Práctica 1

Introducción a Matlab. Vectores y Matrices.

1.1. Introducción a Matlab.

Matlab es un programa que permite realizar cálculos (numéricos y simbólicos), representaciones gráficas, programación de algoritmos ....

Al trabajar en Matlab, hay que tener en cuenta, que el programa distingue las mayúsculas de las minúsculas, es decir, que no es lo mismo escribir $>> x = 1$, que $>> X = 1$, considerará x y X como dos cosas diferentes.

Al escribir en Matlab, las líneas que terminen en ; se ejecutarán pero no mostrará el resultado, es decir, si escribimos $>> x = 2 * 3 + 1;$, no nos mostrará el resultado, aunque si posteriomente escribimos $>> x$, nos dará como resultado 7.

El comando $>>$ *clear all* borra todas las variables con las que se ha trabajado hasta el momento. Si escribimos $>>$ *clear* x, borrará sólo la variable x.

El comando de ayuda en Matlab, es el $>>$ *help*, si escribimos dicho comando seguido del nombre de otro comando o función, nos proporcionará información sobre ello, por ejemplo, al teclear $>>$ *help cos* nos proporciona información sobre cómo funciona la función coseno en matlab, y una lista de funciones relacionadas.

Las operaciones básicas en Matlab son:

- La suma y la diferencia: $+$ y $-$. $3 + 5{,}3 - x$.
- El producto y el cociente: $*$ y $/$. $3 * 2$, $4/3$.
- Las potencias: ^. 3^(1/2).

Además, Matlab tiene predefinidas algunas variables y funciones:

- $sqrt(x)$ nos da la raiz cuadrada de x.
- $sin(x)$, $cos(x)$, $tan(x)$, ... nos da las razones trigonométricas del valor que se introduzca.
- $exp(x)$, nos da e^x. El número e no está definido previamente, por lo que habría que escribir $exp(1)$ para obtenerlo.
- $log(x)$ nos da el logaritmo neperiano de x, para calcular el $log_a(x)$, hay que aplicar las transformaciones habituales: $log_a(x) = \frac{ln(x)}{ln(a)}$.

- $abs(x)$ nos da el valor absoluto de x.
- pi es el número π.
- Inf es el ∞.
- NaN indica indeterminación.
- eps nos da el número más pequeño que reconoce Matlab.

Ejercicio 1.1.1.– Realizar las siguientes operaciones:

1. $\sqrt{3^2+1} - sen(\pi)$
2. $e^{2^2+1} + \frac{3}{2}$.
3. $log_3(9) + ln(e) + log_{10}(0{,}1)$.
4. $\dfrac{3^2+5}{\frac{1}{2}+2}$.
5. $\sqrt[5]{\sqrt{3}+2}$.

1.2. Vectores.

Introducir vectores en Matlab es muy sencillo, basta con escribir las componentes entre corchetes y separadas por un espacio o coma:

$$>> x = [1,2,3];\ \text{ó} >> x = [1\ 2\ 3];$$

Los vectores así definidos, son vectores fila, si deseamos definir un vector columna, habrá que separar los elementos mediante ;.

El comando $>> length(x)$, nos dirá cuántas componentes tiene el vector x, en este caso, 3.

Otras formas de definir vectores son:

- $>> x = 1:3:11$ nos crearía un vector que empezaría en 1, su segunda componente sería $1+3$, la tercera $4+3$, la cuarta $7+3$, y no tendría quinta porque $10+3 > 11$. Es decir, nos daría el vector $x = [1,4,7,10]$. Si escribimos $>> x = 1:5$, nos dará un vector que empieza en 1, termina en 5 y salta de 1 en 1, es decir, $x = [1,2,3,4,5]$.

$$>> x = 5:2:13;\quad \text{es el vector } x = [5,7,9,11,13]$$

- $>> x = \text{linspace}(0,10,6)$ nos da un vector que empieza en 0, termina en 10 y tiene 6 componentes. $x = [0,2,4,6,8,10]$.

Ejercicio 1.2.2.–

1. Definir un vector con 15 componentes que empiece en 3 y acabe en 9.
2. Definir un vector que empiece en 15, acabe en 20 y sus componentes vayan de 3 en 3.

Para seleccionar algunos elementos de un vector, se procede de la siguiente manera:

- Si tenemos el vector $>> x = 1:2:10$, y queremos los elementos primero cuarto y segundo, en ese orden, escribiremos $>> y = x([1,4,2])$.
- Considerando el vector anterior, si deseamos las componentes de la segunda a la quinta, pero saltándonas una de cada dos, podríamos escribir $x(2:2:5)$.
- Al igual que al definir un vector, si en el ejemplo anterior,eliminamos la componente de en medio $(:2)$, y dejamos $x(2:5)$, nos mostrará las componentes segunda, tercera, cuarta y quinta del vector.

Ejercicio 1.2.3.– Dado el vector $>> x = 1:10;$.

1. Definir un vector que tenga las componentes primera, tercera, septima y quinta, y en ese orden.
2. Definir un vector que empiece en la sexta componente, acabe en la décima y se salte dos de cada tres.
3. Definir un vector que tenga las componentes de la tercera a la sexta.

1.2.1. Operaciones con vectores.

Las operaciones $+, -, *, /$, y $\wedge$ entre escalares y vectores, se realizan elemento a elemento del vector (con la operación potencia, suponemos el vector elevado a una determinada potencia, es decir, si x es un vector, y k un escalar, la operación sería x^k).
Por ejemplo, si tengo el vector x, y realizo la operación $x+2$, se sumará 2 a cada elemento de dicho vector x.

Si dos vectores tienen la misma longitud y orientación (vertical u horizontal), las operaciones elemento a elemento se realizan con los comandos $+, -, .*, ./, .\wedge$.

Por ejemplo, dados los vectores $x = [1,2,3]$, e $y = [4,5,6]$:

$x + y = [1+4, 2+5, 3+6]$

$x.*y = [1 \cdot 4, 2 \cdot 5, 3 \cdot 6]$

Para calcular el vector traspuesto a uno dado, se usa el comando $'$ de Matlab (a la derecha del 0 en el teclado convencional).

Si trabajamos con vectores en campo complejo, la traspuesta no sólo traspone el vector, si no que además, da el conjugado de cada elemento, en este caso, si no queremos que nos de los elementos conjugados, hay que poner un punto delante del comando.

Para realizar el producto escalar entre dos vectores, hay que utilizar el comando que realiza el producto matricial ($*$, sin el punto delante), y tener en cuenta que un vector debe ser fila, y otro columna.

Ejercicio 1.2.1.–

1. Crear un vector x que tenga 10 componentes, que empiece en 3 y acabe en 12. Multiplicar este vector por 2.

2. Crear un nuevo vector y que tenga por componentes los números del 3 al 12. Calcular el logaritmo en base 10 de y.

3. Realizar el producto de ambos vectores elemento a elemento.

4. Realizar el producto escalar entre ambos vectores.

1.3. Variables.Resolución de ecuaciones.

Si vamos a asignar a una variable x un determinado valor, dicho valor se asocia usando $=$:

$>> x = 3\text{^}2 + 6/2$
Al escribir esta expresión y pulsar *enter*, Matlab automáticamente nos mostrará:

$x =$

12
Si por el contrario, al terminar la expresión ponemos un ;, no nos mostrará el resultado, aunque sí almacenará la variable x, pudiendo solicitarla con posterioridad:

$>> x = 4 + 6 * 5 - sqrt(2);$
$>> y = x + 5$
$y = 32{,}5858$

Si deseamos que el resultado final nos lo de con más decimales, debemos usar el comando $format\ long$ (15 dígitos), si por el contrario, deseamos que tenga menos decimales, usaremos $format\ short$ (5 dígitos).

Para resolver ecuaciones algebraicas, Matlab cuenta con el comando *solve*. Si queremos obtener las soluciones de la ecuación $x^4 - 3x^3 + 2x^2 - 1 = 0$, escribiremos:

$>> solve('x\text{^}4 - 3 * x\text{^}3 + 2 * x\text{^}2 - 1 = 0')$
Y nos dará:
$ans =$

$$\begin{array}{r} 2{,}178724176105221792565668718459 9 \\ -0{,}512876396864094813769908832208 16 \\ 0{,}670769076539605511266903741704 25 * i \\ 0{,}667076110379436510602120056874 14 \end{array} \quad \begin{array}{r} \\ \\ +0{,}667076110379436510602120056874 14 \\ -0{,}670769076539605511266903741704 25 * i \end{array}$$

Obsérvese que la solución la ha asignado a una variable llamada ans, es lo que ocurre por defecto, si no se le indica lo contrario.

Para reducir el número de decimales visto, se puede utilizar el comando vpa, los argumentos de dicho comando serán, la variable o número cuyos decimales se quieren acortar, y el número de dígitos que se desean obtener. Por ejemplo, si en el caso anterior, queremos que nos de la solución con 3 dígitos, escribiremos:

$>> vpa(ans, 3)$

Y nos dará como resultado:
$ans =$

$$\begin{array}{r} 2{,}18 \\ -0{,}513 \\ 0{,}671 * i + 0{,}667 \\ 0{,}667 - 0{,}671 * i \end{array}$$

El comando solve, también resuleve sistemas de ecuaciones lineales, aunque para observar la solución, hay que asignar la solución a un vector cuyas componentes sean las incógnitas. Si queremos resolver el sistema de ecuaciones:

$$\left.\begin{array}{r} 2x + y - z = 3 \\ 3y - 2z = 1 \\ x - y = 0 \end{array}\right\}$$

Escribiremos:

$>> [x, y, z] = solve('2 * x + y - z = 3', '3 * y - 2 * z = 1', 'x - y = 0')$

Y el resultado saldrá:

$x =$

$5/3$

$y =$

$5/3$

$z =$

2

Por último, el comando *solve*, también puede utilizarse para despejar alguna variable de una ecuación. Por ejemplo, si queremos despejar y en la primera de las ecuaciones anteriores:

$>> solve('2 * x + y - z = 3','y')$

El resultado será:

$ans =$

$z - 2 * x + 3$

Ejercicio 1.3.2.– Dar los resultados con un máximo de 5 dígitos.

1. Resolver la siguiente ecuación $x^3 - 3x^2 + 2x + 3 = 0$.
2. resolver el siguiente sistema de ecuaciones.

$$\left.\begin{aligned} x^2 + 2x - y = 0 \\ x - 2y = 3 \end{aligned}\right\}$$

3. Calcular las raíces de una ecuación de segundo grado.