

# Cálculo Numérico de 1 variable. Curso 2019-2020

## PRÁCTICA 1: PRIMER CONTACTO CON MATLAB.

### ERRORES Y CONDICIONAMIENTO.

**Objetivo:** Realizar diversos cálculos de prueba para familiarizarse con el programa.

#### 1. Programa Octave/Matlab

El entorno de trabajo de Octave/Matlab está formado por:

|   |   |
|---|---|
| 1. Ventana de comandos<br>(Command window)    | Se utiliza para introducir una orden directamente tras el “ <b>prompt</b> ” (>>) (seguida de la tecla <i>enter</i> (↵)).<br>El resultado de la orden se muestra en la misma pantalla salvo que la orden acabe en “;” que lo dejará en la memoria del ordenador.<br>Las líneas que empiezan por “%” son comentarios que no serán ejecutados.<br>La orden “ <b>clear</b> ” borra los datos (variables y funciones) de la memoria del ordenador.<br>La orden “ <b>clc</b> ” limpia la ventana de comandos. |
| 2. Editor<br>(Editor window)                  | Para crear, ejecutar y depurar (reparar errores) de los ficheros “ <b>script</b> ” o “ <b>function</b> ”.   |
| 3. Directorio actual<br>(Current directory)   | Es el directorio de trabajo. Es donde va a buscar los ficheros a ejecutar   |
| 4. Explorador de archivos<br>(File manager)   | Ventana con administrador de archivos para “ <b>seleccionar</b> ” el “ <b>directorio de trabajo</b> ” y los archivos script o function “(.m)” con los que trabajar.   |
| 5. Historial de comandos<br>(Command history) | Las órdenes introducidas en la ventana de comandos quedan grabadas en esta ventana, de forma que, haciendo doble <i>click</i> sobre ellas las podemos volver a ejecutar   |
| 6. Espacio de trabajo<br>(Workspace)          | contiene las variables introducidas y proporciona información sobre el nombre, dimensiones, tamaño y tipo de variable.  |
| 7. Documentación                              | Ventana con sistema de búsqueda y navegación en la documentación del programa. Como alternativa se puede acudir directamente a Google.  |

Todas las ventanas se pueden separar o cerrar pulsando la opción correspondiente en la esquina superior derecha. Para recuperar ventanas cerradas hay que seleccionar la ventana en el desplegable “**Ventana**” de la barra de herramientas



Teclea en la ventana de comandos las ordenes relacionadas, finalizando con la tecla  “Enter”.

### Comandos adicionales:

| Nombre                          | Descripción  | Teclea:   |
|---------------------------------|--|---|
| <b>clc</b>                      | limpia los resultados de la ventana de comandos.   | >> <b>clc</b>                                     |
| <b>clear</b>                    | borra las variables del workspace.   | >> <b>x=1+1</b><br>>> <b>clear</b><br>>> <b>x</b> |
| <b>%</b>                        | Matlab ignora lo que se encuentra a su derecha, por eso se utiliza para escribir comentarios.  | >> % Hola desde Octave/Mathlab                    |
| <b>;</b>                        | Inhibe la visualización en pantalla de la operación realizada.   | >> <b>x= 1+1;</b><br>>> <b>x</b>                  |
| <b>disp('...')</b>              | muestra en pantalla lo que se escribe entre las comillas, al ejecutar cambia de línea.   | >> <b>display('Hola desde Octave/Mathlab')</b>    |
| <b>fprintf('...',variables)</b> | muestra en pantalla lo que se escribe entre las comillas. Admite incluir variables con formato (%d para un dato correspondiente a una variable que se incluye después, con comas). No cambia de línea al ejecutar. | >> <b>fprintf('x = %d ; pi= %d',x,pi)</b>         |
| <b>\n</b>                       | es un salto de línea en la terminal.   | >> <b>fprintf('x = %d ; pi= %d \n',x,pi)</b>      |

### Operaciones elementales con números : +, -, \*, /, ^.

Teclea:

```
>> 7+27^(1/3)
>> (2+3.5)*(2-3.5)/5
```

**Variables:** (Las variables aparecen en el Espacio de trabajo “Workspace”).

Hay dos tipos de variables, las **variables locales y las variables globales**.

Las variables locales se utilizan en el espacio de trabajo en un script concreto.

Cuando se quiere que la misma variable se utilice en varios ficheros se debe declarar como variable global. (Ejemplo: global n)

Teclea:

```
>> a=2;
>> b=3;
>> area= 1/2*a*b
```

### Variables predefinidas en Matlab

| Nombre               | Descripción                                 | Teclea:                          |
|----------------------|---|----------------------------------|
| <b>ans</b>           | es la variable que contiene los resultados, | >> <b>5*ans</b>                  |
| <b>pi</b>            | el número real $\pi$ ,                      | >> <b>2*pi*5</b>                 |
| <b>eps</b>           | el epsilon de la máquina (números “double”) | >> <b>eps</b>                    |
| <b>eps('single')</b> | el epsilon de la máquina (números “simple”) | >> <b>eps('single')</b>          |
| <b>Inf o inf</b>     | infinito,                                   | >> <b>1/0</b><br>>> <b>1/Inf</b> |
| <b>Nan o nan</b>     | indeterminado                               | >> <b>0/0</b>                    |
| <b>i o j</b>         | unidad imaginaria de $\mathbb{C}$ .         | >> <b>(3i)^2</b>                 |

## Funciones predefinidas:

| Nombre            | Descripción   | Teclea:   |
|-------------------|---|---|
| $\text{sqrt}(x)$  | raíz cuadrada de $x$                                    | >> <b>sqrt</b> (81)                             |
| $\text{exp}(x)$   | exponencial $e^x$                                       | >> <b>exp</b> (2)* <b>exp</b> (-pi*i)           |
| $\text{log}(x)$   | logaritmo neperiano                                     | >> <b>log</b> (2)                               |
| $\text{sin}(x)$   | seno (radianes) (sind(x) en grados)                     | >> <b>sin</b> (3)                               |
| $\text{cos}(x)$   | coseno (radianes)                                       | >> <b>sin</b> (3) ^2+ <b>cos</b> (3) ^2         |
| $\text{tan}(x)$   | tangente  | >> 1/(1+ <b>tan</b> (pi/3) ) ^2                 |
| $\text{abs}(x)$   | valor absoluto $ x $                                    | >> <b>abs</b> (-5)<br>>> <b>abs</b> (3+4i)      |
| $\text{fix}(x)$   | redondea al entero más próximo en la dirección del cero | >> <b>fix</b> (-3.7)                            |
| $\text{floor}(x)$ | redondea al entero más próximo en la dirección del -Inf | >> <b>floor</b> (-3.7)                          |
| $\text{ceil}(x)$  | redondea al entero más próximo en la dirección del +Inf | >> <b>ceil</b> (-3.7)                           |
| $\text{round}(x)$ | redondea al entero más próximo                          | >> <b>round</b> (3.7)<br>>> <b>round</b> (-3.7) |

## Declaraciones de formato:

| Nombre         | Descripción  | Teclea:                              |
|----------------|--|--------------------------------------|
| format short   | Formato de punto fijo con 4 dígitos decimales si $0.001 \leq x \leq 1000$ , en otro caso short e | >> <b>format short</b><br>>> 290/7   |
| format long    | Formato de punto fijo con 14 dígitos decimales si $0.001 \leq x \leq 1000$ , en otro caso long e | >> <b>format long</b><br>>> 290/7    |
| format short e | Formato de coma flotante con 4 dígitos decimales   | >> <b>format e</b><br>>> 290/7       |
| format long e  | Formato de coma flotante con 15 dígitos decimales  | >> <b>format long e</b><br>>> 290/7  |
| format short g | La mejor de entre coma flotante y punto fijo con 5 decimales                                     | >> <b>format short g</b><br>>> 290/7 |
| format long g  | La mejor de entre coma flotante y punto fijo con 15 decimales                                    | >> <b>format long e</b><br>>> 290/7  |
| format bank    | dos decimales  | >> <b>format bank</b><br>>> 290/7    |

Las declaraciones son válidas hasta que aparece una nueva declaración y permanecen mientras este activo el núcleo de Octave/MatLab . Se pueden usar otras longitudes, por ejemplo para usar  $d$  dígitos decimales con Octave se pueden ejecutar las siguiente líneas

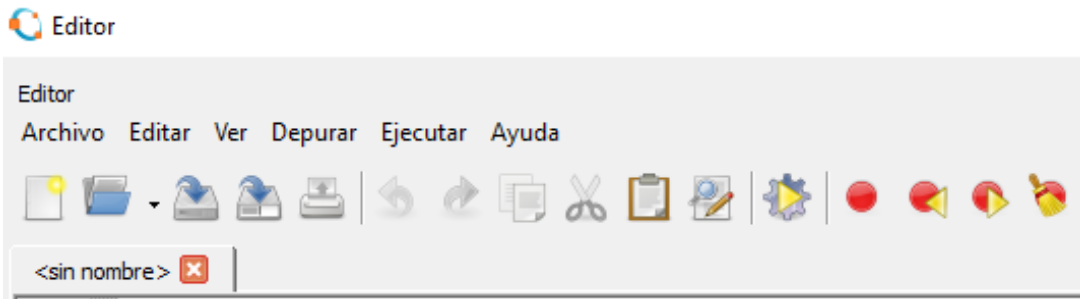
```
format long;
output_precision(d);
```

La función “**single(x)**” también se puede utilizar para trabajar en simple precisión en Matlab.

Ficheros “**script.m**” y “**function.m**”

El cálculo elemental se puede realizar en la ventana de comandos, pero para cálculos que se quieran guardar es preferible utilizar ficheros “**script.m**” o “**function.m**” que se manejan desde del

“Editor”.



- Para abrir un fichero “.m” se puede hacer click en la ventana del Explorador de Archivos de Octave/MatLab o usar el desplegable “**Archivo**” de la barra de herramientas de la ventana del Editor.
- Para crear un fichero “.m”, selecciona en el desplegable “**Archivo**” de la barra de herramientas “Nuevo guión (Script) o Nueva función (function)”, como alternativa puedes usar el icono correspondiente de la barra
- Para guardarlo, selecciona en el desplegable “**Archivo**” de la barra de herramientas “Guardar / Save” o “Guardar como / Save As”.
- Ejecutarlo: selecciona en el desplegable “**Ejecutar**” de la barra de herramientas “Guardar y ejecutar / Save and Run”, como alternativa puedes pulsar el icono correspondiente de la barra
- Se pueden utilizar los *breakpoints* (punto rojo sobre la barra) para detener la ejecución del programa en un punto concreto del fichero. El fichero se sigue ejecutando al pinchar en *continue*. se pueden poner tantos como se quiera y se borran haciendo *click* sobre ellos.

Es conveniente que al principio de cada ejercicio comencéis con algo así como:

```
disp('Ejercicio 1')
clear all
clc
```

De esta forma se borran todas las variables utilizadas hasta el momento y no interfieren en los cálculos siguientes.

**Todos los “scripts” se guardan en una misma carpeta en el directorio de trabajo con extensión “.m” y los ficheros “function” que también tienen extensión “.m” pueden estar en el mismo directorio de trabajo o en otro (en estos casos hay que acceder a ellos añadiendo el nuevo directorio al sistema con la orden “addpath”.**

**NO SE PUEDEN USAR ACENTOS NI ESPACIOS EN BLANCO EN LOS NOMBRES DE LOS DIRECTORIOS NI EN LOS DE LOS FICHEROS.**

Si en un M-file se define una función el fichero debe llamarse con el nombre de la función.

## 2. Ejercicios

1. Construye un programa en Matlab que realice el redondeo simétrico a un número fijo de dígitos (llámalo **fls.m**). Nota: En Octave/Matlab se puede utilizar el comando “**eval (mat2str (x, n)**” siendo *n* una variable global.

En Matlab se puede utilizar también el comando “**round(x,n,'significant')**” siendo  $n$  una variable global.

- Trabajando en aritmética con siete dígitos y redondeo simétrico comprobar que si  $a = 0.5234563$ ,  $b = 0.5711321$  y  $c = -0.5988678$  entonces

$$a \oplus (b \oplus c) \neq (a \oplus b) \oplus c.$$

- Diseña un programa que halle las soluciones de una ecuación cuadrática  $ax^2 + bx + c = 0$  cuando se le introducen los valores  $a$ ,  $b$  y  $c$  y tenga en cuenta la cancelación substractiva. Comprueba tus programas para algunas ecuaciones de segundo grado cuya solución conozcas de antemano.
- Halla el valor de  $A = \sqrt[3]{5556} - \sqrt[3]{5555}$  trabajando en aritmética de 4 dígitos y redondeo simétrico, y con 8 dígitos y redondeo simétrico.
  - Encuentra un algoritmo estable que permita el calculo de  $A$  sin perdida de cifras significativas y utiliza ese algoritmo para calcular  $A$  trabajando en aritmética de 4 dígitos y redondeo simétrico, y con 8 dígitos y redondeo simétrico.
  - Tomando como valor real  $A = 1.062665655226140e - 003$  compara el número de dígitos significativos que tiene cada una de las aproximaciones anteriores.
- Trabajando con aritmética de 8 dígitos y redondeo simétrico aproxima la suma de los  $10^4$  primeros términos de la serie

$$\sum_{k=1}^{\infty} \frac{1}{k^2}.$$

Suma los términos de grandes a pequeños y de pequeños a grandes. Controla el tiempo de ejecución con los comandos `tic`, `toc`.

Determina el número de cifras significativas exactas de las aproximaciones anteriores tomando como valor exacto de la suma el que se obtiene con doble precisión.

Nota: puedes hacer lo mismo con las series  $\sum_{k=1}^{\infty} \frac{1}{k}$  y  $\sum_{k=1}^{\infty} \frac{1}{2^k}$  y ver lo que ocurre.

- Aproxima la derivada de la función  $\sin(x)$  en  $x = 1$  mediante la fórmula

$$\frac{\sin(1+h) - \sin(1)}{h}$$

con  $h = 10^{-i}$  para  $i = 5, 6, \dots, 16$  comparando los resultados obtenidos con el valor exacto de  $\cos(1)$ . Halla el error absoluto y relativo para cada aproximación y comprueba la pérdida de precisión por cancelación. Trabajad con toda la precisión de la máquina.

Nota: puedes ver lo que ocurre si trabajas con sólo 5 dígitos y redondeo simétrico.

- Determina el mayor y menor número máquina positivo cuando se trabaja con doble precisión. Compara con los comandos `realmax` y `realmin`.
- Calcula el épsilon de la máquina. Para ello, calcular  $1 + x$  con  $x = 2^{-n}$  para  $n = 1, 2, \dots$ , mientras que  $1 + x > 1$ . Comparar con el comando `eps` de Matlab.