Máster en Materiales y Sistemas Sensores para Tecnologías Medioambientales (Erasmus Mundus)

PRÁCTICAS DE CÁLCULO NUMÉRICO

Damián Ginestar Peiró

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DEL DISEÑO

UNIVERSIDAD POLITÉCNICA DE VALENCIA

Práctica 1

Introducción al programa MatLab

1.1. Preguntas más frecuentes

¿ Qué es el MatLab y cuáles son sus aplicaciones?

El MatLab es un sistema interactivo orientado al cálculo matricial. Su nombre es una abreviatura de **Mat**rix **Lab**oratory.

Debido a su versatibilidad permite ser utilizado en multitud de aplicaciones de tipo científico o tecnológico como, por ejemplo, en el campo del procesamiento de la señal o de la imagen, en la simulación de sistemas dinámicos y teoría de control, en el estudio de redes neuronales, etc.

¿ Existe algún comando del MatLab que permita trabajar con comandos o ficheros externos al MatLab?

Sí, para trabajar con funciones definidas por el usuario se puede utilizar el comando `path` que permite ampliar el árbol de directorios donde MatLab busca posibles funciones construidas por el usuario. Para ejecutar comandos o ficheros del DOS se utiliza el signo de admiración **!** seguido del nombre del comando.

¿Distingue MatLab entre mayúsculas y minúsculas?

Hay que tener en cuenta que MatLab distingue entre mayúsculas y minúsculas en los nombres de los comandos, funciones y variables, por lo tanto, hay que tener cuidado en escribirlas correctamente.

¿Se puede guardar la sesión de trabajo en un fichero?

El comando `diary` permite guardar el texto de la sesión. La pauta a seguir es la palabra `diary` seguida de la unidad donde se pretende almacenar la información y el nombre del fichero.txt. Para añadir texto le indicaremos donde empieza el parrafo que queremos guardar con el comando `diary on` y donde termina con el comando `diary off`.

El comando `save` guarda las variables en un fichero denominado **matlab.mat** y estas variables se cargan con el comando `load`.

¿Cómo se utiliza la ayuda?

El comando de ayuda del MatLab se denomina `help`. Se ejecuta seguido del nombre de la función sobre la que se quiere obtener ayuda. Además, si se desea buscar toda la información sobre algún tema utilizaremos el comando `lookfor` seguido de la palabra de la que se pretende obtener la información. En versiones superiores a la 5.2 aparecen otros comandos de ayuda como pueden ser el `helpwin`. Tecleando esta palabra se abre una ventana donde podemos encontrar toda la ayuda clasificada por materias, eligiendo la que se desee y pulsando **ENTER** encontraremos la respuesta buscada.

Con el comando `helpdesk`, se abre una ventana de ayuda con el navegador web que se tenga definido por defecto.

¿Cómo se puede salir del MatLab?

El comando `quit` permite abandonar el MatLab.

¿Cómo se puede detener algún cálculo, gráfico o impresión del MatLab sin salir del paquete?

Con el comando **CTRL-C** o con **CTRL-BREAK.**

¿Qué sucede si una instrucción no cabe en una línea?

Las instrucciones de MatLab se terminan pulsando **ENTER**. Si una instrucción no cabe en una línea, se escriben 3 puntos o más y se puede continuar en la línea siguiente.

¿Cómo se hacen los comentarios?

Los comentarios se hacen con el carácter %. Para que MatLab no muestre el resultado de una instrucción ésta debe terminar con ; .

1.2. Operaciones elementales, variables y constantes

Con el MatLab se puede realizar cualquier operación que podríamos hacer con una calculadora. Cada una de estas operaciones por defecto se guarda en la variable `ans`. Si nosotros no queremos trabajar con esta variable, antes de realizar cualquier operación, deberemos hacer una asignación a la variable donde pretendemos que se guarde el resultado. El proceso consiste en escribir el nombre de la variable seguida del signo = y de la operación que pretendemos realizar. Cada vez que escribamos la variable y apretemos **ENTER**, el programa devolverá su valor. El nombre de la variable se puede utilizar tantas veces como deseemos, pero hay que recordar que guardará sólo el último valor.

Al igual que otros paquetes matemáticos el resultado que obtenemos en pantalla se puede visualizar en diferentes formatos. Si no se indica lo contrario, por defecto se obtienen cuatro cifras decimales. Con el comando `format long` aparecen más cifras decimales. Para volver a su formato estándar escribiremos `format short`. Si se pretende obtener el resultado con un formato racional escribiremos `format rat`.

Si se desea conocer las variables que están activas hasta este momento utilizaremos el comando `who`, si además queremos saber de qué tipo son y su tamaño se utiliza el comando `whos`. Si queremos borrar alguna variable teclearemos el comando `clear` seguido del nombre de la variable que deseamos borrar. Utilizando sólo el comando `clear` se borran todas las variables, que se tienen asignadas en la memoria.

El MatLab tiene una lista de las constantes que frecuentemente aparecen en problemas matemáticos o técnicos, por ejemplo el número **pi**, el número **e** que la escribiremos como **exp**(1) , la unidad imaginaria que la podremos escribir como `i` o `j`, etc.

Así si, por ejemplo, escribimos

```
(2+3*i)+(5-2*i)
```

obtendremos como resultado

```
7+1i
```

Si escribimos

```
(2+3*i)*(5-2*i)
```

o bien

```
(2+3*i)/(5-2*i)
```

obtenemos respectivamente

```
    16+11i
0.1379+0.6552i
```

Como vemos, MatLab simplifica las expresiones con números complejos hasta obtener la forma binómica del número complejo resultado. Por otra parte, el MatLab dispone de algunas funciones específicas para operar con complejos como son las siguientes.

`abs( )`	→	Calcula el módulo del número complejo.
`angle( )`	→	Calcula la fase, en radianes del número complejo.
`real( )`	→	Calcula la parte real del número complejo.
`imag( )`	→	Calcula la parte imaginaria del número complejo.
`conj( )`	→	Calcula el conjugado del número complejo.

Veamos algunos ejemplos

`abs(2+3*i)`	=	3.6056
`angle(2+3*i)`	=	0.9828
`real(2+3*i)`	=	2
`imag(2+3*i)`	=	3
`conj(2+3*i)`	=	2-3i

1.3. Funciones

Al igual que otros paquetes matemáticos, MatLab dispone de un catálogo completo con las funciones más utilizadas. Siempre podremos buscar información de dichas funciones con la ayuda. A modo de ejemplo, a continuación daremos una lista de algunas de estas funciones:

`sqrt( )`	$\rightarrow$	raíz cuadrada.
`round( )`	$\rightarrow$	redondeo entero más cercano.
`sign( )`	$\rightarrow$	función signo.
`exp( )`	$\rightarrow$	exponencial.
`log( )`	$\rightarrow$	logarítmo neperiano.
`log10( )`	$\rightarrow$	logarítmo decimal.
`sin( )`	$\rightarrow$	seno.
`cos( )`	$\rightarrow$	coseno.
`tan( )`	$\rightarrow$	tangente.
`asin( )`	$\rightarrow$	arcoseno.
`acos( )`	$\rightarrow$	arcocoseno.
`atan( )`	$\rightarrow$	arcotangente.

Las funciones hiperbólicas

`sinh( )`	$\rightarrow$	seno hiperbólico.
`cosh( )`	$\rightarrow$	coseno hiperbólico.
`tanh( )`	$\rightarrow$	tangente hiperbólica.
`asinh( )`	$\rightarrow$	arcoseno hiperbólico.
`acosh( )`	$\rightarrow$	arcocoseno hiperbólico.
`atanh( )`	$\rightarrow$	arcotangente hiperbólica.

etc.

1.4. Matrices. Operaciones con matrices

La estructura matricial es la forma natural de trabajar en MatLab. Las matrices 1×1 son escalares, las matrices $1 \times n$ ó $n \times 1$ son vectores filas o columna, respectivamente

Para introducir los elemetos de una matriz se comienza con un corchete. Los elementos de las filas se introducen separándolos entre comas, se indica que finaliza la fila con un punto y coma, finalizaremos el proceso cerrando el corchete.

En el siguiente ejemplo asignamos a la variable `A` una matriz 3×3.

```
A=[1,2,3;4,5,6;7,8,9]
```

Otra forma más visual de introducir los datos de la matriz sin utilizar comas o puntos y coma es separando los elementos de la fila por espacios y escribiendo cada fila en un línea diferente. Así, por ejemplo, podemos escribir

```
B =[1
    2
    3]
```

o

```
C =[1 0 0
    2 0 1
    0 0 2]
```

Una vez que hemos definido estas matrices se puede obtener el elemento i, j de la matriz A escribiendo `A(i,j)`.

Se pueden realizar multiples operaciones con matrices y vectores. Los signos que se utilizan son: para sumar +, para restar -, para multiplicar *, la potenciación se indica con ˆ seguido de número al que se pretende elevar la matriz. Todas las operaciones se llevan a cabo entre dos matrices. Si ello no es posible, MatLab devuelve un mensaje de error.

A parte de las operaciones anteriores, MatLab incorpora la división por la izquierda (\ ) y la división por la derecha (/). Así,

$$\begin{array}{lcl} A\backslash B & \rightarrow & \text{es equivalente a } A^{-1}B. \\ A/B & \rightarrow & \text{es equivalente a } AB^{-1}. \end{array}$$

A parte de este tipo de operaciones podemos representar la traspuesta de una matriz con el signo '. El producto escalar y el producto vectorial de dos vectores con los comandos `dot( )` y `cross( )`, respectivamente.

Cuando se quiere hacer operaciones sobre los elementos de las matrices hay anadirles a las matrices un punto. Así, si escribimos

```
A=[1  2  3]
```

y tratamos de calcular $A\hat{}2$ nos dará un error, ya que el cuadrado de A no está bien definido. Si se pretende obtener una nueva matriz cuyos elementos sean los cuadrados de los elementos de A, podemos escribir

```
A.^2
```

obteniendo como resultado

```
[1 4 9]
```

Este resultado se puede obtener también escribiendo

```
A.*A
```

A parte de acceder a un solo elemento de una matriz, es posible acceder a submatrices de una matriz dada. Por ejemplo, dada

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix},$$

si queremos el elemento (1,1) de la matriz escribiremos `A(1,1)` si queremos las dos primeras filas de la tercera columna, escribiremos `A(1:2,3)` y el resultado será

$$\begin{bmatrix} 3 \\ 6 \end{bmatrix},$$

si queremos todos los elementos de la tercera fila escribiremos `A(3,:)` y el resultado obtenido

$$[7\ 8\ 9],$$

si queremos la submatriz formada por la primera y la tercera columna, escribiremos `A(:,[1,3])` el resultado será

$$\begin{bmatrix} 1 & 3 \\ 4 & 6 \\ 7 & 9 \end{bmatrix}.$$

Por último, si escribimos `A([1,3],[1,3])`, el resultado obtenido será

$$\begin{bmatrix} 1 & 3 \\ 7 & 9 \end{bmatrix}.$$

A parte de estas operaciones existen funciones que nos permiten construir matrices, como por ejemplo:

`eye(n)`	$\rightarrow$	matriz identidad $n \times n$.
`zeros(n,m)`	$\rightarrow$	matriz nula $n \times m$.
`diag(A)`	$\rightarrow$	devuelve un vector con la diagonal de `A`.
`diag(x)`	$\rightarrow$	devuelve una matriz con `x` en la diagonal.
`triu(A)`	$\rightarrow$	devuelve la parte triangular superior de `A`.
`tril(A)`	$\rightarrow$	devuelve la parte triangular inferior de `A`.
`rand(n,m)`	$\rightarrow$	devuelve una matriz generada aleatoriamente.

A parte se pueden construir matrices por bloques, por ejemplo la intrucción

```
B=[zeros(2,3),eye(2);zeros(3,2),eye(3)]
```

devuelve una matriz de la forma

$$B = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} .$$

1.5. Ejercicios

1. *a*) Si $z_1 = 3 + 5i$ y $z_2 = 1 - 2i$. Calcular

   $$|z_1|\,,\; z_1 + z_2,\; z_1 * z_2,\; z_1/z_2\ .$$

   Comprobar que $\arg(z_1/z_2) = \arg(z_1) - \arg(z_2)$.

   b) Calcular parte real, imaginaria, módulo y argumento de $\left(1 + \sqrt(3i)\right)^{1-i}$.

   c) Calcular

   $$(8)^{\frac{1}{5}}\ ,\quad e^7 + \ln(5)\,,\quad \frac{\ln(9) + \operatorname{sen}(\pi/5)}{45 - \cos(23)}\ .$$

2. Dadas

   $$A = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 6 \\ 0 & 1 & 9 \end{pmatrix} \text{ y } b = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix},$$

   calcular

a) $A + A$,

b) $2A$,

c) AA,

d) A^2,

e) $A\backslash b$ y comprobar que el resultado coincide con $A^{-1}b$,

f) b^T/A y comprobar que el resultado coincide con $b^T A^{-1}$,

g) Calcular el producto escalar $b \cdot b$ y el producto vectorial $b \wedge b$.

3. Introducir las siguientes matrices

$$D = \begin{bmatrix} 1 & -3 & 4 \\ 2 & -5 & 7 \\ 0 & -1 & 1 \end{bmatrix}, \; E = \begin{bmatrix} 2 & 4 & 6 \\ 4 & 5 & 6 \\ 3 & 1 & 2 \end{bmatrix}, \; c = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}.$$

a) Calcular $D + E, DE, D^T E$.

b) A patir de las matrices D y E anteriores sacar sus vectores fila, sus vectores columna y una submatriz 2×2 formada por las filas 1 y 2 y las columnas 2 y 3.

c) Resolver el sistema
$$E\, x = c$$

d) Construir una matriz diagonal que contenga la diagonal de la matriz E.

e) Construir las matrices por bloques

$$\left[\begin{array}{c|c} D & E \end{array}\right], \quad \left[\begin{array}{c} D \\ \hline E \end{array}\right].$$

4. Dada la matriz
$$E = \begin{bmatrix} 2 & 4 & 6 \\ 4 & 0 & 6 \\ 3 & 1 & 2 \end{bmatrix}$$
obtener su parte estríctamente triangular inferior (sin la diagonal) y su parte estríctamente triangular superior.