



## VECTORES

- Escribiendo un vector: `>>a = [1 2 3 4 5 6 9 8 7]`
- Creando un vector con elementos entre 0 y 20 igualmente espaciados en incrementos de 2 (frecuentemente usado para crear un vector tiempo): `>>t = 0:2:20`
- Aumentando 2 a cada elemento en el vector a: `>>b=a+2`
- Sumando dos vectores: `>>c=a+b`

## FUNCIONES

- MATLAB incluye muchas funciones estándares, tales como `sin`, `cos`, `log`, `exp`, entre otras:  
`>>sin(pi/4)`
- Para determinar el uso de cualquier función, tipear `>> help [nombre de la funcion en ingles]`

## PLOTEOS

- Suponiendo que se quiere plotear una onda senoidal como función del tiempo. Primero construir un vector tiempo y luego calcular los valores de la función seno para cada tiempo.  
`>> t=0:0.25:7; y = sin(t); plot(t,y)`

## POLINOMIOS

- En MATLAB un polinomio se representa como un vector. Por ejemplo, sea el siguiente polinomio:  $s^4 + 3s^3 - 15s^2 - 2s + 9$ . En MATLAB se escribe el siguiente vector: `>> x=[1 3 -15 -2 9]`
- El valor de un polinomio se puede evaluar con la función `polyval`. Por ejemplo, encontrando el valor del polinomio antes descrito en  $s = 2$ : `>> z=polyval([1 3 -15 -2 9],2)`
- Para encontrar las raíces se usa: `>>roots([1 3 -15 -2 9])`
- Para multiplicar dos polinomios se usa `conv`, así: `<< x = [1 2]; y = [1 4 8]; z = conv(x,y)`
- El dividir dos polinomios se realiza con `deconv`. Dividiendo `z` por `y` se obtiene con:  
`<< [xx, R] = deconv(z,y)`

## MATRICES

- Escribir matrices en MATLAB es lo mismo que escribir vectores, excepto que cada fila de elementos es separado por un punto y coma (;): `<< B = [1 2 3 4;5 6 7 8;9 10 11 12]`
- La transpuesta de una matriz se puede obtener con: `>> C = B'`
- Para multiplicar dos matrices B y C, recuerde que el orden de multiplicación importa: `>> D = B * C`
- Otra forma de manipular matrices es que se puede multiplicar los elementos correspondientes de dos matrices usando el operador `.*` (las matrices deben ser del mismo tamaño):  
`>> E = [1 2;3 4], F = [2 3;4 5], G = E .* F`
- Si se tiene una matriz cuadrada, como E se puede multiplicar a la matriz por si misma tantas veces como se quiere usando una potencia: `<< E^3`

- Si se desea elevar al cubo cada elemento de la matriz, usar elevar al cubo elemento por elemento: `<< E.^3`
- La inversa de una matriz: `<< X = inv(E)` o sus autovalores: `<< eig (E)`

## **IMPRESIÓN**

- Para imprimir una grafica o un archivo-m, seleccionar “Print” del menu “File” en la ventana de la grafica o el archivo-m, y luego presione “Enter”.
- Existen otras formas de imprimir...

## **USANDO ARCHIVOS-M EN MATLAB**

- Asegurarse de guardar el archivo con una extensión “.m”
- Se puede escribir el comando directamente en MATLAB, o juntar todos los comandos que se puedan necesitar juntos en un archivo-m, y solo correr el archivo. Si todos los archivos-m estan en el mismo directorio en el que se está corriendo MATLAB no habrá problema para encontrarlos, de otra forma ubicarse en la carpeta donde se encuentran los archivos-m

## **OBTENIENDO AYUDA EN MATLAB**

- Escribir: `>> help [nombre del comando]`