

# Matlab

Autor: Alain Antonio Cabrera González.

## Introducción

**MATLAB** (abreviatura de **MATrix LABoratory**) es un software matemático que permite operaciones de vectores y matrices, funciones, cálculo simbólico, y programación orientada a objetos. Utiliza un lenguaje de programación propio (lenguaje M).

MatLab emplea matrices porque con ellas se puede describir infinidad de cosas de una forma altamente flexible y matemáticamente eficiente.

## Inicio

Después de ejecutar el programa MatLab, aparece el indicador de comandos el cual está listo para recibir instrucciones en lenguaje MatLab. Este indicador es de la siguiente forma:

```
>>
```

Al iniciar el uso de MatLab están disponibles dos comandos de ayuda y

demostración. Para ejecutarlos se escribe el comando en la línea de comandos después del símbolo >> y se presiona la tecla Enter. Por ejemplo:

```
>> help
```

permite obtener una ayuda sobre los diferentes comandos de MatLab.

```
>> demo
```

hace una demostración de las diferentes aplicaciones de MatLab.

Para cerrar o finalizar el uso de MatLab se usa el comando quit.

```
>> quit
```

## Comandos

La primera forma de interactuar con MatLab es a través de la línea de comandos. Puede ejecutarse un comando si este está escrito después del símbolo >> y se presiona la tecla Enter.

MATLAB trabaja esencialmente con matrices numéricas rectangulares. La manera más fácil de entrar matrices pequeñas es enumerando los elementos de ésta de tal manera que:

- los elementos estén separados por espacios ó comas.
- los elementos estén cerrados entre corchetes, [ ].
- muestre el final de cada fila con ; (punto y coma).

Ejemplos:

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

A =

1	2	3
4	5	6

```
B=[7,8,9;10,11,12]
```

```
B =  
     7     8     9  
    10    11    12
```

Matlab guarda estas matrices con los nombres A y B, respectivamente.

Para trabajar con escales se utiliza una asignación como la siguiente:

```
C=1
```

```
C = 1
```

Automáticamente se presenta en pantalla la asignación realizada, en caso de no querer desplegar dicha asignación se utilizará punto y coma al final de la línea. Por ejemplo:

```
D=1;
```

Para desplegar el valor de una variable basta con escribir su nombre:

```
A
```

```
A =  
     1     2     3  
     4     5     6
```

Se pueden redefinir variables, por ejemplo:

```
A=[9,8,7]
```

```
A =  
     9     8     7
```

## Entradas de una matriz

Las entradas de una matriz pueden ser cualquier expresión de MATLAB. Por ejemplo:

```
E=[-1 sqrt(2) 3.5 pi; exp(1) 3*exp(2) inf 5]
```

```
E =  
 -1.0000    1.4142    3.5000    3.1416  
  2.7183   22.1672    Inf     5.0000
```

Para ver más decimales se puede utilizar la siguiente instrucción:

```
format long  
E
```

```
E =  
 -1.0000000000000000    1.414213562373095    3.5000000000000000    3.141592653589793  
  2.718281828459046   22.167168296791949             Inf     5.000000000000000
```

Para volver a los decimales originales utilizamos:

```
format short  
E
```

```
E =  
-1.0000    1.4142    3.5000    3.1416  
 2.7183   22.1672    Inf    5.0000
```

Nos podemos referir a elementos individuales de la matriz con índices entre paréntesis. Por ejemplo:

```
E(2,3)
```

```
ans = Inf
```

Es posible hacer una reasignación de ese elemento:

```
E(2,3)=abs(-8)
```

```
E =  
-1.0000    1.4142    3.5000    3.1416  
 2.7183   22.1672    8.0000    5.0000
```

También es posible obtener renglones o columnas de una matriz. Por ejemplo, para obtener el segundo renglón de nuestra matriz podemos usar el comando:

```
E(2,:)
```

```
ans =  
 2.7183   22.1672    8.0000    5.0000
```

Para obtener la tercer columna de la matriz usamos:

```
E(:,3)
```

```
ans =  
 3.5000  
 8.0000
```

Para trabajar con una matriz que contenga la primer y la tercer columna de la matriz se utiliza la instrucción:

```
[E(:,1) E(:,3)]
```

```
ans =  
-1.0000    3.5000  
 2.7183    8.0000
```

Podemos generar una matriz con entradas aleatorias. Por ejemplo:

```
F=rand(5,8)
```

F =

0.0326	0.3689	0.6448	0.1206	0.2518	0.9827	0.9063	0.0225
0.5612	0.4607	0.3763	0.5895	0.2904	0.7302	0.8797	0.4253
0.8819	0.9816	0.1909	0.2262	0.6171	0.3439	0.8178	0.3127
0.6692	0.1564	0.4283	0.3846	0.2653	0.5841	0.2607	0.1615
0.1904	0.8555	0.4820	0.5830	0.8244	0.1078	0.5944	0.1788

Para obtener la submatriz sin el primer y el último renglón y columnas utilizamos el siguiente comando:

```
G=F(2:4,2:7)
```

G =

0.4607	0.3763	0.5895	0.2904	0.7302	0.8797
0.9816	0.1909	0.2262	0.6171	0.3439	0.8178
0.1564	0.4283	0.3846	0.2653	0.5841	0.2607

Para obtener la dimensión de una matriz usamos la siguiente instrucción:

```
n=size(G)
```

n =

3      6

El resultado es un vector, para obtener el número de renglones tenemos que acceder a la primer entrada de dicho vector:

```
n(1)
```

ans = 3

Alternativamente, para obtener el número de columnas podemos utilizar la función length:

```
length(G)
```

ans = 6

También podemos hacer lo siguiente:

```
[m,n]=size(G)
```

m = 3

n = 6

## Método de Gauss-Jordan

Ejemplo: Considere el sistema de ecuaciones

$$3x_2 - 6x_3 + 6x_4 + 4x_5 = -5$$

$$3x_1 - 7x_2 + 8x_3 - 5x_4 + 8x_5 = 9$$

$$3x_1 - 9x_2 + 12x_3 - 9x_4 + 6x_5 = 15$$

Trabajaremos con la matriz aumentada:

$$\begin{bmatrix} 0 & 3 & -6 & 6 & 4 & -5 \\ 3 & -7 & 8 & -5 & 8 & 9 \\ 3 & -9 & 12 & -9 & 6 & 15 \end{bmatrix}$$

Para trabajar con ella en matlab usaremos:

```
A=[0 3 -6 6 4 -5;3 -7 8 -5 8 9;3 -9 12 -9 6 15]
```

```
A =  
     0     3    -6     6     4    -5  
     3    -7     8    -5     8     9  
     3    -9    12    -9     6    15
```

Nuestro objetivo es encontrar la forma escalonada reducida. Lo podemos hacer de la siguiente manera:

```
rref(A)
```

```
ans =  
     1     0    -2     3     0   -24  
     0     1    -2     2     0    -7  
     0     0     0     0     1     4
```

Cómo llegamos a eso paso a paso:

Intercambiamos el renglón 1 y con el 2

```
aux=A(1,:);  
A(1,:)=A(2,:);  
A(2,:)=aux
```

```
A =  
     3    -7     8    -5     8     9  
     0     3    -6     6     4    -5  
     3    -9    12    -9     6    15
```

Ahora debemos hacer un 0 en el tercer renglón

```
A(3,:)=A(3,:)-A(1,:)
```

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

## Manipulación de matrices

- diag: crea un vector vertical con las entradas de la diagonal de la matriz

- `tril`: crea una matriz triangular inferior correspondiente a la matriz
- `triu`: crea una matriz triangular superior correspondiente a la matriz
- `'`: crea la matriz transpuesta de la matriz

Recuerde que estas funciones no modifican la matriz original.