

CALCULO

GRADO EN INGEN. INFORM. DEL SOFTWARE. 12-13

PRACTICA 1. INTRODUCCIÓN A MATLAB

¿Qué es MATLAB?

MATLAB es la abreviatura de MATrix LABoratory (laboratorio de matrices). Se trata de un software matemático para cálculo científico (aritmético y simbólico) basado en matrices. MATLAB ofrece un entorno de desarrollo integrado (IDE) muy versátil con un lenguaje de programación propio (lenguaje M). MATLAB está disponible para sistemas operativos Windows, Unix y Apple Mac OS X. Entre sus prestaciones básicas se hallan:

- Realizar cálculos aritméticos (como una calculadora).
- Realizar cálculo simbólico (calcular derivadas de funciones, primitivas, etc)
- Realizar gráficos en dos y tres dimensiones
- Programar en un lenguaje no compilado (script).

El entorno de MATLAB para Windows

Si abrimos MATLAB, versión 7 para windows, nos aparecen por defecto 3 ventanas. La ventana de la derecha según miramos (**command window**) es la que usamos para escribir las órdenes. En la de arriba a la izquierda, **Current Directory**, aparecen los ficheros que hay en la carpeta de trabajo. Bajo esta, está oculta una ventana con el título **Workspace**, donde nos aparecen las variables que estamos utilizando y en la de abajo a la izquierda (**Command History**), un historial con las órdenes que hemos escrito. Encima de estas ventanas hay un recuadro (**Current Directory**) donde aparece la carpeta o directorio de trabajo, es decir, el lugar donde MATLAB buscará nuestros programas. Si durante el trabajo perdemos esta estructura de ventanas y la queremos recuperar, basta con pinchar en Desktop – Desktop Layout - Default.

Primeros pasos con MATLAB

Para realizar un cálculo, escribimos en la ventana de comandos a continuación del símbolo `>>` y le damos a la tecla “Enter”.

Ejemplo:

```
>> 5 + 12
ans =
    17
>> a = 5;
>> b = 12;
>> suma = a+b
suma =
    17
```

La orden **clc** sirve para “limpiar” el área de trabajo de la ventana de comandos.

Para obtener ayuda se escribe **help** “orden” ó **helpwin** “orden”.

Ejemplo:

```
>> help clc
```

Para salir de MATLAB se puede escribir **exit** o **quit**

Operaciones aritméticas con MATLAB

Los símbolos utilizados para las operaciones son: +, -, *, /, ^ (suma, resta, multiplicación, división y potenciación respectivamente).

La potenciación tiene el orden de prioridad más alto, seguida de la multiplicación y división con igual prioridad; por último están la suma y la resta con igual prioridad. A igual prioridad las operaciones se realizan de izquierda a derecha. Por supuesto se pueden utilizar paréntesis para alterar esta ordenación; en este caso la evaluación se inicia dentro del paréntesis más interno y se continúa hacia fuera.

Ejemplo: $2^3 + 7 - \frac{8}{4} * 3 + 5 * 3^3 + 2^4 - 7 + \frac{24}{3 * 4}$

```
>> 2^3 + 7 - 8 / 4 * 3 + 5 * 3 ^3 + 2 ^4 -7 + 24 / (3 * 4)
ans =
    155
```

Variables

El nombre de una variable debe comenzar necesariamente por una letra. Puede contener letras, números y el guión de subrayado; no se permiten espacios en blanco. Así, por ejemplo: A7, A_2, a7 son nombres válidos de variables.

A efectos de nombrar variables las mayúsculas son distintas de las minúsculas. Así pues las variables de nombre: area, Area, AREA, ... son distintas.

La orden **who** y la orden **whos** nos dan información sobre las variables definidas en una sesión de trabajo.

Para borrar variables se utiliza la orden **clear** (las borra todas) o, por ejemplo, **clear a b** si deseamos borrar las variables de nombre a y b.

Algunas variables predefinidas en MATLAB son:

ans : variable que, por defecto, contiene los resultados.

pi : contiene el valor del número real π

eps: es el número positivo más pequeño que sumado a 1 genera un número mayor que uno en el ordenador ($2.2204e-016 = 2.2204 \times 10^{-16}$).

Inf o inf: es el infinito

NaN o nan: representa una indeterminación ($0/0, 0 \cdot \infty, \dots$)

i o j : representa la unidad imaginaria $i = j = \sqrt{-1}$

Signos de puntuación, comentarios.

En una misma línea pueden definirse varias variables separadas por comas o por punto y coma. El punto y coma evita (al pulsar intro) la visualización en pantalla.

Ejemplo:

```
>> a = 1; b = 3; c = 7;  
>> a = 1; b = 3; c = 7  
c =  
    7
```

Utilizando tres puntos se puede cambiar de línea sin ejecutar las órdenes escritas. Esto es útil cuando una expresión no cabe en una sola línea de la ventana de comandos.

Ejemplo:

```
>>a = 5 ; b = 2 ; ...  
s = a/b  
s =  
    2.5000
```

Para introducir comentarios se utiliza el símbolo %. MATLAB ignora lo que se encuentra a la derecha de este símbolo.

Funciones predefinidas

exp , **log** , **log10**, **sqrt** (raíz cuadrada) , **abs** (valor absoluto)
sin , **cos**, **tan**, **cot**, **sec**, **csc**, **asin** (inversa del seno), **acos**, **atan**, **acot**, **asec**, **acsc**
floor(x) (parte entera de x , es decir, el mayor entero menor o igual a x)
 redondea hacia $-\infty$
fix redondea hacia cero
round redondea hacia el entero más proximo

Ejemplos:

```
>> sqrt(16)  
    4  
>> 16^(1/2)  
    4  
>> (25)^(1/3)  
    2.9240  
>> -27^(1/3)  
   -3  
>> (-27)^(1/3)  
    1.5000 + 2.5981i  
>> sin(pi/3)  
    0.8660  
>> sqrt(3)/2  
    0.8660
```

```
>> x = cos(pi) , y = log(25) , z = exp(y)
-1
3.2189
25.0000
>> abs(-3)
3
>> floor(-5.7)
-6
>> fix(-5.7)
-5
>> round(-5.7)
-6
```

Ejercicio:

Calcular

parte entera $\left(\sqrt{|\cos(9\pi/14)|}\right)$

Números complejos

Son de la forma $a + bi$ siendo a y b números reales ($i = \sqrt{-1}$)

```
>> sqrt(-25)
ans =
0 +5.0000i
```

A cada nº complejo se le asigna un punto P de coordenadas (a,b) . Si se une el origen de coordenadas O con P , se obtiene el vector OP .

Algunas funciones de argumento complejo son:

$\text{conj}(z)$: conjugado de z (el conjugado de $a+bi$ es $a-bi$ y recíprocamente).

$\text{abs}(z)$: módulo de z , es decir, $\sqrt{a^2 + b^2}$.

$\text{angle}(z)$: argumento en $(-\pi, \pi]$. Es el ángulo que forma el vector OP con el semieje positivo de abscisas. Muchas veces se calcula en $[0, 2\pi)$.

Ejemplo:

```
>> z = -3 - 4i ;
>> abs(z)
ans =
5
>> angle(z)
ans =
-2.2143
```

Formatos numéricos

Si el resultado de un cálculo es un número entero, MATLAB lo muestra en pantalla como entero si tiene menos de 10 cifras. Si tiene 10 o más cifras o es un número decimal, se puede expresar en alguno de los formatos siguientes (format short es el formato por defecto):

format short : hasta 3 cifras la parte entera y 4 cifras la parte decimal.

format short e : 1 cifra la parte entera, 4 cifras la parte decimal y 3 cifras el exponente.

format long : hasta 2 cifras la parte entera y 14 cifras la parte decimal.

format long e : 1 cifra la parte entera, 15 cifras la parte decimal y 3 cifras el exponente.

Ejemplo:

```
>> a=4/3; b=1231231231; c = pi;
```

```
>> format short
```

```
>> a,b,c
```

```
1.3333
```

```
1.2312e+009
```

```
3.1416
```

```
>> format short e
```

```
>> a,b,c
```

```
1.3333e+000
```

```
1.2312e+009
```

```
3.1416e+000
```

```
>> format long
```

```
>> a,b,c
```

```
1.33333333333333
```

```
1.231231231000000e+009
```

```
3.14159265358979
```

Vectores

Se introducen escribiendo, entre corchetes, cada una de sus coordenadas separadas por un espacio en blanco o una coma.

Ejemplo:

```
>> v = [2, 6, pi/2, log(3), -7, 0.25]
```

Se pueden generar vectores sin introducir explícitamente sus coordenadas:

$v = [v_1:h:v_n]$, genera el vector $v = (v_1, v_1+h, v_1+2h, \dots, v_1+kh)$, donde k es el mayor entero tal que $v_1+kh \leq v_n$.

Ejemplo:

```
>> v = [1: 2: 10]
```

```
v =
```

```
1 3 5 7 9
```

$v = \text{linspace}(a,b,n)$, genera un vector de n coordenadas igualmente separadas, empezando por a y terminando por b

Ejemplo.

```
>> v = linspace(0,1,11)
```

v =

```
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
```

Ejercicio:

Usar la orden `linspace` para generar un vector con todos los múltiplos de 7 desde cero hasta 9996 inclusive. Determinar previamente el número de coordenadas.

Operaciones con vectores elemento a elemento

Los operadores empleados son los de las operaciones aritméticas anteponiendo un punto excepto en la suma y producto por un escalar.

Ejemplo:

```
>> x = [1, 2 ,3]; y = [4,5,6] ;
```

```
>> x+y
```

```
>> 3 * x
```

```
>> x.*y
```

```
>> x./y
```

```
>> x.^2
```

```
>> x.^y
```

Bibliografía utilizada:

Notas sobre MATLAB. Aranda, T. y García, J.G. Universidad de Oviedo. Servicio de Publicaciones. 1999