

# INTRODUCCIÓN A MATLAB

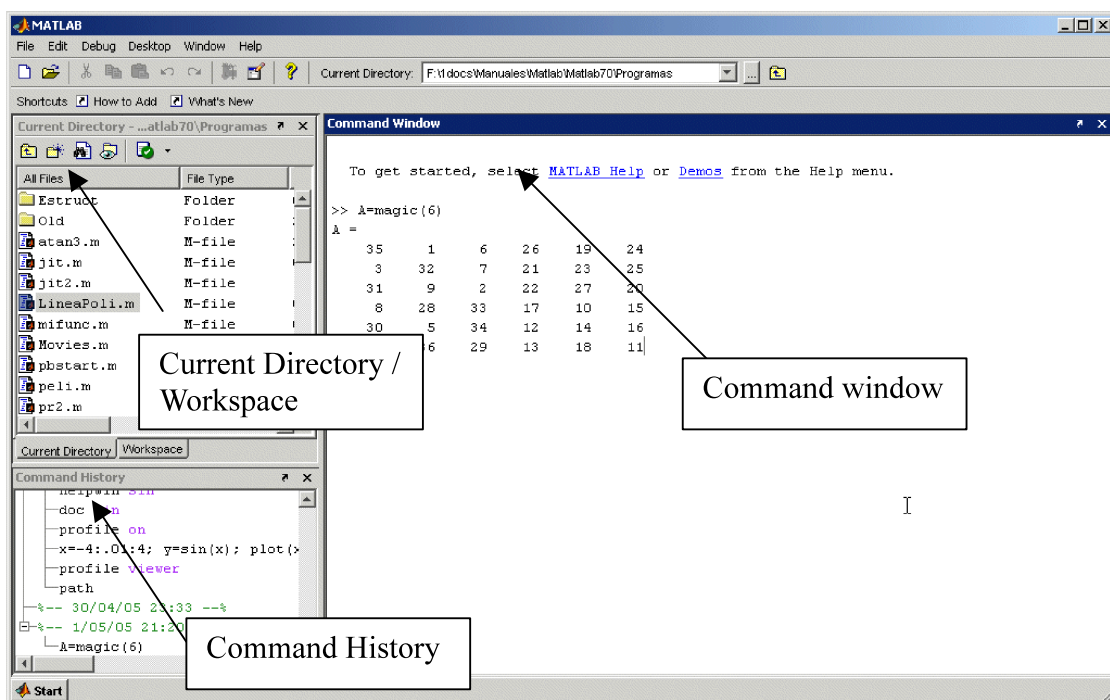
## 1. INTRODUCCIÓN

Matlab es un programa que permite realizar cálculos de una forma rápida y fiable. A diferencia de programas como Maple o Mathematica, Matlab no es un manipulador algebraico y sólo puede operar con variables reales y complejas.

Tal y como su nombre indica, Matlab (MATrix LABoratory) es un programa pensado para trabajar con matrices de manera que permite implementar cómoda y rápidamente cualquier algoritmo que las utilice.

### Cómo entrar y moverse por el programa

Una vez se entra en Matlab aparece una pantalla como la que se puede ver en la siguiente figura. Las instrucciones se escriben detrás del símbolo `>>`.



Se pueden distinguir tres pantallas, que engloban los siguientes cuatro entornos de trabajo:

- **Command Window:** es la parte donde se ejecutan las instrucciones. El símbolo `>>` indica que el programa está preparado para recibir instrucciones.
- **Current Directory:** es la ventana en que se muestran los archivos del directorio activo
- **Workspace:** muestra las variables que se han definido y su valor.
- **Command History:** muestra las últimas instrucciones ejecutadas en la Command Window.

Al empezar a trabajar con Matlab, es importante modificar el directorio de trabajo Current Directory, de manera que en éste estén todos los ficheros que se van a utilizar.

Es útil saber que en el entorno Current Directory se puede utilizar el botón derecho del ratón para crear carpetas. Además, se pueden borrar o crear ficheros como en cualquier carpeta del ordenador.

Como en la mayoría de aplicaciones de Windows, en la parte superior de la ventana de Matlab aparecen diferentes pestañas, entre las cuales se pueden destacar:

- Desktop: da acceso a diferentes módulos de Matlab que pueden estar instalados. También se puede acceder clicando el botón Start que se encuentra en el extremo inferior izquierdo de la ventana.
- Help: Matlab dispone de una ayuda muy potente y útil. En la pestaña Help se pueden encontrar, entre otros, las siguientes opciones
  - Matlab Help: en el apartado de búsqueda (Search) se puede encontrar como implementar cualquier instrucción.
  - Demos: da acceso a algunos ejemplos resueltos con Matlab.
  - About Matlab: Proporciona la información del programa que se tiene instalado.

Para salir de Matlab, se puede utilizar la opción Exit Matlab del menú File o bien escribir directamente en la línea de comandos la instrucción quit. El acceso rápido Ctrl+Q también permite salir del programa.

Finalmente, para detener para la ejecución de un cálculo, nos tenemos que situar sobre la Command Window y apretar las teclas Ctrl+C. En la mayoría de casos, esta opción hace que se interrumpa el proceso actual y aparezca de nuevo el prompt (símbolo >>).

## 2. OPERACIONES BÁSICAS

Aunque Matlab está pensado para trabajar con matrices vamos a ver primero qué permite hacer con escalares.

Podemos utilizar el programa como una calculadora que permite hacer todas las operaciones básicas que conocemos.

```
» 2 -3.75 +1.2
```

```
ans =
```

```
-0.5500
```

```
» 2/3
```

```
ans =
```

```
0.6667
```

```
» 5.7^3
```

```
ans =
```

```
185.1930
```

Si en una línea se escriben varias operaciones, éstas se ejecutan teniendo en cuenta las siguientes reglas de prioridades:

- En primer lugar se ejecutan las operaciones entre paréntesis
- A continuación se calculan las potencias
- Seguidamente se realizan los productos y divisiones, de izquierda a derecha
- Finalmente se calcula las sumas y las restas, de izquierda a derecha.

```
» 2 + 3/4*5
```

```
ans =
```

```
5.7500
```

```
» 3*4-5^2*2-3
```

```
ans =
```

-41

»  $(2/3^2*5)*(3-4^3)^2$

ans =

4.1344e+03

**Observación.** Se puede cambiar el formato de escritura de los números mediante la opción Preferences del menú File.

### 3. ASIGNACIÓN DE VARIABLES

Muchas veces interesa guardar el resultado de una operación para poder utilizarlo posteriormente, cosa que haremos asignando este valor a una variable. La instrucción básica en Matlab es

variable = expresión;

El ; es opcional y sirve únicamente para suprimir la respuesta. Es decir, si una instrucción no se acaba con ; se escribe el resultado de la asignación por pantalla.

» x = 2.24

» y = 3.56

» z = x + y^2;

Si se escribe directamente una expresión, el resultado se almacena en una variable denominada ans (del inglés, answer).

Como nombre de las diferentes variables se puede utilizar cualquier secuencia de caracteres alfanuméricos siempre que empiece por una letra. Matlab utiliza únicamente los primeros 31 caracteres del nombre de la variable y distingue mayúsculas y minúsculas. Hay algunos nombres especiales que conviene no usar porque corresponden a valores ya definidos en el programa (eps, pi, realmax, realmin...)

En la ventana Workspace puede verse un listado de las variables definidas, consultar su valor y modificarlas o borrarlas.

Además, existen algunas instrucciones que permiten realizar estos procesos desde la línea de comandos. La instrucción who proporciona un listado de todas las variables que estamos utilizando. Podemos saber también cuánto ocupa cada una de ellas mediante el comando whos.

» who

Para borrar una variable utilizaremos la instrucción clear seguida del nombre de la variable. Es importante remarcar que si escribimos únicamente clear se borrarán todas las variables de la memoria.

» clear x

» whos

**Observación.** Aunque en este apartado sólo se han definido variables escalares, se procede de la misma manera para definir vectores y matrices.

### 4. FUNCIONES

Matlab dispone de una librería con diversas funciones que el usuario puede utilizar sin necesidad de programarlas. Aunque en la ayuda del programa pueden consultarse todas las funciones disponibles, presentamos aquí algunas de las más

frecuentemente utilizadas en matemáticas:

Abs	Valor absoluto de un número real o módulo de un complejo
acos, asin, atan	Inversa del seno, del coseno y de la tangente
Ceil	Redondea al siguiente entero
Cos, sin, tan	Coseno, seno y tangente de un ángulo expresado en radianes
Exp	Exponencial
Floor	Redondea al entero anterior
Gcd	Máximo común divisor
Lcm	Mínimo común múltiplo
log, log2, log10	Logaritmo natural, en base 2 y en base 10.
Rem	Resto de una división
sqrt	Raíz cuadrada

## 5. VECTORES

En este apartado vamos a ver cómo se definen vectores en Matlab y qué operaciones pueden efectuarse con ellos.

### Vectores fila

Para definir vectores filas en Matlab se escriben sus componentes entre corchetes y separadas por comas o espacios en blanco.

```
» x1 = [1 -5 7 4]
```

```
x1 =
```

```
1 -5 7 4
```

```
» x2 = [1, 0, -3, 5]
```

```
x2 =
```

```
1 0 -3 5
```

También se pueden definir vectores utilizando la expresión inicio:h:fin. Con ella se crea un vector con valores desde inicio hasta fin con paso h. El paso es opcional, si no se especifica toma por defecto el valor 1.

```
» x3 = [-2 : 2]
```

```
x3 =
```

```
-2 -1 0 1 2
```

```
» x4 = [1 : 0.25 : 2]
```

```
x4 =
```

```
1 0.25 0.5 0.75 1 1.25 1.5 1.75 2
```

```
» x5 = [4 : -1 : 1]
```

```
x5 =
```

```
4 3 2 1
```

### Vectores columna

Los vectores columna se definen en Matlab escribiendo sus componentes separadas por ; o saltos de línea.

```

» y1 = [ 1
        2
        3
        4];
» y2 = [1; 0; -1; 3]
y2 =
     1
     0
    -1
     3

```

También pueden definirse vectores columna transponiendo vectores fila.

```

» y3 = x1'
y3 =
     1
    -5
     7
     4

```

### Operaciones con vectores

Al hablar de operaciones con vectores es importante notar que, además de las operaciones algebraicas, Matlab permite efectuar operaciones componente a componente. Para ello sólo hace falta añadir un `.` delante del símbolo de la correspondiente operación.

- Operaciones con escalares  
Podemos multiplicar o dividir un vector por un escalar.

```

» x5 * 2
ans =
     8     6     4     2

```

Si sumamos o restamos un escalar a un vector la operación se efectúa sobre todas las componentes del vector.

Nótese que esta operación no está definida desde un punto de vista algebraico.

```

» y1 - 3
ans =
    -2
    -1
     0
     1

```

- Suma de vectores  
Para poder sumar o restar dos vectores es necesario que sus dimensiones coincidan; es decir, que sean dos vectores fila o columna y que tengan el mismo

número de componentes.

```
» x1 + x2
```

```
ans =
```

```
2 -5 4 9
```

```
» x1 + x3
```

```
??? Error using ==> +
```

```
Matrix dimensions must agree.
```

```
» x1 + y1
```

```
??? Error using ==> +
```

```
Matrix dimensions must agree.
```

- Producto de vectores

El resultado de multiplicar un vector fila por un vector columna es un escalar

```
» x1 * y1
```

```
ans =
```

```
28
```

El resultado de multiplicar un vector columna por un vector fila es una matriz

```
» y1 * x1
```

```
ans =
```

```
1 -5 7 4
2 -10 14 8
3 -15 21 12
4 -20 28 16
```

No se puede multiplicar directamente dos vectores fila o dos vectores columna, porque esta operación no está definida algebraicamente. En cambio, sí se pueden multiplicar dos vectores de la misma dimensión componente a componente:

```
» x1 * x2
```

```
??? Error using ==> *
```

```
Inner matrix dimensions must agree.
```

```
» x1 .* x2
```

```
ans =
```

```
1 0 -21 20
```

Es importante notar que esta operación, al igual que pasaba con la suma de un vector y un escalar, no está definida desde el punto de vista algebraico.

- División componente a componente

Finalmente podemos dividir componente a componente dos vectores de las mismas dimensiones.

**Atención:** observar las diferencias entre el operador / y el operador \

```
» x2 ./ y1'
```

```
ans =
```

```
1.0000 0 -1.0000 1.2500
```

```
» x2 ./ y1'
```

```
Warning: Divide by zero.
```

```
ans =
```

```
1.0000    Inf -1.0000    0.8000
```

```
» x2 ./ y2'
```

```
Warning: Divide by zero.
```

```
ans =
```

```
1.0000    NaN    3.0000    1.6667
```

**Observación.** Si dividimos un número por cero nos da como resultado un valor infinito. En cambio, al dividir 0 entre 0 el resultado es NaN (Not-a-Number) que indica que es una indeterminación.

- Exponenciación

Para poder elevar las componentes de un vector a un determinado exponente es necesario especificar que se trata de una operación componente a componente.

```
» x3 .^ 2
```

```
ans =
```

```
4    1    0    1    4
```

También se puede elevar cada componente de un vector al exponente indicado por los coeficientes de otro vector.

```
» x1 .^ x2
```

```
ans =
```

```
1.000    1.000    2.915451895043732e-03    1024.000
```

- Funciones

Si aplicamos una de las funciones que tiene definidas el programa a un vector, automáticamente ésta se aplica sobre todas las componentes del mismo.

```
» exp(x3)
```

```
ans =
```

```
0.1353    0.3679    1.0000    2.7183    7.3891
```

- Representación gráfica

Para representar una función en Matlab se definen dos vectores: uno con las coordenadas de los puntos sobre el eje de abscisas y otro con las correspondientes imágenes. Una vez definidos los vectores se puede dibujar la función utilizando la instrucción plot.

Por ejemplo, para representar la función  $f(x) = \sin(x) - 2 * \cos(x)$ . En primer lugar, definimos el vector de las x's

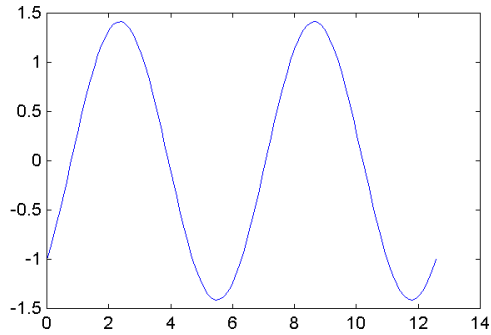
```
» x = linspace(0, 4*pi);
```

A continuación definimos el vector de las imágenes

```
» y = sin(x) -cos(x);
```

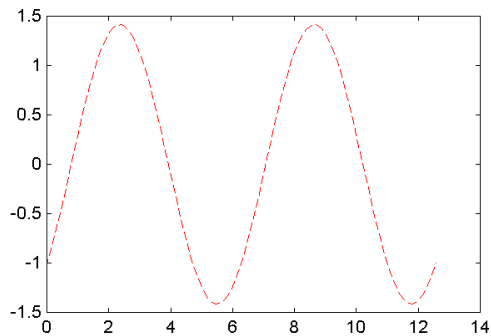
y dibujamos la gráfica utilizando la instrucción plot

```
» plot(x,y)
```



Podemos cambiar el color de la gráfica y el estilo de línea

» `plot(x,y,'r--')`

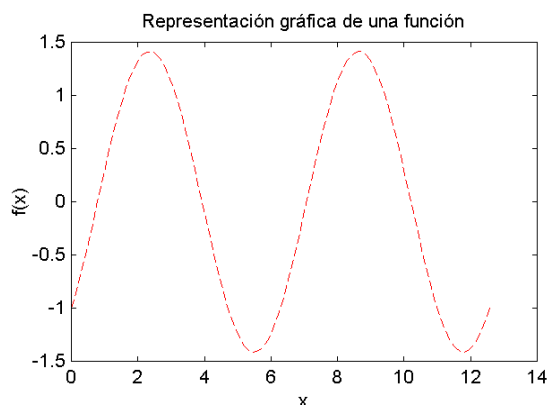


Y añadir títulos o etiquetas a los ejes

» `title('Representación gráfica de una función')`

» `xlabel('x')`

» `ylabel('f(x)')`



## 6. MATRICES

### Definición de matrices

Las matrices pueden definirse escribiendo sus coeficientes correctamente situados empleando como separador de filas un ; o un salto de línea y como separador de columnas una , o un espacio en blanco.

» `A = [1 2; 3 4]`

A =

1 2

3 4



```
» B = [2 -1;1 0]
```

```
B =
```

```
2 -1
```

```
1 0
```

```
» C = [1 1 1
```

```
2 2 2]
```

```
C =
```

```
1 1 1
```

```
2 2 2
```

También pueden definirse matrices a partir de vectores y matrices ya existentes.

```
» D = [x1; x2]
```

```
D =
```

```
1 -5 7 4
```

```
1 0 -3 5
```

```
» E = [A C; C' zeros(3)]
```

```
E =
```

```
1 2 1 1 1
```

```
3 4 2 2 2
```

```
1 2 0 0 0
```

```
1 2 0 0 0
```

```
1 2 0 0 0
```

### Matrices especiales

Matlab permite generar directamente algunas matrices especiales. Así,

- ones(m,n) genera una matriz m x n con todos sus coeficientes iguales a 1.
- zeros(m,n) genera una matriz de m filas y n columnas con todos los coeficientes nulos.
- eye(m,n) genera una matriz m x n con unos en la diagonal y ceros en las demás posiciones.

Si a cualquiera de estas funciones se le pasa un solo argumento n, se genera una matriz cuadrada de dimensión n con las características descritas.

```
» F = eye(3)
```

```
F =
```

```
1 0 0
```

```
0 1 0
```

```
0 0 1
```

### Operaciones con matrices

- Operaciones con escalares

» C + 1

ans =

2 2 2

3 3 3

» B \* 3

ans =

6 -3

3 0

- Producto de matrices

» A \* B

ans =

4 -1

10 -3

- Exponenciación

Podemos multiplicar una matriz cuadrada por ella misma tantas veces como queramos elevándola a un exponente.

» A ^ 2

ans =

7 10

15 22

- División de matrices por la derecha ( $A/B = A*B^{-1}$ )

A/B

ans =

-2 5

-4 11

- División de matrices por la izquierda ( $A\B = A^{-1}*B$ )

A\B

ans =

-3.0000 2.0000

2.5000 -1.5000

### **Observaciones**

- Para matrices también están definidas las operaciones componente a componente. Para realizarlas sólo hace falta añadir un . delante del correspondiente símbolo.

» A . ^ 2

ans =

1 4

9 16

```

» A.* B
ans =
     2  -2
     3   0

```

- Como en el caso de los vectores para poder realizar cualquier operación es necesario que las matrices tengan las dimensiones adecuadas.
- Ya hemos comentado que Matlab trata a todos los elementos como matrices, de manera que igual que se han definido las operaciones entre matrices se pueden realizar operaciones entre matrices y vectores, siempre que tengan las dimensiones apropiadas.

```

» D * y1
ans =
    28
    12
» x3 * E
ans =
   -2  -2  -4  -4  -4

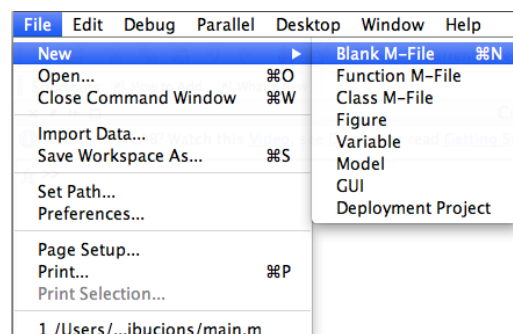
```

**Observación.** Para resolver un sistema de ecuaciones  $Ax = b$  sólo hace falta escribir  $x=A\b{b}=A^{-1}\cdot b$  y Matlab escoge el método más adecuado para resolverlo.

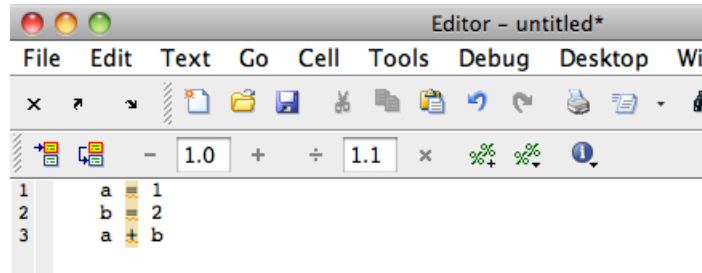
## 7. ARCHIVOS .M

En lugar de trabajar en la Command Window, se pueden crear archivos con las instrucciones que se quieren ejecutar, y después llamarlos desde el programa. Se trata de archivos de tipo.m. Para trabajar con estos archivos, se abre el editor de Matlab y desde aquí se van modificando.

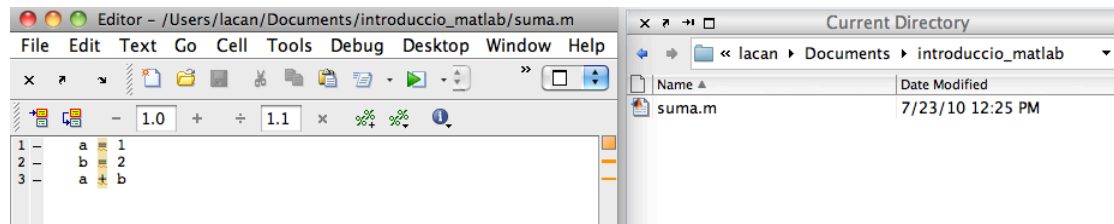
Veamos como hacer un programa que define dos números y calcula su suma. Para abrir el editor de Matlab, se va al menú File y se clica sobre New y posteriormente sobre Blank M-File



Esto abre una ventana del editor de Matab en la que se pueden escribir las operaciones, del mismo modo que en la Command Window.



Al guardar el fichero, es importante recordar que se tiene que guardar en el directorio de trabajo. Si guardamos el archivo generado como suma.m, veremos el nombre en la parte superior del editor. Además, el fichero también tiene que verse en la ventana Current Directory.



Hay diferentes maneras de ejecutar el archivo. La primera es escribir el nombre en la línea de comandos de Matlab. En este caso, es importante asegurarse primero de que el archivo está en el directorio de trabajo, bien mirando el entorno Current Directory o bien utilizando la instrucción dir para listar los ficheros de la carpeta de trabajo.

La segunda manera de ejecutar el archivo es situarse sobre el nombre del archivo en la ventana Current Directory, clicar con el botón derecho y seleccionar Run File.

Finalmente, una opción muy útil es seleccionar en el editor de Matlab las líneas que se quieren ejecutar y apretar F9. Se puede hacer lo mismo si, una vez seleccionadas las líneas del editor, se clicca el botón derecho y se marca la opción Evaluate Selection.

## Funciones

Se pueden definir funciones utilizando un archivo .m. Es importante recordar que el nombre del archivo tiene que ser el mismo que el de la función. Así, en el archivo nombre.m definiremos una función de la siguiente manera:

```
function resultado = nombre (datos)
%
% comentarios que se consideran la ayuda de la función
% y que aparecerán al escribir en la línea de comandos
% » help nombre
%
%
:
resultado = ...
```

## 8.SENTENCIAS DE CONTROL DE FLUJO

### Sentencia condicional IF

La forma general de escribir esta sentencia es

```
if condición 1
    bloque de instrucciones 1
elseif condición 2
```

```

        bloque de instrucciones 2
    :
else
    bloque de instrucciones final
end

```

Esta instrucción funciona de la siguiente manera: se van evaluando las distintas condiciones hasta que se encuentra una verdadera; entonces se ejecuta el correspondiente bloque de instrucciones y se sale del bucle..

Para escribir la condición se pueden utilizar los siguientes operadores relacionales:

>	mayor que	>=	mayor o igual que
<	menor que	<=	menor o igual que
= =	igual a	~=	diferente de

que se pueden combinar con diferentes operadores lógicos:

&	y
	o
~	negación

### **Bucle FOR**

La forma de esta sentencia es

```

for variable = inicio : paso : fin
    instrucciones
end

```

Se van ejecutando las instrucciones del interior del bucle mientras la variable contador toma valores comprendidos entre inicio y fin con paso h. Si no se especifica el paso, éste toma el valor 1.

### **Bucle WHILE**

La forma en que se escribe esta sentencia es

```

while condición
    instrucciones
end

```

En este caso se ejecutan las instrucciones del interior del bucle mientras la condición sea verdadera.