

2020

Manual MATLAB



Duván Mejía

UNIVERSIDAD SURCOLOMBIANA

25-11-2020

Tabla de contenido

Clase semana 1:	2
Clase semana 2	8
Clase semana 3	13

Clase semana 1:

who: Aquí podemos observar las variables que tengo.

```
a =  
  
    4  
  
>> who  
  
Your variables are:  
  
a
```

whos: aquí vemos todas las características de las variables.

```
>> whos  
  
Name      Size      Bytes  Class  Attributes  
  
a         1x1         8  double  
b         1x1         8  double  
c         1x1         8  double
```

Clear + Variable: Borramos la variable seleccionada.

```
>> clear b  
>> who  
  
Your variables are:  
  
a  c
```

Clear all: borramos todas las variables.

```
>> clear all  
>> who  
>> |
```

disp: Método para Imprimir una variable.

```
>> disp(x)  
Duvan  
>> x = 'Duvan';  
>> disp(x)  
Duvan
```

ans: Variable temp Operación del dato anterior.

```
>> 3 + 5

ans =

     8

>> suma = 4 + 1

suma =

     5

>> ans

ans =

     8
```

Save: guardar archivo.mat y muestra la ruta

```
>> save

Saving to: C:\Users\Mr Mejia\Documents\MATLAB\Examples\R2020a\matlab\CallAFunctionThatReturnsOutputExample\matlab.mat
```

cls : limpia la pantalla.

```
>> a = 4;
>> b = 3

b =

     3

>> x = 'Duvan';
>> cls|
.
```

```
f_x >> |
```

Load: cargar el archivo Matlab

```
>> load
```

```
Loading from: C:\Users\Mr Mejia\Documents\MATLAB\Examples\R2020a\matlab\CallAFunctionThatReturnsOutputExample\matlab.mat
```

format –long : Formato de decimal largo

```
>> format long
```

```
>> pi
```

```
ans =
```

```
3.141592653589793
```

format -shortEng: Formato de Ingeniería

```
>> format shortEng
```

```
>> pi
```

```
ans =
```

```
3.1416e+000
```

format -short : Formato de decimales corto

```
>> format short
```

```
>> pi
```

```
ans =
```

```
3.1416
```

format- shortG: Formato más pequeño de decimales

```
>> format shortG
```

```
>> pi
```

```
ans =
```

```
3.1416
```

Eps: Es el valor mas pequeño y equivalente a `eps(1.0)` y `eps('doble')`.

```
>> eps
```

```
ans =
```

```
2.2204e-16
```

Pi: función matematica

```
>> pi
```

```
ans =
```

```
3.1416
```

Inf: Variable infinita resultante de la división entre Zero

```
>> 6 / (4-4)
ans =
    Inf
```

NaN: Cuando la variable no tiene ningún dato o indeterminado.

```
>> nan(5)

ans =

    NaN    NaN    NaN    NaN    NaN
    NaN    NaN    NaN    NaN    NaN
    NaN    NaN    NaN    NaN    NaN
    NaN    NaN    NaN    NaN    NaN
    NaN    NaN    NaN    NaN    NaN
```

Compleja(i+j): utilizamos con valores complejos.

```
>> sqrt(-1)

ans =

    0.0000 + 1.0000i
```

Realmin: más pequeño a utilizar.

```
>> format long e
>> f = realmin

f =

    2.225073858507201e-308
```

Realmax: valor más grande a utilizar.

```
>> format long e
>> f = realmax

f =

    1.797693134862316e+308
```

Clock: formato de fecha.

```
>> format shortg  
c = clock
```

```
c =
```

```
2020      11      26      21      2      38.18
```

Date: formato de fecha

```
>> z = date
```

```
z =
```

```
'26-Nov-2020'
```

Calendar: variable del calendario

```
>> calendar(1998,01)
```

```
          Jan 1998  
   S      M      Tu      W      Th      F      S  
   0       0       0       0       1       2       3  
   4       5       6       7       8       9      10  
  11      12      13      14      15      16      17  
  18      19      20      21      22      23      24  
  25      26      27      28      29      30      31  
   0       0       0       0       0       0       0
```

Funciones de Aproximación

fix: Redondea el valor más cercano a cero.

```
>> x
```

```
x =
```

```
3.2
```

```
>> y = fix(x)
```

```
y =
```

```
3
```

Floor: Valor negativo hacia infinito.

```
>> X = [-1.9 -0.2 3.4; 5.6 7.0 2.4+3.6i];  
>> Y = floor(X)
```

Y =

-2 +	0i	-1 +	0i	3 +	0i
5 +	0i	7 +	0i	2 +	3i

Round: Redondea al valor decimal o entero infinito más cercano.

```
>> y = round(pi,3)
```

y =

3.142

```
>> y = round(pi)
```

y =

3

Clase semana 2

Trigonometría

Sin: Seno dado en radianes.

```
>> sin(pi)

ans =

    1.2246e-16
```

Sind: Seno dado en grados.

```
>> sind(180)

ans =

    0
```

asin: seno inverso en radianes.

```
>> asin(pi)

ans =

    1.5708 - 1.8115i
```

Asind: seno inverso en grados.

```
>> asind(pi)

ans =

    9.0000e+01 - 1.0379e+02i
```

Sinh: seno hiperbólico.

```
>> sinh(pi)

ans =

    11.5487
```

Cos: Coseno en radianes.

```
>> cos(pi)
```

```
ans =
```

```
-1
```

Cosd: coseno en grados.

```
>> cosd(pi)
```

```
ans =
```

```
0.9985
```

Acos: inverso de coseno en radianes.

```
>> acos(pi)
```

```
ans =
```

```
0.0000 + 1.8115i
```

Acosd: inverso de coseno en grados.

```
>> acosd(pi)
```

```
ans =
```

```
0.0000e+00 + 1.0379e+02i
```

Cosh: coseno hiperbólico en radianes.

```
>> cosh(pi)
```

```
ans =
```

```
11.5920
```

Acosh: coseno hiperbólico inverso en radianes.

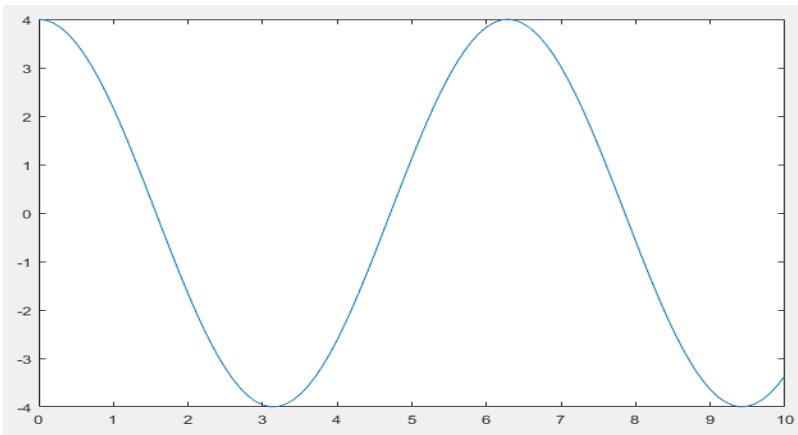
```
>> acosh(pi)
```

```
ans =
```

```
1.8115
```

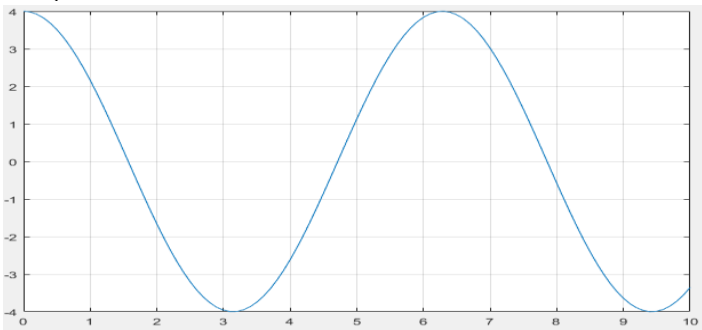
Plot: Graficar funciones.

```
t = 0:0.1:10;  
y = 4 * cos(t);  
plot(t,y);  
% Grafico de la funcion coseno en  
% radianes con amplitud de 4 |
```



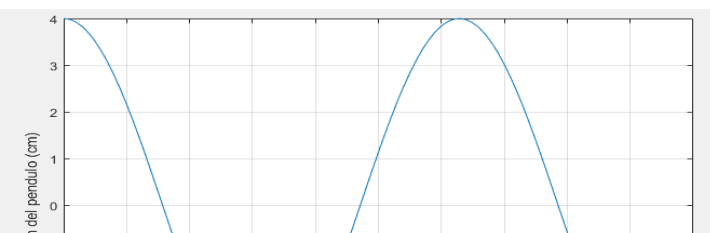
Grid – on - off: Activar cuadrícula o desactivar cuadrícula

```
>> grid
```



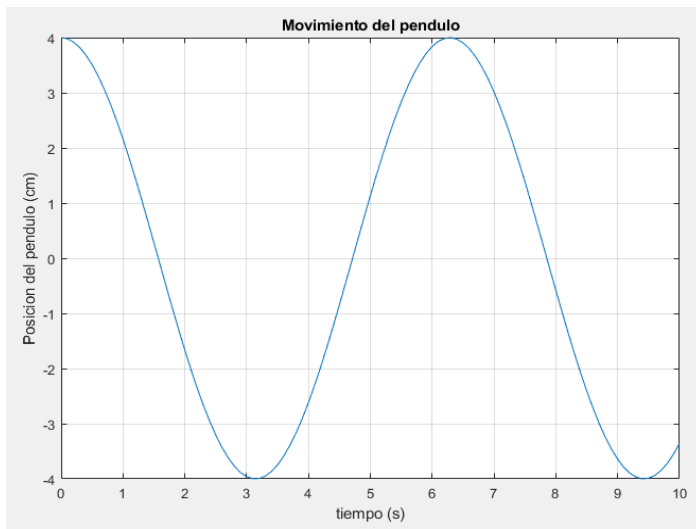
Label : Etiqueta de clase.

```
>> xlabel("tiempo (s)")  
>> ylabel("Posicion del pendulo (cm)")
```



Title: Titulo de la gráfica.

```
>> title("Movimiento del pendulo")
```



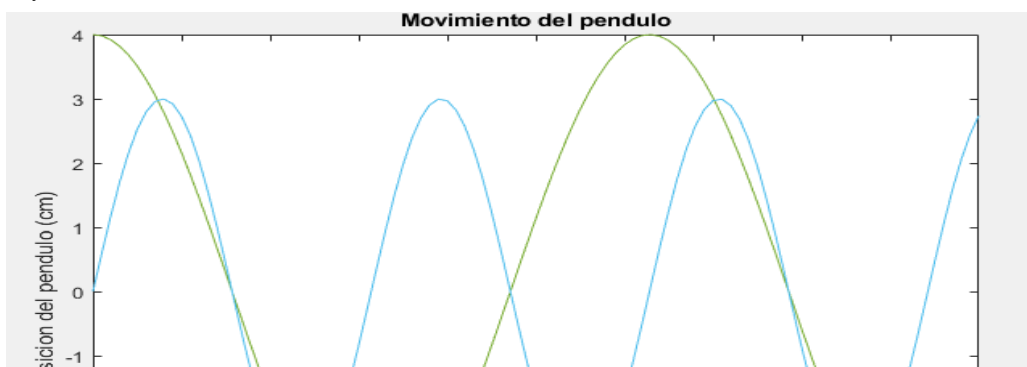
Workspace: Muestra la venta de trabajo donde están las variables

```
>> workspace
```

Workspace	
Name ▲	Value
ans	1.8115
t	1x101 double
x	4
y	1x101 double

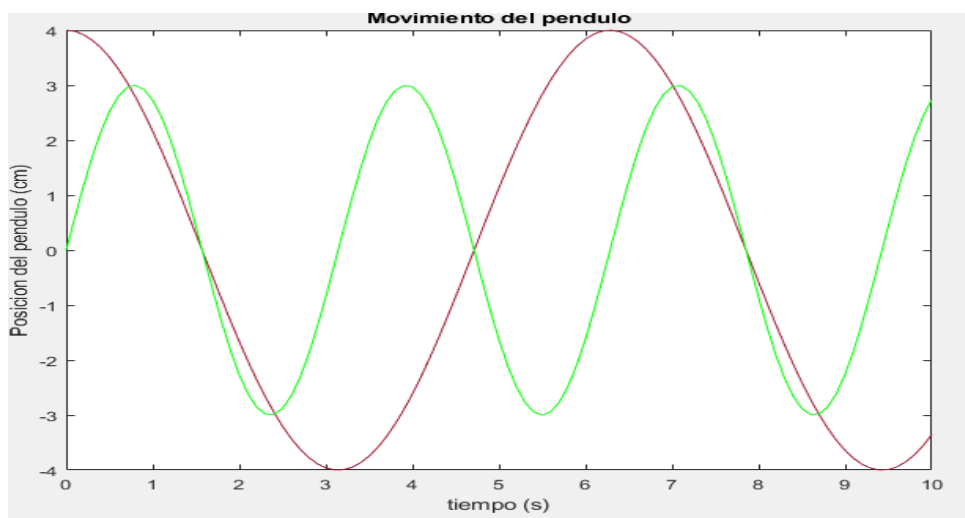
Hold on : Muestro dos graficas distintas en una misma figura.

```
hold on  
y2=3*sin(2*t);  
plot(t,y2);
```



Paleta de colores: se selecciona la letra del color inicial para cambiar en la gráfica.

```
hold on  
y2=3*sin(2*t);  
plot(t,y2,'g');
```



Subplot: Graficar subfunciones.

```
subplot(2,3,4)  
plot(t,y1)
```

Clase semana 3

Cross: Producto cruz entre dos vectores.

```
>> x = [3 2 0];  
y = [2 -1 0];  
>> cross(x,y)  
  
ans =  
  
    0    0   -7
```

dot: Producto escalar entre dos vectores.

```
>> dot(x,y)  
  
ans =  
  
    4
```

det: Hallar la determinante.

```
>> A=[1 2 3; 4 5 6; 7 8 9]  
determinande_A=det(A)  
  
A =  
  
    1    2    3  
    4    5    6  
    7    8    9  
  
determinande_A =  
  
-9.5162e-16
```

inv: Hallar la matriz inversa

```
>> invA=inv(A)
Warning: Matrix is close to singular or badly
scaled. Results may be inaccurate. RCOND =
2.202823e-18.
```

```
invA =

    1.0e+16 *

    0.3153   -0.6305    0.3153
   -0.6305    1.2610   -0.6305
    0.3153   -0.6305    0.3153
```

fliplr: devuelve A con sus columnas volteadas en la dirección izquierda-derecha

```
>> fliplr(A)

ans =

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

flipud: Devuelve las filas en dirección arriba a abajo, alrededor del eje horizontal.

```
>> flipud(A)

ans =

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

reshape: reformula A utilizando el vector de tamaño, sz, para definir size(B)

```
>> A = 1:10;  
B= reshape(A, [5,2])
```

B =

1	6
2	7
3	8
4	9
5	10

rot90: gira la matriz A en sentido contrario a las agujas del reloj en 90 grados.

```
>> rot90(A)
```

ans =

10
9
8
7
6
5
4
3
2
1

Rot90 + n: gira la matriz A n veces en el sentido contrario al reloj.

```
>> rot90(A,2)
```

ans =

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

sqrtm: devuelve la raíz cuadrada principal de la matriz A.

```
A =  
  
     5     -4      1      0      0  
    -4      6     -4      1      0  
     1     -4      6     -4      1  
     0      1     -4      6     -4  
     0      0      1     -4      6  
  
>> X = sqrtm(A)  
  
X =  
  
     2.0015    -0.9971     0.0042     0.0046     0.0032  
    -0.9971     2.0062    -0.9904     0.0118     0.0094  
     0.0042    -0.9904     2.0171    -0.9746     0.0263  
     0.0046     0.0118    -0.9746     2.0503    -0.9200  
     0.0032     0.0094     0.0263    -0.9200     2.2700
```

sqrt: devuelve la raíz cuadrada de cada elemento de la matriz x. Para los elementos de x que son negativos o complejos, `sqrt(x)` produce resultados complejos.

```
>> sqrt(A)
```

```
ans =
```

```
Columns 1 through 4
```

2.2361 + 0.0000i	0.0000 + 2.0000i	1.0000 + 0.0000i	0.0000 + 0.0000i
0.0000 + 2.0000i	2.4495 + 0.0000i	0.0000 + 2.0000i	1.0000 + 0.0000i
1.0000 + 0.0000i	0.0000 + 2.0000i	2.4495 + 0.0000i	0.0000 + 2.0000i
0.0000 + 0.0000i	1.0000 + 0.0000i	0.0000 + 2.0000i	2.4495 + 0.0000i
0.0000 + 0.0000i	0.0000 + 0.0000i	1.0000 + 0.0000i	0.0000 + 2.0000i

```
Column 5
```

0.0000 + 0.0000i
0.0000 + 0.0000i
1.0000 + 0.0000i
0.0000 + 2.0000i
2.4495 + 0.0000i

rand: Números aleatorios distribuidos uniformemente.

```
>> A = rand(2,3,4)
```

```
A(:,:,1) =
```

```
    0.9293    0.1966    0.6160  
    0.3500    0.2511    0.4733
```

```
A(:,:,2) =
```

```
    0.3517    0.5853    0.9172  
    0.8308    0.5497    0.2858
```

```
A(:,:,3) =
```

```
    0.7572    0.3804    0.0759  
    0.7537    0.5678    0.0540
```

```
A(:,:,4) =
```

```
    0.5308    0.9340    0.5688  
    0.7792    0.1299    0.4694
```

```
...
```

permute: Me cambia el orden de las dimensiones en este caso filas, columnas y cantidad.

```
>> B = permute(A,[3 2 1]);
```

```
>> size(B)
```

```
ans =
```

```
     4     3     2
```

diag: Traza Diagonal y el resto 0 en la matriz.

```
>> v = [1 2 3 4]
M2 = diag(v)

v =

    1    2    3    4

M2 =

    1    0    0    0
    0    2    0    0
    0    0    3    0
    0    0    0    4
```

tril: La parte triangular inferior de la matriz.

```
>> M = [1 2 3 4; 5 6 7 8; 9 10 11 12]
tril(M)
```

```
M =

    1    2    3    4
    5    6    7    8
    9   10   11   12
```

```
ans =

    1    0    0    0
    5    6    0    0
    9   10   11    0
```

triu: Matriz triangular superior.

```
triu(M)

ans =
|
    1    2    3    4
    0    6    7    8
    0    0   11   12
```

Guardar un polinomio:

```
p = [1 -1 -26 -24]

p =

     1     -1    -26    -24
```

roots: Calculamos las raíces del polinomio.

```
>> roots(p)

ans =

     6.0000
    -4.0000
    -1.0000
```

polyval: Evaluar el polinomio en un punto dado.

```
>> polyval(p,1)

ans =

    -50
```

conv: devuelve la convolución Circunvolución de los vectores u y v. Si u y v son vectores de coeficientes polinómicos, convolución equivale a multiplicar los dos polinomios.

```
>> u = [1 0 1];
v = [2 7];
>> w = conv(u,v)

w =

     2     7     2     7
```

deconv: División de un polinomio.

```
>> w = conv(u,v)

w =

     2     7     2     7
```

