

Matlab

INSTALACIÓN DE MATLAB

La Universidad de Cantabria ofrece una licencia MATLAB, Simulink con más de 80 herramientas para toda la comunidad universitaria.

Todos los estudiantes pueden descargarse e instalar estos productos tanto en los equipos de la Universidad de Cantabria, como en equipos particulares, siempre y cuando su uso se destine a fines docentes y de investigación, quedando prohibido su uso para fines comerciales privados.

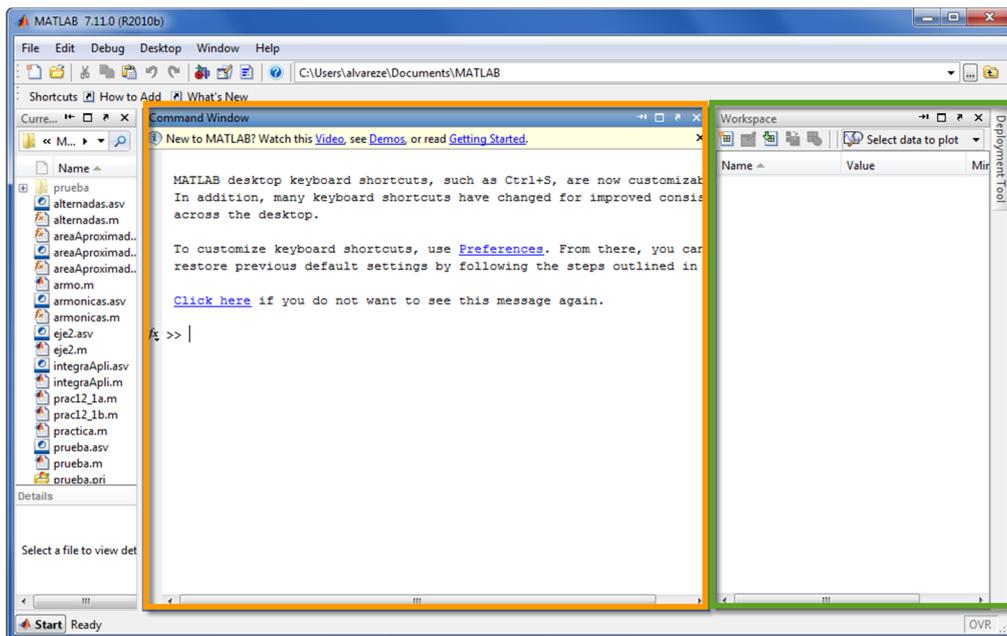
Información:

- Para obtener información de cómo descargar e instalar Matlab consulta el siguiente enlace:
<https://sdei.unican.es/Paginas/servicios/software/Matlab-Campus.aspx>
- También es posible utilizar Matlab a través del portal de aplicaciones Porticada
https://sdei.unican.es/Paginas/servicios/servicios_web/Porticada.aspx
- Porticada también es accesible en nuestros propios equipos
https://sdei.unican.es/Paginas/servicios/servicios_web/Porticada-propios.aspx

INTRODUCCIÓN

1 ¿Qué es Matlab?

El nombre de Matlab es un acrónimo de MATrix LABoratory. Hoy en día Matlab es un programa muy potente con un entorno agradable, que incluye herramientas de visualización gráfica, así como un lenguaje de alto nivel.

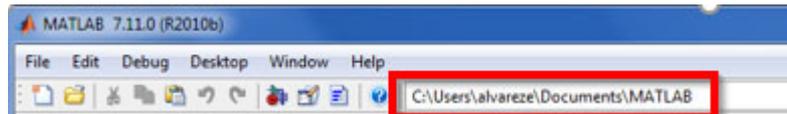


La ventana de Matlab muestra un escritorio dividido en varias partes:

- Las órdenes se escriben en la ventana de comandos, *Command Window*.
- La ventana *Workspace* proporciona información sobre las variables utilizadas y permite explorar datos que cree o importe de archivos.
- *Current Folder*: para acceder a los archivos.

2 Directorio de trabajo. Sesión de trabajo.

El recuadro señalado en la figura siguiente sirve para seleccionar el directorio de trabajo.



A medida que vayamos escribiendo código en ficheros lo guardaremos en algún lugar del ordenador. Para poder utilizarlos en un futuro es importante que Matlab sepa dónde lo hemos dejado. Matlab ya sabe, porque así viene configurado de fábrica, dónde tiene guardadas las funciones propias de la aplicación y de los distintos toolkits pero no sabe dónde están las que hemos escrito.

Advertencia: Matlab busca funciones y scripts en los directorios especificados por la función `path`. El primero de ellos es siempre el especificado en el diálogo `Current Directory`.

Para iniciar una sesión en las salas de uso común de informática se pueden seguir los siguientes pasos :

- Introducir un pendrive y crear una carpeta con el nombre que se corresponda al contenido que contendrá, por ejemplo "*practica1*".
- Entrar en Matlab y hacer que *Current Directory* sea la carpeta creada "*practica1*".
- Para conservar nuestra sesión de trabajo en un fichero de texto, empezar tecleando:


```
>>diary practical
```
- Al terminar la sesión teclear:


```
>>diary off
```

De esta manera en la carpeta se generará un fichero de texto que contendrá todo lo que se haya ejecutado durante la sesión.

3 Operadores aritméticos

Al trabajar en MATLAB, se deben ejecutar comandos para crear variables y llamar a funciones propias de Matlab o creadas por los usuarios. Por ejemplo, para crear una variable denominada `a`, debe introducir esta instrucción en la línea de comandos y pulse la tecla *intro* o salto de línea.

```
>> a = 1
```

MATLAB agrega la variable `a` al área de trabajo y muestra el resultado en la ventana de comandos.

```
a =
```

```
1
```

Creemos algunas variables más.

```
>> b = 2
```

b =

2

```
>> c = a + b
```

c =

3

```
>> d = cos(a)
```

d =

0.5403

Cuando no se especifica una variable de salida, MATLAB utiliza la variable `ans`, abreviatura de *answer* (respuesta), para almacenar los resultados del cálculo.

```
>> sin(a)
```

ans =

0.8415

Si una instrucción finaliza con punto y coma, MATLAB realiza el cálculo, pero elimina la visualización de la salida en la ventana de comandos.

```
>> e = a*b;
```

Para recuperar comandos anteriores, presione las teclas de flecha arriba y abajo: \uparrow y \downarrow . Presione las teclas de flecha en la línea de comandos vacía o después de escribir los primeros caracteres de un comando. Por ejemplo, para recuperar el comando `b = 2`, escriba `b` y luego presione la tecla de flecha arriba.

La forma de representar números y de operar con Matlab es la misma que la de una calculadora de bolsillo.

```
>>3.2+7
```

Las operaciones básicas se hacen con los mismos símbolos y en la misma secuencia que en las calculadoras.

Operador	Utilización	Ejemplo
+	Adición	2+3
-	Sustracción	2-3
*	Multiplicación	2*3
/	División	2/3
^	Potenciación	2^3

Las expresiones se evalúan de izquierda a derecha, la potencia tiene el **orden de prioridad** más alto, seguido del producto y la división (ambas tienen la misma prioridad) y por último la suma y la resta (con igual prioridad entre ellas). Para alterar este orden se deben introducir adecuadamente paréntesis.

Ejemplo: (a) Calcular $2 + 3$, $\cos(\pi)$, $\sqrt{4}$, $1/3$, $\frac{(3^3 + 4)^2 e^{\sqrt{3}}}{1 + \cos(2)}$

(b) ¿son iguales los resultados de $(2^3)^4$ y $2^{(3^4)}$

```
>>2+3
>>cos(pi)
>>sqrt(4)
>>1/3
>>(3^2+4)^2*exp(sqrt(3))/(1+cos(2))
>>(2^3)^4
>>2^(3^4)
```

4 Comentarios

Matlab ignora cualquier texto que vaya precedido por el símbolo % por lo que % sirve para incluir comentarios.

```
>>3+5^3-2 %2+3 no se evalúa
ans=
    126
```

5 Recuerda:

- Se pueden introducir varias órdenes en una misma línea, separadas cada una de las demás por una coma o por un punto y coma.
- Matlab distingue entre minúsculas y mayúsculas. No es lo mismo $\cos(3)$ que $\text{Cos}(3)$
- `clc` (clear console) deja en blanco la ventana de comandos, eliminando las órdenes introducidas. Sin embargo, siguen estando en el historial de comandos.
- Para interrumpir la ejecución de una instrucción se deben pulsar simultáneamente las teclas "Control" y "C"

6 Variables

Una variable es un nombre que se da a una entidad que puede ser una matriz, un vector, un escalar. El valor de esa variable, e incluso el tipo de la entidad que representa, puede variar a lo largo de una sesión de Matlab. Para asignar un valor a una variable se escribirá:

```
nombreVariable=expresión
```

Si quisiéramos guardar el resultado en otra variable escribiríamos

```
>> operacion=3+5^3-2
>> operacion=operacion+5
>> %El valor de operación después de las instrucciones es 131
```

REGLAS PARA NOMBRAR VARIABLES

- El nombre de una variable puede tener como máximo 63 caracteres que pueden ser letras, números y el guion de subrayar
- El primer carácter tiene que ser una letra, `modulo2` es un nombre válido, pero no lo es `2modulo`.
- Las mayúsculas y las minúsculas tienen valor distintivo. La variable `Modulo` es distinta de la variable `modulo`.
- Dentro de un nombre de variable no puede haber espacios en blanco, `modulo1` es un nombre de variable válido, pero no `modulo 1`.
- Existen nombres que deben evitarse porque tienen significado propio en Matlab: `ans`, `pi`, `Inf`, `i`, `j`, ... Para obtener una lista de palabras reservadas teclea

```
>> iskeyword()
```

Ejemplo: Define tres variables $a=1.5$, $b=4$, $c=3.5$ y calcula el valor de

$$d = \frac{a}{\frac{b}{c} + \frac{c^2}{a}}$$

```
>>a=1.5;b=4;c=3.5;
>>d=a/(b/c+c^2/a)
%d=0.1611
```

Ejemplo: El flujo de un gas que escapa de un tanque a presión P_o y en condiciones adiabáticas es

$$\phi = \sqrt{\frac{k}{k-1}} \sqrt{\left(\frac{P_e}{P_o}\right)^{2/k} - \left(\frac{P_e}{P_o}\right)^{(k+1)/k}}$$

donde P_e es la presión externa y K la constante del gas reversible adiabático. Escribe esta expresión en Octave y calcula su valor para los valores de $K=1.4$ y $\frac{P_e}{P_o} = 0.3$.

```
>>k=1.4;A=0.3;
>>phi=sqrt(k/(k-1))*sqrt(A^(2/k)-A^((k+1)/k))
```

```
%Sol:  $\phi = 0.4271$ 
```

Ejemplo: Indica cuáles de las siguientes expresiones son incorrectas y por qué

1. Numero-6+2
2. 8num=3*2
3. Num valores=3+2
4. A234_7899000=3
5. B-32=0

Solución: Son incorrectas 2, 3 y 5 ya que el nombre de una variable no puede empezar por un número, no puede tener un espacio en blanco y la forma de hacer una asignación es

nombreVariable=valor

Ejemplo: ¿Qué resultado esperarías encontrar el valor de x después de ejecutar el siguiente código en Matlab?

```
>> x=5;
>> 2*x;
>> y=x^2;
>> x=y/x;
```

Solución: 5

3 Ficheros .m

En el caso de incluir órdenes complicadas o la repetición de las mismas órdenes con distintos valores de las variables, la utilización de la ventana de comandos no es lo más adecuado. Octave permite utilizar ficheros-M.

La secuencia de órdenes contenida en un fichero-M constituye un programa y se podrá ejecutar fácilmente cuando se desee.

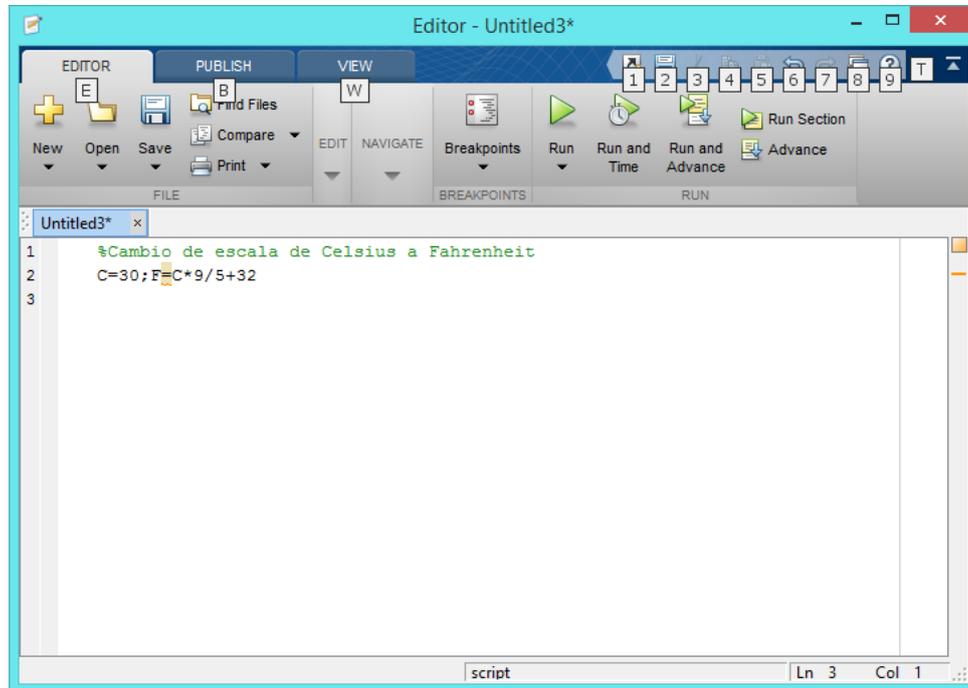
Para crear un fichero-M utilizaremos el editor de texto cualquiera y lo guardaremos como `nombre.m` siendo nombre una secuencia de caracteres que no admite caracteres blancos.

Para abrir el editor de Matlab basta elegir en el menú New Script



Ejemplo de un fichero-M

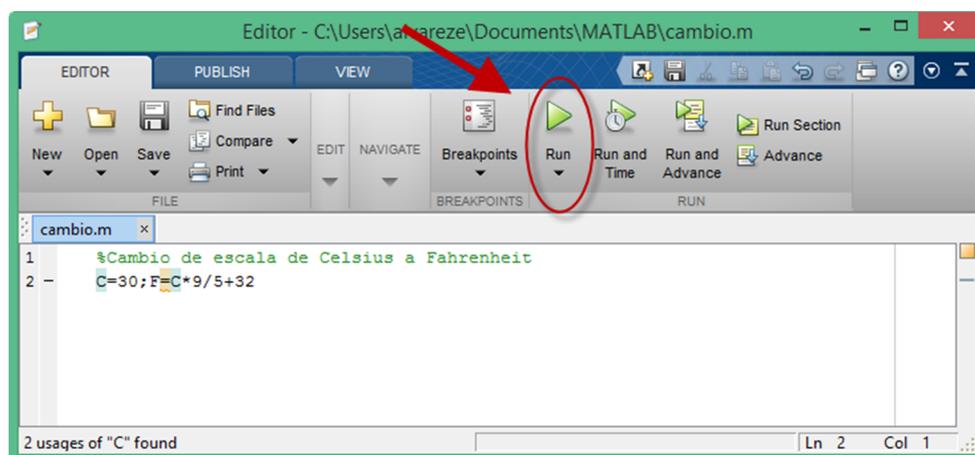
```
%Cambio de escala de Celsius a Fahrenheit
C=30;F=C*9/5+32
```



- Para ejecutar este fichero basta con escribir en la ventana de comandos el nombre de fichero sin la extensión m. Así, si el fichero se llama `cambio.m` se deberá escribir

```
>> cambio
F=86
```

También se puede ejecutar desde el propio editor



AYUDA EN MATLAB

Para obtener ayuda desde la Command Window (ventana de comandos) hay que ejecutar `help`, `lookfor`, `doc`, o `type` seguido del comando del que se requiere la ayuda.

```
>>help plot
>>lookfor graph
>>doc plot
>>type linspace
```

Para encontrar comandos de Matlab :

- Ir a *Product help* en el menú *Help* de la ventana principal de Matlab.
- En *MATLAB* buscar el tema que interese.

FORMATOS DE NÚMEROS

En Matlab, todas las variables se almacenan con doble precisión (16 decimales). Sin embargo, se puede cambiar el formato con el que se muestran las variables. Las siguientes órdenes afectan a la impresión del número.

Orden	Decimales	pi
<code>format short</code>	4 dígitos decimales	3.1416
<code>format long</code>	14 dígitos decimales	3.14159265358979
<code>format short e</code>	4 dígitos decimales	3.1416e+00
<code>format long e</code>	14 dígitos decimales	3.14159265358979e+00
<code>Format rat</code>	forma fraccional	

VECTORES Y MATRICES (ARRAYS)

Matlab es un programa que trabaja principalmente con vectores y matrices.

Para crear un vector con cuatro elementos en una fila única, separe los elementos con una coma (,) o un espacio incluyéndolos entre corchetes.

```
a = [1 2 3 4]
```

$a = 1 \times 4$

```
1 2 3 4
```

Este tipo de array es un *vector fila*.

Para crear una matriz con varias filas, separe las filas con punto y coma.

```
a = [1 2 3; 4 5 6; 7 8 10]
```

$a = 3 \times 3$

```
1  2  3
4  5  6
7  8 10
```

Otra forma de crear una matriz es usando una función, como `ones`, `zeros` o `rand`. Por ejemplo, para crear un vector columna de ceros de 5 por 1 se puede utilizar el comando `zeros`.

```
z = zeros(5,1)
```

$z = 5 \times 1$

```
0
0
0
0
0
```

OPERACIONES ENTRE VECTORES Y MATRICES

MATLAB le permite procesar todos los valores de una matriz mediante el uso de una sola función u operador aritmético. Por ejemplo para sumar a todos los elementos de una matriz el número 10 se escribiría:

```
a + 10
```

$ans = 3 \times 3$

```
11 12 13
14 15 16
17 18 20
```

También es posible aplicar una función a todos los elementos de un vector o una matriz, por ejemplo el seno a cada uno de sus componentes se escribiría:

```
sin(a)
```

$ans = 3 \times 3$

```
0.8415  0.9093  0.1411
-0.7568 -0.9589 -0.2794
0.6570  0.9894 -0.5440
```

Para transponer una matriz, use una comilla simple (`'`):

```
a'
```

$ans = 3 \times 3$

```
1  4  7
```

```
2 5 8
3 6 10
```

Para realizar una multiplicación de matrices estándar, la cual calcula los productos internos entre filas y columnas, se debe usar el operador `*`, por ejemplo, para confirmar que el producto de una matriz por su inversa (comando `inv` en Matlab) es igual a la matriz identidad se podría escribir

```
p = a*inv(a)
```

`p = 3×3`

```
1.0000    0    0
0.0000  1.0000    0
0.0000 -0.0000  1.0000
```

Observe que `p` no es una matriz de valores enteros. MATLAB almacena los números como valores de punto flotante, y las operaciones aritméticas son sensibles a pequeñas diferencias entre el valor real y su representación en punto flotante. Para visualizar más dígitos decimales, use el comando `format`:

```
format long
p = a*inv(a)
```

`p = 3×3`

```
1.0000000000000000    0    0
0.0000000000000002  1.0000000000000000    0
0.0000000000000002 -0.0000000000000004  1.0000000000000000
```

Restablezca la visualización al formato más corto con

```
format short
```

`format` influye solo en la visualización de números, no en la forma en que MATLAB los calcula o los guarda.

OPERACIONES ENTRE VECTORES Y MATRICES ELEMENTO A ELEMENTO

Si `a` y `b` son matrices y `r` es un escalar, la forma de indicar a Matlab que realice las operaciones algebraicas habituales se realiza mediante los operadores ya vistos de suma (+), producto (*) y exponenciación (^). Para poder realizar estos cálculos únicamente es necesario que los vectores y matrices tengan la dimensión adecuada para que la operación pueda realizarse.

```
>>a*b+r*a^2
```

Si queremos realizar estas operaciones algebraicas “elemento a elemento” los operadores que debemos utilizar son los que se muestran a continuación.

Operadores entre vectores	Utilización	Ejemplo
---------------------------	-------------	---------

.*	Multiplicación término a término	$[2\ 3].*[2\ 4]=$ $= [4\ 12]$
./	División término a término	$[2\ 3]./[2\ 4]=$ $= [1\ 0.7500]$
.^	Potenciación término a término	$[2\ 3].^2 = [4\ 9]$

Para realizar una multiplicación elemento por elemento en lugar de una multiplicación de matrices, utilice el operador `.*`. Por ejemplo, para la matriz

```
a = [1 2 3; 4 5 6; 7 8 10]
```

`a = 3x3`

```
1 2 3
4 5 6
7 8 10
```

el producto elemento a elemento es

```
p = a.*a
```

`p = 3x3`

```
1 4 9
16 25 36
49 64 100
```

Cada uno de los operadores de matriz para la multiplicación, la división y la potencia tiene un operador de array correspondiente que opera elemento por elemento. Por ejemplo, eleve cada elemento de `a` a la tercera potencia:

```
a.^3
```

`ans = 3x3`

```
1 8 27
64 125 216
343 512 1000
```

Ejemplo: Calcula las cinco primeras potencias de $x = 2$
Calcula las cinco primeras potencias de exponente par de $x = 2$

```
%Definimos un vector con los exponentes
>>n=1:5;
%Elevamos 2 a cada componente del vector n
>>2.^n
```

```
%Definimos un vector con cinco primeros números pares
>>n=2:2:10;
%Elevamos 2 a cada componente del vector n
>>2.^n
```

CONCATENACIÓN

La *concatenación* es el proceso que consiste en unir arrays para crear otros más grandes. De hecho, el primer array de este capítulo resultó de la concatenación de sus elementos individuales. El operador de concatenación es el par de corchetes: `[]`.

```
A = [a, a]
```

A = 3×6

```
1  2  3  1  2  3
4  5  6  4  5  6
7  8 10  7  8 10
```

La concatenación de arrays uno junto a otro mediante el uso de comas se conoce como concatenación *horizontal*. Cada array debe tener la misma cantidad de filas. De forma similar, cuando los arrays tienen la misma cantidad de columnas, es posible concatenarlos *verticalmente* mediante el uso de punto y coma.

```
A = [a; a]
```

A = 6×3

```
1  2  3
4  5  6
7  8 10
1  2  3
4  5  6
7  8 10
```

FORMAS DE DEFINIR UN VECTOR

Para definir un **vector fila** se puede:

- a) Introducir sus componentes separadas por un espacio o una coma

```
>> w=[1 4 9]
```

- b) Introducir sus componentes, especificando el valor de cada componente

```
>> w[1]=1, w[2]=4, w[3]=9
```

- c) Utilizar el operador, colon (:). El comando
a:h:b

genera un vector fila de primer elemento a y los demás elementos aumentan de h en h hasta no superar b .

```
>> v=2:9
% Devuelve v = 2 3 4 5 6 7 8 9
>> w=-5:2:5
% Devuelve w = -5 -3 -1 1 3 5
```

d) Utilizar el comando

```
linspace(a,b,n)
```

que genera un vector fila de n componentes cuyo primer elemento es a y el último b , siendo todos sus elementos equidistantes.

```
>> w=linspace(-5,5,10)
% Devuelve w= -5.0000 -3.8889 -2.7778 -1.6667 -0.5556
0.5556 1.6667 2.7778 3.8889 5.0000
% w es un vector de 10 números igualmente espaciados entre -5 y 5.
```

Si se quiere definir un **vector columna** basta hacer lo mismo que para un vector fila pero separando por un punto y coma cada fila.

```
>>w=[-1;2;3]
>>a=[1 2 3; 4 5 6] %matriz de 2 filas y 3 columnas
```

También es posible definir un vector fila y luego calcular el vector traspuesto

```
>>w=[-1 2 3]
>>a=w'
```

FUNCIONES

MATLAB® ofrece una amplia gama de funciones que realizan tareas computacionales. Las funciones son equivalentes a las *subrutinas* o *métodos* de otros lenguajes de programación.

Para llamar a una función, como, por ejemplo, `max` (que calcula el máximo), encierre entre paréntesis sus argumentos de entrada:

```
A = [1 3 5];
max(A)
```

```
ans = 5
```

Algunas funciones matemáticas

Funciones	Utilización	Ejemplo
<code>exp(x)</code>	Exponencial de x	<code>exp(1)=2.7183</code>
<code>log(x)</code>	Logaritmo natural	<code>log(2.7183)=1.0000</code>
<code>log10</code>	Logaritmo en base 10	<code>log10(350)=2.5441</code>
<code>sin(x)</code>	Seno de x	<code>sin(pi/6)=0.500</code>

Funciones	Utilización	Ejemplo
cos(x)	Coseno de x	cos(0)=1
tan(x)	Tangente de x	tan(pi/4)=1.000
asin(x)	Arco seno de x con imagen en el rango $[-\pi/2, \pi/2]$	asin(1)=1.5708
acos(x)	Arco coseno de x con imagen en $[-\pi/2, \pi/2]$	acos(1)=-6.1257e-17
atan(x)	Arco tangente de x con imagen en el rango $[-\pi/2, \pi/2]$	atan(1)=0.7854
atan2(y,x)	Arco tangente de y/x con imagen en el rango $[-\pi, \pi]$	atan2(0,-1)=3.1416
sinh(x)	Seno hiperbólico de x	sinh(3)=10.0179
cosh(x)	Coseno hiperbólico de x	cosh(3)=10.0677
tanh(x)	Tangente hiperbólica de x	tanh(3)=0.9951

NÚMEROS COMPLEJOS

```
sqrt(-1)
```

```
ans = 0.0000 + 1.0000i
```

i (ó j)

Es la unidad imaginaria en Matlab

```
>> z=2+3*i; w=7+5*i; z+w
```

abs(s)

Valor absoluto de los elementos de "s" o módulo en el caso de ser complejos.

```
>> z=2+3*i; w=5+7*i;
>> abs(z) % Devuelve 3.6056
>> abs([z,w]) % Devuelve 3.6056 86023
```

angle(h)

Retorno el ángulo de fase en radianes de cada elemento de la matriz h con elementos complejos.

```
>> z=2+3i; w=5+7i;
>> angle(z) % Devuelve 0.9828
>> angle([z,w]) %Devuelve 0.9828 0.9505
```

`real(z)`

Devuelve la parte real de z

```
>> z=2+3i; w=5+7i;
>> real(z)           % Devuelve 2
>> real([z,w])      % Devuelve 2    5
```

`imag(z)`

Devuelve la parte imaginaria de z

```
>> z=2+3i; w=5+7i;
>> imag(z)          % Devuelve 3
>> imag([z,w])     % Devuelve 3    7
```

`conj(z)`

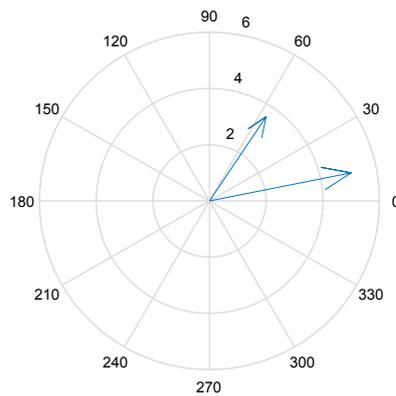
Devuelve el conjugado de z

```
>> z=2+3i; w=5+7i
>> z=2+3i; w=5+7i
>> conj(z)
% Devuelve 2.0000-3.0000i
>> conj([z,w])
% Devuelve 2.0000-3.0000i  5.000-7.000i
```

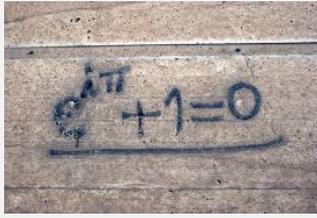
`compass(z)`

Representa los números complejos almacenados en la matriz z

```
>> z=2+3i; w=5+i; compass([z w])
```



Ejemplo: Una de las fórmulas más bella de las matemáticas ¹es



ya que interviene 5 de los números más importantes. Comprueba con Matlab esta expresión. Nota: La constante π se escribe en Matlab como pi.

Ejemplo: Elige dos números complejos y comprueba que el módulo del producto es el producto de sus módulos y que el argumento es la suma de sus argumentos.

Nota: Utiliza las funciones abs y angle.

Ejemplo: (a) Calcular la parte real y la parte imaginaria del siguiente número

$$z = (1 + \sqrt{2} i)^8 e^{1+\pi i}$$

(b) Calcula también el número complejo w resultado de girar 120 grados el número z . Recuerda que para girar un número complejo α radianes debes multiplicar al número por un número complejo de módulo 1 y argumento α .

```
%Apartado a
z=(1+sqrt(2)*i)^8*exp(1+pi*i)
real(z)
imag(z)
%Apartado b
w=z*exp(2*pi/3*i)
```

Ejemplo: Calcula las diez primeras potencias de $z = 1 + i$
Calcula las diez primeras potencias de $w = 1 - i$

- Escribe un vector con las componentes 1 a 10 y guárdalo en una variable de nombre exponente

```
>>n=1:10
```

También podría escribirse

```
>>n=linspace(1,10,10)
```

- Escribe

¹ Ver información en <https://www.microservos.com/archivo/ciencia/e-pi-i-formula-mas-bella-explicada.html>

```
>> (1+i).^n
```

REPRESENTACIÓN DE PUNTOS EN EL PLANO

`plot(x,y)`

Dibuja una línea que une los puntos de abscisas el vector "x" y ordenadas "y".

`plot(y)`

Dibuja una línea que une los puntos del vector "y" considerado como abscisas su índice. Si "y" es complejo es equivalente a dibujar `plot(real(y),imag(y))`.

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

Dibuja los puntos que tienen de abscisas las componentes del vector "x" y con ordenadas las componentes del vector "y"

```
>> x=1:0.5:5;
>> y=x.^2;
>> plot(x,y,'o') %Dibujaría solo los puntos (x,y)
>> plot(x,y) %Dibuja la gráfica de la función uniendo los puntos
```

`plot(x,y,s)`

Realiza el gráfico con el estilo indicado en "s". Para ello "s" debe ser una cadena de caracteres formada por uno o ningún elemento de las tres columnas siguientes:

y	yellow	.	point	-	solid
m	magenta	o	circle	:	dotted
c	cyan	x	x-mark	-.	dashdot
r	red	+	plus--		dashed
g	green	*	star		
b	blue	s	square		
w	white	d	diamond		
k	black	v	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		
		h	hexagram		

```
>> n=1:10
>> a=2.^n;
>> plot(a,'bo')
>> %Para ver más opciones teclea la orden:
>> help plot
```

Ejercicio. Representar la función $f(x) = \frac{10x+5}{x^4+x^3+2x-4}$

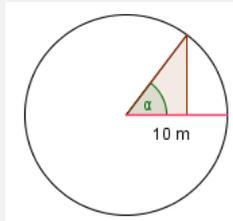
```
x=linspace(-5,5);
y=(10*x+5)./(x.^4+x.^3+2*x-4);
plot(x,y)
```

Ejercicio. Gráfica de una función a trozos. Dibujar la gráfica de la siguiente función:

$$f(x) = \begin{cases} \operatorname{tg}(x) & \text{si } -\pi/4 \leq x \leq \pi/4 \\ \cos(x - \pi/4) & \text{si } \pi/4 < x \leq \pi/2 \\ e^x & \text{si } \pi/2 < x \leq 3 \end{cases}$$

```
x1=-pi/4:pi/200:pi/4;
y1=tan(x1);
x2=pi/4:pi/200:pi/2;
y2=cos(x2-pi/4);
%Linspace genera 100 puntos entre dos números dados, se puede
%indicar también el número de puntos de la forma linspace(x1,x2,n)
x3=linspace(pi/2,3);
y3=exp(x3);
x=[x1,x2,x3];
y=[y1,y2,y3];
```

Ejercicio. Considera la región sombreada



Expresa el área de la región en función de α , $A = f(\alpha)$, cuando $\alpha \in [0, \pi]$.

Teniendo en cuenta la figura, la función área será:

$$A = \begin{cases} f(\alpha) = \frac{10 \operatorname{sen} \alpha \cdot 10 \cos \alpha}{2} = 50 \operatorname{sen} \alpha \cos \alpha = 25 \operatorname{sen}(2\alpha) & \text{si } \alpha \in \left[0, \frac{\pi}{2}\right] \\ f(\pi - \alpha) & \text{si } \alpha \in \left[\frac{\pi}{2}, \pi\right] \end{cases}$$

```
t1=0:0.01:pi/2;
t2=pi/2:0.01:pi;
t=[t1 t2];
% f en el intervalo [0, pi/2]
f1=25*sin(2*t1);
% f en el intervalo [pi/2, pi]
f2=25*sin(2*(pi-t2));
plot(t1,f1,t2,f2,'b')
```

figure(n)

Para crear una ventana de dibujo

```
>> x=-pi : 0.1: pi;
>> figure(1);
>> plot(x,sin(x),'b. ');
>> figure(2);
>> plot(x,cos(x), 'gd-');
```

hold on hold off

Permite dibujar dos gráficas en una misma ventana de dibujo.

```
>> x=-pi : 0.1: pi;
>> hold on
>> figure(1);
>> plot(x,sin(x),'b. ');
>> plot(x,cos(x), 'gd-');
>> hold off
```

compass(z)

Representa el número complejo como una flecha que tiene su origen en el punto (0,0).

```
>> z=3+2*i;
>> figure(1);
>> plot(z);
>> figure(2);
>> compass(z);
>> % Esto es equivalente a:
>> compass(real(z), imag(z));
```

fplot(f, [a,b], opciones)

```
fplot('sin(x^2)', [-2,2], '*--g')
```

REPRESENTACIÓN DE FUNCIONES IMPLÍCITAS

```
ezplot(f, [a,b], fig)
```

```
ezplot(f, [a,b,c,d], fig)
```

```
>> %El segundo y el tercer parámetro son opcionales.
>> ezplot('x^2+y^2-1', [-2,2])
```

Ejercicio. Representación de funciones.

a) Forma explícita: $y = |x^2 - 9| + 4$ en el intervalo $[-5, 5]$

b) Forma implícita: $x^2 + 4y^2 - 3xy - 5 = 0$

c) Forma paramétrica: $\begin{cases} x(t) = a \cos(nt) \cos(t) \\ y(t) = a \cos(nt) \sin(t) \end{cases}$ con el valor del parámetro t en un

cierto intervalo, por ejemplo, $t \in [0, 2\pi]$.

Según distintos valores de a y n se obtienen distintas curvas. Por ejemplo si $n=1$ la curva es una circunferencia. Puedes probar con los valores: $n=1/2$, $n=2/3$, $n=9/2$ y $a=5$.

```
% Apartado a
x1=-5:0.1:5;
y1=abs(x1.^2-9)+4;
plot(x1,y1)
axis([-5 5 -1 19])
grid on

% Apartado b
ezplot('x^2+4*y^2-3*x+y-5', [-5,5])
grid on

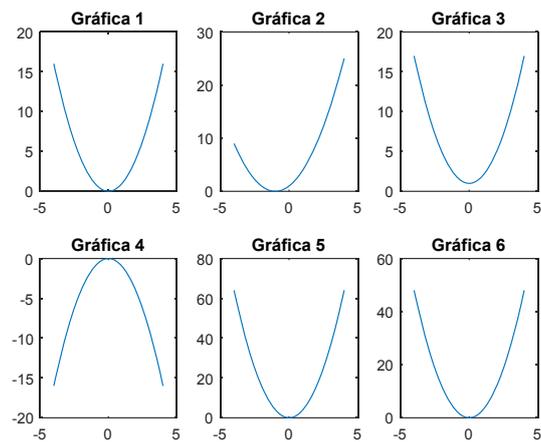
% Apartado c
% Linspace genera 100 puntos entre dos números dados,
% se puede indicar también el número de puntos de la
% forma linspace(x1,x2,n)
% Consideramos 200 puntos en el intervalo [0, 4*pi]
t=linspace(0,4*pi,200);
a=5;n=9/2;
x3=a*cos(n*t).*cos(t);
y3=a*cos(n*t).*sin(t);
plot(x3,y3)
```

MATRICES DE GRÁFICOS

El comando `subplot` permite subdividir la ventana gráfica en una retícula de m filas y n columnas. La función `subplot(m,n,p)` separa la figura en una matriz $m \times n$. 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.

```
x=-4:0.1:4;
subplot(2,3,1);plot(x,x.^2);title('Gráfica 1')
subplot(2,3,2);plot(x,(x+1).^2);title('Gráfica 2')
subplot(2,3,3);plot(x,x.^2+1);title('Gráfica 3')
subplot(2,3,4);plot(x,-x.^2);title('Gráfica 4')
subplot(2,3,5);plot(x,(2*x).^2);title('Gráfica 5')
subplot(2,3,6);plot(x,3*x.^2);title('Gráfica 6')
```



DEFINICIÓN DE FUNCIONES EN LÍNEA

`inline (expresion,argumento1,argumento2,...)`

```
>> f=inline('x^2')
>> f(2) %devuelve 4
>> %Si se quiere evaluar en un vector deberá escribirse
>> g=inline('x.^2')
>> x=-0:10;
>> g(x)
```

IMPRESIÓN EN PANTALLA

Con la instrucción `disp` se puede imprimir en pantalla el valor de una variable. Algunos ejemplos de su uso se muestran a continuación

```
>>a=[1 2 3 4];
>>disp('El valor de a es')
>>disp(a)
```

Si se quiere mostrar con `disp` texto y números hay que generar un vector concatenando las cadenas de caracteres convirtiendo el número en una cadena de caracteres que represente ese valor con el comando `num2str`

```
>>a=3;
>>disp(['El valor de a es', num2str(a)])
```

Para imprimir varias constantes o variables controlando la forma en la que se imprimen los datos se puede utilizar `fprintf`. Su sintaxis es

```
fprintf(formato, lista_datos)
```

El formato es una combinación de texto y códigos para escribir los datos numéricos que se aplican a la lista de datos según el orden en el que aparecen. El formato puede contener texto literal a imprimir y códigos de conversión formados por el símbolo `%`, una letra (f,e,i,s) y números para indicar el número de espacios que ocupará el dato a imprimir.

```
>>r=3.2;area=pi*r^2;
>>fprintf('El área del círculo de radio %3.1f es %10.7f
\n',r,area)
```

Matlab devolvería como resultado

```
>>El área del círculo de radio 3.2 es 32.1699088
```

La impresión ha consistido en la concatenación de

- La cadena de caracteres 'El área del círculo de radio'
- El valor de r en tres espacios siendo 1 para la cifra decimal
- La cadena de caracteres 'es'
- El valor de la variable area ocupando 10 espacios siendo siete para las cifras decimales
- Un salto de línea correspondiente al carácter `\n`

CÁLCULO SIMBÓLICO CON MATLAB

Aunque los cálculos en Matlab son por defecto numéricos. Por ejemplo, si calculamos $1/3 + 1/2$ el valor que nos devuelve Matlab es un valor aproximado en vez del valor exacto

```
>>1/2+1/3 %Devuelve 0.8333 en lugar de 5/6
```

Sin embargo, también podemos trabajar también de forma simbólica declarando objetos simbólicos.

1 Objetos simbólicos

`sym (expresion)`

Convierte a variable simbólica

```
>> sym(1/2+1/3) %Devuelve 1/6
```

`syms nombreVariable`

Crea una variable simbólica

```
>> syms x
>> x+x %Devuelve 2*x
```

También es posible indicar el tipo de objeto simbólico

```
syms arg1 arg2 ... real
```

Es la forma abreviada de escribir:

```
arg1 = sym('arg1','real');
arg2 = sym('arg2','real'); ...
```

```
syms arg1 arg2 ... positive
```

Es la forma abreviada de escribir:

```
arg1 = sym('arg1','positive');
arg2 = sym('arg2','positive'); ...
```

```
syms arg1 arg2 ... unreal
```

Es la forma abreviada de escribir:

```
arg1 = sym('arg1','unreal');
arg2 = sym('arg2','unreal'); ...
```

`double (expresion)`

Convierte a variable numérica

```
>> a=sym(pi) %Almacena en a la constante simbólica pi
>> a+1 %Devuelve pi+1
>> double(a+1) %Devuelve 4.1416
```

Una vez definidas las variables simbólicas podemos realizar distintas operaciones, calcular su valor en un punto, derivarlas, integrarlas...

2 Sustituir la variable simbólica en un punto

`subs (f, var, valor)`

```
>> syms x
>> y=sin(x)+3^x+8/(x+1)
>> subs(y, x, 2)
```

Cuando se quiere sustituir varias variables por diferentes valores se debe escribir tanto las variables como sus valores entre llaves

```
>> syms a b c
>> f=a^2+b^2+c^2; subs(f, {a,b,c}, {1,2,3})
```

2 Polinomio de Taylor de una función simbólica

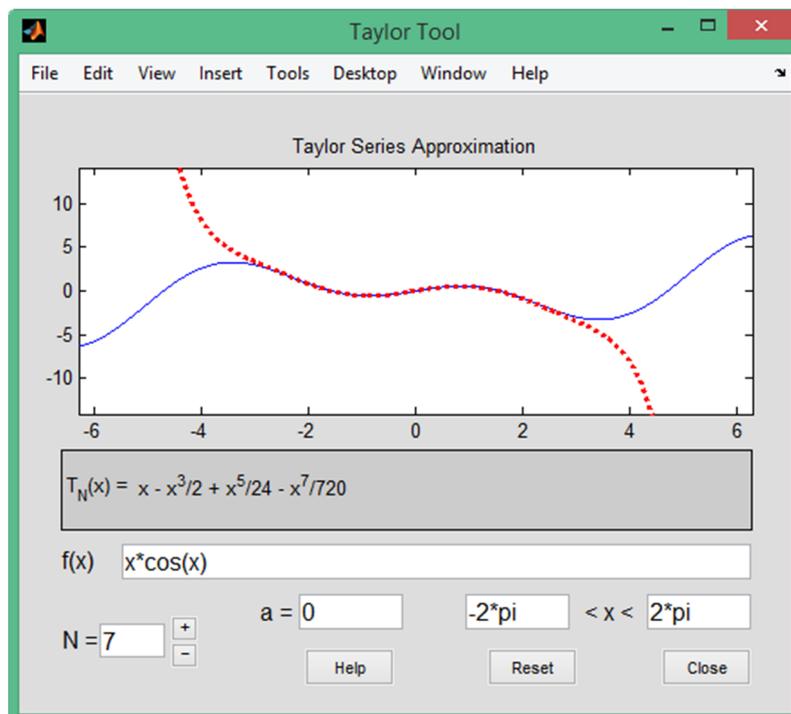
```
taylor(f, var, 'ExpansionPoint', a, 'Order', n);
```

Calcula el polinomio de Taylor de la función f en el punto a de grado $n-1$.

```
syms x
f=x*sin(x+1);
taylor(f,x, 'ExpansionPoint', 0, 'Order', 7);
%Devuelve el polinomio de Taylor de f en el punto 0 de grado 6.
```

Matlab también dispone de la herramienta `taylortool` que se invoca desde la ventana de comandos de la forma siguiente:

```
>> taylortool
```



3 Derivar una función simbólica

```
diff(f);
```

Deriva la función f respecto de la variable simbólica preferente

```
diff(f, var);
```

Deriva la función f respecto de la variable simbólica var

```
syms x y
f=x*sin(x+1);
diff(f)
g=x*sin(y+1)+cos(y)
diff(g, a)
```

4 Integrar una función simbólica

```
int(f);
```

Calcula una primitiva de la función f respecto de la variable simbólica preferente

```
int(f, var);
```

Calcula una primitiva la función f respecto de la variable simbólica var

```
int(f, a, b);
```

Integra la función f en el intervalo $[a,b]$ respecto de la variable simbólica preferente

```
int(f, var, a, b);
```

Integra la función f en el intervalo $[a,b]$ respecto de la variable simbólica var

```
syms x y
f=x*sin(x+1);
int(f) %Calcula una primitiva de f respecto de la variable x
int(f,2,3) %Integra f en el intervalo [2,3] respecto de x
g=x*sin(y+1)+cos(y)
int(f,y) %Calcula una primitiva de f respecto de y
int(f,y,2,3) %Calcula f respecto de y en el intervalo [2,3]
```

CURVAS Y SUPERFICIES EN EL ESPACIO

1 Curvas en el espacio

```
plot3(X,Y,Z,S)
```

Dibuja el conjunto de puntos (X,Y,Z) donde X , Y y Z son vectores fila y S son las opciones de dibujo.

```
plot3(X1,Y1,Z1,S1,X2,Y2,Z2,S2,...)
```

Dibuja sobre los mismos ejes los gráficos definidos por las tripletas (X_i, Y_i, Z_i) con las opciones de dibujo por Si.

```
t=linspace(0,4*pi,100);
x=cos(t);
y=sin(t);
z=t;
plot3(x,y,z)
```

Ejemplo. Representar la hélice circular $\begin{cases} x = \cos(t) \\ y = \text{sen}(t) \\ z = \frac{t}{5} \end{cases}$ con $t \in \mathbb{R}$

Solución:

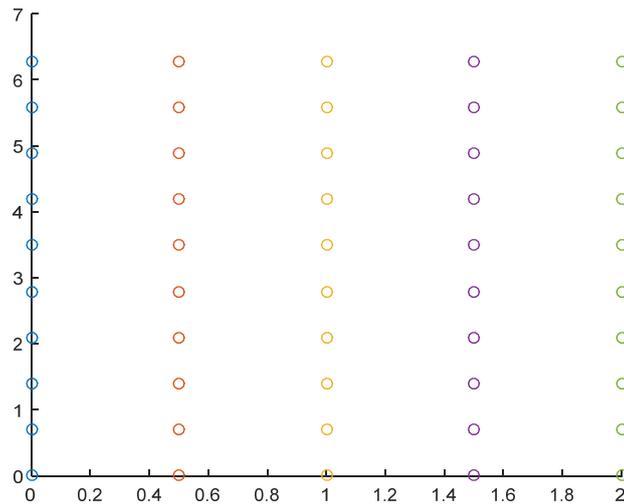
```
%Se define el rango del parámetro. Se declaran
%las variables en función del mismo y, por la
%instrucción 'plot3' se dibuja la hélice circular sobre el
espacio.
t=linspace(0,4*pi,1000);x=cos(t);y=sin(t);z=t/(2*pi);
plot3(x,y,z)
%Declarando el parámetro simbólicamente podemos
%representar la hélice circular a través de la
%instrucción 'ezplot3'
syms t
x=cos(t);y=sin(t);z=t/(2*pi);
ezplot3(x,y,z,[0,4*pi])
%Se explicita el dominio del parámetro, ya que
%por defecto MATLAB hubiese tomado el intervalo [0,2*pi]
```

2 Malla de puntos en una región rectangular del plano

```
meshgrid(x,y)
meshgrid(x) %Es equivalente a meshgrid(x,x)
```

```
%Para evaluar la función  $f(x,y)=x^2*y$  en
%el dominio  $-2 < x < 2$ ,  $-3 < y < 3$ 
>>[X, Y]=meshgrid(-2:.2:2,-3:0.5:3);
>>Z=X.^2.* Y
```

```
%Representación de los puntos (x,y,z) de la figura siguiente
x=linspace(0,2,5);
y=linspace(-1,1,6);
[X,Y]=meshgrid(r,phi);
Z=0*X;
plot3(X,Y,Z,'o')
```



3 Representación de una superficie sobre un dominio rectangular

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

Representa el gráfico de superficie de la función $z=f(x,y)$ con los colores especificados en C (este último parámetro se puede ignorar).

`surfc(X,Y,Z,C)`

Representa el gráfico de superficie de la función $z=f(x,y)$ junto con el gráfico de contorno correspondiente (curvas de nivel)

Representación de $z = f(x,y) = x^2 + y^2$ sobre el rectángulo

$$R = [-1,1] \times [-2,2] = \{(x,y) \in \mathbb{R}^2 / -1 \leq x \leq 1 \quad -2 \leq y \leq 2\}$$

```
% Se genera una malla de puntos en el dominio R
[X,Y]=meshgrid(-1:0.1:1,-2:0.1:2);
Z=X.^2+Y.^2;
% Se representa la superficie
surf(X,Y,Z)
```

Prueba a cambiar la última orden `surf` por `mesh`, `contour`, `contour3`.

`contour(Z,n)`

Representa el gráfico de contorno para la matriz Z usando n líneas. El segundo parámetro es opcional.

`contour3(Z,n)`

Representa el gráfico de contorno en tres dimensiones para la matriz Z usando n líneas. El segundo parámetro es opcional.

```
>>%Para evaluar la función f(x,y)=x^2+y^2 en el
>>%dominio -2<x<2, -3<y<3
>>[X, Y]=meshgrid(-2:.2:2,-3:0.2:3);
>>Z=X.^2.+Y.^2;
>>figure(1)
>>contour(Z)
>>figure(2)
>>contour3(Z)
```

Ejercicio. Representa en una matriz de gráficos 1x2 la superficie en el lado izquierdo y diez curvas de nivel de $f(x,y)=\sqrt{64-4x^2+y^2}$ cuando $x\in[-2,2]$, $y\in[-3,3]$ en la parte derecha.

Solución

```
%Para representar unas líneas de contorno utilizaremos el
%comando contour(X,Y,f(X,Y),n), siendo n las líneas de contorno
%equiespaciados. Se puede sustituir el argumento n por un vector
%con las coordenadas que nos interesen
x=-2:.5:2;
y=-3:.5:3;
[X,Y]=meshgrid(x,y);
f=inline('sqrt(64-4*x.^2-y.^2)','x','y');
Z=f(X,Y);
subplot(1,2,1); surf(X,Y,Z);
subplot(1,2,2)
contour(X,Y,Z,10)
```

Ejercicio. Dada la superficie S de ecuación $z = x^2 + 3xy + y^2$ y el punto $P(1,1,5)$. Se pide:

- Calcula la pendiente de la recta r que es paralela al plano ZX y tangente a la superficie S en el punto P . Obtén también la ecuación de dicha recta. Representa con Matlab la superficie en las proximidades del punto P junto con la recta r .
- Calcula la pendiente de la recta r que es paralela al plano ZY y tangente a la superficie S en el punto P .
- Obtén también la ecuación de dicha recta. Representa con Matlab la superficie en las proximidades del punto P junto con la recta r .

Superficie

```
[X,Y]=meshgrid(-2:.25:2);
Z=X.^2+3*X.*Y+Y.^2;
h1=surf(X,Y,Z);
%Dibujamos la superficie en color magenta, con transparencia %0.5,
y color de la retícula en blanco
set(h1,'FaceColor','magenta','FaceAlpha',0.5,'EdgeColor','w')

%Etiquetamos los ejes
xlabel('Eje X')
ylabel('Eje Y')
zlabel('Eje Z')
title('Gráfica de f(x,y) = 9 - x^2 - y^2')
view(150,20)
```

Puntos y C Recta en el plano z=0

```
hold on
%Dibujamos el punto (1,1,0) y su imagen
plot3(1,1,0,'bo')
plot3(1,1,5,'bo')
%Utilizamos coordenadas paramétricas para representar en el %plano
z=0, la recta y=1.
t=linspace(-3,1)
X1=1+t; Y1=1+0*t; Z1=0*t;
%La dibujamos con grosor 2 y color azul
line(X1,Y1,Z1,'linewidth',2,'color','blue')
```

Imagen de la curva C

```
%Dibujamos la curva C imagen de la recta anterior
Z1= X1.^2+3*X1.*Y1+Y1.^2;
line(X1,Y1,Z1,'linewidth',2,'color','blue')
```

Recta tangente

```
%Dibujamos la recta tangente a C en el punto (1,1,f(1,1))
lambda=linspace(-1,1);
```

```
X3=1+lambda;
Y3=1+lambda*0;
Z3=5+5*lambda;
line(X3,Y3,Z3,'LineWidth',2,'color','black')
```

pcolor(X,Y,Z)

Representa el gráfico de contorno para la matriz (X,Y,Z) utilizando densidades de colores.

```
>>%Para evaluar la función f(x,y)=x^2+y^2 en el
>>%dominio -2<x<2, -3<y<3
>>[X, Y]=meshgrid(-2:.2:2,-3:0.2:3);
>>Z=X.^2.+Y.^2;
>>pcolor(X,Y,Z)
```

mesh(X,Y,Z,C)

Representa el gráfico de malla de la función $z=f(x,y)$ con los colores especificados en C (este último parámetro se puede ignorar).

meshc(X,Y,Z,C)

Representa el gráfico de malla de la función $z=f(x,y)$ junto con el gráfico de contorno correspondiente (curvas de nivel)

meshz(X,Y,Z,C)

Representa el gráfico de malla de la función $z=f(x,y)$ junto con una especie de cortina en la parte inferior.

```
>>%Para evaluar la función f(x,y)=x^2*y en el
>>%dominio -2<x<2 -3<y<3
>>[X, Y]=meshgrid(-2:.2:2,-3:0.5:3);
>>Z=X.^2.*Y;
>>figure(1)
>>mesh(X,Y,Z)
>>figure(2)
>>meshc(X,Y,Z)
>>figure(3)
>>meshz(X,Y,Z)
```