

Guía Rápida MATLAB y Octave

Martin Mellado

1. Introducción

La herramienta CAE MATLAB es un poderoso entorno para álgebra lineal y representación gráfica, disponible para una amplia gama de plataformas. MATLAB es un lenguaje de programación intuitivo acompañado de un entorno de computación orientado a ingeniería. Ha sido desarrollado por la casa Mathworks. El nombre MATLAB viene de “Matrix laboratory” o laboratorio de matrices, de lo que se deduce que este programa es especialmente ventajoso para el trabajo con matrices. De hecho, en MATLAB se tratan de la misma forma escalares, vectores y matrices. Es importante resaltar que MATLAB no es un compilador sino un interprete. La funcionalidad básica de MATLAB puede ampliarse con librerías (*Toolbox*). Las Toolbox son librerías de funciones desarrolladas para un campo específico de aplicación. Existen, por ejemplo, Toolbox de finanzas, optimización, identificación, lógica borrosa, redes neuronales, control adaptativo, robótica, etc. Las funciones en los Toolbox suelen ser escritas en MATLAB con código accesible y modificable por el usuario. En esta práctica no se utilizará ninguna de las librerías disponibles en el área de robótica, sino que se desarrollarán herramientas propias de modelado a partir de funciones comunes de MATLAB. La versión básica de MATLAB contiene más de 500 funciones matemáticas y gráficas, además de otras herramientas gráficas. Existen herramientas adicionales como el compilador de MATLAB, enlazador con C/C++, etc.

Arranque:

Para empezar, cada grupo debe crear su propio directorio en el disco duro (p.ej., C:\MiMATLAB). Una vez arrancado el programa, teclear desde MATLAB: "`cd C:\MiMATLAB`". De esta forma nos aseguramos que el programa trabaja en nuestro directorio. Cuando se invoca un **comando** desde MATLAB, el intérprete busca el fichero **comando.m** en el directorio actual y en los directorios incluidos en su PATH personalizado.

Ayuda:

Utilizando el comando **help** de MATLAB aparece un listado de los posibles comandos por los cuales nos podemos mover (`help comando`) en busca de un comando concreto. Es aconsejable utilizar este comando para recordar la sintaxis de una orden o para ver los parámetros de entrada y salida de un comando, por ejemplo:

```
>>help if
```

```
IF IF statement condition.
The general form of the IF statement is
```

```
IF expression
    statements
ELSEIF expression
    statements
ELSE
    statements
END
```

```
The statements are executed if the real part of the expression
has all non-zero elements. The ELSE and ELSEIF parts are optional.
Zero or more ELSEIF parts can be used as well as nested IF's.
The expression is usually of the form expr rop expr where
rop is ==, <, >, <=, >=, or ~=.
```

```
Example
```

```
if I == J
    A(I,J) = 2;
elseif abs(I-J) == 1
    A(I,J) = -1;
else
```

```
A(I,J) = 0;  
end
```

See also RELOP, ELSE, ELSEIF, END, FOR, WHILE, SWITCH.

```
>>help inv
```

```
INV      Matrix inverse.  
INV(X) is the inverse of the square matrix X.  
A warning message is printed if X is badly scaled or  
nearly singular.
```

See also SLASH, PINV, COND, CONDEST, NNLS, LSCOV.

```
Overloaded methods  
help zpk/inv.m  
help tf/inv.m  
help ss/inv.m  
help lti/inv.m
```

2. Comandos básicos

Los comandos básicos se pueden introducir directamente en la consola de comandos, tras el símbolo de MATLAB >>.

2.1. Operaciones con matrices

Las matrices se introducen en MATLAB abriendo corchetes, tecleando los elementos de la matriz, y cerrando corchetes.

Por ejemplo, para introducir una matriz fila, teclead:

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

MATLAB responderá:

```
a =  
  
     1     2     3
```

Una matriz fila también se puede introducir separando sus elementos por comas:

```
>> b=[4, 5, 6]
```

Para introducir una matriz columna sus elementos tienen que estar separados por “;”. Teclead en MATLAB:

```
>> c=[7;8;9]
```

En MATLAB, el signo “;” (punto y coma) significa “nueva fila” cuando se encuentra entre corchetes.

De esta forma se puede introducir una matriz por ejemplo 3X3:

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

El carácter “;” al final de una línea significa que no se produzca eco, con lo que MATLAB no muestra el resultado del comando.

Probad con:

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

Las operaciones aritméticas con matrices se hacen de la misma forma que con escalares:

```
>> a+b
```

```
ans =
```

```
5      7      9
```

Notad que el programa le asigna por defecto el nombre ans (de “answer”- respuesta en inglés) al resultado de la última operación, a menos que el usuario le asigne otro. De esta forma se puede reutilizar el último resultado en la siguiente operación:

```
>> ans*3
```

Notad igualmente que para que una operación con matrices sea posible, es necesario que las dimensiones de las matrices sean las correspondientes. Por ejemplo:

```
>> a+c
```

dará un error.

De forma similar las matrices se pueden multiplicar. Introducid:

```
>> a *c
```

Probar el mismo producto invirtiendo el orden de los factores. Luego calculad a*b.

Si el usuario se olvida del valor de una variable, basta con teclear su nombre y pulsar “enter” y su valor aparecerá en la pantalla:

```
>> d
```

Se pueden extraer de forma fácil los elementos de una matriz. Teclead:

```
>> d(1)
```

Probad con d(2) y d(4).

También se pueden extraer por fila y columna. Por ejemplo:

```
>> d(2,3)
```

devolverá el elemento de la segunda fila y tercera columna de la matriz d.

Para extraer una fila entera, por ejemplo, la tercera, teclead:

```
>> d(3,:) 
```

El carácter “:” significa *todas*.

Introduce un comando para extraer la primera columna.

También se puede extraer cualquier submatriz de una matriz. Para, por ejemplo, obtener las primeras dos filas y las últimas dos columnas, teclead:

```
>> d(1:2,2:3)
```

Probad el resultado de:

```
>> d(:, [1 3])
```

Se pueden formar matrices a partir de otras matrices:

```
>> e=[a;b;a]
```

Para encontrar la traspuesta de una matriz se utiliza el signo “'” (que se encuentra debajo del punto de interrogación en el teclado).

Probad:

```
>> e'
```

```
>> d'
```

Para encontrar la inversa de una matriz hay que teclear:

```
>> inv(e)
```

```
>> inv(f)
```

Recordad que solamente las matrices cuadradas y no singulares pueden tener matriz inversa.

Para elevar a una potencia se utiliza el signo “^”.

Probad:

```
>> f^2
```

```
>> f*f
```

Para conocer las dimensiones de una matriz se utiliza el comando `size`. Por ejemplo:

```
>> size(f)
```

devuelve el número de filas y de columnas de la matriz `f`.

Probad con `size(a)` y `size(c)`.

Si se quiere saber solo el número de filas, introducid:

```
>> size(a,1)
```

Y para obtener el número de columnas, teclead:

```
>> size(a,2)
```

Probar lo mismo con la matriz `c`.

Por tanto, una de las grandes ventajas de MATLAB es que permite definir matrices y operar con ellas con facilidad, como, por ejemplo:

```
>>A = [1 2 3 4; 5 6 7 8; 9 10 11 12];
>>B = A;
>>C = A + B
C =
     2     4     6     8
    10    12    14    16
    18    20    22    24
>>D = A * B'           // A por B traspuesta
D =
    30    70   110
    70   174   278
   110   278   446
```

2.2. Operadores relacionales, condiciones y bucles

Operadores relacionales

El MATLAB contiene los seis siguientes operadores relacionales:

- Menor que: <
- Mayor que: >
- Mayor o igual: >=
- Menor o igual: <=
- Igual: ==
- Diferente de: ~=

Su función es comparar dos expresiones. Su modo de funcionamiento es simple. Teclead, por ejemplo:

```
>> 3<5
```

Y luego

```
>> 3>5
```

Obviamente, devuelven un 1 cuando la expresión es verdadera y un 0 cuando es falsa.

Las expresiones relacionales se pueden combinar o concatenar mediante los siguientes operadores lógicos:

- AND: &
- OR: |
- NOT: ~

Teclead:

```
>> a(1)<4 & b(2)>1
```

Condiciones:

La forma más simple de condición es:

```
if <expresión relacional>
    instrucciones
end
```

Lo que hace el interprete en este caso es evaluar la expresión relacional. Si es verdadera ejecutará las instrucciones dentro de la condición. Si no es verdadera, seguirá con el programa a partir del **end**.

Teclead:

```
>> if a(3)==3
x=1
end
```

Las condiciones complementarias se pueden concatenar de la siguiente forma:

```
if <expresión relacional1>
    instrucciones1
...
elseif <expresión relacional2>
    instrucciones2
```

```

        ...
    elseif <expresión_relacional3>
        instrucciones3
        ...
    else
        instrucciones4
        ...
    end

```

Teclead una secuencia de líneas de código que hagan lo siguiente:

- asignar a la variable b el valor de 0 si a es menor que 0.
- asignar a la variable b el valor de 1 si a está entre 0 (incluido) y 2 (excluido).
- asignar a la variable b el valor de 2 si a está entre 2 (incluido) y 5 (excluido).
- asignar a la variable b el valor de 3 si a es mayor o igual a 5.

Bucles:

Existen dos tipos de bucles en MATLAB, los bucles **for** y los bucles **while**. En esta guía se explicarán exclusivamente los segundos.

Su formato es el siguiente:

```

while condición
    comandos
    ...
end

```

Los comandos se van ejecutando mientras se cumple la condición.

Ejemplo:

Vamos a hacer un bucle que sume todos los elementos de la matriz "a". Teclead:

```

>> suma=0;
>> i=1;
>> while i < 4
suma= suma+ a(i)
i= i+1;
end

```

3. Ficheros de MATLAB

MATLAB utiliza varios tipos de ficheros de los cuales vamos a ver uno, los ficheros de código que tienen la extensión ".m". Su contenido es una secuencia de comandos de MATLAB como las que hemos utilizado hasta ahora. Se pueden introducir desde el editor del MATLAB (File -> New -> M-file) o desde cualquier editor ASCII.

Abrir un fichero nuevo, teclear en él las líneas de código que utilizamos para sumar los elementos de la matriz a. Guardar el fichero en C:\MiMATLAB (o el nombre que le dimos a nuestro directorio) como "sumar.m".

Teclead en MATLAB sumar y ver el efecto.

No hay absolutamente ninguna diferencia entre ejecutar las instrucciones desde el MATLAB o desde un fichero ".m".

Haced un fichero ".m" que sume todos los elementos de la matriz d.

Recordar que los ficheros *.m se interpretan, no se compilan. Para que el resultado de una línea de código no salga en la pantalla, se debe poner un ";" al final de la misma.

Las líneas de comentario tienen que empezar por "%".

4. Funciones de MATLAB

Se ha visto que las líneas de código de MATLAB se pueden introducir desde un fichero ".m". Sin embargo, existe el inconveniente de que las variables de estos ficheros no están separadas de las del Workspace. Es decir, si hay una variable en el Workspace que se llama igual que otra en un fichero ".m", al modificar una se modifica también la otra. Esto puede crear problemas a medida que hay más código, si se utiliza el código de otros, etc.

La forma de superarlo es mediante las funciones. Estas no son otra cosa que ficheros ".m", con la única diferencia de que comienzan por la línea de código:

```
function [salida1,salida2,...]=nombre(entrada1, entrada2,..)
```

Donde salida1,salida2, ... son las variables de salida, nombre es el nombre de la función y entrada1, entrada2,.. son las variables de entrada de la función.

Al principio del fichero sumar.m que se ha implementado anteriormente añadir la línea:

```
function [suma]=sumar(a)
```

Ejecutas desde el MATLAB:

```
>> sumar(a)
>> sumar(b)
>> s=sumar(c)
```

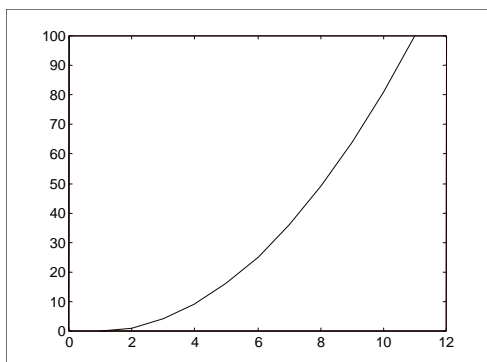
Hacer una función sumprod que devuelva dos valores, el producto y la suma de todos elementos de una matriz mXn.

5. Comandos Gráficos

MATLAB tiene bastantes recursos para visualización. Es posible dibujar directamente en una ventana tanto en 2D como en 3D. Por ejemplo, la siguiente secuencia:

```
>>q = [] ;
>>for t=0:10
q = [q;t*t];
end
>>plot(q)
```

Dibuja la función $q=t^2$ de la forma siguiente:



6. **Octave**

Octave o GNU Octave es un programa libre para realizar cálculos numéricos. Como su nombre indica, es parte del proyecto GNU. Es considerado el equivalente libre de MATLAB. Entre varias características que comparten, se puede destacar que ambos ofrecen un intérprete, permitiendo ejecutar órdenes en modo interactivo. Octave está orientado al análisis numérico.

Octave tiene un intérprete de su propio lenguaje (de sintaxis casi idéntica a MATLAB), y permite una ejecución interactiva o por lotes.

La licencia de Octave es una Licencia pública general de GNU, por lo que puede ser compartido y utilizado libremente.

Octave se puede descargar de <https://www.gnu.org/software/octave/>.