

# Guía rápida de octave

Marduk Bolaños Puchet  
octubre de 2012

## Uso de octave

help comando	ayuda sobre <i>comando</i>
run archivo.m	procesar archivo de comandos
Ctrl-c	interrumpir octave
quit	salir de octave

## Edición de la línea de comandos

TAB	Completar el nombre de un comando
Flecha hacia arriba	Línea de comandos anterior
Flecha hacia abajo	Línea de comandos siguiente
Ctrl-e	Cursor al final de la línea
Ctrl-a	Cursor al principio de la línea
Ctrl-k	Borrar línea de comandos actual

## Constantes predefinidas

Inf	Infinito
NaN	No es un número
ans	El resultado de la última expresión
eps	epsilon de la máquina
pi	$\pi$
ii	$\sqrt{-1}$

## Funciones predefinidas

mod(x,y)	residuo del cociente $x/y$
sqrt(x)	raíz cuadrada de $x$
sin(x)	$\sin(x)$ , $x$ en radianes
asin(x)	$\arcsin(x)$ , función inversa de sin
exp(x)	$\exp(x)$ , función exponencial
log(x)	logaritmo natural de $x$

## Asignaciones

var = expr	asignar una expresión a una variable y mostrar el resultado
var = expr;	asignar una expresión a una variable y <b>no</b> mostrar el resultado

## Comentarios

% Este es un comentario de una línea

## Estructuras de control

<b>for</b> <i>variable</i> = <i>expr</i> <i>lista de expresiones</i> <b>end</b>	Ejecuta la <i>lista de expresiones</i> una vez, por cada valor en el intervalo <i>expr</i> . <i>variable</i> tiene el valor actual de <i>expr</i> en cada iteración
<b>while</b> ( <i>condicion</i> ) <i>lista de expresiones</i> <b>end</b>	Ejecuta la <i>lista de expresiones</i> siempre que la variable booleana (es decir, vale 0 ó 1) <i>condicion</i> sea verdadera (vale 1).
<b>if</b> ( <i>condicion1</i> ) <i>instrucciones1</i> <b>[elseif</b> ( <i>condicion2</i> ) <i>instrucciones2</i> <b>[else</b> <i>instrucciones0</i> <b>end</b>	Si <i>condicion1</i> es verdadera, se ejecuta <i>instrucciones1</i> . Si <i>condicion1</i> es falsa y <i>condicion2</i> es verdadera, se ejecuta <i>instrucciones2</i> . Si ambas son falsas, se ejecuta <i>instrucciones0</i> . Estos dos casos son opcionales.

## Definir funciones

En el archivo `funcion.m`

```
function [ lista_retorno ] = funcion ( [ lista_args ] )  
cuerpo de la función  
end
```

La *lista\_retorno* puede ser una sola variable o una lista de variables. La o las variables deben ser utilizadas en el *cuerpo de la función*. *lista\_args* es la lista de argumentos de la función, puede no tener elementos.

### Funciones anónimas

```
f = @(x) f(x);
```

## Definir matrices y vectores

v = [x, y, ...]	vector renglón
v = [x; y; ...]	vector columna
M = [a, b; c, d]	matriz de $2 \times 2$
eye(N)	matriz identidad de $N \times N$
ones(1,N)	vector renglón con $N$ unos
zeros(1,N)	vector renglón con $N$ ceros
M(i,j)	elemento $M_{ij}$ de la matriz M
v(i)	elemento $v_i$ del vector v

## Operadores aritméticos para matrices

$x + y$	adición elemento por elemento
$x - y$	substracción elemento por elemento
$x * y$	multiplicación de matrices
$x . * y$	multiplicación elemento por elemento
$x ./ y$	división elemento por elemento
$x.^y$	eleva a una potencia elemento por elemento

## Definir intervalos

```
1:k:N          Los números de 1 a  $N$  en pasos de  $k$   
linspace(1,N,k)  $k$  números equiespaciados entre 1 y  $N$ 
```

## Operadores lógicos (booleanos)

```
==  igual          ~=  diferente  
&& conjunción      ||  disyunción
```

## Ecuaciones diferenciales ordinarias con ode45 (odepkg)

**EDO de primer orden**  $\dot{x} = f(t, x)$

```
ec_p = @(t,x) f(t,x);
```

```
[tsol, xsol] = ode45(ec_s,[t_ini,t_fin],x0);
```

**EDO de segundo orden**  $\ddot{x} = f(t, x, \dot{x})$

```
ec_s = @(t,x) [x(2); f(t,x,v)];
```

```
[tsol, sol] = ode45(ec_s,[t_ini,t_fin],[x0;v0]);
```

```
xsol = sol(:,1); vsol = sol(:,2);
```

**EDO de segundo orden acopladas**

$$\ddot{x} = f(t, x, \dot{x})$$
$$\ddot{y} = g(t, y, \dot{y})$$

```
ec_s = @(t,x) [x(2); f(t,x,vx); x(4); g(t,y,vy)];
```

```
[tsol, sol] = ode45(ec_s,[t_ini,t_fin],[x0;vx0;y0;vy0]);
```

```
xsol = sol(:,1); vxsol = sol(:,2);
```

```
ysol = sol(:,3); vysel = sol(:,4);
```

**Detener la integración cuando ocurre un evento**

```
eventos.m
```

```
function [valor, terminar, direccion] = eventos(t,r)  
    valor = r(1) - x; % buscamos ceros de esta función  
    terminar = 1; % dejamos de integrar cuando encontramos un cero  
    direccion = 1; % buscamos ceros cuando la función crece  
end
```

```
opciones = odeset('Events', @eventos);
```

```
[t,sol] = ode45(ec_r,[t_ini,t_fin],[x0;v0],opciones);
```

## Ecuaciones diferenciales ordinarias con lsode

```
ec_p = @(x,t) f(t,x);
```

```
xsol = lsode(ec_p,x0,[t_ini:dt:t_fin]);
```

## Gráficos en 2D

```
plot(x,f(x),x,g(x))    Grafica  $x$  vs.  $f(x)$  y  $x$  vs.  $g(x)$   
xlabel='Eje x'          Etiqueta del eje  $x$   
legend('f(x)', 'g(x)')  Leyenda  
xlim([xmin, xmax])     Región del eje  $x$  mostrada
```

## Múltiples gráficos en una figura

Consideramos una matriz de gráficos con  $r$  renglones y  $c$  columnas.

```
subplot(r,c,1)          Primera gráfica  
plot(x,sin(x))  
subplot(r,c,2)          Segunda gráfica  
plot(x,cos(x))
```

## Gráficos en 3D

$$z = f(x, y)$$

```
x = linspace(xmin,xmax,Nx);  
y = linspace(ymin,ymax,Ny);  
[X,Y] = meshgrid(x,y);  
surf(X,Y,f(X,Y))
```

**Superficies paramétricas**

*Esfera*

```
fx = @(s,t) cos(s) .* cos(t);  
fy = @(s,t) sin(s) .* cos(t);  
fz = @(s,t) sin(t);  
ezsurf(fx, fy, fz, [-pi, pi, -pi/2, pi/2], 20);
```

## Exportar gráfico a un archivo

```
set(0, 'defaultlinelength',4);          línea gruesa  
figure(N);                               Crea la figura número  $N$   
plot(x,f(x));                             Grafica una función en la figura  
print -dpdfwrite -tight -FArial:14 archivo.pdf  Guarda la figura en un archivo .pdf
```

## Instalación de paquetes

```
pkg install -forge paquete  odepkg  Solución de ecuaciones diferenciales  
                             audio    Grabar, procesar y reproducir sonido  
                             optim    Estimación de parámetros con mínimos cuadrados
```

## .octaverc

```
pkg prefix ~/octave_pkg ~/octave_arch;  necesario para instalar paquetes  
format long;                             muestra todas las cifras significativas  
setenv("DISPLAY","");                   necesario para el paquete audio
```