

 UNIVERSIDAD DE OVIEDO DEPARTAMENTO DE MATEMÁTICAS	Asignatura	Análisis Numérico	Página 1 de 7
	Tema	MATLAB-Introducción: Interface, operaciones y representación	
	Práctica	1	
	Autor	César Menéndez Fernández	

1.- Interface, funciones y condicionales

En esta práctica aprenderemos a entrar en MATLAB y utilizarlo como una potente calculadora. También veremos cómo representar pares de puntos.

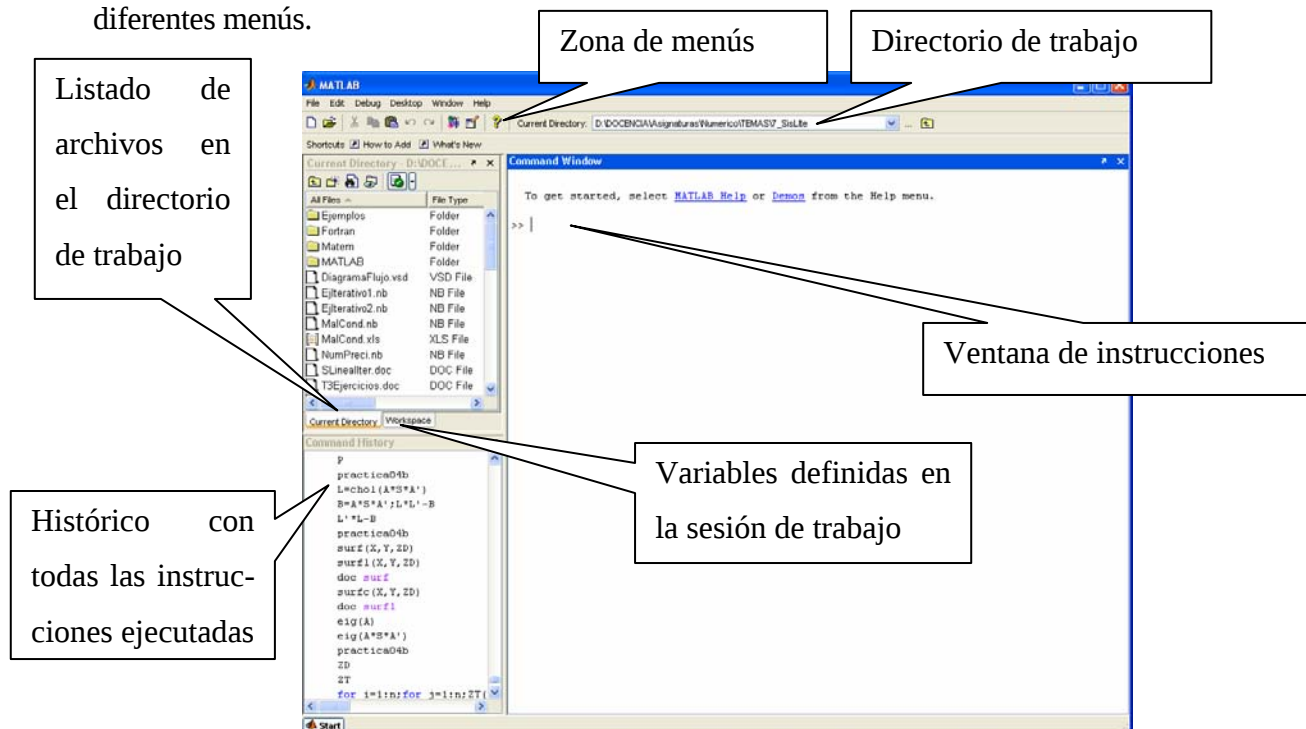
(a) Entrar en Matlab

Si aparece el icono de MATLAB como acceso directo en la pantalla inicial, es suficiente pulsar sobre el con el ratón. En otro caso, será necesario buscarlo a partir del menú de inicio.



Icono de MATLAB

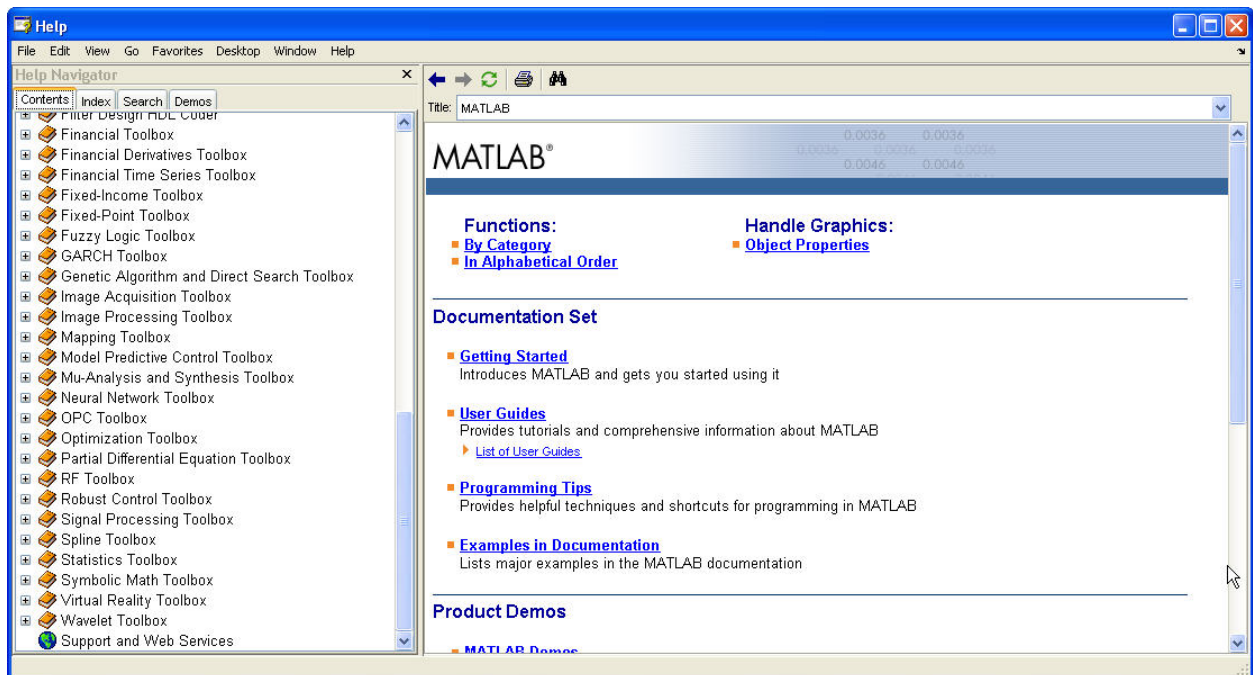
Una vez iniciado MATLAB, nos encontramos con la pantalla de la figura, donde se observan los diferentes menús.



Pantalla de entrada de MATLAB

(b) La ayuda en MATLAB

A la ayuda se accede mediante la opción “Full Product Family Help” o “Help Window” del menú *Help*. El mismo efecto se logra con la tecla “F1”.



Pantalla de ayuda (Help Window) de MATLAB

Se puede recorrer el índice seleccionando los diferentes tópicos o bien pedir información específica sobre una función determinada utilizando las opciones “index” o “search”. Cuando se conoce el nombre de la instrucción (p.e. `rand`) se pueden utilizar las instrucciones **help** o **doc**.

```
» help rand
» doc rand
```

(c) MATLAB como calculadora

Se puede usar MATLAB como si fuera una potente calculadora, almacenando variables y realizando operaciones entre ellas. MATLAB trabaja habitualmente con valores matriciales, de ahí que su definición y manejo sean fundamentales. Para definir una matriz en MATLAB, basta con introducir entre corchetes todas sus filas separadas por punto y coma. Los vectores se pueden introducir separando sus componentes por espacios en blanco o por comas. Su sintaxis es la siguiente:

`vector=[a, b, c, d, ...m]` Define un vector fila, cuyos elementos son los valores a, b, c, d, ...m.

`vector=[a; b; c; d; ...m]` Define un vector columna, cuyos elementos son los valores a, b, c, d, ...m.

En resumen, las comas separan elementos de un vector, mientras que el punto y coma separa las filas. MATLAB indica un error cuando las filas tienen diferente número de elementos. En el

apartado 2.-II. Selección de los elementos de un vector x o matriz se muestra la forma de manejar los elementos de este tipo de variables.

Veamos algunos ejemplos. Para generar la variable A que almacena una matriz de dos filas y tres columnas con los valores dados, introducimos

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

Para obtener la submatriz resultante de quitar la primera columna, añadir una matriz a sí misma, obtener sus todos sus elementos como un único vector o trasponerla, haremos respectivamente:

```
» A(:,2:3)
» [A A]
» A(:)
» A'
```

Existen en MATLAB dos tipos de operaciones aritméticas: Las operaciones aritméticas matriciales, que se rigen por las reglas del álgebra lineal, y las operaciones aritméticas con vectores, que se realizan elemento a elemento.

Símbolo	Operación	Símbolo	Operación
+	Suma de escalares, vectores o matrices	-	Resta de escalares, vectores o matrices
*	Producto matricial	.*	Producto elemental
^	Potenciación matricial	.^	Potenciación elemental $A.^B \rightarrow a_j^k \wedge b_j^k$
/	Cociente matricial, $B/A=B*\text{inv}(A)$	./	Cociente elemental $A./B \rightarrow a_j^k / b_j^k$
\	Cociente matricial, $A\backslash B=\text{inv}(A)*B$	.\	Cociente elemental $A.\backslash B \rightarrow b_j^k / a_j^k$

Hay que destacar la suma y multiplicación de un escalar por una matriz. En ambos casos se aplica la operación a cada elemento de la matriz:

```
»10*A, 10+A
ans =
10 20 30
40 50 60
ans =
11 12 13
14 15 16
```

Notemos que, como el resto de lenguajes de programación, la asignación (signo “=” en MATLAB) se puede interpretar como: “Realizar las operaciones a la derecha del símbolo = y almacenar su resultado en la variable indicada a la izquierda del =”. Esto permite realizar operaciones como:

```
» A=A*A'
» i=10, i=i+2
```

Para informarnos de las variables almacenadas que estamos usando pondremos *who* o *whos*.

```
» whos
Name      Size      Bytes  Class
A         2x2         32  double array
ans       2x2         32  double array
Grand total is 8 elements using 64 bytes
```

Al igual que la calculadora, MATLAB dispone de una librería muy completa de funciones predefinidas que se corresponden con las funciones matemáticas más utilizadas. Algunos

ejemplos se muestran en el anexo. Utilizándolas, se puede calcular el máximo de todos los elementos de A o la raíz cuadrada de los mismos, como

```
» max(A(:))  
» sqrt(A)  
» x=10;prod(1:x) % Calcula el factorial del numero 10  
» x=20;sum(1:x) % Calcula el sumatorio de numero 10
```

También dispone de instrucciones para representaciones gráficas en 2 y 3 dimensiones, de las cuales la más sencilla es **Plot** (ver anexo). Para representar una función en un intervalo, se generan dos vectores, uno para las abscisas y otro para las ordenadas, utilizando la orden *plot* para unir todos los pares de puntos. La representación de $y=e^x$ en $[-2,2]$ se realiza así mediante

```
» x=linspace(-2,2);y=exp(x);plot(x,y)
```

O bien

```
» x=linspace(-2,2); plot(x,exp(x))
```

Se pide:

1. Utilizando las instrucciones del anexo, representar la función $y=\sin(x)$ en el intervalo $[-\pi,\pi]$
2. Representar la función $y=|x|$ en el intervalo $[-2,2]$
3. Representar las funciones $\sin(x)$, $\cos(x)$ y $\tan(x)$ en el intervalo $[-\pi,\pi]$
4. Representar las funciones $\sinh(x)$, $\cosh(x)$ y $\tanh(x)$ en el intervalo $[-\pi,\pi]$
5. Crear los siguientes vectores
 - a. Vector ordenado de pares entre 6 y 26.
 - b. Vector ordenado de 10 múltiplos de tres comenzando en 21.
 - c. Vector en orden inverso desde 10 hasta 0.
6. Matriz de orden 4 con todos sus elementos 3 salvo la diagonal que es nula.
7. Matriz de orden 8 con la tabla de multiplicar desde el 1 hasta el 8 por columnas

2.- Anexo: Instrucciones necesarias

I. **Introducciones de manejo de vectores y matrices**

Repasamos aquí las instrucciones ya introducidas cursos anteriores.

`vector=[a, b, c, d, ...m]` Define un vector fila, cuyos elementos son los valores dados.

`vector=[a; b; c; d; ...m]` Idem con un vector columna.

`variable=[primer_elemento:último_elemento]` Define el vector cuyos primeros y último elementos son los especificados, y los elementos intermedios se diferencian en una unidad.

`variable=[primer_elemento:incremento:último_elemento]` Define el vector cuyos primeros y último elementos son los especificados, y los elementos intermedios se diferencian en la cantidad especificada por el incremento

`variable=linspace(primer_elemento,último_elemento,n)` Define el vector cuyos primeros y último elementos son los especificados, y que tiene en total n elementos uniformemente espaciados.

`variable=logspace(primer_elemento,último_elemento,n)` Define el vector cuyos primeros y último elementos son los especificados, y que tiene en total n elementos en escala logarítmica uniformemente espaciados entre sí.

Para definir una matriz en Matlab, basta con introducir entre corchetes todos sus vectores fila separados por punto y coma. Los vectores se pueden introducir separando sus componentes por espacios en blanco o por comas.

II. **Selección de los elementos de un vector x o matriz A**

<code>x(n)</code>	Devuelve el n-ésimo elemento del vector x
<code>x([n,m,p])</code>	Devuelve los elementos del vector x situados en las posiciones n-ésima, m-ésima y p-ésima.
<code>x(n:m)</code>	Devuelve los elementos del vector x situados entre el n-ésimo y el m-ésimo, ambos inclusive
<code>x(n:p:m)</code>	Devuelve los elementos del vector x situados entre el n-ésimo y el m-ésimo, ambos inclusive pero separados de p en p unidades
<code>A(m,n)</code>	Devuelve el elemento (m,n) de la matriz A (fila m y columna n)
<code>A([m, n],[p, q])</code>	Devuelve la submatriz de A formada por la intersección de las filas n-ésima y m-ésima y las columnas p-ésima y q-ésima.
<code>A(n,:)</code>	Devuelve la fila n-ésima de la matriz A
<code>A(:,p)</code>	Devuelve la columna p-ésima de la matriz A
<code>A(:)</code>	Devuelve un vector columna cuyos elementos son las columnas de A situadas por orden
<code>A(:,:)</code>	Devuelve toda la matriz A
<code>[A,B,C]</code>	Devuelve la matriz formada por las submatrices A,B,C,...

III. Funciones trigonométricas e hiperbólicas

Trigonométrica	sin(Z)	cos(Z)	tan(Z)	sec(Z)	csc(Z)	cot(Z)
Trig. Inversa	asin(Z)	acos(Z)	atan(Z), atan2(Z)	asec(Z)	acsc(Z)	acot(Z)
Hiperbólica	sinh(Z)	cosh(Z)	tanh(Z)	sech(Z)	csch(Z)	coth(Z)
Hip. Inversa	asinh(Z)	acosh(Z)	atanh(Z)	asech(Z)	acsch(Z)	acoth(Z)

IV. Funciones exponenciales

exp(Z)	Función exponencial de base e	log(Z)	Función Logaritmo neperiano
sqr(Z)	Función Raíz cuadrada	log10(Z)	Función Logaritmo decimal

V. Funciones específicas de variable numérica

abs(Z)	Módulo o valor absoluto	rem(a,b)	Da el resto de la división entre los reales a y b
angle(Z)	Argumento	fix(x)	Elimina la parte decimal del real x
conj(Z)	Complejo conjugado	floor(x)	Redondea los decimales al menor entero más cercano
imag(Z)	Parte imaginaria	ceil(x)	Redondea los decimales al mayor entero más cercano
real(Z)	Parte real	round(x)	El entero más próximo al real x
sign(x)	Signo del real x (1 si x>0, -1 si x<0)		

VI. Funciones Matriciales

max(x)	Máximo de los números los elementos de x	gcd(x)	Máximo común divisor de los elementos de x
min(x)	Mínimo de los números los elementos de x	lcm(x)	Mínimo común múltiplo de los elementos de x
size(A)	Filas y columnas de la matriz A	length(x)	Longitud del vector x

VII. Operaciones de matrices

transpose(A)	Traspuesta de la matriz A	det(A)	Determinante de la matriz A
A'	Traspuesta de la matriz A	trace(A)	Suma de los elementos de la diagonal de A
inv(A)	Inversa de la matriz A (A^{-1})	norm(A)	Norma de A (mayor valor singular de la matriz A)
zeros(m,n)	Crea una matriz con m filas y n columnas cuyos elementos son ceros	ones(m,n)	Crea una matriz con m filas y n columnas cuyos elementos son unos
eye(n)	Crea la matriz identidad de orden n	diag(v)	Crea una matriz diagonal con los elementos de v

VIII. Funciones lógicas vectoriales

isempty	Cierto si el vector es vacío	find(A)	Cierto si encuentra índices de elementos no nulos de A
isequal	Cierto si ambos son iguales	any(A)	Cierto si algún elemento de A es no nulo
isfinite	Cierto si es un valor finito	all(A)	Cierto si todos los elementos de A son no nulos

IX. Variable especiales

En MATLAB existen variables de uso común, cuyo valor viene ya preasignado.

pi	3'1415926535897...	NaN	Indeterminación (Not a Number, por ejemplo 0/0)
i ó j	Unidad imaginaria (raíz cuadrada de -1)	realmin	El menor número real positivo utilizable
inf	Infinito, por ejemplo 1/0	realmax	El mayor número real positivo utilizable
eps	Menor valor positivo que sumado a la unidad tiene representación diferente a 1.	ans	Variable creada automáticamente para representar el último resultado

X. Gráficos bidimensionales (2-D)

plot(X)	Representa los puntos (k,X _k). Si X es una matriz, hace lo mismo para cada columna de la matriz. Si X es un vector complejo, representa Real(X) frente a IMAG(X)..				
plot(X,Y)	Representa el conjunto de puntos (X,Y). Si X o Y son matrices, representa por filas o columnas los datos de X frente a los datos de Y, dependiendo si el otro vector es fila o columna. Para valores complejos de X e Y, se ignoran las partes imaginaria.				
grid	Sitúa rejillas en los ejes de un gráfico 2-D o 3-D. La opción grid on coloca las rejillas y grid off las elimina. La opción grid permuta entre on y off				
hold	Permite mantener el gráfico existente con todas sus propiedades, de modo que el siguiente gráfico que se realice se sitúe sobre los mismos ejes y se superponga al existente. La opción hold on activa la opción y hold off la elimina. La opción hold permuta entre on y off. Válido para 2-D y 3-D				
subplot(m,n,p)	Divide la ventana gráfica en mxn subventanas y coloca el gráfico corriente en la ventana p-ésima, empezando a contar por la parte superior izquierda y de izquierda a derecha hasta acabar la línea, para pasar a la siguiente				
plot(X,Y,S)	Gráfica de plot(X,Y) con la opciones definidas en S. Usualmente, S se compone de dos caracteres entre comillas simples, el primero de los cuales fija el color de la línea del gráfico, y el segundo fija el carácter a usar en el graficado. Los valores posibles de colores y caracteres son, respectivamente, los siguientes:				
y amarillo	g verde	.	puntos	-	sólido
m magenta	b azul	o	círculos	:	a puntos
c cyan	w blanco	x	x-marcas	-.	guiones y puntos
r rojo	k negro	+	signo más	--	semisólidos
		*	estrellas		

XI. Títulos, etiquetas, mallas y textos

title('texto')	Añade el texto como título del gráfico en la parte superior del mismo	text(x,y,'texto')	Sitúa el texto en el punto (x,y)
xlabel('texto')	Sitúa el texto al lado del eje x	ylabel('texto')	Sitúa el texto al lado del eje y
gtext('texto')	Permite situar el texto en un punto seleccionado con el ratón	ginput(n)	Permite recuperar las coordenadas de n puntos de un gráfico mediante ratón