

1. Introducción

GNU Octave es un lenguaje de alto nivel destinado para el cálculo numérico, sólo opera con números. Provee una interfaz sencilla, orientada a la línea de comandos (consola), que permite la resolución de problemas numéricos, lineales y no lineales, además permite la ejecución de scripts y puede ser usado como lenguaje orientado al procesamiento por lotes.

Octave nació alrededor del año 1988, y fue concebido originalmente para ser usado en un curso de diseño de reactores químicos. El desarrollo real de comenzó en 1992. La primera alfa fué publicada en 1993, y en 1997 se publicó la versión 1.0.

2. Iniciar y salir de Octave

Para iniciar octave hay que ejecutar la instrucción *octave* en una consola. Aparecerá la siguiente ventana:

```
GNU Octave, version 3.6.2
Copyright (C) 2012 John W. Eaton and others.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. For details, type 'warranty'.
```

```
Octave was configured for "x86_64-pc-linux-gnu".
```

```
Additional information about Octave is available at http://www.octave.org.
```

```
Please contribute if you find this software useful.
For more information, visit http://www.octave.org/help-wanted.html
```

```
Read http://www.octave.org/bugs.html to learn how to submit bug reports.
```

```
For information about changes from previous versions, type 'news'.
```

```
octave:1>
```

Para salir de octave usaremos el comando *quit* o *exit*.

2.1. Instrucciones de utilidad

- `>pwd`: para mostrar el directorio en el que nos encontramos.
- `>ls`: para mostrar una lista de los ficheros y los directorios del directorio actual.
- `>cd ruta`: para cambiar de directorio.
- `>>cd ..`: para ir al directorio padre.
- `>diary`: volcar lo mostrado en el transcurso de la ejecución del programa al fichero 'diary' de la carpeta por defecto.
- `>>diary fichero`: el fichero 'diary' se guardará en el fichero indicado.

- >>diary off: para el guardado.
- >help comando: muestra la ayuda sobre el comando indicado.
- >history: muestra una lista con los comandos ejecutados.
- >save archivo: guardar la sesión en un archivo.
- >load archivo: cargar la sesión del archivo.

Los comandos pueden recuperarse con la *flecha hacia arriba*, o escribiendo el inicio del comando y pulsar la *flecha hacia arriba*, y navegar por los comandos ejecutados.

2.2. Operaciones básicas

- = *asignacion*
- + *suma*
- - *resta*
- * *multiplicación*
- .* *multiplicación de matrices elemento por elemento (deben coincidir el número de columnas con el número de filas)*
- / *división derecha*
- exp *exponencial*
- log *logaritmo neperiano*
- log10 *logaritmo base 10*
- sin *seno*
- cos *coseno*
- abs *valor absoluto*
- sqrt *raíz cuadrada*
- round *redondeo al entero más cercano*
- floor *redondea por defecto*
- ceil *redondea por exceso*
- ' *transpuesta*

3. Variables

Podemos asignar variables con determinados nombres a las expresiones numéricas (números, constantes). Los nombres son sensibles a mayúsculas y minúsculas, el máximo de caracteres que deben tener es de 31 caracteres, deben empezar por una letra y pueden contener letras, números y el símbolo '_'. Las variables se crean escribiendo el nombre que le queramos dar y asignándole el valor con el operador de asignación. El acceso a la variable se realiza escribiendo el nombre asignado.

```
>a=1/3
```

Cuando se asigna otro valor se machaca en anterior

```
>a=1/3
```

```
>a=1/6
```

```
>a
```

```
a=0.16667
```

Las variables creadas se guardan en una lista de variables.

- `whos`: obtener la lista de variables guardadas.

Las variables guardadas pueden ser borradas.

- `>clear`: borra todas las variables
- `>>clear a`: borra la variable indicada (en este caso `a`).

Pueden colocarse varios comandos en una misma línea separándolos con ';', si no se quiere que se muestre el resultado de algún comando se finaliza el comando con ';'.

Se permiten comentarios, de gran ayuda a la hora de realizar scripts para ejecutar.

- `%` línea de comentario, todo lo que esté a su derecha se considera comentario.

Se puede extender un comando a más de una línea con '...'.

```
>esto\_es\_muy\_largo...
```

```
>=17
```

```
esto\_es\_muy\_largo = 17
```

3.1. Variables predefinidas

- `ans` último resultado
- `pi` = 3.1416
- `e` = 2.7183
- `i`, `j` número imaginario
- `Inf` infinito
- `NaN` indeterminado

4. Vectores

Un vector es definido como un conjunto de datos a los cuales se accede por medio de índices. Es una matriz de una dimensión.

La forma en la que octave se definen vectores es utilizando corchete []. Los elementos de una fila se separan con un espacio ' ' o una coma ','.

```
> v= [1,2,3,5,7,11]
```

```
v =
```

```
1    2    3    5    7   11
```

Las columnas por su parte se separan mediante puntos y comas ';'.
';

```
> w= [1;2;3;5;7;11]
```

```
w =
```

```
1
2
3
5
7
11
```

4.1. Secuencias

En octave podemos crear vectores de secuencias utilizando los dos puntos, p:q:r, donde p sería el valor en el que se iniciaría la secuencia, r el valor final y q el intervalo. El valor q es opcional y si se obvia el intervalo por defecto es 1.

```
> 1:10
```

```
ans =
```

```
1    2    3    4    5    6    7    8    9   10
```

```
> 1:2:10
```

```
ans =
```

```
1    3    5    7    9
```

Además hay otras dos funciones para crear vectores con secuencias separadas x valores, *linspace* y *logspace*. El primero separa los números uniformemente y el segundo logarítmicamente.

```
> linspace(0,15,6)
```

```
ans =
```

```
0    3    6    9   12   15
```

Hay dos funciones especiales para crear filas o columnas de unos o ceros, *ones* y *zeros*

```
> ones(1,5)
```

```
ans =
```

```
1    1    1    1    1
```