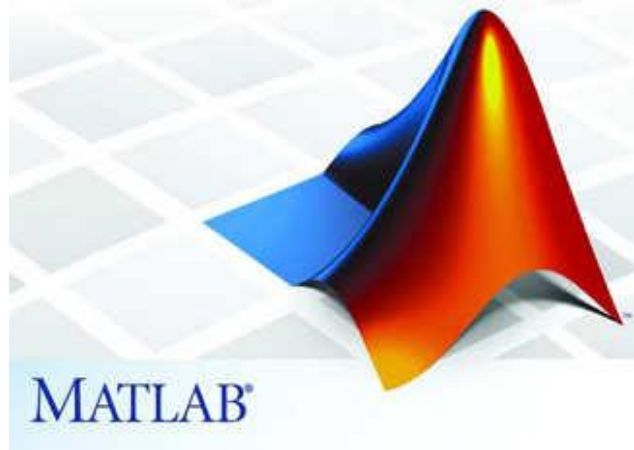
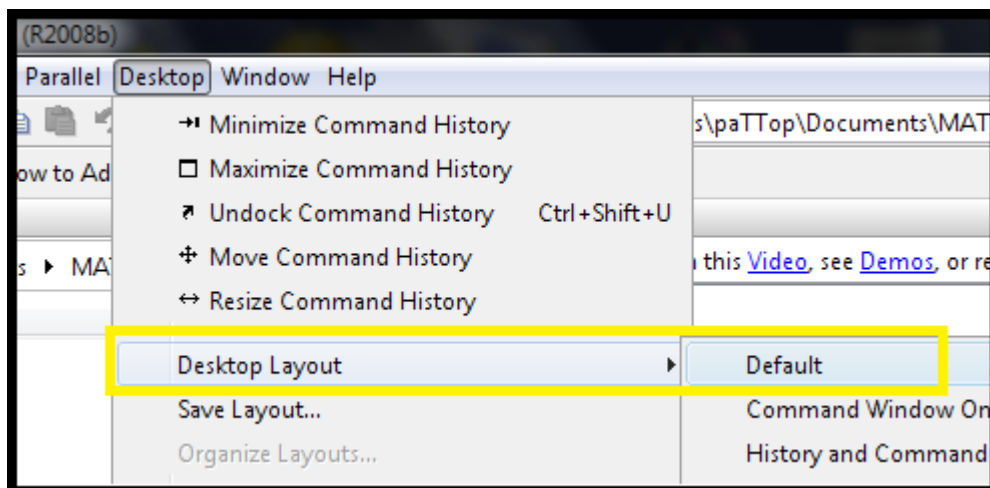


TUTORIAL INTRODUCTORIO DE MATLAB (PVB Parte I)



- Consejos y tips para tener en cuenta.
 - Si la forma de la interface o la disposición de las ventanas cambia de configuración, ya sea que se desordenen o se eliminen por algún motivo, iremos a la pestaña *Desktop* → *Desktop Layout* → *Default*. De esta manera restauraremos la configuración que viene por defecto en matlab.



- Para evitar que matlab muestre automáticamente el resultado en la ventana de comandos, será necesario colocar ";" (sin comillas) al final de la sentencia.

- **Funciones básicas para matrices:**

➤ Creación de una matriz nxn

```
A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]
```

```
A =  
    16     3     2    13  
     5    10    11     8  
     9     6     7    12  
     4    15    14     1
```

➤ Sumatoria de columnas:

```
sum(A)
```

```
ans =  
    34    34    34    34
```

➤ Para sumatoria de filas ocuparemos el apostrofe para definir matriz traspuesta.

```
A'
```

```
ans =  
    16     5     9     4  
     3    10     6    15  
     2    11     7    14  
    13     8    12     1
```

```
sum(A')'
```

```
ans =  
    34  
    34  
    34  
    34
```

➤ Definir los elementos de la diagonal de la matriz.

```
diag(A)
```

```
ans =  
    16  
    10  
     7  
     1
```

- Para la sumatoria de la diagonal y de la antidiagonal.

```
sum(diag(A))
```

```
ans =  
    34
```

```
sum(diag(fliplr(A)))
```

```
ans =  
    34
```

- Suma de elementos individuales de una matriz

```
A(1,4) + A(2,4) + A(3,4) + A(4,4)
```

```
ans =  
    34
```

- Operador *Colon*

```
1:10
```

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

```
100:-7:50
```

```
100    93    86    79    72    65    58    51
```

```
0:pi/4:pi
```

```
0    0.7854    1.5708    2.3562    3.1416
```

- Operador *Colon* Aplicado a la matriz A

```
sum(A(:,end))
```

```
ans =  
    34
```

```
sum(1:16)/4
```

```
ans =  
    34
```

- **Expresiones**

➤ Variables. Para el ejemplo le llamaremos var1.

```
var1=5
```

```
var1 =  
    5
```

```
7*var1
```

```
ans =  
    35
```

➤ Números.

```
3          -99          0.0001  
9.6397238  1.60210e-20    6.02252e23  
1i         -3.14159j    3e5i
```

➤ Funciones.

Para una lista elemental de funciones matemáticas para matlab.

```
help elfun
```

Y para una lista más avanzada.

```
help specfun  
help elmat
```

➤ Además.

| | |
|---------|---|
| pi | 3.14159265 |
| i | Unidad imaginaria, $\sqrt{-1}$ |
| j | Lo mismo que i |
| eps | Precisión relativa punto flotante $\epsilon = 2^{-52}$ |
| realmin | Numero de punto flotante mínimo, 2^{-1022} |
| realmax | Numero de punto flotante máximo, $(2-\epsilon)2^{1023}$ |
| Inf | Infinito |
| NaN | <i>Not a number</i> |

- Algunas expresiones básicas.

```
rho = (1+sqrt(5))/2
rho =
    1.6180
```

```
a = abs(3+4i)
a =
    5
```

```
z = sqrt(besselk(4/3,rho-i))
z =
    0.3730+ 0.3214i
```

```
huge = exp(log(realmax))
huge =
    1.7977e+308
```

```
toobig = pi*huge
toobig =
    Inf
```

- **Trabajo de matrices.**

- Ejemplos de generación básica de matrices.

```
Z = zeros(2,4)
Z =
    0    0    0    0
    0    0    0    0
```

```
F = 5*ones(3,3)
F =
    5    5    5
    5    5    5
    5    5    5
```

```
N = fix(10*rand(1,10))
N =
    9    2    6    4    8    7    4    0    8    4
```

```
R = randn(4,4)
```

```
R =
```

```
    0.6353    0.0860   -0.3210   -1.2316  
   -0.6014   -2.0046    1.2366    1.0556  
    0.5512   -0.4931   -0.6313   -0.1132  
   -1.0998    0.4620   -2.3252    0.3792
```

- Concatenación, o agrupación de variables en una matriz.

```
A = 16.0    3.0    2.0    13.0  
     5.0   10.0   11.0    8.0  
     9.0    6.0    7.0   12.0  
     4.0   15.0   14.0    1.0
```

```
B = [A A+32; A+48 A+16]
```

```
B =
```

```
    16     3     2    13    48    35    34    45  
     5    10    11     8    37    42    43    40  
     9     6     7    12    41    38    39    44  
     4    15    14     1    36    47    46    33  
    64    51    50    61    32    19    18    29  
    53    58    59    56    21    26    27    24  
    57    54    55    60    25    22    23    28  
    52    63    62    49    20    31    30    17
```

- Eliminación de filas y columnas.

```
X = A;
```

```
X(:,2) = []
```

```
X =
```

```
    16     2    13  
     5    11     8  
     9     7    12  
     4    14     1
```

- Eliminación de filas y columnas.

```
X(1,2) = []
```

- Determinante de una matriz.

$$d = \det(A)$$

- Matriz reducida por filas.

$$R = \text{rref}(A)$$

- Matriz Inversa.

$$X = \text{inv}(A)$$

- Cálculo valores propios de la matriz.

$$e = \text{eig}(A)$$

$$e = \begin{array}{c} 34.0000 \\ 8.0000 \\ 0.0000 \\ -8.0000 \end{array}$$

- Operaciones elemento por elemento.

+
-
.*
./
.\
.^

- Ejemplos.

$$A = [1 \ 2; 3 \ 4]$$

$$A = \begin{array}{cc} 1 & 2 \\ 3 & 4 \end{array}$$

$$R = A .* A$$

$$R = \begin{array}{cc} 1 & 4 \\ 9 & 16 \end{array}$$

- Hacer tablas.

```
n = (0:9)';
```

```
pows = [n n.^2 2.^n]
```

```
pows =  
    0     0     1  
    1     1     2  
    2     4     4  
    3     9     8  
    4    16    16  
    5    25    32  
    6    36    64  
    7    49   128  
    8    64   256  
    9    81   512
```

- Ejemplo para desarrollar una tabla de logaritmos.

```
format short g
```

```
x = (1:0.1:2)';
```

```
logs = [x log10(x)]
```

```
logs =  
    1.0     0  
    1.1    0.04139  
    1.2    0.07918  
    1.3    0.11394  
    1.4    0.14613  
    1.5    0.17609  
    1.6    0.20412  
    1.7    0.23045  
    1.8    0.25527  
    1.9    0.27875  
    2.0    0.30103
```

- Ejemplo para uso de información estadística básica.

Se tomaran tres variables: Ritmo cardiaco, peso y horas de ejercicio a la semana. A partir de esto se desarrollara una matriz aplicada a 5 pacientes.

```
D = 72     134     3.2  
    81     201     3.5  
    69     156     7.1  
    82     148     2.4  
    75     170     1.2
```


Obtener la media.

```
mu = mean(D)
mu =
    75.8    161.8    3.48
```

Desviación estándar.

```
sigma = std(D)
sigma =
    5.6303    25.499    2.2107
```

Para obtener una lista de funciones de análisis de datos

```
help datafun
```

Para obtener herramientas estadísticas.

```
help stats
```

➤ Encontrar números primos usando *find* e *isprime*.

Para el ejemplo usaremos un cuadro mágico con la sentencia *magic*.

```
B = magic(4)
B =
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

Para encontrar la ubicación usaremos *find*.

```
k = find(isprime(A))'
k =
     2     5     9    10    11    13
```

Para identificarlos.

```
A(k)
ans =
     5     3     2    11     7    13
```

- La función *fix* redondea una cifra al número entero menor.

```
fix(4.1)
ans =
     4

fix(4.9)
ans =
     4
```

- Uso de función *format*.

```
x = [4/3 1.2345e-6]

format short

    1.3333    0.0000

format short e

    1.3333e+000    1.2345e-006

format short g

    1.3333    1.2345e-006

format long

    1.3333333333333333    0.00000123450000

format long e

    1.3333333333333333e+000    1.2345000000000000e-006

format long g

    1.3333333333333333    1.2345e-006

format bank

    1.33    0.00

format rat

    4/3    1/810045

format hex

    3ff5555555555555    3eb4b6231abfd271
```

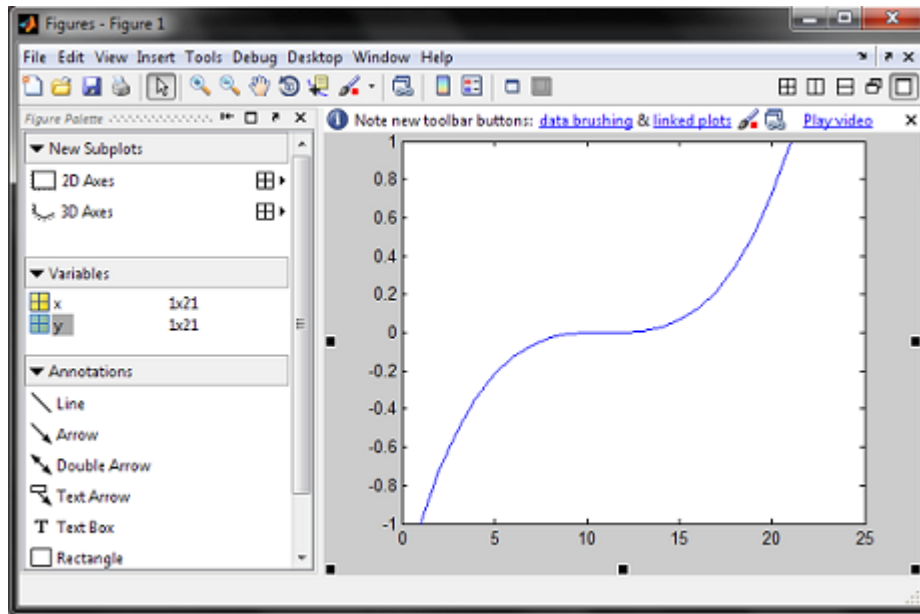
- **Funciones Gráficas:**

➤ Ejemplo de gráficas en matlab

```
x = -1:.1:1;
```

```
y = x.^3;
```

```
plottools
```



➤ Otro ejemplo de gráficas en matlab.

```
t = 0:pi/20:2*pi;
```

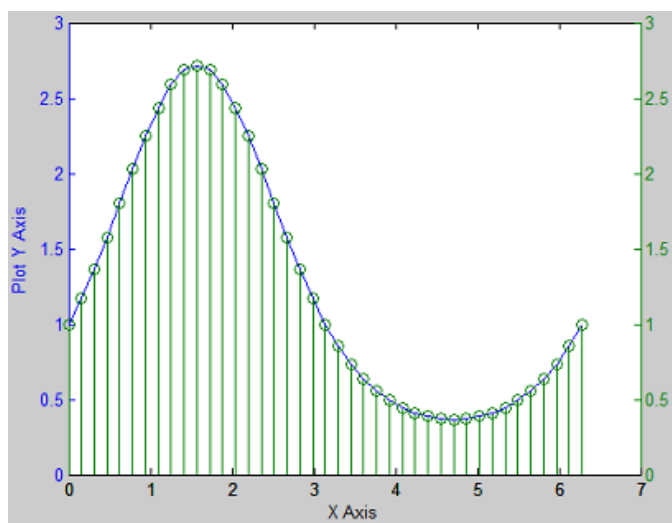
```
y = exp(sin(t));
```

```
plotyy(t,y,t,y,'plot','stem')
```

```
xlabel('X Axis')
```

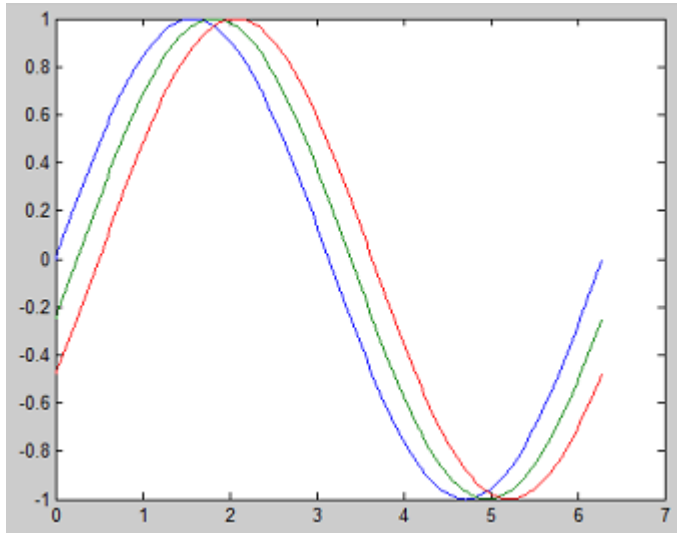
```
ylabel('Plot Y Axis')
```

```
title('Two Y Axes')
```



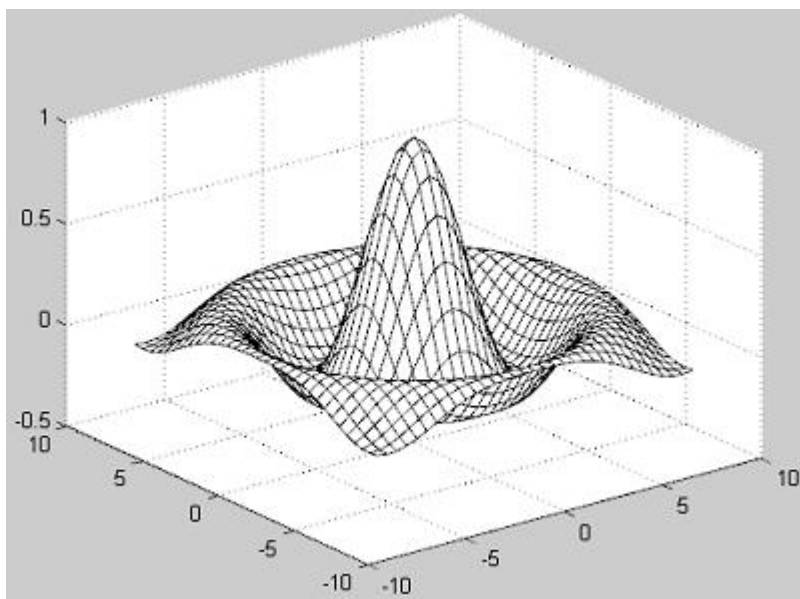
➤ Gráficas múltiples.

```
x = 0:pi/100:2*pi;  
y = sin(x);  
y2 = sin(x-.25);  
y3 = sin(x-.5);  
plot(x,y,x,y2,x,y3)
```



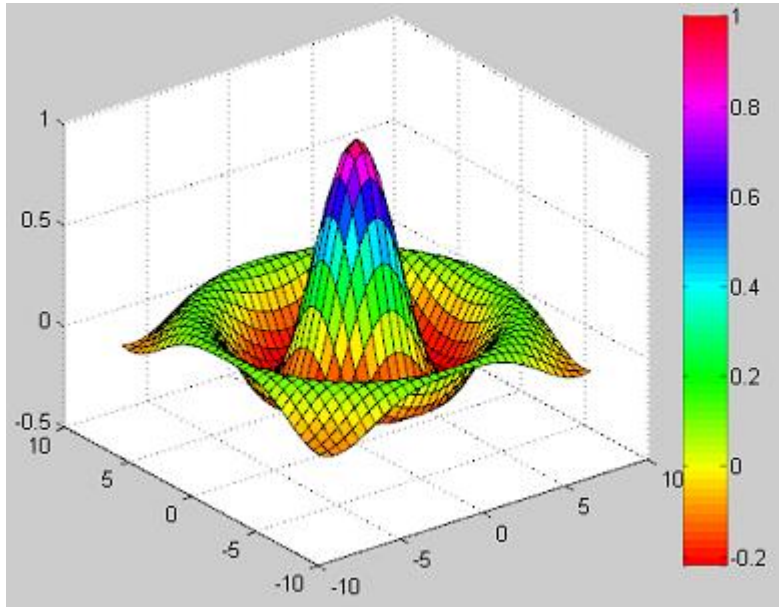
➤ Gráficas 3D.

```
[X,Y] = meshgrid(-8:.5:8);  
R = sqrt(X.^2 + Y.^2) + eps;  
Z = sin(R)./R;  
mesh(X,Y,Z, 'EdgeColor', 'black')
```



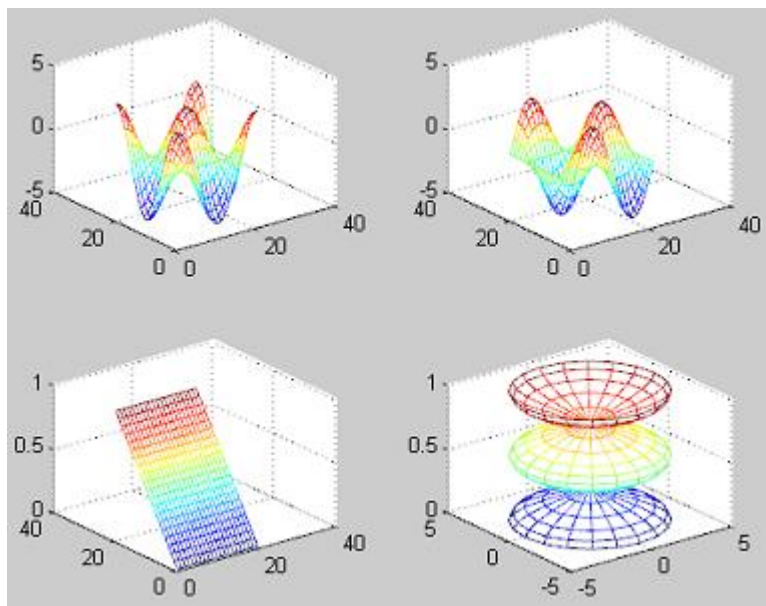
- Para rango de color.

```
surf(X,Y,Z)  
colormap hsv  
colorbar
```



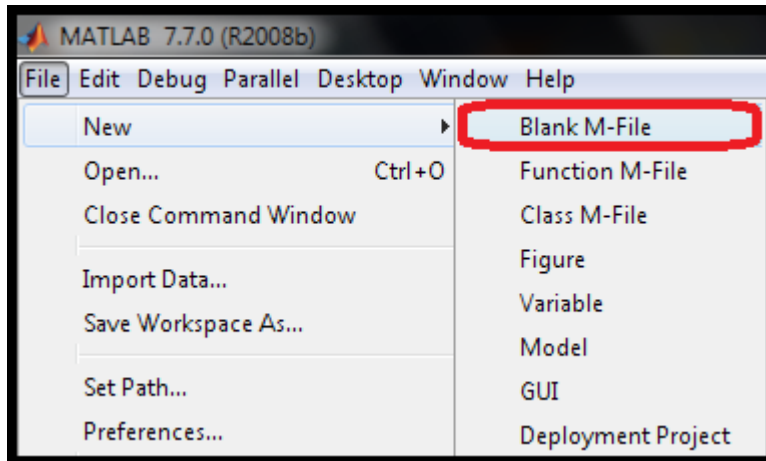
- Desplegar varios gráficos en un cuadro.

```
t = 0:pi/10:2*pi;  
[X,Y,Z] = cylinder(4*cos(t));  
subplot(2,2,1); mesh(X)  
subplot(2,2,2); mesh(Y)  
subplot(2,2,3); mesh(Z)  
subplot(2,2,4); mesh(X,Y,Z)
```



- **Programar en matlab.**

- Matlab cuenta con un editor incorporado para desarrollar programas, para acceder a él iremos a *File* → *New* → *Blank M-file*.



- Para tener una idea más inmediata desarrollaremos un programa sencillo para calcular los valores de una ecuación cuadrática.

```

1      %ec: Script para calcular ecuacion cuadraticas ]1
2      %ax^2 + bx + c = 0
3
4      clear all ]2
5      clc
6
7      format compact; ]3
8
9      %Ingreso de coeficientes a b c
10     a = input('Ingresar coeficiente a: ');
11     b = input('Ingresar coeficiente b: ');
12     c = input('Ingresar coeficiente c: ');
13     disp(' ') ]5
14
15     % ingreso de la formula general ]6
16     x = -b/(2*a);
17     y = sqrt(b^2-4*a*c)/(2*a);
18
19     % Soluciones
20     s1 = x+y;
21     disp('Valor de x1: '),disp(s1);
22     s2 = x-y;
23     disp('Valor de x2: '),disp(s2); ]7

```

1.- Para dar nombre al programa y establecer comentarios con información sobre el mismo, sin que esto se muestre cuando lo ejecutemos, utilizamos este símbolo “%” al comienzo. Esto nos servirá para tener una noción de lo que hacen las sentencias y así identificar problemas o reprogramación de algoritmos de una manera más rápida.

2.- Sentencia aplicada para borrar variables y la ventana de comandos (*command window*) respectivamente.

3.- Definición del formato numérico, notaciones, cantidad de decimales, etc.

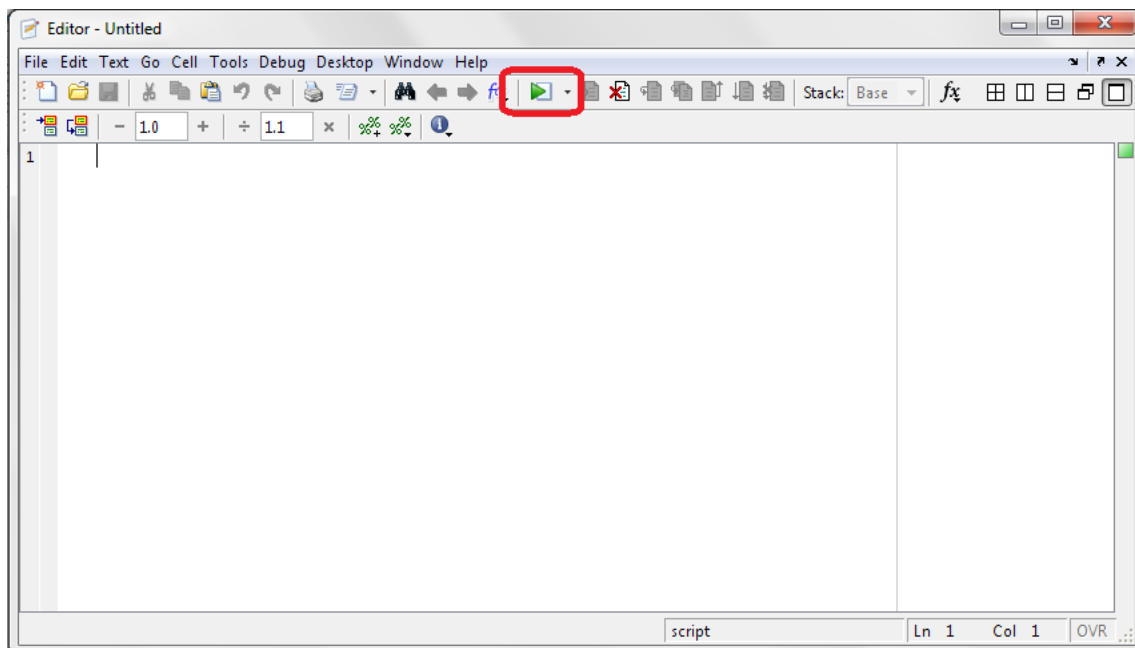
4.- Sentencia utilizada para ingresar variables en el programa.

5.- Se utiliza para dejar espacios entre sentencias con un fin visual.

6.- Ingreso de las formulas con el cual el programa trabajará en función de las variables expresadas en el punto 4.

7.- Sentencia comúnmente utilizada para mostrar los resultados, en base a los cálculos efectuado en 6.

- Finalmente para correr el programa hacemos click en *save file and run*.



- **Bibliografia**

- MathWorks Documentation - MATLAB V7 Introductory and Programming
- MATLAB Programming - David Kuncicky