

PRÁCTICA 0: INTRODUCCIÓN A MATLAB

El acceso a Matlab se realiza presionando o “pinchando” dos veces en el botón izquierdo del ratón en el icono Matlab. Al terminar cada sesión se debe comprobar que se graba el trabajo realizado en vuestro dispositivo personal de almacenamiento (disco extraíble).

Para garantizar que se va a trabajar en la unidad o directorio elegido se debe escribir en la línea de comandos:

```
>> cd [unidad o directorio]:
```

Con ello garantizamos que desde el comienzo de la sesión de prácticas trabajamos con nuestro disco de trabajo.

Una vez que se ha realizada una práctica, cada alumno de forma individual debe entregar un informe de resultados por escrito al profesor para que sean evaluadas, indicando el nombre y fecha.

1.- ÓRDENES EN MATLAB

1. En la siguiente tabla se indican algunas órdenes útiles en MATLAB. En general, se distingue entre mayúsculas y minúsculas, aunque muchos comandos estén definidos tanto en mayúsculas como en minúsculas.

help	<ul style="list-style-type: none">• Ayuda sobre órdenes y funciones de Matlab.• Ayuda sobre nuestras propias funciones .m
ver	<ul style="list-style-type: none">• Muestra la versión de Matlab que se esta utilizando
what	<ul style="list-style-type: none">• Da una lista de funciones .m en el directorio especificado. <pre>>> what a:\ejemplos</pre>
pwd	<ul style="list-style-type: none">• Indica el directorio en el que estoy trabajando <pre>>> pwd</pre>
cd	<ul style="list-style-type: none">• Cambia al directorio que se especifique. <pre>>> cd .. >> cd a:\ejemplos</pre>
which	<ul style="list-style-type: none">• Localiza funciones. <pre>>> which hilb</pre>
dir	<ul style="list-style-type: none">• Da una lista del contenido del directorio en el que me encuentro

2. Haciendo uso de los comandos **help** y **helpwin** ejecuta y anota el significado de lo que se te indica al ejecutar las siguientes sentencias:

```
>>helpwin help  
>>helpwin cd  
>>helpwin dir
```

3. *Más ayuda:* Haciendo uso del interfaz gráfico del que dispone MATLAB (parte superior sombreado en gris), presiona o pincha una vez en el botón izquierdo en Help e indica qué alternativas dispone.

4. Utilizando las ventanas de MATLAB (parte superior sombreada en gris), indica el contenido de las ventanas: File, Edit, View, Windows.

5. En MATLAB existen diferentes modos o formatos de visualizar un resultado numérico. Dicho formato de salida se selecciona con el comando **format**. Usando la ayuda puedes conocer qué opciones de formato de salida existen.

Como has comprobado existen varios modos de seleccionar el formato de salida:

1. Tecleando explícitamente el tipo de formato en la línea de comandos. Por ejemplo, ejecuta estas dos sentencias:

```
>>format long
>>x=1.2
```

2. Escribe el resultado de x con el formato *long* y con el formato *short*.

3. Escribe el resultado que se obtiene tras hacer:

```
>>format long e
>>x
```

¿Cuál es la diferencia con el formato *long*?

2.- OPERACIONES CON ESCALARES

Operaciones	Ejemplos
Suma +	+ >>a=4 >>b=2 >>s=a+b
Resta -	- >>r=4-2
Multiplicación *	>>m=a*b
División /	/ >>56/8
División \	\ >>8\56
Potencia ^	>>a^2

Ejecutar los ejemplos que se indican en la tabla indicando los resultados en cada caso.

1. Ejecuta:

```
>>clear all
>> a = 4
>> a
>> clc
>> a
>> clear a
>> a
```

- (a) ¿Cuál es el valor de a después de ejecutar cada comando?
(b) ¿Qué acción realizan los comandos **clear** y **clc**?

2. Busca el significado de las siguientes funciones intrínsecas de Matlab. Realiza una tabla en donde en la primera columna se indique la función, en la segunda se indique el significado, en la tercera se indique un ejemplo y en una cuarta columna el resultado de dicho ejemplo.

abs	sin	cos	tan	asin
sinh	exp	log	log10	log2
rem	round	sign	sqrt	pi

3.- GRÁFICAS

1. Busca el significado de las órdenes **linspace** y **plot**. Ejecuta las siguientes instrucciones, indicando el resultado de cada una de ellas:

```
>>x=linspace(0,2*pi,30)
>>y=sin(x)
>>disp('Voy a dibujar funciones sinusoidales')
>>plot(x,y)
>>plot(x,y,x,cos(x))
>>plot(x,y,x,y,'*',x,cos(x),x,cos(x),'+')
>>grid
>>xlabel('Variable independiente')
>>ylabel('Variables dependientes')
>>title('Primer ejemplo de gráficas')
```

2. Con la siguiente instrucción se puede generar una figura en tres dimensiones:

```
>>plot3(x,y,cos(x))
```

3. ¿Qué pasaría si añadimos un punto y coma (;) al final de las dos primeras instrucciones indicadas en el apartado (1)?.

4.- CREACIÓN DE UN FICHERO .m

1. Creación de un fichero **.m** (o cualquier otro fichero en el sistema MATLAB).

Para crear un fichero **.m** se debe hacer lo siguiente:

- (1) Pinchar **File** en el menú de ventanas de MATLAB.
- (2) Seleccionar la opción **New** y **M-file** (con ello entraremos en el editor de MATLAB).

- (a) Edita el siguiente fichero:

```
function grafica
% Creación de una figura conteniendo las funciones seno y coseno.
x=linspace(0,2*pi,30);
y=sin(x);
disp('Voy a dibujar funciones sinusoidales')
plot(x,y,x,y,'*',x,cos(x),x,cos(x),'+')
grid
xlabel('Variable independiente') %Crear la etiqueta del eje X
ylabel('Variables dependientes')
title('Primer ejemplo de gráficas')
```

Vete a la ventana **File** y selecciona **Save as**. Graba el nombre de este fichero en tu directorio con el nombre **grafica.m**

IMPORTANTE: el nombre del fichero debe ser el mismo que aparece en la línea **function**. La extensión **.m** es obligatoria.

- (b) Para saber lo que hace este fichero o programa ejecuta:

```
>> help grafica
```

¿Qué hace el programa **grafica.m** que has editado?. ¿cuál es el significado del **%** en el editor de Matlab?. Si desea que aparezca en pantalla el contenido total del fichero utiliza el comando **type**.

(c) Para ejecutar este programa haz:

```
>> grafica
```

Escribe el resultado o resultados de la ejecución.

2. Ejercicio

(a) Partiendo del programa **grafica.m**, realiza las siguientes modificaciones:

- Sustituye la primera sentencia, **function grafica**, por **function grafica2(x,y)**
 - Añade el símbolo % a la 1ª columna de las sentencias número 3 y número 4.
- Graba este nuevo programa con el nombre de **grafica2.m** (Recuerda que debes hacerlo con Save as).

(b) Ejecuta ahora el siguiente conjunto de instrucciones en la línea de comandos de Matlab:

```
>>x=linspace(0,2*pi,30);  
>>y=sin(x);  
>>grafica2
```

¿Obtienes los mismos resultados que con el programa grafica.m?

(c) Explica el significado de la primera modificación, teniendo en cuenta las instrucciones del apartado (b).

(d) Explica el significado de la segunda modificación.

3. Ejercicio.

(a) Crea un programa llamado **medesv.m** que contenga las siguientes sentencias:

```
function [media,desviacion] = medesv(x)  
% Da el valor medio y la desviacion estandar del vector x.  
n = length(x);  
media = sum(x) / n;  
desviacion = sqrt(sum((x - media).^2)/n);
```

(b) Busca el significado de las órdenes que no entiendas.

(c) Ejecuta las sentencias:

```
>>x=[1 2 3 4 3 2 1]
>>[media,desviacion]=medesv(x)
```

Escribe los resultados.

(d) Vuelve a preguntar por el valor de las magnitudes anteriores, recuerda que se hace ejecutando:

```
>>media
>>desviación
```

Estas instrucciones nos indican que hemos almacenado los valores de la media y la desviación en las variables MEDIA y DESVIACION, lo cual nos permitirá poderlas usar más adelante.

5.- OPERACIONES E INSTRUCCIONES RESERVADAS EN MATLAB

A continuación tienes un resumen de algunas de las instrucciones básicas de Matlab .

OPERACIONES RELACIONALES Y LÓGICAS

Operaciones LÓGICAS

&	AND
	OR

Operaciones RELACIONALES

<	Menor que
<=	Menor o igual que
>	Mayor que
>=	Mayor o igual que
= =	Igual que
~ =	Distinto

BUCLES (Permiten que un conjunto de órdenes se repitan un número de veces predeterminado)

	Estructura	Ejemplo sencillo	Ejemplo de anidamiento
for	for x=matriz Ordenes end	for n=1:10 x(n)=sin(n*pi/10) end	for n=1:5 for m=5:-1:1 a(n,m)=n^2+m^2 end disp(n) end
while	while expresion Ordenes end	num=0;eps=1; while (1+eps)>1 eps=eps/2 num=num+1 end	

ESTRUCTURAS IF-ELSE-END

Estructura	Ejemplo sencillo	Ejemplo de anidamiento
if expresion Ordenes end	M=10; if M>5 C=(1-20/100); end	
if expresion Ordenes si expresion verdadera else Ordenes si la expresion es falsa end	M=10; if M>5 C=(1-20/100); else C=(1-10/100); end	function [C0]=ejemif(M,C) if M < 0 disp(' M No puede ser negativo ') break else if (M > 0) & (M <= 5) disp('M positivo y menor o igual que 5') C0=(1-10/100); else disp('M mayor que 5') C0=(1-20/100); end end

1. Crea tres programas **.m** que contengan los tres ejemplos que se indican dentro de la sección BUCLES. Estos programas se llamarán:

ejemfor1.m (contendrá el conjunto de sentencias del ejemplo sencillo de **for**).

ejemfor2.m (contendrá el conjunto de sentencias del ejemplo más complejo de **for**).

ejemwhile.m (contendrá el conjunto de sentencias del ejemplo de **while**).

¿Qué hace cada uno de estos programas? ¿Qué resultados obtienes?

2. Crea el programa **ejemif.m** que contenga el ejemplo de anidamiento de IF-ELSE-END. Explica los resultados que obtienes, indicando si son lógicos, para los siguientes valores de M:

- (a) -3.0
- (b) 0
- (c) 1.3
- (d) 1
- (e) 5

6.- Operaciones básicas de vectores y matrices:

1.-Funciones definidas sobre matrices:

Haciendo uso del comando **help** conoce el significado de los siguientes comandos:

zeros	ones	rand	eye	size
diag	triu	tril	lu	det
inv	rcond	vander	det	trace
rank	eig	norm(A,1)	norm(A,2)	norm(A,inf)
norm(A,'fro')	cond(A,1)	cond(A,2)	cond(A,inf)	

2.- Definiciones básicas:

```
>> A = [1 2 1;4 5 2;7 8 9]      >> v = [1 0 1]      >> v2 = 1:1:10
>> v3 = 10:-1:-2                >> v4 = 0:pi/10:pi    >> t = 1:5
>> t = [0,t]                     >> t2 = [t,t]      >> t3 = [t;t]
```

3.- Acceso a elementos de una matriz o vector:

```
>> A      >> A(2,2)      >> A(3,1)      >> A(1,1)*A(2,2)
>> v(1,3) >> A([1,2],3)  >> A([1,2,3],2) >> A([1:3],2)
>> A(:,1) >> A(2,:)      >> A(2,:)+A(3,:) >> C = [A(1,:); A(2,:)]
>> D = [A(:,1); A(:,3)]
```

4.- Tipos especiales de matrices:

```
>> M1 = eye(3)      >> M2 = zeros(2,4)      >> M3 = ones(4,2)
>> M4 = rand(4,3)   >> M5 = rand(6)         >> M6 = diag([1 2 3 4 5])
```

5.- Operaciones con matrices:

```
>> A      >> A*A      >> A.*A      >> v
>> v*v'   >> v'*v     >> A*v'      >> A\v
>> inv(A)*v'
```

Ejercicio 1.- El siguiente ejemplo muestra como asignar valores a una matriz:

```
>>A= [1 2 3; 4 5 6]
```

(a) ¿Cuál es la salida que nos proporciona MATLAB? ¿Cuál es el valor numérico del elemento $A(1,2)$, es decir del elemento que está en la primera fila y segunda columna?

(b) ¿Cuál es el resultado del siguiente comando? ¿Es correcto?

```
>>A(1,2)
```

(c) ¿Por qué tras ejecutar el siguiente comando se indica que la variable **a** no está definida?

```
>> a(1,2)
```

(d) Ejecuta las siguientes órdenes en la línea de comandos e indica el resultado

comentando el por qué:

```
>> zeros(3,2)
```

```
>> zeros(3,3)
```

```
>> y=zeros(3,3)
```

```
>> size(A)
```

```
>> A=[1 2 3;4 5 6;7 8 9]
```

¿Qué orden o comando de Matlab te indicaría el tamaño de esta matriz?

(e) Construye una matriz de ceros de las mismas dimensiones que A.

Construye una matriz del mismo tamaño que A, pero cuyos elementos sean todos igual a 1.

Construye una matriz del mismo tamaño que A, pero donde los elementos de la diagonal sean igual a 1 y el resto igual a 0.

Construye una matriz triangular superior, teniendo en cuenta A.

(f) Busca en Matlab una función interna que dé el valor máximo de cada uno de los elementos de las columnas de A.

Ejercicio 2.-

(a) Asigna los siguientes valores numéricos a los elementos de la matriz de tres dimensiones B. Primera fila: 7,4,6. Segunda fila: 3,5,8. Tercera fila: 8,9,10.

Para confirmar que no has cometido ningún error puedes ejecutar: `>> B`

(b) Indica los resultados de las siguientes operaciones, explicando el significado de la operación:

`>> A*B` `>> A.*B` `>> A+B` `>> A-B` `>> A/B`

`>> A\B` `>> A.^2` `>> inv(A)` `>> det(A)` `>> A'`

Ejercicio 3.- Teniendo en cuenta la matriz A definida en el ejercicio anterior, ejecuta:

`>> Z=diag(A)`

Pregunta, tras ejecutar esta última sentencia, por el valor de Z. ¿Z será una matriz, un vector o un escalar?

Ejercicio 4.- Teniendo en cuenta la matriz A de los dos ejercicios anteriores, ejecuta:

`>> C=(0:2:10)`

Esta orden construye un vector. ¿Cuáles son sus elementos? Explica qué hace esta orden ¿Qué orden debes especificar para saber el valor del elemento 3 de este vector?

Averigua o recuerda si existe una función interna que almacene el número PI y explica el resultado de la orden:

`>> C=(0:2:10)*pi`

Ejercicio 5.- Escribe los comandos de MATLAB necesarios para definir la siguiente matriz y el siguiente vector:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & -1 & 0 \\ 2 & 2 & 3 \end{bmatrix} \quad b = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$$

Ejercicio 6.- Resuelve el sistema $Ax = b$ con una única sentencia de MATLAB y sin usar $\text{inv}(A)$.

Ejercicio 7. Se considera el siguiente código MATLAB. ¿Qué es lo que calcula?

```
>> A=[1 2 3 1 ; 0 0 2 -3 ; 1 1 -4 1 ; 2 2 5 4];
```

```
>> b=[1 0 -1 2]';
```

```
>> [ L,U ]=lu(A);
```

```
>> y=L\b;
```

```
>> x=U\y;
```