MATLAB.

INTRODUCCIÓN A LAS OPERACIONES CON MATRICES

Luis Vicente PÉREZ ARRIBAS
Departamento de Química Analítica
Facultad de C. Químicas
Universidad Complutense de Madrid

Madrid 2018

INTRODUCCION

Notación utilizada

negrita	Términos importantes
cursiva en negrita	Término nuevos
Negrita con la inicial en mayúscula	Nombres de teclas o del menú
`Constantes`	Comandos, funciones, nombres de fichero y presentación en pantalla
Cursiva	Nombres de ventanas, libros, herramientas, ejemplos y notaciones matemáticas

Iniciación del programa

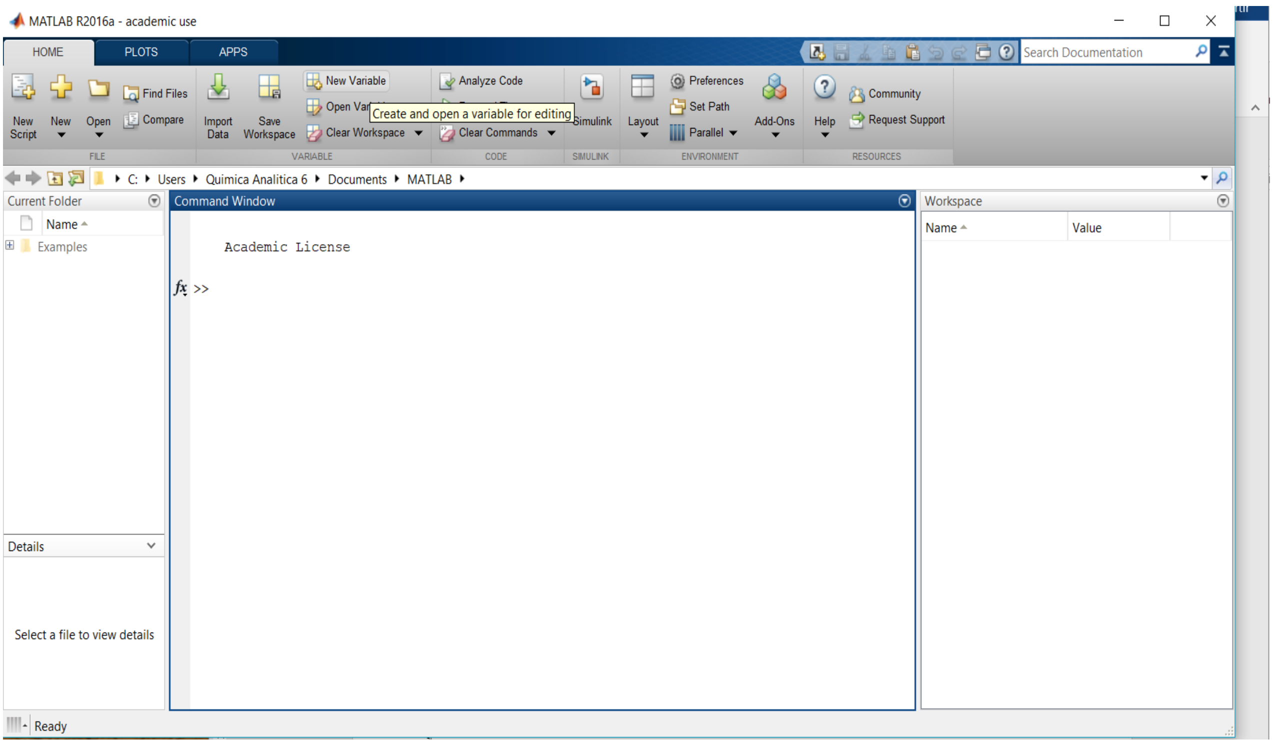
El programa MATLAB se hace correr marcando sobre el icono correspondiente en el escritorio o desde el menú de programas. Una vez que se ha activado y se encuentra corriendo pueden aparecer una o más ventanas en el monitor. De ellas la más importante en la de Comandos (*Command Window*). Una vez que esta ventana está abierta aparecerá en ella la leyenda:

```
Academic License
```

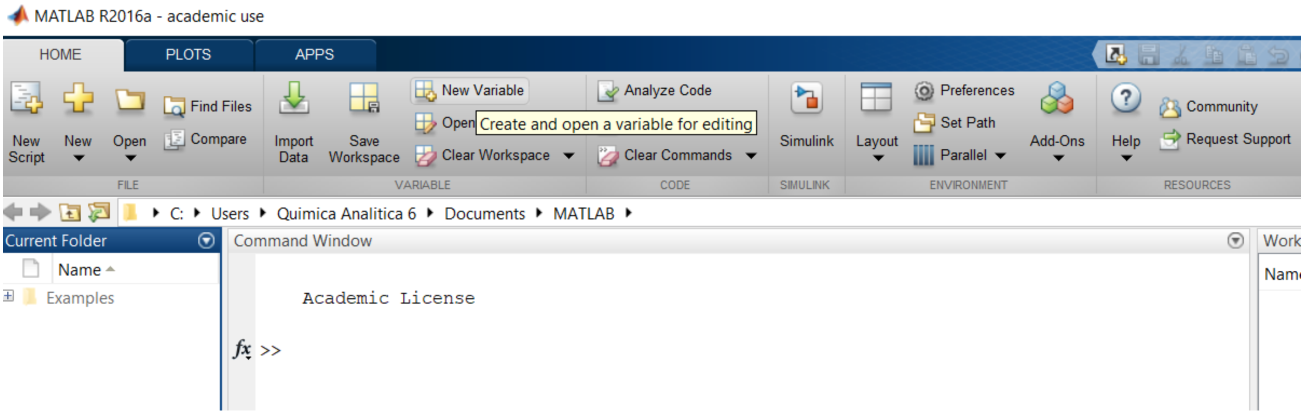
u otra similar, dependiendo del tipo de licencia que se posea, seguido de

fx >>

donde *fx* >> es el "prompt" o línea de comandos de MATLAB. Cuando la ventana *Command* está activa, el cursor aparecerá a la derecha del "prompt", lo que significa que el programa está preparado para contestar cualquier pregunta matemática.



Mediante el "prompt" o línea de comandos de MATLAB se pueden introducir tanto operaciones matemáticas simple como comandos propios de MATLAB que ejecutan operaciones matemáticas más complejas. También se pueden introducir o declarar variables en diferentes formatos. Las variables que van a ser utilizadas en los cálculos matemáticos también pueden ser declaradas e introducidas a través de la opción **New variable**, accesible desde la cinta de opciones HOME.



<u>CÁLCULOS MATEMÁTICOS SIMPLES</u>

MATLAB puede realizar cálculos simples como si de una calculadora se tratara. Si queremos sumar 4+6+2, simplemente habrá que escribir a la derecha de la línea de comandos:

fx >> `4+6+2` y tras presionar **Intro** aparecerá:

```
ans =
     12
```

De igual manera se pueden utilizar otros operadores matemáticos sencillos

fx >> `4*25 - 6*52 + 2*99` **Intro**

```
ans =
     -14
```

Los operadores matemáticos que se pueden utilizar con MATLAB son:

Operación	Símbolo	Ejemplo
Suma, *a+b*	+	5+3
Resta, *a-b*	-	23-12
Multiplicación, *a.b*	*	3.14*0.85
División, a:b	/	56/8
División inversa, b:a=a:b	\	8\56
Exponente, a^b	^	5^2

Estas expresiones son evaluadas por MATLAB leyendo de izquierda a derecha, teniendo la exponencial preferencia sobre la multiplicación, ésta sobre la división y ésta sobre la suma o resta, siendo estas últimas operaciones equivalentes entre sí. Este orden puede ser alterado encerrando entre paréntesis parte de la expresión.

Cuando se va a utilizar una misma variable varias veces, esta se puede definir con un nombre, bien desde la línea de comandos o bien desde la opción **New variable**, de la cinta de opciones HOME.

$f\!x$ `>> borra=4` **Intro**

```
borra =
     4
```

$f\!x$ `>> cuaderno=6` **Intro**

```
cuaderno =
          6
```

Si ahora se quiere utilizar estas variables en operaciones diferentes, simplemente hay que introducirlas por su nombre, p.e.:

$f\!x$ `>>  item=borra+cuaderno` **Intro**

```
item =
     10
```

$f\!x$ `>>  coste=borra*25+cuaderno*52` **Intro**

```
 coste =
     412
```

$f\!x$ `>>  coste_medio=coste/ítem` **Intro**

```
coste_medio =
          41.2000
```

Cuando una variable está formada por más de una palabra, esta deberá estar unida mediante un **Guión bajo largo**. (**_**). MATLAB guardará en memoria todas las variables definidas con anterioridad y se mostrarán en la ventana *Workspace*. Si se desean listar en la ventana *Command Window* se puede teclear el comando `who`. A la hora de nombrar variables hay que tener en cuenta que MATLAB diferencia entre mayúsculas y minúsculas por lo que `item` e `Item`, por ejemplo, son dos variables diferentes. Existen algunos nombres de variables que están reservadas para MATLAB y que no pueden ser definidas por el usuario:

Variable	Valor
`ans`	Resultado de una operación
`pi`	Número pi (3.1416...)
`eps`	El número más pequeño que puede ser añadido a otro para que cambie su valor (2.2204e-016)
`flops`	Contador de operaciones con punto flotante
`inf`	Valor del infinito (∝)
`NaN` o `nan`	Número indeterminado
`i` o `j`	Valor imaginario en un número complejo ($-1^{1/2}$)
`nargin`	Número o función utilizada como argumento de entrada
`nargout`	Número o función utilizada como argumento de salida
`realmin`	Número real positivo más pequeño que puede ser utilizado (2.2251e-308)
`realmax`	Número real positivo más grande que puede ser utilizado (1.7977e+308)

Una vez definidas las variables, estas pueden ser borradas mediante el comando `clear`:

fx >> `clear borra` elimina la variable `borra`

fx >> `clear cuaderno` elimina la variable `cuaderno`

fx >> `clear` borra todas las variables definidas anteriormente

Puntuación e inclusión de comentarios

Se pueden añadir comentarios, con fines aclaratorios, a una línea de cálculo precediendo este con el signo (`%`) sin que MATLAB lo tenga en cuenta:

```
fx >>  borra=4        % número de borradores   Intro
borra =
     4
```

También se pueden colocar en una sola línea varios comandos o variables separadas por comas o punto y coma. Cuando se teclea **Intro** todos los comandos se ejecutan en orden de izquierda a derecha. Los que van seguidos de coma se mostrarán en la pantalla, mientras que los que van seguidos de punto y coma quedarán ocultos:

```
fx >>  borra=4, cuaderno=6; celo=2
borra =
     4
celo =
     2
```

En otras palabras, las comas se utilizan cuando se quiere mostrar el resultado de la operación, el punto y coma cuando no. Esto tiene gran importancia en el cálculo con MATLAB, sobre todo cuando es necesario realizar multitud de cálculos intermedios, ya que, de no utilizarse el punto y coma, se llenaría la pantalla con los resultados que se van produciendo en cada operación.

OPERACIONES CON MATRICES MATEMÁTICAS

Una de las principales aplicaciones de las matrices matemáticas está relacionadas con operaciones de álgebra lineal, como, por ejemplo, la resolución de sistemas de ecuaciones. Supongamos que queremos resolver el sistema siguiente:

$$x + 2y - z = 1$$
$$2x - y + z = 3$$
$$x + y + z = 4$$

Desde el punto de vista del álgebra lineal, este sistema de ecuaciones puede expresarse como un producto de matrices cuyo resultado es otra matriz:

$$\begin{bmatrix} 1 & 2 & -1 \\ 2 & -1 & 1 \\ 1 & 1 & 1 \end{bmatrix} . \begin{bmatrix} x1 \\ x2 \\ x3 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 4 \end{bmatrix}$$

A . x = b

La solución matemática a este sistema de ecuaciones es:

$$\mathbf{x = A^{-1} . b}$$

donde x representa el vector que contiene los resultados, $\mathbf{A^{-1}}$ es la matriz inversa de **A** (matriz de términos dependientes) y **b** es el vector de términos independientes del sistema de ecuaciones

La resolución de este sistema mediante MATLAB se reduce a esta pequeña serie de instrucciones:

```
fx >>  A=[1, 2,-1;2,-1,1;1,1,1]; b=[1;3;4];x=inv(A)*b
 x =
   1.0000
   1.0000
   2.0000
```

El primer resultado de los mostrados corresponde a la incógnita x1, el segundo a incógnita x2 y el tercero a x3. A esta misma solución se podría haber llegado utilizando en operador (\) y sin necesidad de invertir la matiz **A**:

```
fx >>  x=A\b
x =
    1.0000
    1.0000
    2.0000
```

Este mismo operador puede utilizarse para la regresión por mínimos cuadrados de aquellos sistemas de ecuaciones que tienen más ecuaciones que incógnitas (sistemas superdeterminados).

```
fx >>  A=[1, 2,-1;2,-1,1;1,1,1;1,-1,1], b=[1;3;4;2]
A =
    1     2    -1
    2    -1     1
    1     1     1
    1    -1     1

b =
    1
    3
    4
    2
```

```
fx >>  x=A\b
x =
   1.0000
   1.0000
   2.0000
```

mientras que la solución matemática mediante regresión de mínimos cuadrados habría sido:

$$\mathbf{x} = (\mathbf{A}'.\mathbf{A})^{-1}.\mathbf{A}'.\mathbf{b}$$

que por supuesto también podría haberse llevado a cabo mediante MATLAB, operando con matrices inversas y transpuestas, como se indica en el ejemplo siguiente:

```
fx >>  x=inv(A'*A)*A'*b
x =
   1.0000
   1.0000
   2.0000
```

Para operar con una matriz transpuesta en MATLAB, simplemente hay indicar el nombre asignado a la matriz seguido de apóstrofo.

Otra posibilidad que ofrece MATLAB es la de dar una solución a sistemas con más incógnitas que ecuaciones. Estos sistemas tienen infinitas soluciones, pero mediante la función `pinv(A)` se puede obtener la solución de la mínima norma de la matriz:

```
fx >>  A=[1,2,3,4;1,1,2,1;2,3,1,2];b=[1;2;3]; x=pinv(A)*b
    x =
   0.9648
   0.6652
   0.5551
  -0.7401
```

También se puede utilizar el operador (\). En este caso se obtiene la solución con el mínimo número de ceros

```
fx >>  x=A\b
x =
       0
  1.4545
  0.8182
 -1.0909
```

Finalmente, existen algunas funciones para crear o realizar cálculos con matrices especiales.

```
fx >>  rand(4,2)
ans =
  0.9501    0.8913
  0.2311    0.7621
  0.6068    0.4565
  0.4860    0.0185
```

crea una matriz de números aleatorios comprendidos entre 0 y uno y de dimensiones 4 por 2

```
fx >>  eye(4)  % matriz identidad de 4 por 4
ans =
   1     0     0     0
   0     1     0     0
   0     0     1     0
   0     0     0     1
```

```
fx >>  eye(3,2)  % matriz identidad de 3 por 2
ans =
   1     0
   0     1
   0     0
```

En la siguiente tabla se muestran las funciones para crear y operar con matrices especiales:

[ ]	Matriz vacía
compan	Matriz que acompaña al polinomio de coeficientes P.
eye	Matriz identidad
gallery	Diversos test matriciales
hadamard	Matriz de Hadamard
hankel	Matriz de Hankel
hilb	Matriz de Hilbert
invhilb	Matriz inversa de Hilbert
magic	Cuadrado mágico
ones	Matriz de unos
pascal	Matriz triangular de Pascal
rand	Matriz formada por números aleatorios entre 0 y 1
randn	Matriz de números aleatorios distribuidos normalmente con varianza igual a 1
rosser	Test de los valores propios simétricos
toeplitz	Matriz de Toeplitz
vander	Matriz de Vandermonde
wilkinson	Test de los valores propios de Wilkinson
zeros	Matriz de ceros