Máster en Materiales y Sistemas Sensores para Tecnologías Medioambientales (Erasmus Mundus)

PRÁCTICAS DE CÁLCULO NUMÉRICO

Damián Ginestar Peiró

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DEL DISEÑO

UNIVERSIDAD POLITÉCNICA DE VALENCIA

Práctica 1

Gráficas con Matlab

El paquete MatLab permite obtener las gráficas de cualquier función matemática tanto si representa una curva plana o una superficie. Además, permite agrupar y superponer gráficas. Otras opciones típicas de los programas de gráficos como colores, marcos, etc. se pueden utilizar en este paquete.

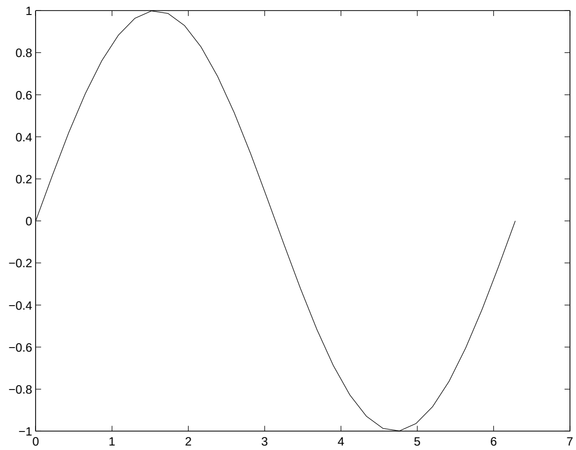
1.1. Gráficas 2D

La representación gráfica 2D de una función se puede obtener cuando la función se expresa en coordenadas cartesianas o paramétricas.

El comando `plot(x,y)` representa los pares que tienen como abcisas los elementos del vector `x` y como ordenadas los elementos del vector `y`. Con el comando `plot(y)` toma como abcisas los números naturales 1,2, $\ldots, n$. El comando `linspace(a,b,N)` genera `N` puntos igualmente espaciados comprendidos entre `a` y `b`. Así, para la generación de gráficas de funciones se procede, por ejemplo, del siguiente modo:

```
x=linspace(0,2*pi,30);
y=sin(x);
plot(x,y)
```

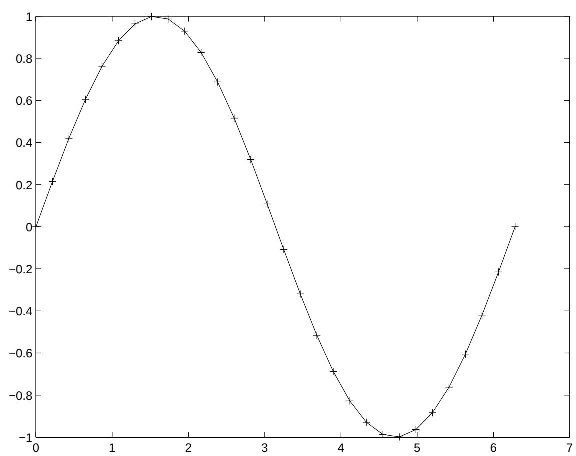
obtenemos la siguiente gráfica



Es posible generar la misma gráfica con líneas y cruces escribiendo

```
plot(x,y,x,y,'+')
```

con lo que se obtiene:



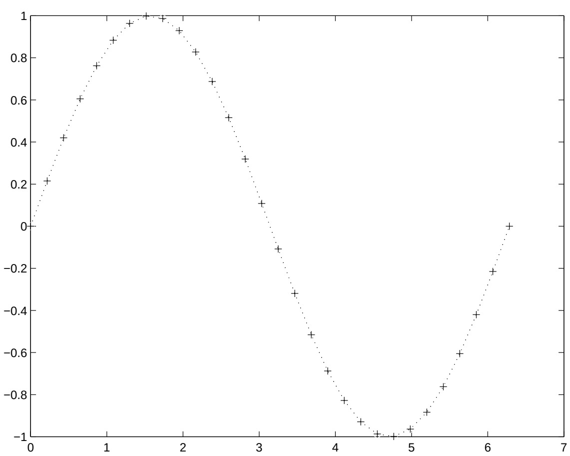
Por otra parte, se pueden controlar los colores y los estilos de las líneas utilizando los símbolos de la siguiente tabla:

Símbolo	Color	Símbolo	Estilo
y	amarillo	.	puntos
m	magenta	o	círculos
c	cyan	x	aspas
r	rojo	+	cruces
g	verde	*	estrella
b	azul	-	línea
w	blanco	:	línea de puntos
k	negro	- .	línea y puntos
		- -	línea de guiones

Por ejemplo, se puede escribir:

```
y=sin(x)
plot(x,y,'g:',x,y,'wo',x,y,'r:',x,y,'c+')
```

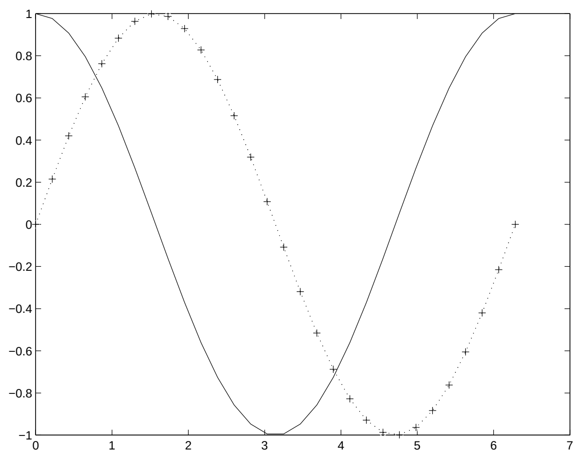
obteniéndose



Cada vez que se ejecuta el comando `plot` desaparece la figura anterior. Si pretendemos superponer gráficas el comando `hold` nos permite mantener la gráfica de la función donde estamos trabajando. Por ejemplo, si escribimos:

```
plot(x,y,'g:',x,y,'wo',x,y,'r:',x,y,'c+')
hold on
z=cos(x)
plot(x,z)
```

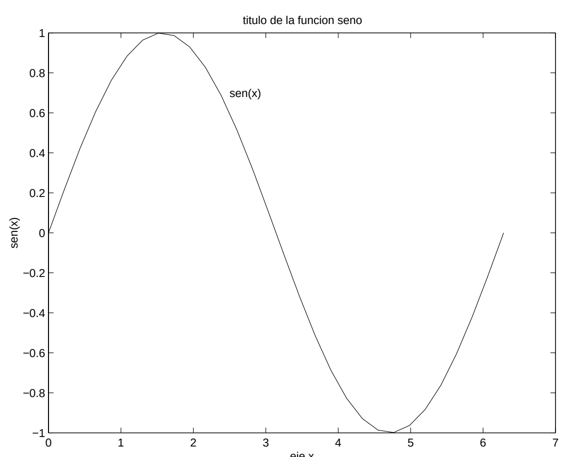
se obtiene la gráfica siguiente:



Hay distintas funciones que controlan la apariencia del gráfico. Así la función `grid on` introduce un mallado del gráfico y `grid off` quita el mallado. Las fuciones `xlabel` e `ylabel` generan un título para el eje x y el eje y, respectivamente. La función `title` genera un título para el gráfico. La función `text` permite poner texto en una zona del gráfico. Un ejemplo para la utilización de estas funciones es el siguiente

```
plot(x,y)
title('titulo de la funcion seno')
xlabel('eje x')
ylabel('sen(x)')
text(2.5,0.7,'sen(x)')
```

obteniendo la siguiente gráfica:



La función `axis` tiene control sobre la apariencia de los ejes, así, por ejemplo:

```
axis([xmin, xmax, ymin, ymax])
```

fija el eje x y el eje y de forma que el valor mínimo para las x-s sea `xmin`, el valor máximo para las x-s sea `xmax`, el valor mínimo para las y-s sea `ymin`, el valor máximo para las y-s sea `ymax`.

`axis auto`: devuelve la escala a los valores de defecto.

`axis equal`: usa la misma escala para las x-s que para las y-s.

`axis normal`: devuelve la escala a sus valores de defecto.

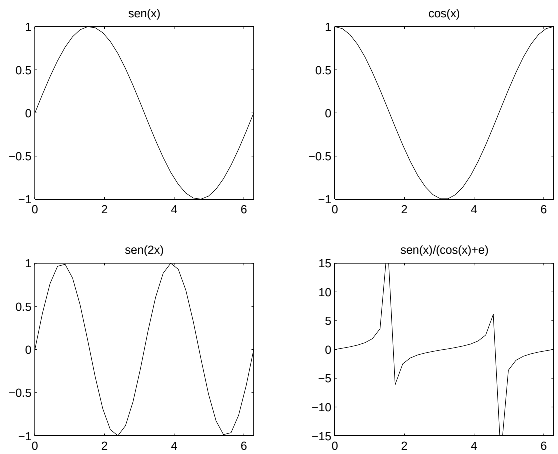
`axis off`: no dibuja los ejes.

`axis on`: vuelve a dibujar los ejes.

El comando `subplot` permite dividir la zona de dibujo en zonas y en cada zona dibujar una curva distinta. Su funcionamiento se muestra en el siguiente ejemplo

```
x=linspace(0,2*pi,30);
y=sin(x);
z=cos(x);
a=2*sin(x).*cos(x);
b=sin(x)./(cos(x)+eps);
subplot(2,2,1) % se divide la zona de dibujo
       % en 2 x 2 graficos y se selciona
       % la primera zona (arriba izquierda).
plot(x,y)
axis([0, 2*pi, -1, 1])
title('sen(x)')
subplot(2,2,2) % se seleciona la segunda zona
plot(x,z)
axis([0, 2*pi, -1, 1])
title('cos(x)')
subplot(2,2,3) % se seleciona la tercera zona
plot(x,a)
axis([0, 2*pi, -1, 1])
title('sen(2x)')
```

```
subplot(2,2,4) % se seleciona la cuarta zona
plot(x,b)
axis([0, 2*pi, -15, 15])
title('sen(x)/(cos(x)+e'))
```



El Matlab permite también realizar gráficas usando escalas logarítmicas y semilogarítmicas. Así,

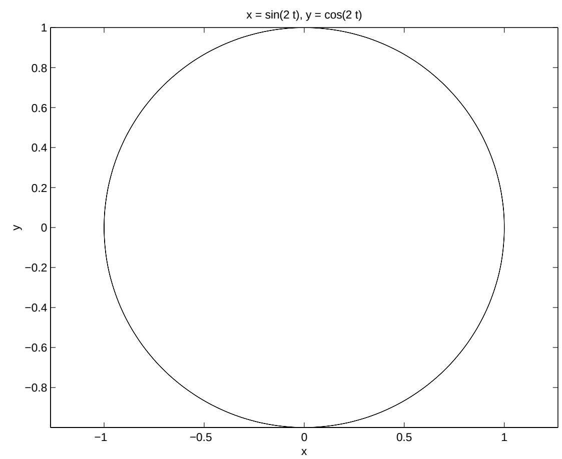
`loglog` : Es una función igual que `plot` pero usa escalas logarítmicas para el eje x y el eje y.

`semilogx` : Es una función igual que `plot` pero usa escala logarítmica para el eje x y lineal para el eje y.

`semilogy` : Es una función igual que `plot` pero usa escala logarítmica para el eje y y lineal para el eje x.

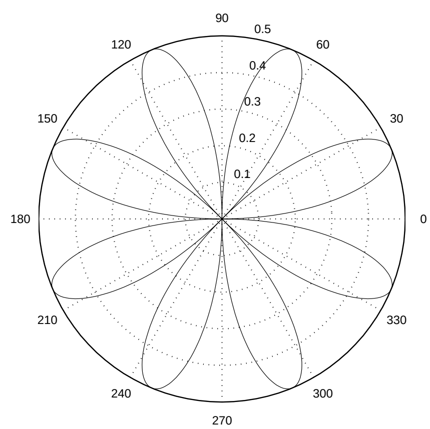
Si pretendemos dibujar curvas que vienen representadas en coordenadas paramétricas podremos utilizar el comando `ezplot` la instrucción sería como sigue:

```
ezplot('sin(2*t)','cos(2*t)',[0,2*pi])
```



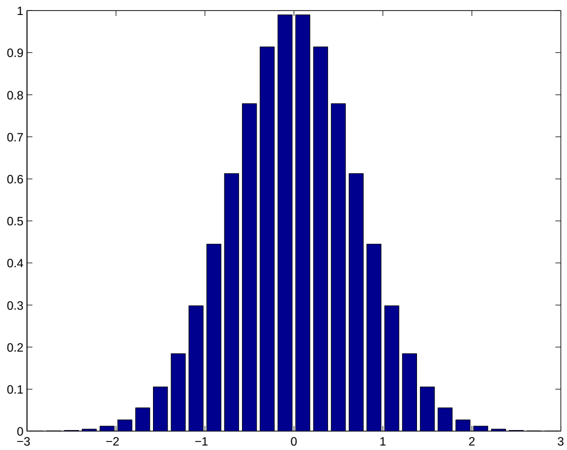
Una gráfica de una función en coordenadas polares puede crearse usando la función `polar`, como muestra el siguiente ejemplo:

```
t=0:0.01:2*pi;
r=sin(2*t).*cos(2*t);
polar(t,r)
```



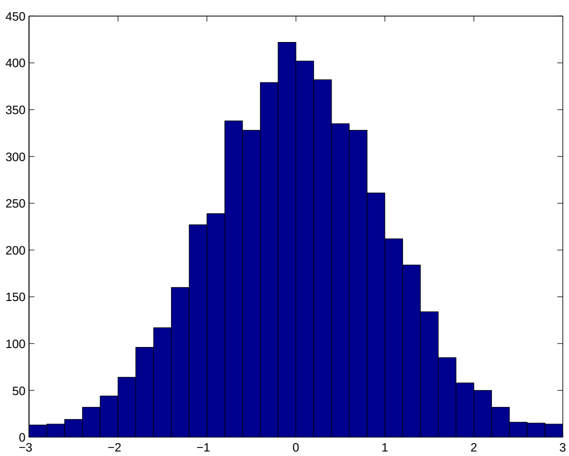
La función `bar` genera un diagrama de barras. Un ejemplo del diagrama de barras asociado a una gaussiana es el siguiente:

```
x=-2.9:0.2:2.9; % especifica el numero de divisiones
y= exp(-x.*x);
bar(x,y)
```



La función hist genera el histiograma asociado a los datos contenidos en un vector. Un ejemplo de su uso para un vector de números aleatorios distribuidos normalmente es

```
x=-2.9:0.2:2.9; % especifica el numero de divisiones
y= randn(5000,1);
hist(y,x)
```

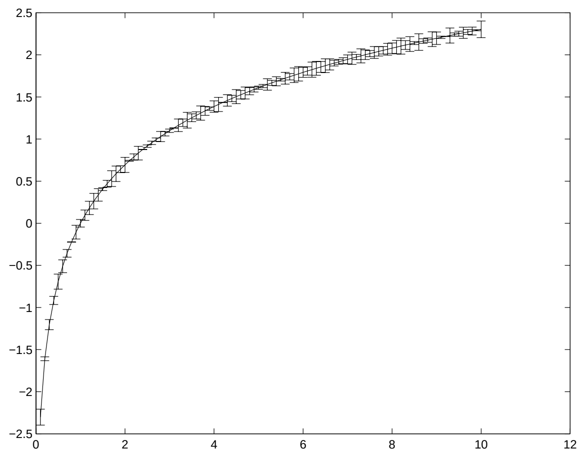


¿Qué pasa si se sustituye la función `randn` por la función `rand`?.

Se pueden dibujar gráficas con barras de error. Para ello, se usa la función `errorbar`. Un ejemplo de su uso es el siguiente

```
x=0.1:0.1:10;
y= log(x);
```

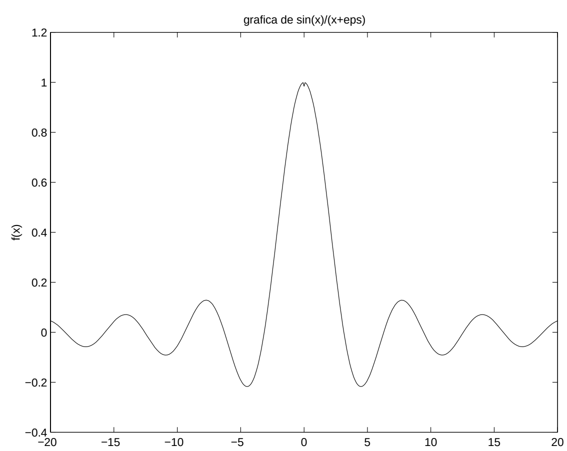
```
e=rand(size(x))/10;  % crea un vector de errores aleatorio
errorbar(x,y,e)
```



La función `fplot` permite generar la gráfica de una función de una variable, sin necesidad de generar vectores con los datos. La sintaxis es de la forma `fplot('fun',[xmin xmax])` o bien `fplot('fun',[xmin xmax ymin ymax])`.

Un ejemplo de su uso es el siguiente:

```
fplot('sin(x)./(x+eps)',[-20 20 -0.4 1.2]);
title('grafica de sin(x)/(x + eps)');
xlabel('x')
ylabel('f(x)')
```



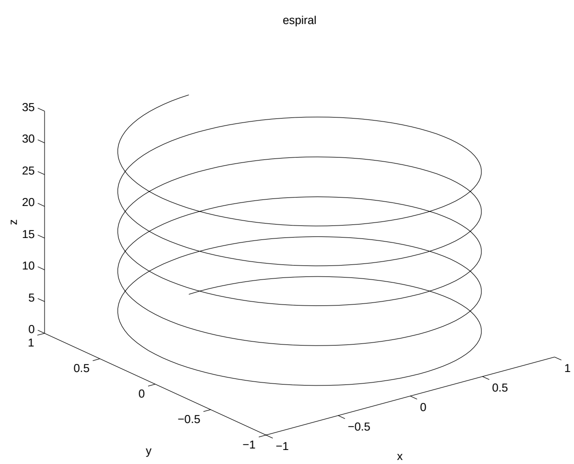
1.2. Gráficos 3D

Para generar la representación de una curva en el espacio, se puede utilizar la función plot3. Supongamos que se quiere dibujar la espiral

$$E = \begin{cases} x(t) = \cos(t) \\ y(t) = \mathrm{sen}(t) \\ z(t) = t \ , \quad t \in [0, 10\pi] \ . \end{cases}$$

Para ello, se puede escribir

```
t=0:pi/50:10*pi;
x=sin(t);
y=cos(t);
z=t;
plot3(x,y,z,'r');
title('espiral')
xlabel('x'), ylabel('y'),zlabel('z')
```



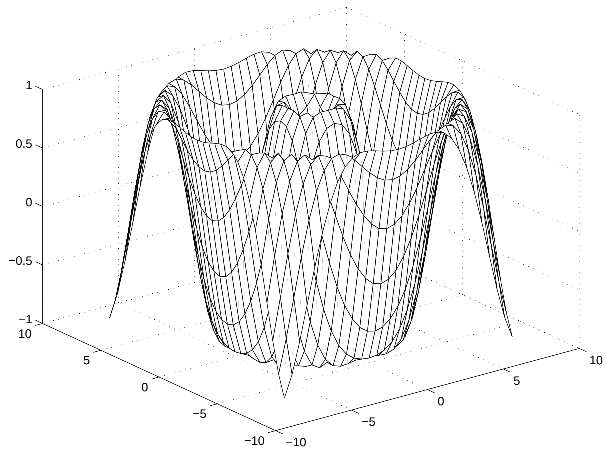
Se puede controlar la escala de los ejes mediante la función `axis([xmin, xmax, ymin,ymax,zmin,zmax])` de forma similar a como se hace para los gráficos bidimensionales. También la función `text(x,y,z,'texto')` nos permite escribir un texto en las coordenadas (x, y, z) del gráfico.

Supongamos ahora que se quiere representar una superficie $Z = z(x, y)$. Para ello, existen distintas funciones en Matlab que permiten hacer distintas representaciones. Consideremos la función

$$z(x, y) = \mathrm{sen}\left(\sqrt{x^2 + y^2}\right) \ .$$

Una representación de esta función puede hacerse como se muestra en el siguiente ejemplo:

```
x=-7.5:0.5:7.5;
y=x;
[X,Y]=meshgrid(x,y);
Z=sin(sqrt(X.^2+ Y.^2));
mesh(X,Y,Z)
```



Otra posibilidad es escribir

```
mesh(Z)
```

Una alternativa a la función `mesh` es la función `surf`. Así, se puede usar

```
surf(X,Y,Z)
```

Se pueden generar curvas de nivel de una determinada función mediante las funciones `contour` y `contour3`.

```
contour(X,Y,Z,20)
```

genera 20 curvas de nivel de la función $z(x, y)$ en el plano y

```
contour3(X,Y,Z,20)
```

da la representación de estas curvas en el espacio.

Se puede obtener una representación bidimensional mediante colores de una función $z(x, y)$ mediante la función `pcolor`. Así,

```
pcolor(Z)
```

produce esta representación con el juego de colores que tiene Matlab por defecto. El juego de colores que usa Matlab se puede cambiar mediante la función `colormap`. Distintas posibilidades de colores que tiene matlab implementadas se muestran en la siguiente tabla

función	Descripción
hsv	Saturación de tonos
hot	Negro, rojo, amarillo y blanco
pink	Sombras pastel rosa
gray	Escala de grises
bone	Escala de grises con matices azules
jet	Una variante de hsv
copper	Tonos cobre
prism	prisma
flag	rojo, blanco, azul y negro alternativos

Se pueden combinar estas funciones de la siguiente manera:

```
colormap(hot)
pcolor(X,Y,Z)
shading flat   % quita la rejilla
hold on
contour(X,Y,Z,20,'k')
hold off
```

Se puede cambiar el punto de vista de las representaciones tridimensionales de funciones mediante el comando `view`. Una posibilidad es pasarle a la función el azimut y la elevación en grados de la direccion en la que se quiere mirar

```
view(phi,theta)
```

o bien se la pasa un vector en esa dirección

```
view([x1,x2,x3])
```

1.3. Ejercicios

1. Representar gráficamente las siguientes curvas:

   a) $y = x^5 + 3x + 5, \quad x \in [0, 1]$,

   b) $y = e^{3x} + \operatorname{sen}(x)$, $x \in [-0{,}5, 0{,}5]$,

   c) $\left\{ \begin{array}{l} x(t) = t - \operatorname{sen}(t) \\ y(t) = t - \cos(t) \end{array} \right. , t \in [0, 4\pi]$.

   d) $\rho = \cos(\theta) + \operatorname{sen}(\theta/4)$, $\theta \in [0, 2\pi]$.

2. Los polinomios de Legendre se definen mediante la siguiente relación de recurrencia

$$(n+1)P_{n+1}(x) - (2n+1)xP_n(x) + nP_{n-1}(x) = 0 \ ,$$

   con $P_0(x) = 1$, $P_1(x) = x$ y $P_2(x) = (3x^2 - 1)/2$. Calcula los tres polinomios de Legendre siguientes y dibuja los 6 primeros polinomios de Legendre en el intervalo $[-1, 1]$.

3. Utilizando el comando adecuado divide la ventana de gráficos en cuatro ventanas y dibuja en cada una de estas vetanas, una de las siguientes funciones:

$$\begin{aligned} y &= \tan(x) \ , \quad x \in [0, 0{,}5] \ , \\ y &= \cosh(5 * x) \ , \quad x \in [0, 10] \ , \\ y &= e^{2x} + 5\operatorname{sen}(2x) \ , \quad x \in [0, \pi] \ , \\ y &= x^2 + 3x + 4, \ x \in [0, 1] \ . \end{aligned}$$

4. Averigua, aproximadamente, el punto de corte de las siguientes funciones en el intervalo [0,2]

$$\begin{aligned} y &= 2x^3 + 5{,}4x^2 + 4{,}8x + 1{,}4 \ , \\ y &= 7x + 10 \ . \end{aligned}$$

5. Obtener una representación de la superficie

$$z(x, y) = \sqrt{x^2 + y^2} \ ,$$

   para $(x, y) \in [-1, 1] \times [-1, 1]$.

6. Dibuja la gráfica de la superficie

$$z(x, y) = x^2 + y^2 \ ,$$

   en el dominio $[-1, 1] \times [-1, 1]$ y superpón a la representación 5 curvas de nivel.