Figura 19

El Ingeniero proyectista quiere graficar los datos contenidos en el fichero datosviga.txt para mejor explorar el conjunto de soluciones en el proceso de la toma de decisión acerca del mejor diseño-solución.

El código propuesto es el siguiente:

```
1  %soluciones para diseño óptimo viga
2  clear all;   %limpia memoria de variables
3  clc;         %limpia pantalla
4  close all;   %cierra todas las ventanas de figuras abiertas
5
6  M=load('datosviga.txt'); %cargar en matriz M datos de fichero
7  [fil,col]=size(M);       %obtener número filas columnas de M
8  x=M(1:fil,1);            %vector x con columna1 de M (sección viga)
9  y=M(1:fil,2);            %vector y con columna2 de M (deformación viga)
10
11 box off;                      %quitar caja exterior de la figura
12 hold on;                      %habilitar varias gráficas en misma figura
13 plot(x,y,'-k','LineWidth',1.5);%dibujar soluciones de datosviga.txt
14 plot(3.0,1.27,'sb','MarkerSize',8,'markerfacecolor','r','MarkerEdgeColor','r');
15 set(gcf,'Color','w');    %fondo figura en blanco
16 axis([0 10 0 10]);       %limitar ejes. También xlim([0 10]),ylim([0 10])
17 xlabel('Sección transversal cm^2','FontSize', 12,'FontName','Times');%etiqueta eje x
18 ylabel('Deformación cm','FontSize', 12,'FontName','Times');          %etiqueta eje y
19 title('Decisión Diseño Óptimo Viga','FontSize', 12,'FontName','Times');%Título
20 text(1.9,4,'soluciones óptimas','FontSize',9,'FontName','Times');     %texto
21 text(3.2,1.5,'solución adoptada','FontSize',9,'FontName','Times');    %texto
22 saveas(gcf,'Miimagen','fig');                                        %guardar imagen
```

La gráfica 20 es el resultado tras ejecutar el código anterior.

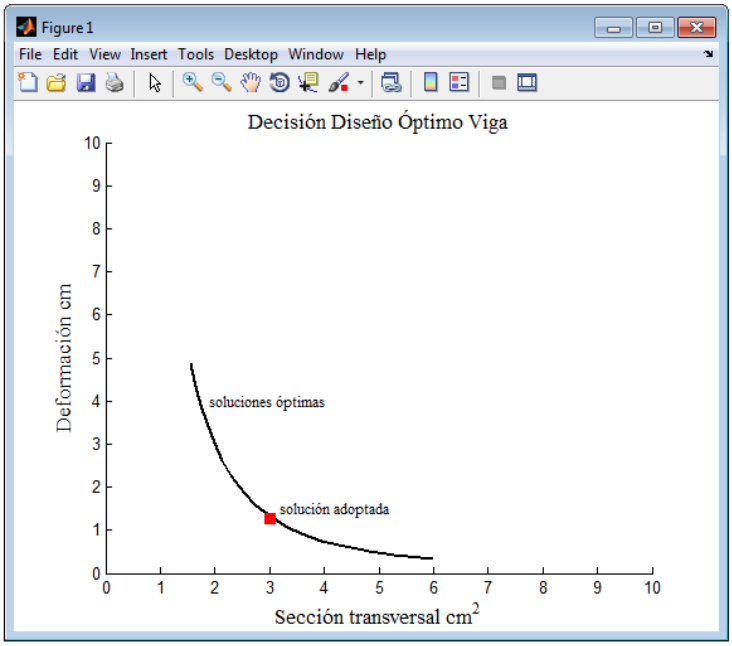

Figura 20

REVISA-ANALIZA-ESTUDIA lo siguiente:

1. ¿Qué hace el comando close all en la línea 4?
2. Mira la línea 6. Haz **>> help load** en la venta command window de Matlab para ver que hace el comando **load** (probablemente es nuevo para ti). Fíjate que sencillo cargar los datos del fichero datosviga.txt en la matriz M.
3. Repasa las líneas 7,8,9. Lo practicamos en la clase anterior.
4. El resto de comandos se han visto en esta clase.
5. Observa todos los comentarios que se han puesto. Es una buena práctica y te puede ser muy útil documentar tu programa.
6. Por último, y como ingeniero que algún día serás, analiza en la figura 20, el efecto de adoptar otra solución de menor sección transversal (más barata).

7. ANEXO: EDITOR DE PROGRAMAS DE MATLAB

Si estamos interesados en crear un fichero .m que contenga una serie de comandos, debe crearse con cualquier editor (bloc de notas, Wordpad, Notepad++, etc) un fichero de extensión .m. La interfaz de Matlab también proporciona su propio editor. Para abrir el editor de Matlab seleccionamos *File->New M File* y aparecerá la ventana de editor como se observa en la figura 13. Una vez tecleados en el fichero los comandos a ejecutar, lo guardamos en disco. La ejecución de un fichero o programa .m es muy sencilla, basta con asegurarse de que está en nuestra carpeta actual (*current directory*), y escribir su nombre (sin extensión) en la consola de comandos de Matlab.

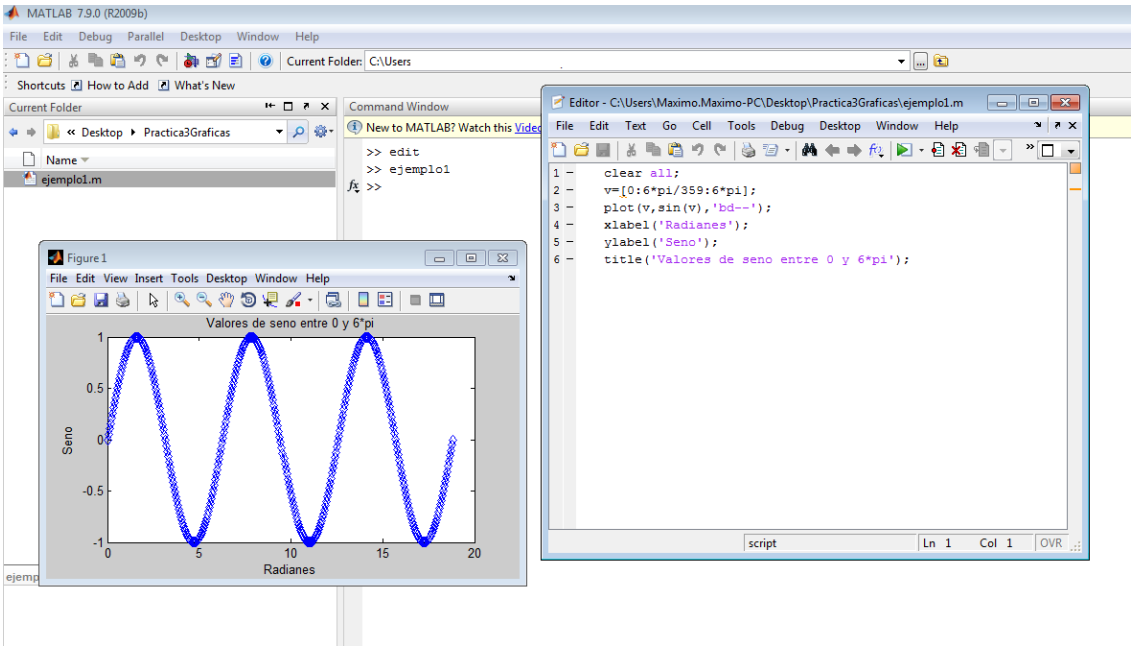

Figura 20

Para elaborar las gráficas anteriores hemos necesitado varios comandos de Matlab. Un ejemplo de fichero .m que genere la función seno con todos sus parámetros podría ser:

```
clear all;
v=[0:6*pi/359:6*pi];
plot(v,sin(v),'bd--');
xlabel('Radianes');
ylabel('Seno');
title('Valores de seno entre 0 y 6*pi');
```

La gran ventaja del uso de un fichero es que podemos lanzar una secuencia de comandos mucho más fácilmente, y su modificación es sencilla para obtener otras gráficas.