

Comenzando con MATLAB

1. Introducción

MATLAB (Matrix laboratory) es un sistema de software interactivo para computaciones numéricas y gráficos. Como su nombre sugiere, MATLAB está específicamente desarrollado para realizar computaciones matriciales. Además, dispone de capacidades gráficas, y puede ser extendido a través de programas escritos en su propio lenguaje de programación.

La primera versión de Matlab data de los años 70, y fue diseñada como herramienta de apoyo para los cursos de Teoría de Matrices, Álgebra Lineal y Análisis Numérico. Hoy en día, Matlab es un programa muy potente, con un entorno agradable, que incluye herramientas de cálculo científico y técnico, de visualización gráfica, así como un lenguaje de programación de alto nivel.

MATLAB está diseñado para resolver problemas numéricos, esto es, trabaja en precisión aritmética finita. Por tanto, produce soluciones aproximadas, y no debe ser confundido con un sistema de computación simbólica (SCS) como son Mathematica o Maple. Esto no hace a MATLAB mejor o peor que un SCS; es una herramienta diseñada para realizar tareas distintas y no es directamente comparable.

MATLAB es muy utilizado por ingenieros para llevar a cabo análisis y diseño. Hay diferentes toolboxes disponibles que extienden las funciones básicas del MATLAB en muchos campos de aplicación.

En las siguientes secciones, veremos una introducción a algunas de las características más útiles de MATLAB.

La mejor forma de aprender a usar MATLAB es leer mientras se ejecuta MATLAB, intentando los ejercicios y ejemplos, y experimentando.

2. Comenzando con MATLAB

Al entrar en MATLAB (haciendo doble click en el icono correspondiente) se abrirá una ventana de comandos (Command Window) donde veremos el

prompt >> que significa que el programa MATLAB está listo para recibir las órdenes de ejecución del usuario.

El *prompt* >> indica que MATLAB está a la espera de un comando. Para realizar una ejecución se debe teclear un comando y seguidamente presionar *Enter*.

Todos los comandos MATLAB deben ser escritos en letras minúsculas.

La mejor manera de aprender el uso de MATLAB es tecleando, ejecutando ejemplos y ejercicios, y probando.

Ahora, consideremos diferentes cuestiones importantes relativas al MATLAB.

Comandos de ayuda

Una de las cosas más útiles en MATLAB es su sistema de ayuda,

- *help* da una lista de temas básicos de ayuda.
- *help nombre-comando* informa del funcionamiento de dicho comando.

También obtenemos la ayuda seleccionando en la barra de menú Help.

Operaciones aritméticas

La forma de representar números y de operar de Matlab es la misma que la de las calculadoras de bolsillo. Por ejemplo:

3 -99 .001 9.63 1.62e-020

Observa que se usa el punto como separador decimal, en lugar de la coma.

Las operaciones usuales se realizan con los mismos símbolos y en la misma secuencia que en las calculadoras,

$a + b$ Suma

$a - b$ Resta

$a * b$ Multiplicación

$a / b = b \setminus a$ División

a^b Potencia a^b

Para que Matlab ejecute una orden, es preciso pulsar la tecla Enter. Por ejemplo, para calcular el valor de $3 + 5 \cdot 2 + 1$, se ejecuta la instrucción

>> 3 + 5 * 2 + 1

Y se obtiene como respuesta

ans = 14

Esto quiere decir que el resultado se ha almacenado en la variable *ans*. En cambio,

```
>> s=(3 + 5) * 2 + 1
```

indica a Matlab que el resultado de esa operación ha de guardarse en la variable *s*. Observa la diferencia con el caso anterior, los paréntesis alteran el valor.

Variables

- Matlab no requiere ningún tipo de comando para definir variables. Sencillamente crea la variable mediante asignación directa de su valor. Una vez declarada la variable podemos utilizarla en los cálculos.
- Las variables son sensibles a las letras mayúsculas.
- Deben comenzar con una letra, seguida por cualquier número de letras, dígitos o guiones de subrayado. No se permiten espacios en blanco.
- Su nombre no debe coincidir con instrucciones predefinidas en Matlab.
- Por defecto, Matlab almacena resultados en la variable *ans*.
- Algunas variables ya están predefinidas en Matlab, por ejemplo *pi* es el número π , *inf* es ∞ , *i* es la unidad imaginaria ó *j* que es también la unidad imaginaria.
- Preguntamos por una variable poniendo su nombre en la línea de comandos.
- Preguntamos por las variables que tenemos definidas con las órdenes
 - *who* nos enseña las variables que tenemos definidas
 - *whos* nos enseña el detalle de las variables que tenemos definidas
- En el espacio de trabajo de Matlab se pueden borrar las variables
 - *clear nombre-var* borra sólo la variable *var*
 - *clear* borra todas las variables

Otras herramientas

- El signo punto y coma al final de una instrucción hace que el programa ejecute dicha instrucción sin mostrar el resultado.

- Para introducir comentarios, los iniciamos con el signo tanto por ciento. Así, al ejecutar la entrada, el programa ignora todo lo que está escrito después del tanto por ciento y ejecuta el resto.
- Se pueden realizar varias entradas en la misma línea separándolas por comas.
- Utilizando las teclas de edición: Las teclas con flechas permiten modificar los datos introducidos,

↑ recupera la línea previa

↓ recupera la línea siguiente

← mueve el cursor hacia la izquierda un carácter

→ mueve el cursor hacia la derecha un carácter

- Si escribimos una secuencia de caracteres determinada y pulsamos la tecla ↑ recuperamos la última entrada que comienza con dicha secuencia de caracteres.
- Si pinchamos dos veces en una línea del "Command History", se vuelve a ejecutar dicha línea en la ventana de comandos. Además, podemos copiar y pegar, lo que permite corregir líneas.
- La orden *clc* borra la ventana de comandos (no borra las variables).
- Con *exit* o *quit* finalizamos la sesión de MATLAB. También finaliza la sesión si simplemente cerramos la ventana del MATLAB.

3. Cálculo numérico y simbólico

MATLAB está diseñado para resolver problemas de forma numérica, pero existe una toolbox para simbólico que permite realizar algunos cálculos simbólicos.

Formatos numéricos más utilizados

- *format short* Formato corto y decimal fijo con 4 dígitos después del punto decimal (por defecto). Por ejemplo, el número π aparecería 3.1416.

- *format long* Formato decimal fijo y largo con 15 dígitos después del punto decimal para los valores de double y 7 dígitos después del punto decimal para los valores de single. Por ejemplo, el número π aparecería 3.141592653589793.
- *format rat* Proporciona una aproximación racional. Por ejemplo, π aparecería como 355/113.
- *short e* Notación científica corta con 4 dígitos después del punto decimal. Por ejemplo, el número π aparecería 3.1416e+00.
- *long e* Notación científica larga con 15 dígitos después del punto decimal para los valores de double y 7 dígitos después del punto decimal para los valores de single. Por ejemplo, el número π aparecería 3.141592653589793e+00.

Objetos simbólicos

Tecleando *syms x* creamos la variable simbólica x .

Un objeto simbólico puede ser una constante, una variable o una expresión en la que intervengan constantes, variables, operadores aritméticos y funciones predefinidas. Veamos algunos comandos.

- *syms a b c* declara a , b y c como variables simbólicas
- $M = \text{sym}(m)$ crea el objeto simbólico M a partir de m
- *pretty(f)* muestra la expresión simbólica f con la notación matemática usual
- *expand(f)* desarrolla la expresión simbólica f
- *simplify(f)* simplifica la expresión simbólica f
- *subs(f,x,s)* substituye en la expresión simbólica f la x por s .
- *solve(f)* resuelve la ecuación $f = 0$
- *factor(f)* factoriza la expresión simbólica f
- *factor(n)* descompone un número entero n en factores primos
- *finverse(f)* calcula la función inversa de f
- *compose(g,f)* calcula la función f compuesta con g

- *compose(f,g)* calcula la función g compuesta con f

Algunas funciones matemáticas predefinidas en MATLAB

La librería Matlab dispone de una gama muy completa de funciones predefinidas que se corresponden con las funciones matemáticas más utilizadas. Algunos ejemplos son,

- *abs(x)* modulo de x (valor absoluto de x si es real)
- *angle(x)* argumento de x (ángulo en radianes entre $(-\pi, \pi]$)
- *conj(x)* conjugado de x
- *real(x)* parte real de x
- *imag(x)* parte imaginaria de x
- *sin(x)* seno de x
- *cos(x)* coseno de x
- *tan(x)* tangente de x
- *asin(x)* arcoseno de x
- *acos(x)* arcocoseno de x
- *atan(x)* arcotangente de x
- *exp(x)* exponencial de x
- *log(x)* logaritmo neperiano (en base e) de x
- *sqrt(x)* raíz cuadrada de x

Nota: Los argumentos de las funciones trigonométricas son siempre ángulos en radianes. Se tiene que

- *sind(x)* es el seno de x expresado en grados
- *cosd(x)* es el coseno de x expresado en grados
- *tand(x)* es el tangente de x expresado en grados
- *asind(x)* es el arcoseno de x expresado en grados
- *acosd(x)* es el arcocoseno de x expresado en grados
- *atand(x)* es el arcotangente de x expresado en grados

3.1. Ejercicios

Ejercicio 1. Realiza los siguientes cálculos

$$a = 2^3 - 4 + \frac{6}{3} \cdot 2,$$

$$b = 2^3 - 4 + \frac{6}{3 \cdot 2},$$

$$c = 2 \cdot 3^2 + 6,$$

$$d = (2 \cdot 3)^2 + 6,$$

$$e = \frac{b - a}{d^3}.$$

Ejercicio 2. Determina la superficie comprendida entre un cuadrado de lado $la=2$ y la circunferencia inscrita en él. Repite el ejercicio con $la=7$.

Ejercicio 3. Realiza las siguientes ejecuciones y observa los resultados:

```
>> syms x
>> expand((1+x)*(3-x))
>> pretty(expand((1+x)*(3-x)))
>> f=2*x-2-2*(x-1)^2+(x-1)^3
>> solve(f)
>> simplify(f)
>> subs(f,x,7)
>> syms s
>> subs(f,x,s)
>> factor(f)
>> factor(6552)
```

Ejercicio 4. Calcula $z = \frac{(2\sqrt{3}-2i)(\cos 28^\circ + i \sin 28^\circ)}{(\cos 31^\circ + i \sin 31^\circ)}$

Ejercicio 5. Sean $f(x) = x^2 + 1$ y $g(x) = x^3 + 2x - 3$. Calcula $f + g$, $f - g$, $f * g$, f/g , f^{-1} , $g \circ f$ y $f \circ g$.