

PRÁCTICA 1: VECTORES Y MATLAB

JESÚS GAGO E IGNACIO OJEDA

INTRODUCCIÓN.

Esta y todas las demás prácticas están pensadas para ser trabajadas delante de un ordenador con MATLAB instalado, y no para ser leídas como una novela. En vez de eso, cada vez que se presente un comando de MATLAB, se debe introducir el comando, pulsar la tecla “Enter” para ejecutarlo y ver el resultado. Más aún, se desea que se verifique el resultado. Asegúrese de que se comprende perfectamente lo que se obtiene antes de continuar con la lectura.

Aunque MATLAB es un entorno que trabaja con matrices, en esta práctica se aprenderá cómo introducir vectores por filas o por columnas y a manejar algunas operaciones con vectores.

Prerrequisitos: ninguno.

1. VECTORES FILA.

La introducción de vectores fila en MATLAB es muy fácil. Introdúzcase el siguiente comando en la pantalla de MATLAB¹

```
>> v=[1 2 3]
```

Hay una serie de ideas a destacar en este comando. Para introducir un vector, se escribe una apertura de corchete, los elementos del vector separados por espacios y un cierre de corchete. Se pueden usar también comas para delimitar las componentes del vector

```
>> v=[1,2,3]
```

El signo = es el operador de asignación de MATLAB. Se usa este operador para asignar valores a variables. Para comprobar que el vector fila [1,2,3] ha sido asignado a la variable v introdúzcase el siguiente comando en el indicador de MATLAB.

```
>> v
```

¹El símbolo >> es el indicador de MATLAB. Se debe introducir lo que aparece tras el indicador. Entonces se pulsa la tecla “Enter” para ejecutar el comando.

Rangos.

Algunas veces es necesario introducir un vector con componentes a intervalos regulares. Esto se realiza fácilmente con **MATLAB** con la estructura `inicio:incremento:fin`. Si no se proporciona un incremento, **MATLAB** asume que es 1.

```
>> x1=0:10
```

Se puede seleccionar el propio incremento.

```
>> x2=0:2:10
```

Se puede ir incluso hacia atrás.

```
>> x3=10:-2:1
```

O se le puede echar imaginación.

```
>> x4=0:pi/2:2*pi
```

Hay veces, sobre todo cuando hay que pintar funciones, que se precisan un gran número de componentes en un vector.

```
>> x=0:.1:10
```

Elimina la salida.

Se puede suprimir la salida de un comando de **MATLAB** añadiendo un punto y coma.

```
>> x=0:.1:10;
```

Es muy útil cuando la salida es muy grande y no se desea verla.

Espacio de trabajo de MATLAB..

Es posible obtener una lista de las variables en el espacio de trabajo en cualquier momento mediante el comando

```
>> who
```

Se puede obtener incluso más información acerca de las variables con

```
>> whos
```

Se elimina la asignación hecha a una variable con

```
>> clear x
>> who
```

Obsérvese que también se da el tamaño de cada variable. Es posible mantener una ventana con la lista de variables usadas y su tamaño. Para ello, en la barra superior selecciónese el menú **View** y actívese la opción **Workspace**.

Se puede obtener el tamaño de un vector **v** con el comando

```
>> size(v)
```

La información que devuelve indica que el vector **v** tiene 1 fila y 3 columnas. Aunque se puede entender al vector **v** como una matriz con 1 fila y 3 columnas, también se puede entender como un vector fila de longitud 3. Por ejemplo, pruébese el siguiente comando:

```
>> length(v)
```

2. VECTORES COLUMNA.

Es también fácil escribir vectores columna en MATLAB. Introdúzcase el siguiente comando en el indicador.

```
>> w=[4;5;6]
```

Observe que los símbolos de punto y coma delimitan las filas de un vector columna. Pruébense los siguientes comandos.

```
>> w
>> who
>> whos
>> size(w)
```

El resultado indica que el vector **w** tiene 3 filas y 1 columna. Aunque se puede ver al vector **w** como una matriz de 3 filas y 1 columna, también es posible pensar en él como un vector columna de longitud 3. Pruébese el siguiente comando.

```
>> length(w)
```

Transposición.

El operador en MATLAB para transponer es el apóstrofe simple **'**. Se puede cambiar así un vector fila a un vector columna.

```
>> y=(1:10)'
```

O un vector columna a un vector fila.

```
>> y=y'
```

Indexado de vectores.

Una vez que se ha definido un vector, es posible acceder fácilmente a cada una de sus componentes con los comandos de MATLAB. Por ejemplo, introdúzcase el siguiente vector.

```
>> x=[10,13,19,23,27,31,39,43,51]
```

Ahora pruébense los siguientes comandos.

```
>> x(2)
```

```
>> x(7)
```

Se puede cambiar fácilmente el contenido de una componente.

```
>> x(6)=100
```

Se puede también acceder a un rango de elementos

```
>> x([1,3,5])
```

```
>> x(1:3)
```

```
>> x(1:2:length(x))
```

3. OPERACIONES CON VECTORES.

Un gran número de operaciones en las que intervienen vectores y escalares se pueden ejecutar con MATLAB.

Operaciones entre vector y escalar.

Las operaciones entre escalares y vectores son directas. Desde el punto de vista teórico, no se puede sumar un escalar a un vector. Sin embargo, MATLAB sí lo permite. Por ejemplo, si y es un vector, el comando $y+2$ añadirá 2 a cada componente del vector. Estúdiense las salidas de los siguientes comandos.

```
>> y=1:5
```

```
>> y+2
```

```
>> y-2
```

```
>> 2*y
```

```
>> y/2
```

Por supuesto, estas operaciones son igualmente válidas para vectores columna.

```
>> w=(1:3:20)'  
>> w+3  
>> w-11  
>> .1*w  
>> w/10
```

Operaciones entre vectores.

En primer lugar, considérense los siguientes vectores.

```
>> a=1:3  
>> b=4:6
```

La adición y sustracción de vectores es natural y fácil. Introdúzcanse los siguientes comandos.²

```
>> a,b,a+b  
>> a,b,a-b
```

De nuevo, estas operaciones son válidas para vectores columna.

```
>> a=(1:3)',b=(4:6)'  
>> a+b,a-b
```

Sin embargo, se pueden obtener resultados no esperados si no se recuerda que MATLAB es un entorno que trabaja con matrices.

```
>> a,b,a*b
```

El último comando devuelve un error porque `*` es el símbolo de MATLAB para la multiplicación de matrices, y en este caso hay un problema de compatibilidad entre los ordenes de las “matrices” a y b . También pueden ocurrir errores si se intenta añadir vectores de diferente tamaño.

```
>> a=1:3,b=4:7,a+b
```

²Como no aparece punto y coma que suprima la salida, el comando `a,b,a+b` mostrará primero el vector a , luego el vector b y por último el $a+b$

Operaciones con componentes.

Para multiplicar los vectores **a** y **b** componente a componente, ejecútase el siguiente comando de MATLAB.

```
>> a=(1:3)',b=(4:6)'
>> a,b,a.*b
```

El símbolo `.*` es el operador de MATLAB para la multiplicación elemento a elemento. La salida se calcula multiplicando las primeras componentes de los vectores **a** y **b**, a continuación las segundas componentes, etc. El operador de MATLAB para la división componente a componente es `./`

```
>> a,b,a./b
```

Para elevar cada componente de un vector a una potencia, úsese `.^`

```
>> a,a.^2
```

Expresiones más complicadas.

Con un poco de práctica se aprenderá como evaluar expresiones más complejas. Supongamos, por ejemplo, para evaluar la expresión $x^2 - 2x - 3$ para valores de x entre 1 y 10, con incremento de 1 escríbase

```
>> x=1:10
>> y=x.^2-2*x-3
```

Supóngase ahora que se quiere evaluar la expresión $\sin(x)/x$ para valores de x entre -1 y 1 con incrementos de $0,1$ unidades.³

```
>> x=-1:.1:1
>> y=sin(x)./x
```

Los operadores por componentes también funcionan con vectores columna.

```
>> xdata=(1:10)'
>> xdata.^2
```

³Escribiendo `help elfun` se obtiene una lista de las funciones elementales de MATLAB.

EJERCICIOS.

- Introduce el disco con tu nombre en la disquetera del ordenador.
- Escribe

```
>> clear all
>> diary a:\practica01.txt
>> % Práctica 01 de NOMBRE APELLIDOS
```

donde pone NOMBRE APELLIDOS debes escribir tu nombre y apellidos en mayúsculas.

Como acabas de comprobar, el símbolo % sirve para introducir comentarios que no serán evaluados por MATLAB. De este modo puedes responder aquellos ejercicios que requieran alguna explicación.

Ejercicio 1.1. Escribe el comando MATLAB que genera cada uno de los siguientes vectores.

1. $\begin{pmatrix} 1 \\ 2 \\ -3 \end{pmatrix}$.

2. $(1, 2, -1, 3)$.

3. Un vector columna que contenga los números impares entre 1 y 1000.

4. Un vector fila que contenga los números pares entre 2 y 1000.

Ejercicio 1.2. Si $x=0:2:20$, escribe el comando de MATLAB que eleva al cuadrado cada componente de x .

Ejercicio 1.3. Si $x=[0,1,4,9,16,25]$, escribe el comando MATLAB que calcula la raíz cuadrada de cada componente de x .

Ejercicio 1.4. Si $x=0:.1:1$, escribe el comando de MATLAB que eleva cada componente de x a $2/3$.

Ejercicio 1.5. Si $x=0:\pi/2:2\pi$, escribe el comando MATLAB que calcula el coseno de cada componente de x .

Ejercicio 1.6. Si $x=-1:.1:1$, escribe el comando MATLAB que calcula el arcoseno de cada componente de x .

Ejercicio 1.7. Si $x=\text{linspace}(0,2\pi,1000)$, ¿cuál es la entrada 50 de x ? ¿Cuál es la longitud de x ?

Ejercicio 1.8. Si $k=0:100$, ¿cuál es la entrada número 12 de $y=0.5.^k$?

- Escribe `diary off`