

Guía rápida de octave

Marduk Bolaños Puchet
octubre de 2012

Uso de octave

<code>octave</code>	ejecutar octave
<code>quit</code>	salir de octave
<code>Ctrl-c</code>	interrumpir octave
<code>octave archivo.m</code>	procesar archivo de comandos

Obtener ayuda

<code>help comando</code>	ayuda sobre un comando
---------------------------	------------------------

Edición de la línea de comandos

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

Constantes predefinidas

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

Funciones predefinidas

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

Asignaciones

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

Comentarios

`% Este es un comentario de una línea`

Estructuras de control

<code>for variable = expr</code> <code>lista de expresiones</code> <code>end</code>	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
<code>while (condicion)</code> <code>lista de expresiones</code> <code>end</code>	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).
<code>if (condicion1)</code> <code>instrucciones1</code> <code>[elseif (condicion2)</code> <code>instrucciones2]</code> <code>[else</code> <code>instrucciones0]</code> <code>end</code>	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

`function [lista retorno] = funcion ([lista args])`
cuero 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 *cuero 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

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

Operadores aritméticos para matrices

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

Definir intervalos

<code>1:k:N</code>	Los números de 1 a N en pasos de k
<code>linspace(1,N,k)</code>	k números equiespaciados entre 1 y N

Operadores lógicos (booleanos)

<code>==</code>	igual
<code>~=</code>	diferente
<code>&&</code>	conjunción
<code> </code>	disyunción

Gráficos en 2D

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

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

<code>figure(N);</code>	Crea la figura número N
<code>plot(x,f(x));</code>	Grafica una función en la figura
<code>print -dpdfwrite -tight archivo.pdf</code>	Guarda la figura en un archivo con extensión <code>.pdf</code>

Instalación de paquetes

<code>pkg install -forge paquete</code>	<code>odepkg</code>	Solución de ecuaciones diferenciales
	<code>audio</code>	Para grabar, procesar y reproducir sonido
	<code>optim</code>	Estimación de parámetros con mínimos cuadrados

`.octaverc`

<code>pkg prefix ~/octave_pkg ~/octave_arch;</code>	para instalar paquetes
<code>format long;</code>	muestra todas las cifras significativas
<code>setenv("DISPLAY"," :0.0");</code>	necesario para el paquete <i>audio</i>