

GNU Octave / Matlab®

Inicio rápido

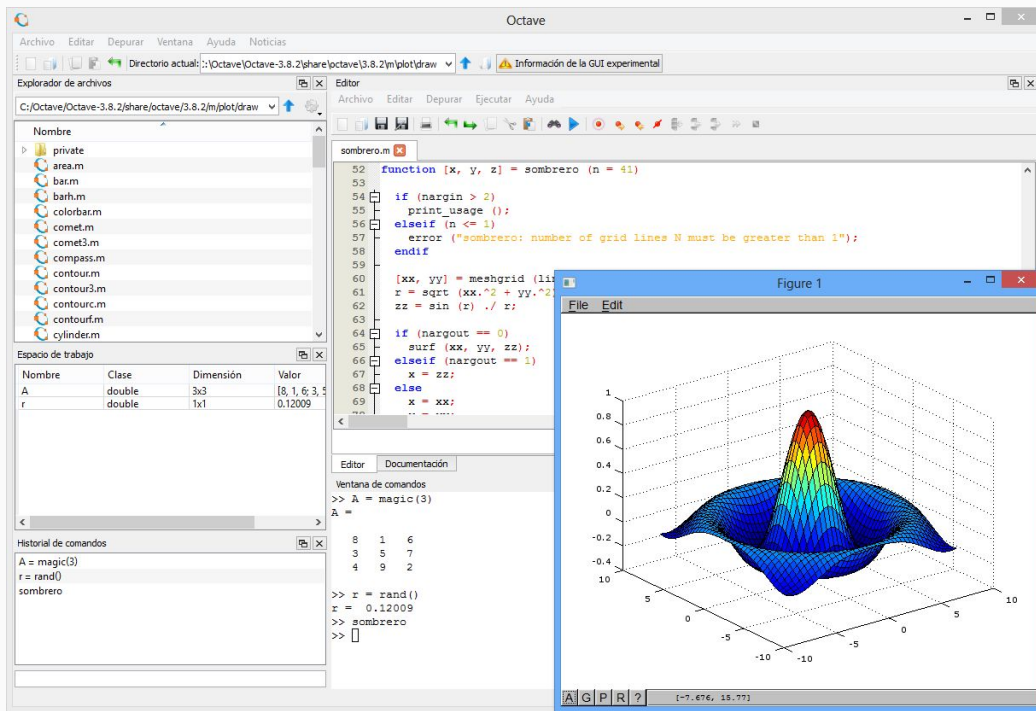
GNU Octave

- Octave o GNU Octave es un programa y lenguaje de programación para realizar cálculos numéricos. Como su nombre indica, Octave es parte del proyecto GNU*.
- Permite ejecutar órdenes en modo interactivo o interpretar scripts.
- Está orientado al análisis numérico.
- Sitio web: <https://www.gnu.org/software/octave/> (Descargas, documentación, código fuente, etc)

* wikipedia: https://es.wikipedia.org/wiki/GNU_Octave

GNU Octave

- Se puede ejecutar en modo GUI o consola



```
K53E ~$ octave --no-gui
GNU Octave, version 4.2.2
Copyright (C) 2018 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/get-involved.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> A = [1 2 3; 4 5 6; 7 8 9]
A =

     1     2     3
     4     5     6
     7     8     9

octave:2>
```

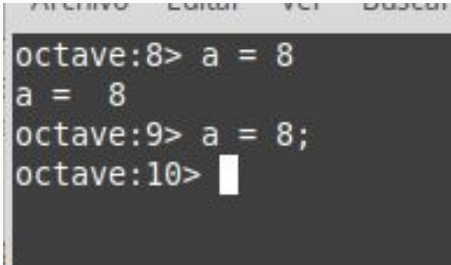
Cálculos elementales

- Operaciones aritméticas: +, -, *, /
- Exponenciación: ^ o **
- Raíz cuadrada: sqrt() , cúbica: cbrt() o n-ésima: nthroot()
- Logaritmos: log(), log10(), log2() y e^x : exp()
- Trigonómicas: rad2deg(), deg2rad(), sin(), cos(), tan(), ... , asin(), acos(), atan(), ..., sinh(), cosh(), ... , asinh(), acosh()...
- Constantes: pi, e
- Complejos: abs(), angle(), real(), imag()... (los complejos se anotan: 3+4i o 3+4j)

```
octave:15> sin(pi/3) == sqrt(3)/2  
ans = 1
```

El lenguaje

- Diferencia minúsculas y mayúsculas: $x \neq X$
- La asignación se realiza con = ej.: rate = 67.5
(la variable rate vale 67.5)
- La salida de cada comando se muestra en la consola, **excepto** si se finaliza con ;



```
octave:8> a = 8
a = 8
octave:9> a = 8;
octave:10> 
```

El lenguaje - comentarios

```
{
  Este script es para mostrar
  varios comentarios
}

# título
display("Cálculo de cuadriláteros")

%{
  un comentario
  de varias líneas
}%

% otro título
disp("Ingrese lado b: ")
```

- Se pueden incluir comentarios hasta el fin línea con % o #
- O hacer bloques de comentarios con %{ ... }% o #{ ... }#

salida

```
Cálculo de cuadriláteros
Ingrese lado b:
>> |
```

Funciones

```
> x = linspace(0, 360, 9)      % 9 valores entre 0 y 360
```

```
x =
```

0.00	45.00	90.00	135.00	180.00	225.00	270.00	315.00	360.00
------	-------	-------	--------	--------	--------	--------	--------	--------

```
> [x; sin(deg2rad(x)); cos(deg2rad(x))] % valores de seno y coseno
```

```
ans =
```

0.00	45.00	90.00	135.00	180.00	225.00	270.00	315.00	360.00
0.00	0.71	1.00	0.71	0.00	-0.71	-1.00	-0.71	-0.00
1.00	0.71	0.00	-0.71	-1.00	-0.71	-0.00	0.71	1.00

```
>
```

Funciones

```
function nombre  
  ...  
  ...  
endfunction
```

Matrices

```
> A = [1 2 3; 4 5 6; 7 8 9]
```

A =

1	2	3
4	5	6
7	8	9

```
> B = [1 0 -1; 2 1 -3]
```

B =

1	0	-1
2	1	-3

2 filas, 3 columnas

traspuesta

```
> size(B)
```

ans =

2 3

```
> A'
```

ans =

1	4	7
2	5	8
3	6	9

Matrices

```
> zeros(2, 4)
```

```
ans =
```

```
0 0 0 0
0 0 0 0
```

```
> ones(3, 3)
```

```
ans =
```

```
1 1 1
1 1 1
1 1 1
```

identidad

```
> eye(2, 4)
```

```
ans =
```

Diagonal Matrix

```
1 0 0 0
0 1 0 0
```

```
> eye(3)
```

```
ans =
```

Diagonal Matrix

```
1 0 0
0 1 0
0 0 1
```

Matrices

```
> V = [1 2 3]
```

```
V =
```

```
1 2 3
```

```
> diag(V)
```

```
ans =
```

Diagonal Matrix

```
1 0 0
0 2 0
0 0 3
```

genera matrices con
números aleatorios

```
> rand(3) * 10
```

```
ans =
```

```
5.5484 9.6397 9.5811
2.1340 6.1965 1.7173
1.1788 7.0846 7.9574
```

```
> randi([-4, 4], 2, 3)
```

```
ans =
```

```
4 1 -4
-4 -4 3
```

genera una matriz
cuya diagonal es el
vector V

Matrices

```
> A = [1 2 3; 4 5 6; 7 8 9]
```

A =

1	2	3
4	5	6
7	8	9

```
> B = [-3 0 1; -2 1 -2; -4 2 3]
```

B =

-3	0	1
-2	1	-2
-4	2	3

```
> A + B           % suma
```

ans =

-2	2	4
2	6	4
3	10	12

```
> A - B           % resta
```

ans =

4	2	2
6	4	8
11	6	6

Matrices

```
> C = [1 0 -2; 2 1 -1; 1 -1 0]
```

```
C =
```

```
1    0   -2
2    1   -1
1   -1    0
```

```
> D = [0 -1 2; -2 3 0; -3 1 -1]
```

```
D =
```

```
0   -1    2
-2    3    0
-3    1   -1
```

```
> C * D % producto de matrices
```

```
ans =
```

```
6   -3    4
1    0    5
2   -4    2
```

```
> D * C % producto de matrices
```

```
ans =
```

```
0   -3    1
4    3    1
-2    2    5
```

Matrices

```
> C = [1 0 -2; 2 1 -1; 1 -1 0]
```

```
C =
```

```
1    0   -2
2    1   -1
1   -1    0
```

```
> E = [3 1; -2 2; -1 0]
```

```
E =
```

```
3    1
-2    2
-1    0
```

```
> C * E    % producto de matrices
```

```
ans =
```

```
5    1
5    4
5   -1
```

```
> E * C    % producto de matrices !!
```

```
error: operator *: nonconformant
arguments (op1 is 3x2, op2 is 3x3)
```

Matrices

```
> C = [1 0 -2; 2 1 -1; 1 -1 0]
```

```
C =
```

```
1  0 -2
2  1 -1
1 -1  0
```

```
> 2 * C
```

```
ans =
```

```
2  0 -4
4  2 -2
2 -2  0
```

determinante



```
> det(C)
```

```
ans = 5
```

```
> det(2 * C)
```

```
ans = 40
```

```
> det(4 * C)
```

```
ans = 320
```

```
> det(1/2 * C)
```

```
ans = 0.62500
```

Matrices

```
> C = [1 0 -2; 2 1 -1; 1 -1 0]
```

```
C =
```

1	0	-2
2	1	-1
1	-1	0

matríz inversa



```
> inv(C)
```

```
ans =
```

-0.20000	0.40000	0.40000
-0.20000	0.40000	-0.60000
-0.60000	0.20000	0.20000

```
> C * inv(C)
```

```
ans =
```

1.00000	0.00000	0.00000
0.00000	1.00000	-0.00000
-0.00000	0.00000	1.00000

```
> inv(C) * C
```

```
ans =
```

1.00000	0.00000	0.00000
0.00000	1.00000	0.00000
0.00000	0.00000	1.00000

Matrices

```
> C = [1 0 -2; 2 1 -1; 1 -1 0]
```

```
C =
```

```
1    0   -2
2    1   -1
1   -1    0
```

```
> D = [0 -1 2; -2 3 0; -3 1 -1]
```

```
D =
```

```
0   -1    2
-2    3    0
-3    1   -1
```

```
> D(1, 3)      % elemento f=1, c=3
ans = 2
```

```
> D(1:3, 3)    % f de 1 a 3, c=3
ans =
```

```
2
0
-1
```

```
> D(2, 1:2)    % f=2 c de 1 a 2
ans =
```

```
-2    3
```

Matrices

```
> C = [1 0 -2; 2 1 -1; 1 -1 0]
```

```
C =
```

```
1    0   -2
2    1   -1
1   -1    0
```

```
> D = [0 -1 2; -2 3 0; -3 1 -1]
```

```
D =
```

```
0   -1    2
-2    3    0
-3    1   -1
```

```
> [C, D]
```

```
ans =
```

```
1    0   -2    0   -1    2
2    1   -1   -2    3    0
1   -1    0   -3    1   -1
```

```
> who
```

```
Variables in the current scope:
```

```
A    B    C    D
```

```
> clear all
```

```
> who
```

Sistemas de ecuaciones

$$\begin{cases} 2x - y = 5 \\ x + 2y - z = -1 \\ 3y - z = -4 \end{cases}$$

(2, -1, 1)

```
> A = [2 -1 0; 1 2 -1; 0 3 -1]
```

A =

```
2  -1  0
1   2 -1
0   3 -1
```

```
> b = [5; -1; -4]
```

b =

```
5
-1
-4
```

```
> A \ b
```

ans =

```
2.00000
-1.00000
1.00000
```

Sistemas de ecuaciones

$$\begin{cases} 2x - y = 5 \\ x + 2y - z = -1 \\ 3y - z = -4 \end{cases}$$

(2, -1, 1)

```
> A = [2 -1 0; 1 2 -1; 0 3 -1]
```

A =

```
2  -1  0
1   2 -1
0   3 -1
```

```
> b = [5; -1; -4]
```

b =

```
5
-1
-4
```

método de matriz inversa

```
> inv(A) * b
```

ans =

```
2.00000
-1.00000
1.00000
```

Matrices

```
> A
```

```
A =
```

```
 2  -1  0
 1   2 -1
 0   3 -1
```

matriz de cofactores

```
> det(A) * inv(A)'
```

```
ans =
```

```
 1.0000  1.0000  3.0000
-1.0000 -2.0000 -6.0000
 1.0000  2.0000  5.0000
```

guarda en un archivo

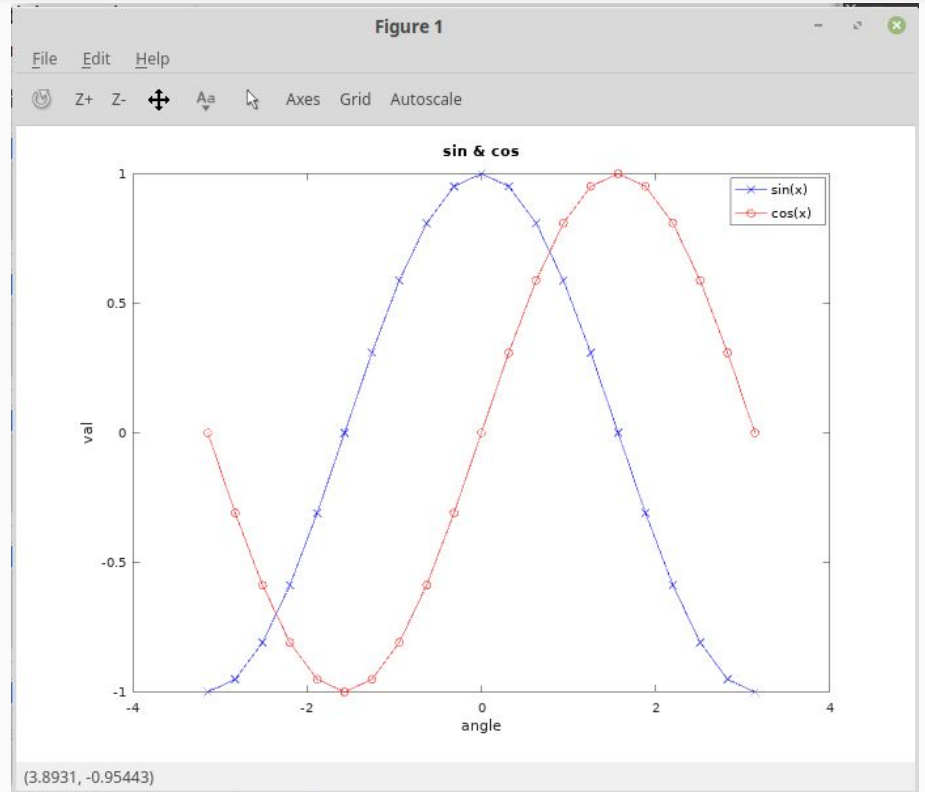
```
> save workspace.mat
```

```
> clear all % borra workspace
```

```
> load workspace.mat
```

carga de un archivo

```
> figure(1)
> x = linspace(-pi, pi, 21);
> y = sin(x);
> z = cos(x);
> plot(x, z, 'b-x', x, y, 'r-o');
> legend("sin(x)", "cos(x)")
> title("sin & cos")
> xlabel("angle")
> ylabel("val")
> saveas(gca, "sin_cos.pdf", "pdf")
```



Gráficos

```
> figure(2)
> t = linspace(-2*pi, 2*pi, 200);
> plot3(sin(t), cos(t), t, 'b-');
> hold on
> plot3(sin(t), -cos(t), t, 'r-');
> xlabel("x")
> ylabel("y")
> zlabel("z")
> saveas(gca, "helices.jpg", "jpg")
> clf      % borra la figura
```

