

Prácticas Matlab

Práctica 1 (30- IX-2015)

Objetivos

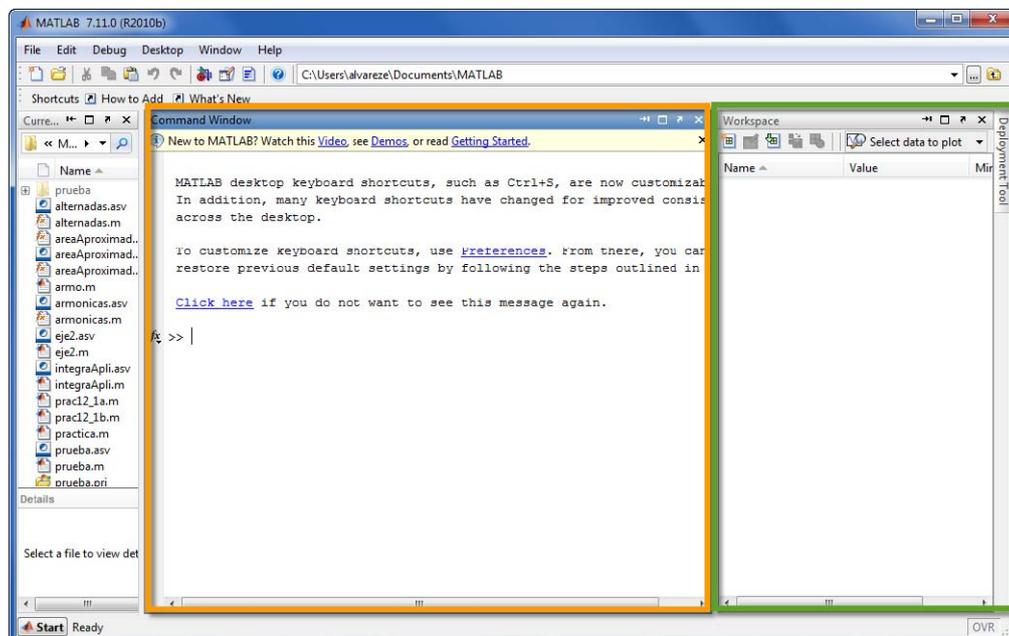
- Iniciarse en el uso de Matlab.
- Conocer comandos básicos de Matlab para realizar cálculos con números reales y números complejos.
- Representar puntos en el plano

¿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.



Inicio de sesión

- Introducir un pendrive y crear una carpeta de nombre "practica1".
- Entrar en Matlab y hacer que *Current Directory* sea la carpeta "practica1".

- Para conservar nuestra sesión de trabajo en un fichero de texto, empezar tecleando:


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


```
>>diary off
```

Operaciones elementales

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

```
>>3.2
```

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

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

Para que Matlab ejecute una orden en la ventana de comandos es necesario pulsar la tecla *intro* o salto de línea.

```
>>3+5^3-2
ans=
    126
```

En el ejemplo anterior el resultado se ha guardado en la variable *ans*. Si al final de la orden se escribe un punto y coma (;) su resultado se calcula pero no se escribe en pantalla. Por ejemplo, si escribimos

```
>>3+5^3-2;
```

el valor de *ans* sería 126 pero no nos lo mostraría.

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

```
>>s=3+5^3-2
```

El nuevo valor de la variable *s* es 126.

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 `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`, . . .

Ayuda de Matlab

En esta primera práctica trabajaremos únicamente en la *Command Window* (Ventana de Comandos) de Matlab.

1.- Cómo obtener ayuda desde la Command Window :

Ejecutar `help`, `lookfor`, `doc`, o `type` seguido del comando del que se requiere la ayuda.

Ejemplo

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

2.- Cómo 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.

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> |
| <code>cos(x)</code> | Coseno de x | <code>cos(0)=1</code> |
| <code>tan(x)</code> | Tangente de x | <code>tan(pi/4)=1.000</code> |
| <code>asin(x)</code> | Arco seno de x con imagen en el rango $[-\pi/2, \pi/2]$ | <code>asin(1)=1.5708</code> |
| <code>acos(x)</code> | Arco coseno de x con imagen en $[-\pi/2, \pi/2]$ | <code>acos(1)=-6.1257e-17</code> |
| <code>atan(x)</code> | Arco tangente de x con imagen en el rango $[-\pi/2, \pi/2]$ | <code>atan(1)=0.7854</code> |
| <code>atan2(y,x)</code> | Arco tangente de y/x con | <code>atan2(0,-1)=3.1416</code> |

| | | |
|------------|----------------------------------|--------------------|
| | imagen en el rango $[-\pi, \pi]$ | |
| $\sinh(x)$ | Seno hiperbólico de x | $\sinh(3)=10.0179$ |
| $\cosh(x)$ | Coseno hiperbólico de x | $\cosh(3)=10.0677$ |
| $\tanh(x)$ | Tangente hiperbólica de x | $\tanh(3)=0.9951$ |

Comandos para trabajar con vectores

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

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
```

Si a y b son matrices y r es un escalar, la forma de indicar a Matlab que realice las operaciones algebraicas habituales es 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 | $[2\ 3].^2 = [4\ 9]$ |

Las funciones matemáticas que se han visto en el apartado anterior pueden aplicarse también a un vector. Por ejemplo, si se quiere calcular el seno a cada componente de un vector basta escribir

```
>>v=0:pi/4:pi;
>>sin(v)
```

Como hemos comentado anteriormente, uno de los aspectos más destacables de Matlab es su capacidad para trabajar con vectores y matrices y, en consecuencia, son muchos los comandos de los que se dispone para su manipulación. A modo de ejemplo:

```
>>v=1/2:1/3:3;
>>sum(v)           %suma las componentes del vector v
>>length(v)       %devuelve el número de elementos de v
```

Comandos para operar con números complejos

`i` (ó `j`)

Es la unidad imaginaria en Matlab

`abs(s)`

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

Ejemplo:

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

Ejemplo:

```
>> 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

Ejemplo:

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

```
>> real([z,w])      % Devuelve 2      5
```

imag(z)

Devuelve la parte imaginaria de z

Ejemplo:

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

```
>> imag(z)          % Devuelve 3
```

```
>> imag([z,w])     % Devuelve 3      7
```

conj(z)

Devuelve el conjugado de z

Ejemplo:

```
>> 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
```

Ejercicios

1

Utilizando Matlab como una calculadora.

(a) Calcula el argumento principal de $z = \frac{\sqrt[6]{2} e^{-\frac{\pi}{2}i} \overline{1-\sqrt{3}i}}{(1+i)^3}$

(b) Ordena los siguientes números complejos

$$z_1 = e^{3+4i} \quad z_2 = (-\sqrt{3} + i)^5 \quad z_3 = \frac{\sqrt{2} e^{i\frac{\pi}{2}} (1-i)}{(2i)^{10}}$$

c.1 En orden creciente a sus módulos

c.2 En orden creciente a sus argumentos principales.

(d) Ordena los siguientes números complejos según su

argumento principal $z_1 = \frac{2}{1-i} + e^{1+2\pi i}$ $z_3 = \frac{(-1+\sqrt{3}i)^{12}}{\left| e^{\frac{7}{3}\pi i} \right|}$

Nota: El argumento principal de un número complejo es aquel argumento φ que está comprendido entre $-\pi$ y π , esto es, $-\pi \leq \varphi < \pi$

(e) Calcula el valor de $z = (\sqrt{3} + i)^8 + (\sqrt{3} - i)^8$

- o Utiliza las funciones
 - `abs` para calcular su módulo,
 - `angle` para calcular su argumento
 - `real` para calcular su parte real
 - `imag` para calcular su parte imaginaria
 - `conj` para calcular su conjugado.

2

Trabajando con vectores

- (a) Calcula las diez primeras potencias de $z = 1 + i$
- (b) Calcula las diez primeras potencias de $w = 1 - i$
- (c) Calcula los valores de $z^k + w^k = (1 + i)^k + (1 - i)^k$ para $k=1,2,\dots,8$
- (d) $z_2 = \sum_{n=0}^{122} i^n$ donde $z_2 = \sum_{n=0}^{122} i^n = i^0 + i + i^2 + i^3 + \dots + i^{122}$

Indicaciones Matlab

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

```
>>exponente=1:10
```

 También podría escribirse


```
>>exponente=linspace(1,10,10)
```
- o Escribe


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

Resumen de comandos

Estos son los comandos utilizados en esta práctica que se darán por conocidos en las prácticas siguientes y que conviene retener porque se podrán preguntar en las distintas pruebas de evaluación.

- Para crear vectores: `:` `linspace`
- Operadores aritméticos
 - Aritméticos `+, -, *, /, ^`
 - Entre vectores `+, -, .* , ./, .^`
- Funciones matemáticas `sin, cos, tan, exp`
- Para sumar componentes de un vector `sum`
- Para operar con complejos `abs, angle, real, imag, conj`

Octave

Octave es una alternativa libre y gratuita a Matlab. Puedes descargar el programa en la siguiente dirección: <https://mat.camino.upm.es/octave/>