



Álgebra Lineal y Geometría

Introducción a MATLAB

Dpto. Matemática Aplicada y Ciencias de la Computación

¿Qué es MATLAB?

MATLAB es un lenguaje de programación de alto nivel enfocado inicialmente al cálculo numérico. Por lo tanto, tiene diversas herramientas para el cálculo matricial y la resolución numérica de problemas lineales y no lineales. Del mismo modo, presenta diferentes herramientas gráficas para visualizar los resultados obtenidos. Se puede utilizar a través de su línea de comandos mediante la interfaz de usuario.

MATLAB es un programa comercial y cerrado, que requiere el pago de una licencia. La Universidad de Cantabria tiene licencia en las aulas de informática de la EPIME y también puede instalarse en los ordenadores personales siguiendo [estas instrucciones](#) y el usuario de la UC.

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.

Se pueden consultar manuales en la [web oficial](#).

Empezando a usar MATLAB

La ventana de MATLAB muestra un escritorio dividido en varias partes, entre ellas:

- Las órdenes a ejecutar se escriben en la ventana de comandos, `Command Window`.
- En el editor (ventana `Editor`) escribimos código que guardaremos en scripts con extensión `.m` y ejecutaremos en la ventana de comandos.
- La ventana `Workspace` proporciona información sobre las variables utilizadas.

Hay ciertas reglas para definir variables en MATLAB:

- 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, `matriz2` es un nombre válido, pero no lo es `2matriz`.

- 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`, . . .

A tener en cuenta:

- Al abrir sesión seleccionaremos el **directorio de trabajo**, donde almacenaremos los scripts.
- Si se escribe `;` al final de una línea de código, no se muestra el resultado por pantalla. Esta característica es la deseada cuando se trabaja con grandes conjuntos de datos.
- **Obtener ayuda** en Matlab: El comando `help` es la orden básica de ayuda con la cual consultar el uso de una función, incluyendo los argumentos de entrada/salida, una breve descripción de qué realiza y de su uso.
- Se pueden incluir **comentarios** (aclaraciones que no se ejecutan) dentro de los programas añadiendo `%` delante de lo que se desea comentar.
- Se puede **ejecutar** un programa entero pulsando la tecla "Run", o bien una(s) línea(s) seleccionadas pulsando F9 o seleccionando "Evaluate selection" en el menú que sale al pinchar con el botón derecho del ratón.
- Es **esencial leer los errores** que obtenemos cuando algo no funciona, ya que contienen información acerca del error. Además, la(s) línea(s) afectada(s) tiene una señal roja a la derecha del editor.

Tipos de datos

MATLAB trabaja con los tipos básicos de datos (números, caracteres y booleanos) a partir de los cuales se construyen vectores, matrices, cadenas de caracteres, etc. Del mismo modo, a partir del tipo de datos se definen diferentes operaciones básicas (suma, producto, intersecciones, etc.) que permiten la construcción y definición por composición de operaciones complejas.

Además de los tipos de datos básicos existen diferentes tipos de variables que sirven de contenedores de datos, como son el tipo `cell` o `struct`.

Para saber qué variables están definidas y a qué tipo de datos corresponde tenemos el comando `whos`. La asignación de una variable con un valor se hace mediante el operador `=`.

Comandos básicos

Cabecera de un programa

```
% Limpia el espacio de trabajo
clear all,
% Limpia la ventana de comandos
clc
% Cierra las ventanas/figuras abiertas
close all
% Lista las variables definidas en el espacio de trabajo
```

```
whos
```

Operaciones básicas

La forma de representar números y de operar con Matlab es la misma que la de una calculadora de bolsillo, usando las operaciones +, -, *, /, ^.

```
17+5 %Suma
```

```
ans = 22
```

```
17/5 %Division con decimales
```

```
ans = 3.4000
```

```
cos(pi) %Coseno de pi
```

```
ans = -1
```

```
sin(pi/2) %Seno de pi/2
```

```
ans = 1
```

```
log(exp(1)^5) %Función logaritmo
```

```
ans = 5
```

```
a=4*atan(1) %atan: Funcion arcotangente con resultado en radianes.
```

```
a = 3.1416
```

```
% Se guarda el resultado en la variable a
```

```
1+2*3 %Jerarquia de las operaciones: multiplicacion y division, suma y resta
```

```
ans = 7
```

```
1/(2+3) %Uso del parentesis
```

```
ans = 0.2000
```

Crear y manipular vectores

Hay varias maneras de construir vectores fila:

```
u=[-7,3,4,-2] %Corchetes y comas
```

```
u = 1×4  
-7 3 4 -2
```

```
v=[-7 3 4 -2] %Corchetes sin comas
```

```
v = 1×4  
-7 3 4 -2
```

```
t= 1:3:20 % para crear secuencias de números tomados de tres en tres,
```

```
t = 1×7
```

1 4 7 10 13 16 19

```
% tales que inicial:incremento:final  
z= linspace(1,2,9) % genera un vector fila de 9 elementos cuyo primer elemento
```

```
z = 1×9  
 1.0000  1.1250  1.2500  1.3750  1.5000  1.6250  1.7500  1.8750 ...
```

```
% es 1 y el último 2, siendo todos sus elementos equidistantes:  
% linspace(inicial, final, n)
```

Y vectores columna:

```
w=[-7;3;4;-2] %Vector vertical
```

```
w = 4×1  
 -7  
  3  
  4  
 -2
```

```
x= u' % Trasponiendo un vector fila
```

```
x = 4×1  
 -7  
  3  
  4  
 -2
```

Se puede comprobar el tamaño de los vectores y extraer elementos:

```
length(u) %Numero de elementos del vector
```

```
ans = 4
```

```
length(w)
```

```
ans = 4
```

```
size(u) %Numero de filas y columnas
```

```
ans = 1×2  
 1  4
```

```
size(w)
```

```
ans = 1×2  
 4  1
```

```
u(1) %Primer elemento del vector u
```

```
ans = -7
```

```
w(4) %Cuarto elemento del vector w
```

```
ans = -2
```

```
u(1)=-4 %Cambia el elemento 1 del vector u
```

```
u = 1x4
    -4     3     4    -2
```

Crear y manipular matrices

```
A=[3,-7;2,5] %Corchetes y comas. Definida por filas
```

```
A = 2x2
     3    -7
     2     5
```

```
B=[3 -7; 2 5] %Corchetes sin comas
```

```
B = 2x2
     3    -7
     2     5
```

```
C=[6 0 -3 8 9; 5 1 7 -4 2; 3 1 -6 2 -1]
```

```
C = 3x5
     6     0    -3     8     9
     5     1     7    -4     2
     3     1    -6     2    -1
```

```
C(2, 3) %Elemento de la fila 2 y columna 3 de la matriz c
```

```
ans = 7
```

```
size(C) %Numero de filas y columnas de la matriz c
```

```
ans = 1x2
     3     5
```

```
length(C) %Numero de columnas de la matriz c
```

```
ans = 5
```

```
C(1, 4)=11 %Cambio del elemento de la fila 1 y 4 columna de la matriz c
```

```
C = 3x5
     6     0    -3    11     9
     5     1     7    -4     2
     3     1    -6     2    -1
```

Matrices especiales

Son aquellas formadas por ceros, unos o la identidad.

```
zeros(3,5) %Nula de 3 filas y 5 columnas
```

```
ans = 3x5
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
```

```
zeros(4) %Nula 4x4
```

```
ans = 4x4
    0     0     0     0
    0     0     0     0
    0     0     0     0
    0     0     0     0
```

```
ones(2,4) %Unidad de 2 filas y 4 columnas
```

```
ans = 2x4
    1     1     1     1
    1     1     1     1
```

```
ones(3) %Unidad 3x3
```

```
ans = 3x3
    1     1     1
    1     1     1
    1     1     1
```

```
eye(4) % Matriz identidad de orden 4 (i.e. diagonal de 1 4x4)
```

```
ans = 4x4
    1     0     0     0
    0     1     0     0
    0     0     1     0
    0     0     0     1
```

```
A = diag([-7 2 4]) %Diagonal con los valores dados
```

```
A = 3x3
   -7     0     0
    0     2     0
    0     0     4
```

Submatrices

Se pueden extraer submatrices de una matriz mayor indicando la fila o columna que se desea. : indica todos los elementos.

```
A=[6 0 -3 8 9; 5 1 7 -4 2; 3 1 -6 2 -1] %Define matriz
```

```
A = 3x5
    6     0    -3     8     9
    5     1     7    -4     2
    3     1    -6     2    -1
```

```
A(2, :) %Fila 2 de la matriz a
```

```
ans = 1x5
    5     1     7    -4     2
```

```
A(:, 5) %Columna 5 de la matriz a
```

```
ans = 3x1
    9
    2
   -1
```

```
A(3, :)= -7 %Cambia todos los elementos de la fila 3 por -7
```

```
A = 3x5
    6     0    -3     8     9
    5     1     7    -4     2
   -7    -7    -7    -7    -7
```

Operaciones con matrices

Cuando trabajamos con matriz, para operar elemento a elemento hay que usar los comandos: `.*`, `./`, `.^`, ya que el producto de matrices como tal no está definido elemento a elemento.

```
A=[-3 7; 2 5] %Define matriz
```

```
A = 2x2
   -3     7
    2     5
```

```
A+100 %Suma 100 a cada elemento de la matriz. % Igual para restas, multiplicaciones y d
```

```
ans = 2x2
    97    107
   102    105
```

```
A=[-3 4 2; 1 7 -5] %Define matriz
```

```
A = 2x3
   -3     4     2
    1     7    -5
```

```
B=[1 -2 -9; 4 2 6] %Define matriz
```

```
B = 2x3
    1    -2    -9
    4     2     6
```

```
C=A+B % guarda el resultado en la variable c
```

```
C = 2x3
   -2     2    -7
    5     9     1
```

```
A+B %Opera sin almacenar
```

```
ans = 2x3
   -2     2    -7
    5     9     1
```

```
0.3*A %Multiplica todos los valores de a por 0.3
```

```
ans = 2x3
  -0.9000    1.2000    0.6000
   0.3000    2.1000   -1.5000
```

```
C=A.*B %Multiplica matrices elemento a elemento
```

```
C = 2x3
   -3    -8   -18
    4    14   -30
```

Funciones matemáticas elementales

- $\exp(x)$: exponencial de x
- $\log(x)$: logaritmo natural o neperiano.
- $\log_{10}(x)$: logaritmo en base 10.
- $\sin(x)$: seno de x .
- $\cos(x)$: coseno de x .
- $\tan(x)$: tangente de x .
- $\text{asin}(x)$: arco seno de x con imagen en el rango $[-\pi/2, \pi/2]$.
- $\text{acos}(x)$: arco coseno de x con imagen en el rango $[-\pi/2, \pi/2]$.
- $\text{atan}(x)$: arco tangente de x con imagen en el rango $[-\pi/2, \pi/2]$.
- $\text{atan2}(y, x)$: arco tangente de y/x con imagen en el rango $[-\pi, \pi]$.
- $\sinh(x)$: seno hiperbólico de x .
- $\cosh(x)$: coseno hiperbólico de x .
- $\tanh(x)$: tangente hiperbólica de x .
- $\text{sum}(x)$: suma de los elementos del vector x .
- $\text{prod}(x)$: producto de los elementos del vector x .

Importante: en estas funciones el ángulo viene en radianes.

Ejercicio resuelto

1. Usar Matlab como una calculadora.

- Calcula la suma de los primeros 100 números naturales.
- Calcula la suma de los cubos de los primeros 100 números naturales.
- Calcula la suma de todos los números pares de 3 cifras y la suma de sus cuadrados.
- Calcula la suma de las raíces cuadradas de los primeros 50 números naturales.

- Calcula la suma de los primeros 100 números naturales.

```
k=1:100; % crea un vector con los números de 1 al 100.  
suma=sum(k) % suma los elementos de k y guarda el resultado en la variable suma.
```

```
suma = 5050
```

- Calcula la suma de los cubos de los primeros 100 números naturales.

```
k3=k.^3; % eleva al cubo cada elemento del vector k  
suma=sum(k3) % suma los elementos de k3
```

```
suma = 25502500
```

- Calcula la suma de todos los números pares de 3 cifras y la suma de sus cuadrados.

```
pares=100:2:998; % crea el vector de números pares de 3 cifras
suma=sum(pares) % suma los elementos del vector pares
```

```
suma = 247050
```

```
suma_cuadrados=sum(pares.^2) % suma de los cuadrados de pares
```

```
suma_cuadrados = 166005300
```

d) Calcula la suma de las raíces cuadradas de los primeros 50 números naturales.

```
a=1:50; % vector formado por los primeros 50 números naturales
raiz=sqrt(a); % calcula la raíz cuadrada de los elementos de a
suma=sum(raiz) % calcula la suma de las raíces cuadradas
```

```
suma = 239.0358
```

Ejercicios propuestos

1. Calcula la suma de las raíces cúbicas de los primeros 50 números naturales.
2. Calcula la suma de los cuadrados de todos los números impares de dos cifras.
3. Calcula la suma de las raíces cuartas de todos los números pares de dos cifras.

```
% Ejercicio 1
k=1:50;
raices=k.^(1/3);
suma3=sum(raices)
```

```
suma3 = 139.7179
```

```
% Ejercicio 2
k=11:2:99;
cuadrados = k.^2;
suma5=sum(cuadrados)
```

```
suma5 = 166485
```

```
% Ejercicio 3
k=10:2:99;
raices=k.^(1/4);
suma7=sum(raices)
```

```
suma7 = 118.6799
```