

Cálculo Simbólico con MATLAB. Introducción a la Programación.

Bloque I: Introducción. Variables. Operaciones Aritméticas. Funciones Elementales. Números Complejos.

Cristina Solares

Universidad de Castilla-La Mancha

4 de septiembre de 2022

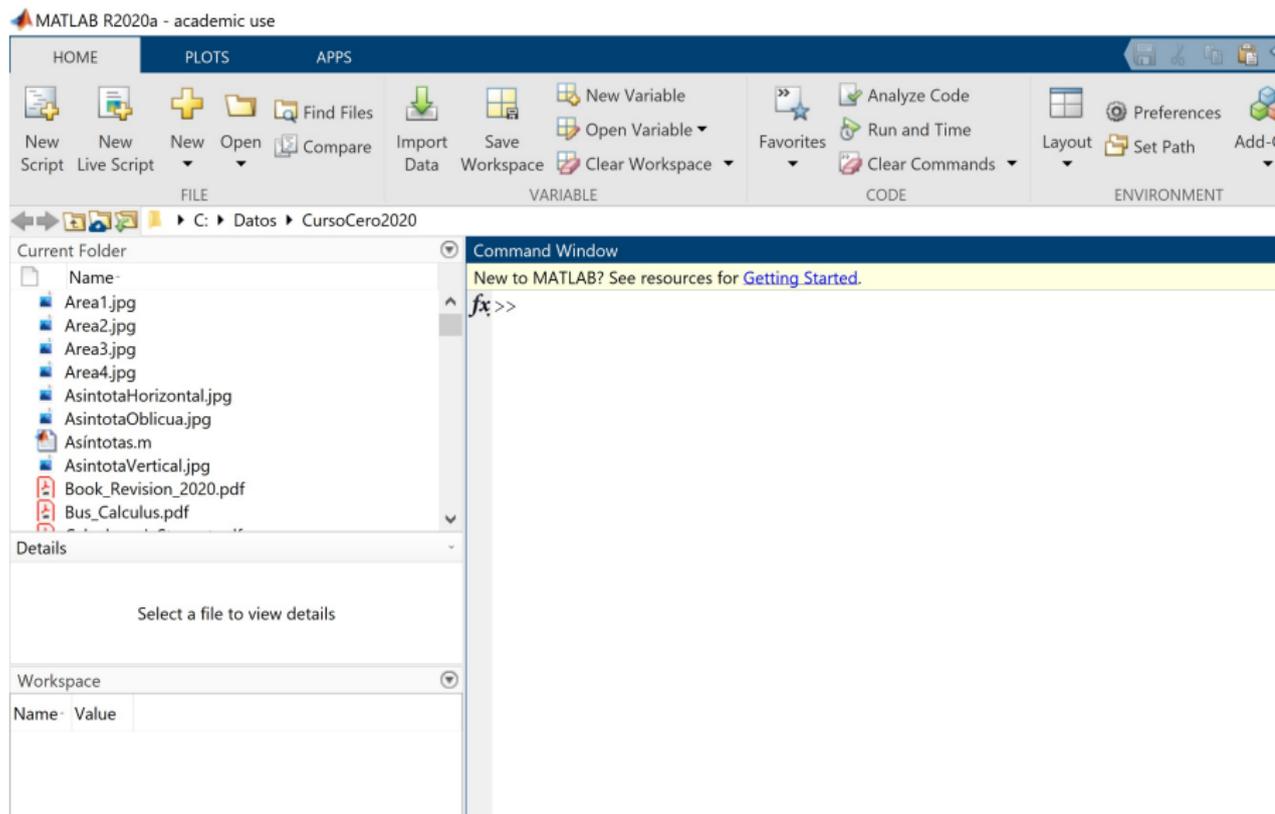
¿Qué es MATLAB (MATrix LAB)?

- Programa de cálculo numérico.
- Programa de cálculo simbólico (Symbolic Math Toolbox).
- Herramienta para trabajar con gráficos.
- Lenguaje de programación.

Inventado por el Prof. Cleve Moler (University of New Mexico, finales de los años 70) para facilitar la programación a sus alumnos.

Empezando con MATLAB

Cuando arrancamos el programa MATLAB en el escritorio aparece la siguiente interfaz de usuario.



En dicha interfaz aparecen las ventanas:

- 1 **Current Folder:** En dicha ventana se muestra el directorio de trabajo actual y se puede acceder a sus ficheros.
- 2 **Command Window:** En la ventana de comandos, a través del prompt `>>`, se pueden introducir operaciones como si se tratara de una calculadora, se pueden introducir expresiones, comandos y programas sencillos. También es posible escribir programas en ficheros de instrucciones (o de funciones) denominados M-files que posteriormente serán llamados desde la ventana de comandos.
- 3 **Workspace:** En el espacio de trabajo se puede explorar las variables que se han creado en la ventana de comandos o han sido importadas de ficheros. Utilizando los comandos `who` y `whos` en la ventana de comandos se muestran dichas variables (tamaño, tipo y atributos).

```
>> who
```

```
Your variables are:
```

```
a b c
```

```
>> whos
```

Name	Size	Bytes	Class	Attributes
a	1x1	8	double	
b	1x1	112	sym	
c	1x1	112	sym	

El comando `diary data.txt` crea el fichero `data.txt` y guarda todos los comandos ejecutados en la sesión de MATLAB hasta que se escribe `diary off`. Para salir del programa se puede escribir `quit` o `exit` en la ventana de comandos.

Variables. Asignación de Valores.

Se puede utilizar una **variable para almacenar un valor numérico**. Para crear una variable numérica y asignarle un valor hacemos

`variable=expresión.`

Comencemos creando la variable `a=2` en la línea de comandos. El resultado se muestra en la ventana de comandos (Command Window). En el espacio de trabajo (Workspace) también se muestra la variable `a` y su valor, que se puede modificar directamente en dicha ventana

```
>> a=2 %Creamos la variable a y le asignamos el valor 2
a =
    2
```

A continuación creamos la variable `b=5` y realizamos la operación `a+b`

```
>> b=5;
>> a+b
ans =
    7
```

Si queremos incrementar el valor de `a` en 3 unidades podemos hacer

```
>> a=a+3
a =
    5
```

- Si ponemos un `;` (punto y coma) al final de la sentencia, se realiza la operación pero no se muestra el resultado.

- Si no asignamos el resultado de la operación a una variable, el programa utiliza la variable por defecto `ans` que también se mostrará en el espacio de trabajo.
- Presionando en el teclado las teclas `[↑]` y `[↓]` se pueden recuperar las variables y comandos que se han utilizado en la sesión actual o en sesiones anteriores.
- Se utiliza `%` para colocar comentarios en el código del programa. Todo lo que sigue a `%` en la misma línea es ignorado por MATLAB.
- Existen restricciones a la hora de nombrar las variables numéricas y simbólicas, se recomienda que el nombre esté relacionado con el significado de cada variable. Las variables siempre empiezan por letras, se recomienda que no coincidan con nombres de funciones definidas en MATLAB y su longitud máxima viene dada por `namelengthmax`.
- Para borrar el contenido de la ventana de comandos se hace `>>clc`.
- Se pueden escribir varias expresiones o comandos en la misma línea, separados por `,` (coma) o `;` (punto y coma).
- MATLAB diferencia entre mayúsculas y minúsculas. Todos los comandos que ya están definidos en MATLAB se escriben en minúsculas.

Para borrar el contenido de una variable escribimos

```
clear nombrevariable.
```

Para borrar el contenido de varias variables escribimos

```
clear nombrevariable1 nombrevariable2
```

y para borrar el contenido de todas las variables hacemos `clear`.

Cada variable en MATLAB tiene un tipo asociado (class). Se pueden definir variables de tipo: double (tipo numérico por defecto), char (caracter o cadena/array de caracteres), logical (valores lógicos true o false).

```
>> a=1;
>> b=5.3;
>> c='x';
>> d=true;
>> f='hola';
>> h=1+i;
>> whos
```

Name	Size	Bytes	Class	Attributes
a	1x1	8	double	
b	1x1	8	double	
c	1x1	2	char	
d	1x1	1	logical	
f	1x4	8	char	
h	1x1	16	double	complex

Errores de sintaxis:

```
>> 1=x
```

```
1=x
```

Error: Incorrect use of '=' operator. To assign a value to a variable, use '='. To compare values for equality, use '=='.

MATLAB intenta ayudarnos con los errores de sintaxis mostrando mensajes de error.

Errores de tipo semántico:

```
>> x=5;y=7;
```

```
>> x=y;
```

```
>> y=x;
```

Deberíamos escribir:

```
>> x=5;y=7;
```

```
>> aux=x;
```

```
>> x=y;
```

```
>> y=aux;
```

Se pueden crear **variables simbólicas para almacenar expresiones simbólicas o valores exactos**. Las variables simbólicas deben ser declaradas con el comando `syms`. Si queremos introducir un valor exacto o una expresión simbólica en dichas variables debemos utilizar el comando `sym`. Si escribimos `syms x` es equivalente a `x=sym('x')`. A continuación mostramos un ejemplo en el que se trabaja con las variables simbólicas `x` e `y`

```
>> syms x y
>> x^2
ans =
x^2
>> y=sym(3/4) %introducimos el valor exacto 3/4 en la variable y
y =
3/4
```

Todas las operaciones con las variables `x` e `y` se realizarán de modo exacto (no aproximado)

```
>> x+y
ans =
x + 3/4
>> y^2
ans =
9/16
```

El comando `subs` permite sustituir una variable simbólica en una expresión por un valor

```
>> expr=x^2+x+1
expr =
x^2 + x + 1
>> subs(expr,x,1)
ans =
3
```

MATLAB reconoce los **operadores aritméticos** habituales de suma (+), diferencia (-), producto (*), división (/) y potencia (^). Se pueden realizar operaciones con valores y variables, numéricos y simbólicos. Al combinar varios operadores en una misma instrucción han de tenerse en cuenta los criterios de prioridad habituales entre ellos. El orden de evaluación normal puede alterarse agrupando expresiones entre paréntesis.

```
>> a=2;b=7;
```

```
>> a/b
```

```
ans =  
    0.2857
```

```
>> syms x;
```

```
>> 2*3-(4+x)
```

```
ans =  
    2 - x
```

```
>> (x+7)/(4+x)
```

```
ans =  
(x + 7)/(x + 4)
```

```
>> 2^2+3*8
```

```
ans =  
    28
```

```
>> syms y; y=sym(2/3);
```

```
>> y^2+4
```

```
ans =  
    40/9
```

En MATLAB podemos realizar operaciones de forma aproximada (numérica) o de forma exacta (simbólica). Las siguientes operaciones se realizan de forma aproximada

```
>> (2/3-3/5)*5/2  
ans =  
    0.1667
```

```
>> sqrt(2)  
ans =  
    1.4142
```

Por defecto, se muestran 4 cifras decimales pero este valor se puede cambiar a 15. Para ello se utiliza el comando `format` con los valores `short` o `long`, respectivamente.

```
>> format long  
>> sqrt(2)  
ans =  
    1.414213562373095  
>> format short  
>> pi  
ans =  
    3.1416
```

Además para mostrar números decimales se puede utilizar una notación científica o exponencial. Si queremos escribir $2 * 10^4$ podemos hacerlo de las dos formas siguientes

```
>> 2*10^4
ans =
    20000
```

```
>> 2e4
ans =
    20000
```

Utilizando el comando `sym` se pueden crear expresiones simbólicas y trabajar de forma exacta. Las siguientes operaciones se realizan de forma exacta

```
>> sym(sqrt(2))
ans =
 2^(1/2)
>> sym(2/3)
ans =
 2/3
```

El comando `double` puede cambiar la expresión simbólica a numérica

```
>> sym(1/2+1/4)
ans =
 3/4
>> double(ans)
ans =
    0.7500
```

Algunas **constantes definidas en MATLAB** son

- **pi**: Número π .
- **i** o **j**: Constante imaginaria que se utiliza para trabajar con números complejos.

```
>> i*i  
ans =  
    -1
```

- **Inf**: Infinito. Aparece al dividir un número por cero.

```
>> 2/0  
ans =  
    Inf
```

- **NaN**: Número indefinido. Por ejemplo al realizar la operación $0/0$.

```
>> 0/0  
ans =  
    NaN
```

Algunas **funciones definidas en MATLAB** son

- Funciones numéricas: `sign` (signo), `abs` (valor absoluto), `sqrt` (raíz cuadrada), `round` (redondeo a entero más próximo).
- Funciones trigonométricas: `sin` (seno), `cos` (coseno), `tan` (tangente).
- Funciones hiperbólicas: `sinh` (seno hiperbólico), `cosh` (coseno hiperbólico)
- Función exponencial y logarítmica: `exp` (función exponencial), `log` (función logaritmo)

Con la función `help` se puede obtener información sobre cómo utilizar una función o comando concretos.

```
>> help sin
```

```
sin - Sine of argument in radians
```

```
This MATLAB function returns the sine of the elements of X.
```

```
Y = sin(X)
```

```
.....
```

Los argumentos de las funciones se pasan entre paréntesis y separados por coma. Los argumentos pueden ser numéricos (reales o complejos) o simbólicos.

```
>> sin(pi/4)
```

```
ans =
```

```
0.7071
```

```
>> syms x; sin(x)
```

```
ans =
```

```
sin(x)
```

Si escribimos `help elfun`, se obtiene información sobre todas las funciones matemáticas definidas en MATLAB.

En MATLAB se puede **trabajar con números complejos** definidos de forma numérica, simbólica o exacta como $a+bi$ siendo a la parte real, b la parte imaginaria e i la unidad imaginaria.

```
>> (2+3i)+(1+5i) %suma
ans =
    3.0000 + 8.0000i
>> syms a
>> (a+i)*(3+i) %producto
ans =
a*(3 + 1i) - 1 + 3i
>> (1+2i)/i %cociente
ans =
    2.0000 - 1.0000i
```

Además, se pueden utilizar las funciones `abs`, `angle`, `conj`, `imag` y `real` para obtener el módulo, argumento, el conjugado, la parte imaginaria o real de un número complejo

```
>> abs(1+i) % mod. forma aproximada
ans =
    1.4142
>> sym(abs(1+i)) % mod. forma exacta
ans =
2^(1/2)
```

```
>> angle(1+i) % argumento forma aproximada (radianes)
ans =
    0.7854
>> sym(ans) % argumento forma exacta
ans =
pi/4

>> real(1+i) % parte real
ans =
    1
>> syms a
>> real(a+2i) % parte real
ans =
real(a)
>> imag(a+2i) % parte imaginaria
ans =
imag(a) + 2
```

Nótese que en el caso del número $a+2i$ se considera que a puede ser un número real o complejo.