

# TP 1 MATLAB - INTRODUCCION

MASTER T.E.C.I.

## 1 Operaciones basicas

```
2+5
7-2
3*8
2/5
3^2
10^(-5)
1e-5
sqrt(4)
4^(0.5)
pi
cos(pi)
sin(pi/2)
tan(pi)
exp(0)
log(1)
inv(3)
abs(-1)
sign(-10)
round(0.4)
round(0.7)
ceil(0.4)
floor(0.4)
factorial(10)
```

## 2 Formato de escritura

```
2+5;
2+5,
2+5,7-2,3*8
format short
1/3
format long
1/3
```

### 3 Asignacion de variables

```
x=5
y=10
x*y
clear x
clear all
```

### 4 Vectores, matrices y sus operaciones

#### 4.1 Creacion

```
[1 2 3 4]
[1,2,3]
[1;2;3]
[1,2,3]'
[1,2;3,4]
M=[1,2;3,4]
V=[1,2]
MATO=ones(4,4)
VECO1=ones(1,4)
VECO2=ones(4,1)
MATZ=zeros(4,4)
MATD=diag([1,2,3,4])
MATV=[]
```

#### 4.2 Vectores y listas

```
V2=1 :10
V3=1 :0.5 :3
V4=10 :-1 :5
V5=5 :-1 :10
```

#### 4.3 Operaciones

```
M*V'
det(M)
trace(M)
eig(M)
norm(M,2)
norm(V,1)
norm(V,2)
M2=[0,pi/2;pi,3*pi/2]
cos(M2)
2*V+3
M2i=M2^(-1)
M2i*M2
```

```

M2i2=inv(M2)
M2i2*M2
M3=[1,2,3;4,5,6]
size(M3)
length(V)
M3^2
M3.^2
cos(M3).^2+sin(M3).^2

```

## 5 Coordenadas

```

M3(2,2)
M3(2,2)=55
M3(:,1)
M3(1,:)
V3=[10;11]
M3(:,1)=V3
M3(1,1:2)
i=2
V3(i)
V3(end)
V3(end-1)
VECA=[1,2,3]
VECB=[4,5,6]
VECC=[VECA,VECB]

```

## 6 Polinomios

```

P=[-1,0,2,0,1] (A nivel de notaicon defeneriamos el polinomio  $-x^4 + 2x^2 + 1$ )
polyval(P,0)
polyval(P,1)
roots(P)

```

## 7 Numeros aleatorios con distribucion normal

```

rand,rand,rand
rand(5,3)
randperm(6)

```

## 8 Condiciones

### 8.1 Tests logicos

```

x=1
(x>1)

```

```

(x>=1)
(x<5)
(x<=10)
(x==1)
(x =1)
isa(x,'numeric')
y='aaa'
isa(y,'char')
isa(y,'logical')
isa((y>1),'logical')
exist x
exist xx

```

## 8.2 Operaciones

```

x=1
y=2
(x==1)&(y==2)
(x==1)|(y>2)
xor((x==1),(y>2))

```

## 9 Outputs

### 9.1 Output textual

```

disp('Hola')
MESS1='El valor de x es : '
x=10
MESS2=[MESS1 num2str(x)];
disp(MESS2)

```

### 9.2 Output en ficheros

```

diary('test.txt')
diary on
1+1
MESS1='Se acabo!';
disp(MESS1)
diary off
result=[1 3 10]
fid = fopen(['results.txt'],'w');
fprintf(fid,'Final point :\n');
fprintf(fid,'%0.12f\n ',result);
fprintf(fid,'or :\n');
fprintf(fid,[num2str(result) '\n '],result);
fclose(fid);

```

## 9.3 Output grafico

### 9.3.1 En dimension 1 :

```
dat=1 :.1 :10;
resdat=cos(dat);
figure(1)
clf
plot(dat,resdat)
grid on
xlabel('Data')
ylabel('Result')
dat2=1 :.1 :10;
resdat2=sin(dat2);
figure(2)
clf
subplot(2,1,1)
plot(dat,resdat,'color',[1 0 0])
title('Cos')
subplot(2,1,2)
plot(dat2,resdat2,'color',[0 1 0],'linewidth',3)
title('Sin')
figure(3)
clf
hold on
plot(dat,resdat,'color',[1 0 0])
plot(dat2,resdat2,'color',[0 1 0],'linewidth',3)
legend('Cos','Sin')
saveas(gcf,'test','jpg')
close all
```

### 9.3.2 En dimension 2 :

```
x=-20 :1 :20;
y=-30 :1 :30;
A=rand(length(x),length(y));
[Xgrid,Ygrid]=meshgrid(x,y);
figure(1)
clf
surface(Xgrid,Ygrid,A')
xlabel('xdata')
ylabel('ydata')
zlabel('result')
view(45,45)
```

## 10 Programacion

### 10.1 Creacion de script con Algunos comandos utiles

*Nota : Se tiene que crear en un script a parte :*

```
clc
disp('Hola')
x=10;
disp(['x vale ' num2str(x)])
x=x+1;
disp(['Y ahorra x vale ' num2str(x)])
disp([])
tic
a=input('Entrar el valor de a : ');
atime=ceil(toc);
disp(['a vale ' num2str(a) ' y has tardado ' num2str(atime) ' segundos en responder'])
tic
pause(.5)
toc
AA=[ 1 2 3 ...
4 5 6 7 8]
disp('Escritura en el DD')
fid = fopen(['results.txt'],'w');
fprintf(fid,'Vector AA : \n');
fprintf(fid,'%0.12f \n',AA);
fclose(fid);
disp('Lectura en el DD')
fid = fopen(['results.txt'],'r');
text = fgets(fid)
AAR(1) = str2num(fgets(fid))
AAR(2) = str2num(fgets(fid))
afin = fscanf(fid, '%g')
AAR=[AAR afin]
fclose(fid);
pause
```

### 10.2 El uso del 'if'

*Nota : Se tiene que crear en un script a parte :*

```
clc
disp('Vamos a ver si X es mas grande que Y')
x=input('Valor de x : ');
y=input('Valor de y : ');
disp(['x vale ' num2str(x)])
disp(['y vale ' num2str(y)])
```

```

if (x>y)
disp(['x mas grande que y '])
elseif (x==y)
disp(['x igual a y '])
else
disp(['x mas pequeño que y '])
end

```

### 10.3 El uso del 'for'

*Nota : Se tiene que crear en un script a parte :*

```

clc
disp('Vamos a calcular la raiz de los diez priemros numeros enteros y lso guardamos en una lista')
lr=[] ; La lista es vacia
for i=1 :1 :10
lr(i)=sqrt(i) ;
disp(['la raiz de ' num2str(i) ' es ' num2str(lr(i))])
end

```

### 10.4 El uso del 'while'

*Nota : Se tiene que crear en un script a parte :*

```

clc
disp('Vamos a ver cual es el primer numero entero mas grande o igual que sqrt(99)')
NMG=1 ;
while(NMG<sqrt(100))
NMG=NMG+1 ;
end
resp=[' sqrt(100) vale ' num2str( sqrt(100))...
' yel primer numero entero mas grande o igual es ' num2str(NMG)] ;
disp(resp)

```

### 10.5 Creacion de una funcion

*Nota : Se tiene que crear en un script a parte :*

*Obs. : vamos a crear una funcion de  $R^2$  en  $R$   $(x,y) \rightarrow x^2+y^3$*

```

function [J]=mifunc(x,y)
J=x^2+y^3 ;
end

```

*Nota : Se tiene que ir al workspace :*

mifunc(2,3)

*Nota : Se tiene que crear en un script a parte :*

*Obs. : vamos a crear una funcion de function de R en  $R^2$  ( $x$ )-> ( $sum(x)$ ,  $norm(x,2)$ )*

```
function [J,N]=mifunc2(x)
```

```
N=norm(x,2)
```

```
J=sum(x)
```

```
end
```

*Nota : Se tiene que ir al workspace :*

```
[J,N]=mifunc2(1)
```

## 11 Ejercicios

### 11.1 Ejercicio 1 :

a) Programar una funcion  $f(x, y) = \cos(x) + 2\sin(y)$

b) Dibujar esta funcion en  $[-\pi, \pi] \times [-2\pi, 2\pi]$  con una paso de 0.1

### 11.2 Ejercicio 2 :

Programar una funcion nprimo(n) que calcula el n-esimo numero primo verificando si la entrada 'n' dada es un numero natural, sino devuelve un mensaje de error 'n no es un numero natural'.