

Primera aproximación a Octave

Taller de Modelización Matemática y Computacional 2021

01/09/2021

¿Qué es Octave?

- Programa y lenguaje de alto nivel diseñado principalmente para el cálculo numérico.
- Es software libre bajo la GNU General Public License:
<https://www.gnu.org/licenses/gpl-3.0.html>
 - “El software libre es un software cuyo código fuente puede ser estudiado, modificado, y utilizado libremente con cualquier finalidad y redistribuido con cambios o mejoras sobre ellas.”
Wikipedia
- Mayormente compatible con MATLAB
 - Esto significa que es muy probable que una solución a un problema en MATLAB se pueda implementar sin cambios en Octave.

Interfaz gráfica de Octave

- File Browser
- Workspace
- Command History
- Command Window
- Documentation
- Variable Editor
- Editor

1. Octave como calculadora

- Operaciones aritméticas: suma (+), resta (-), multiplicación (*), división (/), potenciación (^).
- Constantes matemáticas e y π .
- Prioridad de las operaciones y el uso de paréntesis: 2^3^2 $2^{(3^2)}$ $(2^3)^2$
- Primera función: *sqrt()*
- Comando de cabecera: *help*
- Funciones matemáticas básicas: trigonométricas (*sin()*, *cos()*, *tan()*), exponencial (*exp()*), logarítmicas (*log()*, *log10()*).
- Números complejos (*complex()*).

Ejercicio 1.

- Plantear la operación $3 + 5 / 2 * 3^2$ y calcular el resultado poniendo paréntesis en diferentes lugares. Deducir el orden de las operaciones.
- Explorar las funciones trigonométricas, exponenciales y logarítmicas con el comando `help`. Utilizar la función exponencial para comprobar la identidad de euler:
 $e^{i\pi} = -1$
- El número imaginario $i = \sqrt{-1}$ se representa en Octave como *i* o *j*.
- π se representa como *pi*.

2. Variables

- Una variable es un sector de la memoria que hemos reservado y al que podemos asignar diferentes valores.
- Nombre de las variables
 - Deben comenzar con una letra, se puede utilizar el guión bajo como separación.
 - Se recomienda utilizar nombres que reflejen el contenido de la variable.
 - Comando *exist*. Tener cuidado de no utilizar nombres de funciones de Octave.
- Tipos de datos
 - Numéricos, caracteres, lógicos.

Ejercicio 2.

- Crear una variable con nombre *sqrt* como la raíz cuadrada de 4.
- Calcular la raíz cuadrada de 9.
- Utilizar *help* con *sqrt*
- Crear una función que calcule: $x/1+x$
- Sintaxis para crear una función:

***function* <salida> = <nombre de la función>(<entrada>)**

%Comentario sobre lo que hace la función

<Comandos que lleva a cabo la función>

endfunction

Esto hay que guardarlo en un archivo <nombre de la función>.m Este archivo debe estar en la carpeta en la que estamos trabajando.

3. Vectores y Matrices

- Formas de definir vectores y matrices
- Operaciones con vectores y matrices

Ejercicio 3.

- Crear un vector x que vaya del 0 al 100 con un paso de 0.1.
- Crear una matriz $A_{8 \times 8}$ con números aleatorios.
- Multiplicar A por 10.
- Crear un vector a que sea la tercera columna de la matriz A .
- Extraer el número de la posición 7 de a y ponerlo en la variable b .
- Modificar la función creada en el ejercicio 2 para que pueda operar con vectores.

4. Gráficos

- Función *plot()*
- *title*("nombre del gráfico")
- *xlabel*("nombre del eje x")
- *ylabel*("nombre del eje y")
- Rango de los ejes: *axis*([xmin xmax ymin ymax])
- hold on/off
- grid

Ejercicio 4.

- Graficar la función creada en el ejercicio 2.
- Modificar color y grosor de la línea.
- Modificar ejes.
- Poner título al gráfico y a los ejes.
- Graficar la función seno y sobreponer la función coseno.

5. scripts

- Series de instrucciones que se pueden guardar.
- Su ejecución es análoga a introducir los comandos en la Command Window.
- Sus variables son globales, por lo que cualquier variable que aparezca en el script aparecerá en el Workspace.

Ejercicio 5.

- Escribir un script que grafique la función creada en el ejercicio 2. Utilizar variables para establecer el rango de valores a graficar.
- Probar diferentes valores para estas variables.

6. *if, for, while*

- Nos permiten repetir una serie de comandos una cantidad de veces n (*for*), o hasta que se cumpla una determinada condición lógica (*while*), o ejecutar un grupo diferente de comandos según se cumpla una condición lógica determinada (*if*).
- Operadores lógicos: == (igual) ~= (distinto) < (menor) > (mayor) <= (menor o igual) >= (mayor o igual) & (and) ~ (not) | (or)

Ejercicio 6.

- Crear un script que incluya una bifurcación *if* y determine si dos números son iguales, el primero es mayor al segundo o el segundo mayor al primero.
- Crear un loop *while* para demostrar que el 0 no es el único neutro de la suma en Octave.
- Crear un loop *for* que genere los primeros n números de la sucesión de Fibonacci:

$$F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}$$

Recursos

- Software libre: https://es.wikipedia.org/wiki/Software_libre
- GNU General Public License: <https://www.gnu.org/licenses/gpl-3.0.html>
- Manual Octave, wiki y otros recursos:
<https://www.gnu.org/software/octave/support>
- MATLAB notes: <http://math.niu.edu/~ammar/MatlabNotes.pdf>