



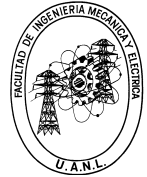
UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

LABORATORIO DE INGENIERÍA DE CONTROL

PRACTICA N° 1

INTRODUCCIÓN Y COMANDOS DEL MATLAB



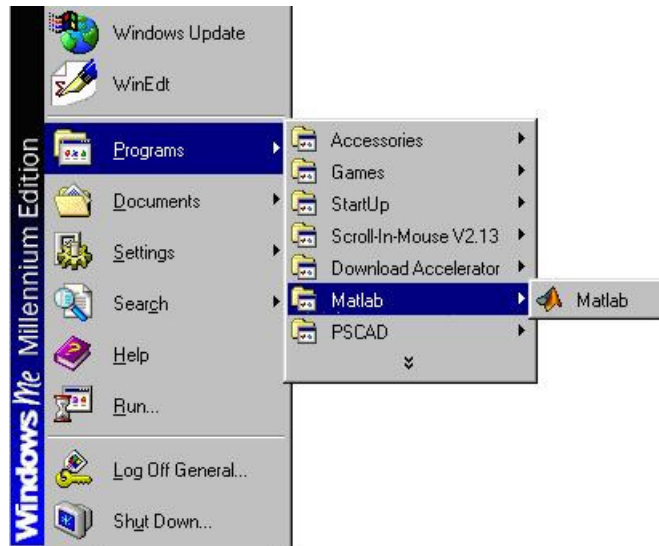
OBJETIVO

Familiarizarse con el modo de operación del matlab y conocer los comandos principales que permitan obtener un mejor provecho al manejar el mismo.

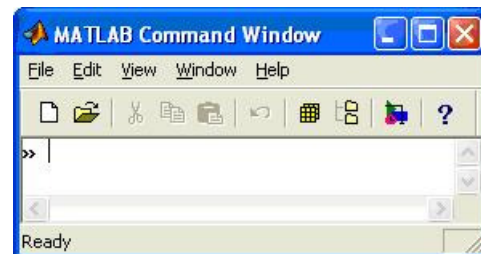
SESIÓN SIMPLE DE MATLAB

Para iniciar el matlab oprima dos veces sobre el icono del escritorio de Windows.

O bien, por el menú inicio



Al iniciar el matlab observará la siguiente pantalla.



COMANDOS Y EXPRESIONES

Los comandos y las expresiones se teclean en matlab como se muestra en la siguiente figura. El resultado de una expresión se almacena automáticamente en la variable **ans** y se despliega con cuatro cifras decimales por default.



VARIABLES

Matlab reconoce la diferencia entre minúsculas y mayúsculas, así volts, Volts y VOLTS son tres variables diferentes. Las variables deben empezar con letra y no deben contener signos de puntuación puesto que estos tienen otro significado en matlab.

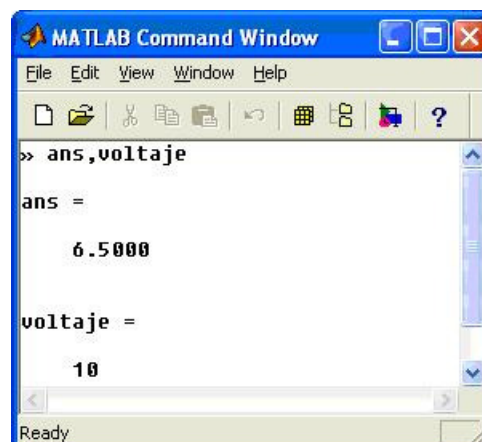


Nótese en la figura anterior como la definición de las variables voltaje y corriente van seguidas por punto y coma (;), no así la declaración de la expresión para la potencia. Para matlab, todo lo que termina en punto y coma el resultado no se muestra en la pantalla. Puede usarse también la coma (,) como separador de variables, pero el resultado se mostrará en la pantalla.

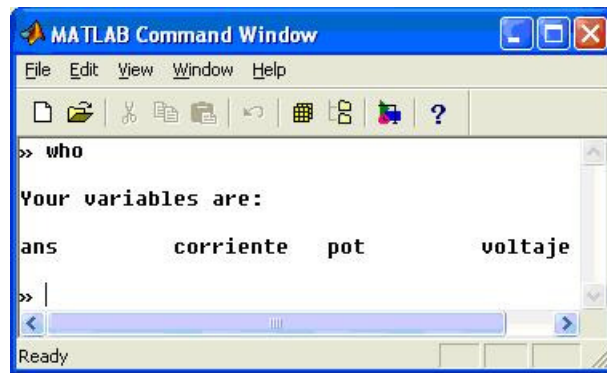
EL ESPACIO DE TRABAJO DE MATLAB (WORKSPACE)

Matlab mantiene el valor de todas las variables utilizadas, para saber que valor tiene una variable solo teclee la variable y **ENTER**. Por ejemplo, para verificar el valor de las variables **ans** y **voltaje**, dadas anteriormente.

Nota: Los valores mostrados pueden cambiar en función de los valores que se hayan tecleado



Si no se recuerdan las variables utilizadas puede usarse el comando **who**, que nos muestra todas las variables que hasta ahora se han usado.

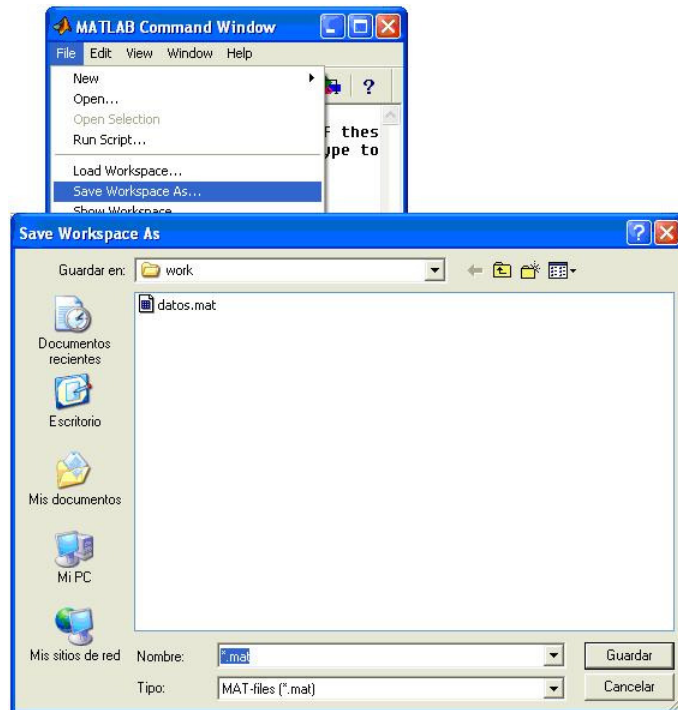


SALVANDO Y RECUPERANDO DATOS

En adición a mantener el valor de las variables, Matlab puede almacenar como archivo todas las variables usadas y recuperarlas posteriormente para otra sesión.

Para guardar las variables se usa el comando **Save Workspace As** del menú File.

Esto abre el siguiente cuadro de diálogo donde se dará el nombre del archivo en el que se guardarán los datos.



Para borrar todas o una de las variables almacenadas se usa el comando **clear**. o **Clear**, para borrar todas las variables se usa **clear** o **clear all**, se requiere especial cuidado con este comando, ya que no podrán recuperarse las variables.

Para borrar una o más variables el comando **clear** va seguido del nombre de la variable o variables.

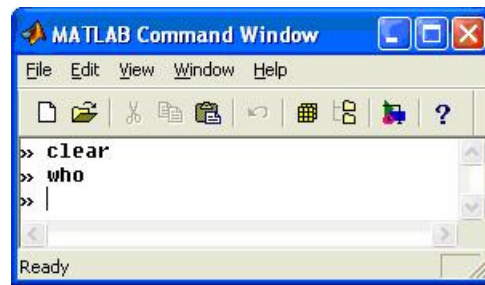
El siguiente ejemplo borra las variables **ans** y **pot**



Nótese que los comandos de matlab deben teclearse en minúsculas, de no hacerse así, se mandará un mensaje de error.

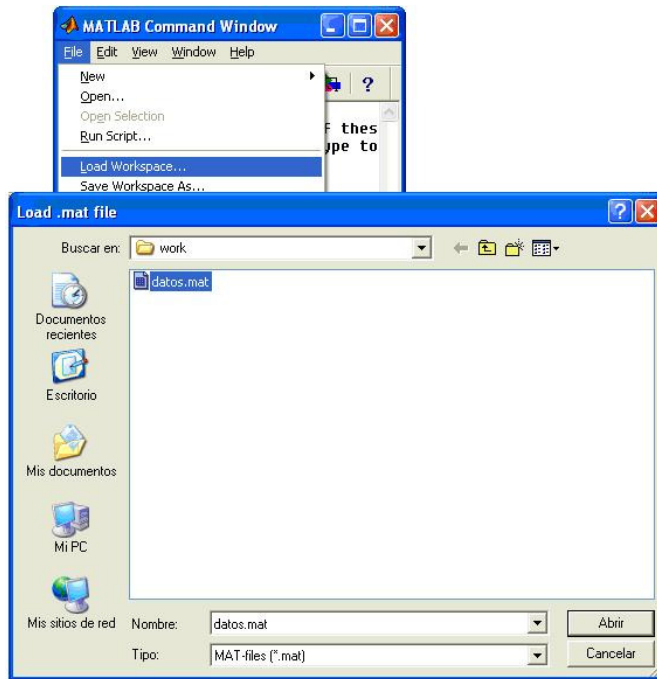
Para borrar todas las variables, se usa **clear** o **clear all**.

Note que el comando **who** nos muestra que no hay ninguna variable almacenada.



Para recuperar las variables que se almacenaron anteriormente con el comando **Save Workspace As** se usa el comando **Load Workspace**, del menú *File*

Esto abre el cuadro de diálogo donde se dará el nombre del archivo donde se almacenaron los datos (datos.mat).



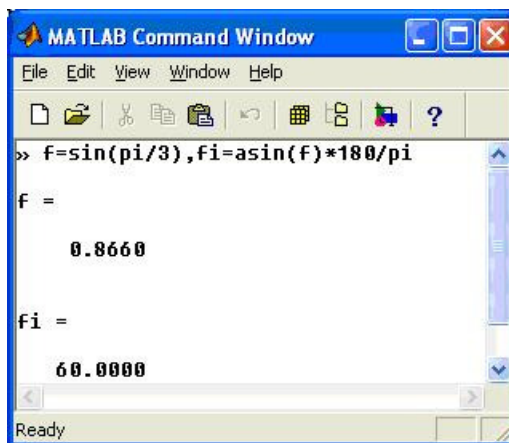
Puede verificar que las variables se hayan recuperado usando el comando **who**.

Otros comandos de matlab que puede probar son: **whos**, **clc**, **help**.

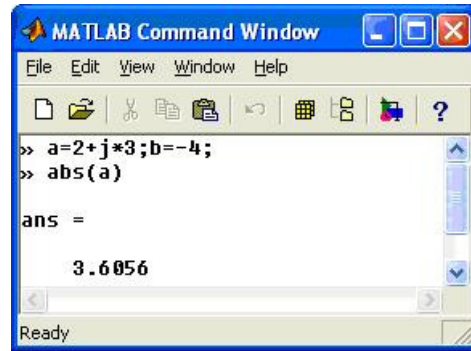
FUNCIONES

Las funciones de matlab son muchas, pero consideraremos aquí solo las básicas como: **abs()**, **sin()**, **asin()**, **angle()**, **exp()**, **log(x)**, **log10()**, **sqrt()** e **imag()**.

Función Seno y seno inverso (El ángulo debe estar en radianes), comandos **sin** y **asin**.



Valor absoluto de un número, comando **abs**



```
MATLAB Command Window
File Edit View Window Help
>> a=2+j*3;b=-4;
>> abs(a)

ans =

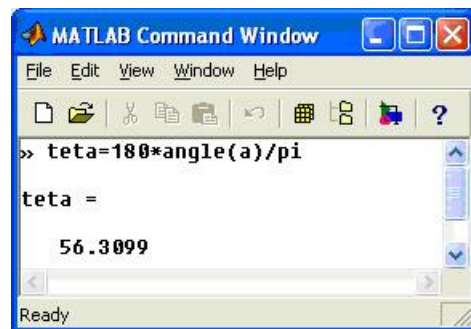
    3.6056

Ready
```

Ángulo en radianes de un número complejo, comando **angle**

Para convertir un ángulo de radianes a grados

$$\text{grados} = \text{radianes} * 180^\circ / \pi$$



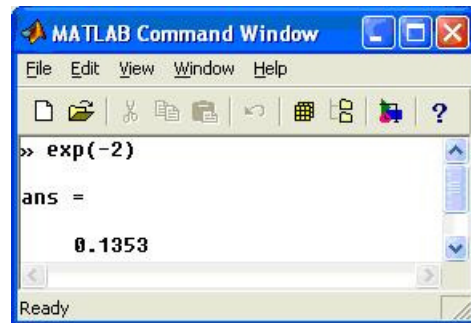
```
MATLAB Command Window
File Edit View Window Help
>> teta=180*angle(a)/pi

teta =

    56.3099

Ready
```

Exponencial de un número, comando **exp**



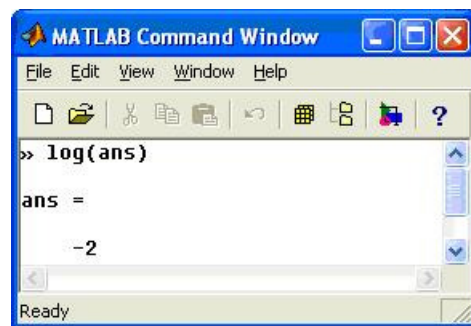
```
MATLAB Command Window
File Edit View Window Help
>> exp(-2)

ans =

    0.1353

Ready
```

Logaritmo natural o base e, comando **log**



```
MATLAB Command Window
File Edit View Window Help
>> log(ans)

ans =

    -2

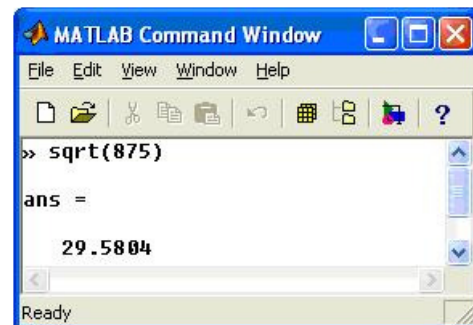
Ready
```

Logaritmo base 10, comando **log10**



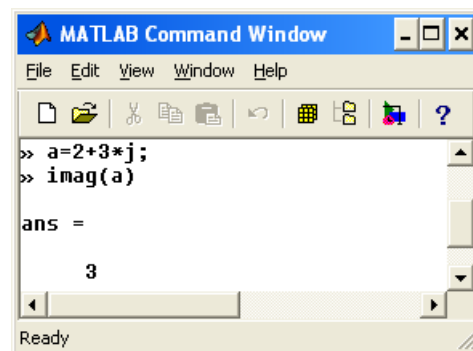
```
MATLAB Command Window
File Edit View Window Help
>> log10(0.1)
ans =
    -1.0000
Ready
```

Raíz cuadrada, comando **sqrt**



```
MATLAB Command Window
File Edit View Window Help
>> sqrt(875)
ans =
    29.5804
Ready
```

Obtención del número imaginario de un número complejo, comando **imag**



```
MATLAB Command Window
File Edit View Window Help
>> a=2+3*j;
>> imag(a)
ans =
     3
Ready
```

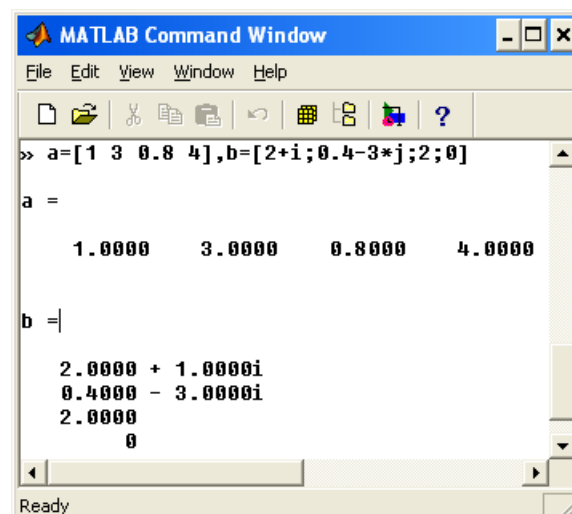
Otras funciones son **cos()**, **acos()**, **tan()**, **atan()**, **real()**, **conj()**

Otras funciones son **cos()**, **acos()**, **tan()**, **atan()**, **real()**, **conj()**

ARREGLOS SIMPLES (VECTORES)

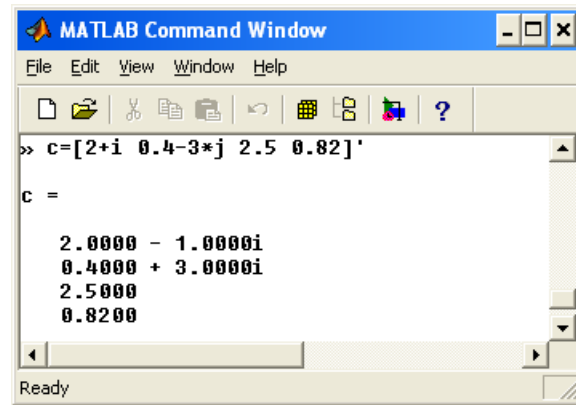
Los vectores en matlab se definen por medio de paréntesis rectangulares, los elementos de la fila están separados por espacios o coma (,).

Para definir un vector columna, cada elemento del vector es separado por punto y coma (;)



```
MATLAB Command Window
File Edit View Window Help
>> a=[1 3 0.8 4],b=[2+i;0.4-3*j;2;0]
a =
    1.0000    3.0000    0.8000    4.0000
b =
    2.0000 + 1.0000i
    0.4000 - 3.0000i
    2.0000
     0
Ready
```

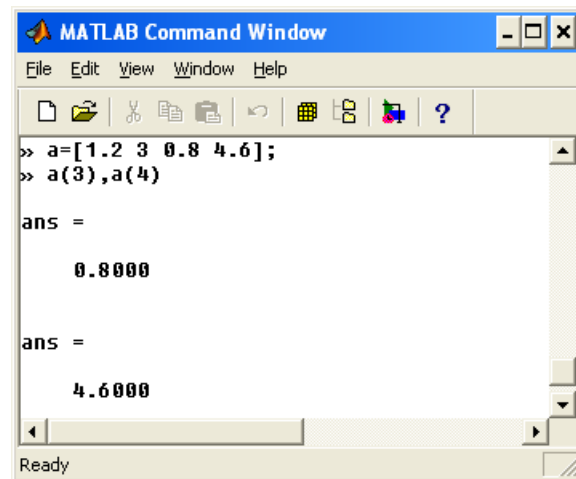
Para obtener la transpuesta de un vector se utiliza la comilla ('). Cuando se obtiene la transpuesta de un vector que tiene elementos complejos se obtiene la transpuesta conjugada.



```

MATLAB Command Window
File Edit View Window Help
>> c=[2+i 0.4-3*j 2.5 0.82]';
c =
    2.0000 - 1.0000i
    0.4000 + 3.0000i
    2.5000
    0.8200
  
```

Para direccionar los elementos de un vector, por ejemplo, el tercero y cuarto elemento de **a**. Se utilizará **a(3)** y **a(4)**



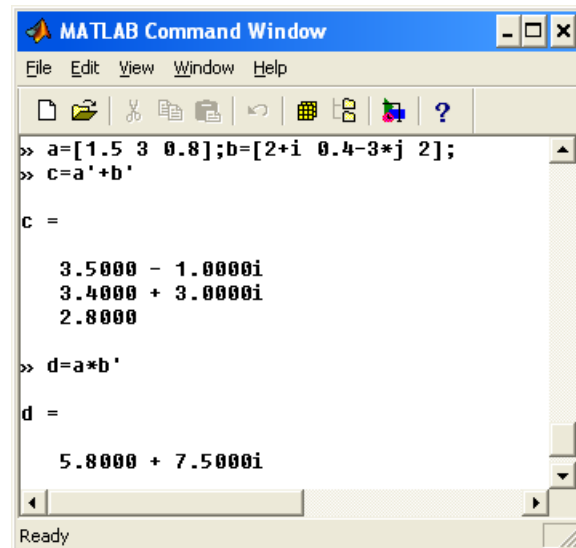
```

MATLAB Command Window
File Edit View Window Help
>> a=[1.2 3 0.8 4.6];
>> a(3),a(4)

ans =
    0.8000

ans =
    4.6000
  
```

La operación entre vectores obedece las reglas del álgebra de vectores, ejemplos



```

MATLAB Command Window
File Edit View Window Help
>> a=[1.5 3 0.8];b=[2+i 0.4-3*j 2];
>> c=a'+b'

c =
    3.5000 - 1.0000i
    3.4000 + 3.0000i
    2.8000

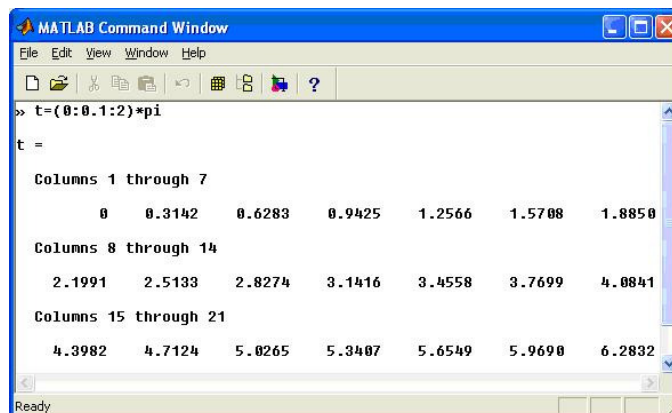
>> d=a*b'

d =
    5.8000 + 7.5000i
  
```


Otra forma de generar vectores de una forma simple es

$t = (0:0.1:2) * \pi$

el cual nos genera el vector **t** con 21 elementos, los elementos comienzan en 0, tienen incrementos de 0.1 hasta llegar a 2, todos los elementos son multiplicados por π (3.1416), esto es:



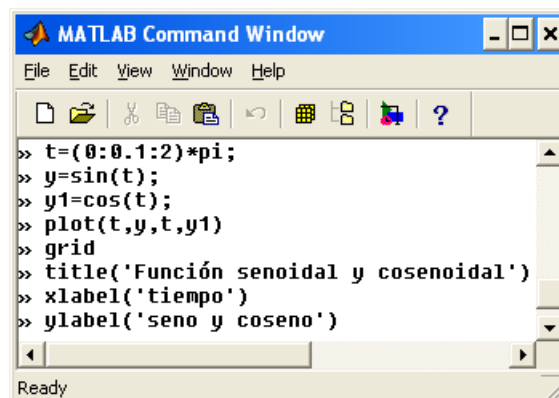
GRÁFICOS SIMPLES

Con el vector **t** definido anteriormente se puede obtener la función seno y la función coseno

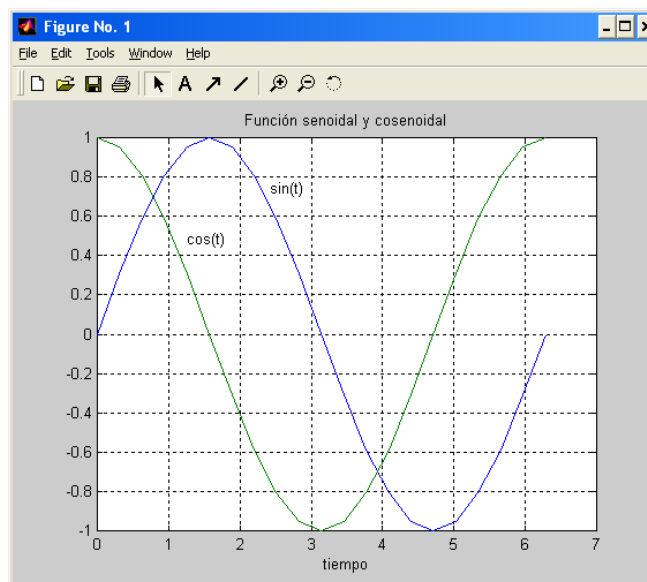
**$y = \sin(t);$
 $y1 = \cos(t)$**

Este comando evalúa la función seno y función coseno para cada uno de los 21 elementos de **t**, y genera un vector **y** y **y1** con 21 elementos cada uno.

Se grafica estas funciones con el comando **plot**



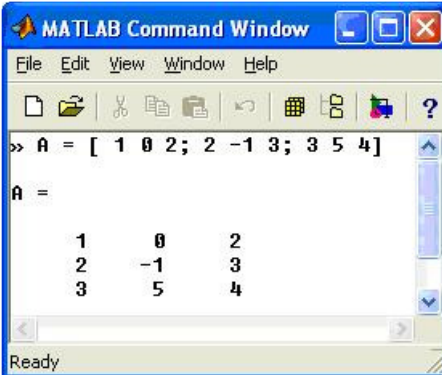
El procedimiento anterior nos da la siguiente gráfica.



MATRICES

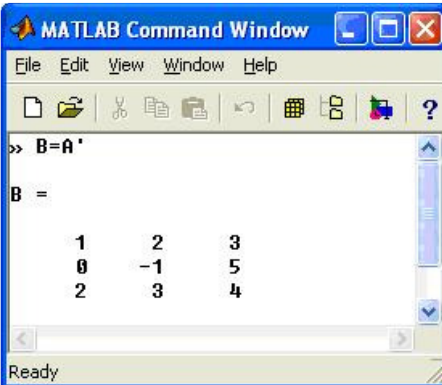
La forma de definir matrices en matlab es parecida a la de vectores; cada elemento de una fila está separado ya sea por espacio o bien por coma, un renglón se determina por un punto y coma al final del mismo, por ejemplo para crear la matriz A

$$A = \begin{bmatrix} 1 & 0 & 2 \\ 2 & -1 & 3 \\ 3 & 5 & 4 \end{bmatrix}$$



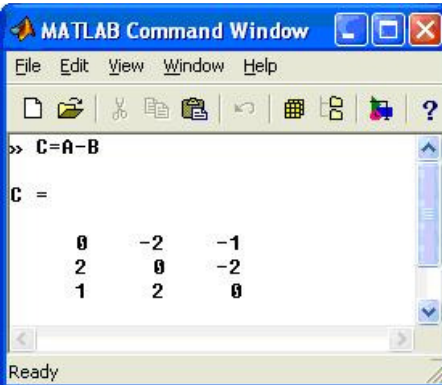
```
MATLAB Command Window
File Edit View Window Help
>> A = [ 1 0 2; 2 -1 3; 3 5 4]
A =
     1     0     2
     2    -1     3
     3     5     4
Ready
```

La transpuesta (') de esta matriz esta dada por:



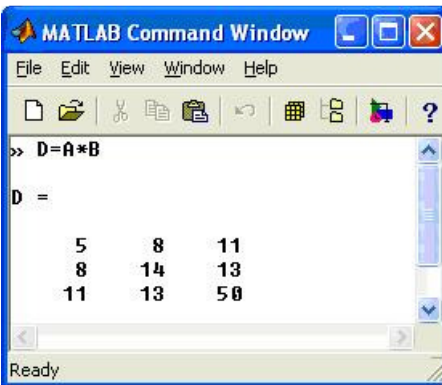
```
MATLAB Command Window
File Edit View Window Help
>> B=A'
B =
     1     2     3
     0    -1     5
     2     3     4
Ready
```

También, las matrices obedecen las leyes del álgebra de matrices, la resta de las matrices anteriores es



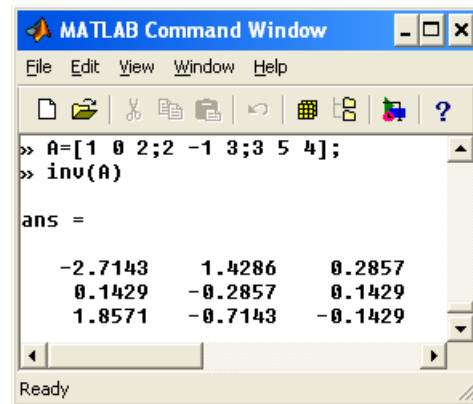
```
MATLAB Command Window
File Edit View Window Help
>> C=A-B
C =
     0    -2    -1
     2     0    -2
     1     2     0
Ready
```

El producto de éstas matrices es



```
MATLAB Command Window
File Edit View Window Help
>> D=A*B
D =
     5     8    11
     8    14    13
    11    13    50
Ready
```

La inversa de la matriz **A**, comando **inv**

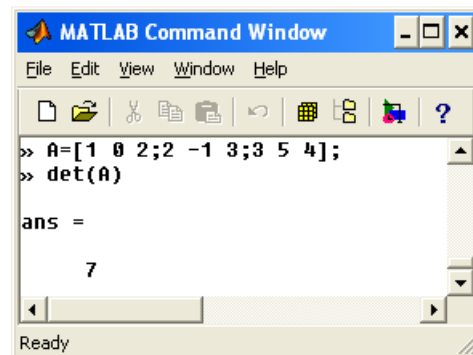


```
MATLAB Command Window
File Edit View Window Help
>> A=[1 0 2;2 -1 3;3 5 4];
>> inv(A)

ans =

    -2.7143    1.4286    0.2857
     0.1429   -0.2857    0.1429
     1.8571   -0.7143   -0.1429
```

El determinante de la matriz **A**, comando **det**



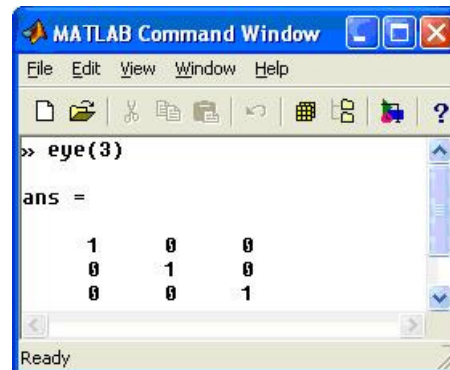
```
MATLAB Command Window
File Edit View Window Help
>> A=[1 0 2;2 -1 3;3 5 4];
>> det(A)

ans =

     7
```

MATRICES ESPECIALES

Matriz identidad o unitaria, comando **eye**

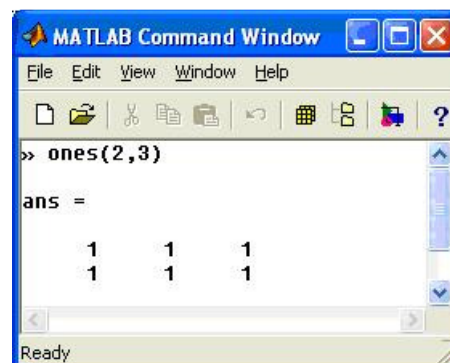


```
MATLAB Command Window
File Edit View Window Help
>> eye(3)

ans =

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

Matriz de unos, comando **ones**



```
MATLAB Command Window
File Edit View Window Help
>> ones(2,3)

ans =

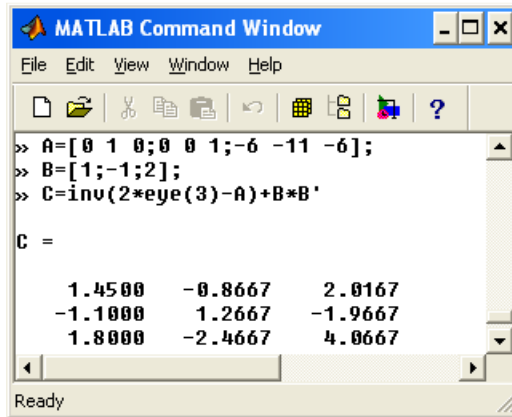
     1     1     1
     1     1     1
```

Ejemplo 1

Resolver la siguiente ecuación matricial

$$C = (2I - A)^{-1} + B * B'$$

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -6 & -11 & -6 \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}$$



```
MATLAB Command Window
File Edit View Window Help
>> A=[0 1 0;0 0 1;-6 -11 -6];
>> B=[1;-1;2];
>> C=inv(2*eye(3)-A)+B*B'

C =

    1.4500    -0.8667    2.0167
   -1.1000     1.2667   -1.9667
    1.8000    -2.4667    4.0667
```

Ejemplo 2

Grafique la siguiente ecuación
(Respuesta a una entrada escalón de un sistema de 2do. Orden bajo amortiguado)

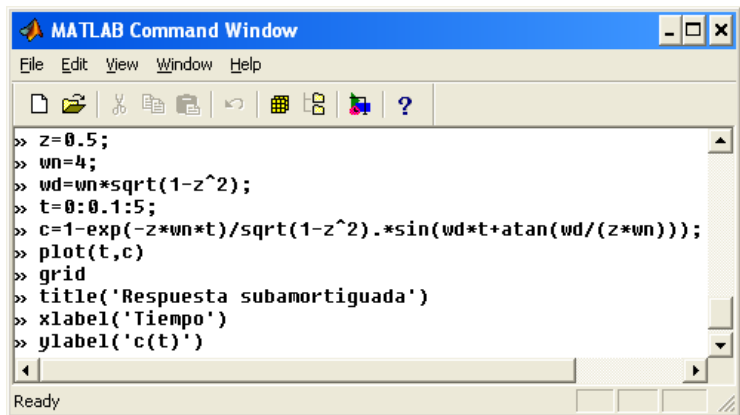
$$c = 1 - \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}} \sin\left(\omega_d t + \tan^{-1} \frac{\omega_d}{\zeta\omega_n}\right)$$

para un tiempo t de 0 a 3, considerando una $\zeta = 0.5$ y $\omega_n = 4$

$$\omega_d = \omega_n \sqrt{1-\zeta^2}$$

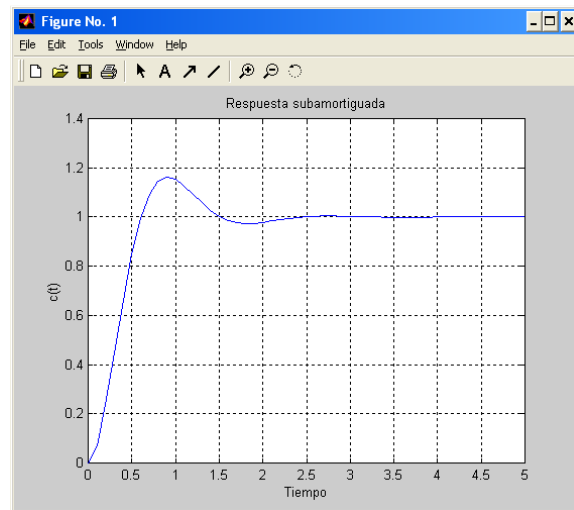
El procedimiento en el matlab para obtener la grafica de c sería

Nota: el operador $.$ * sirve para multiplicar las matrices elemento por elemento.



```
MATLAB Command Window
File Edit View Window Help
>> z=0.5;
>> wn=4;
>> wd=wn*sqrt(1-z^2);
>> t=0:0.1:5;
>> c=1-exp(-z*wn*t)/sqrt(1-z^2).*sin(wd*t+atan(wd/(z*wn)));
>> plot(t,c)
>> grid
>> title('Respuesta subamortiguada')
>> xlabel('Tiempo')
>> ylabel('c(t)')
```

La grafica de respuesta $c(t)$ quedaría



REPORTE

1. Resolver las siguientes ecuaciones matriciales

$$D = (C')^{-1} + |C| * I + A * B$$

$$E = B * C - 2 * A'$$

$$F = A .* B'$$

donde

$$A = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix} \quad B = \begin{bmatrix} -2 & 1 & 3 \end{bmatrix} \quad C = \begin{bmatrix} 1 & -1 & 0 \\ 3 & 2 & -1 \\ 2 & 0 & 1 \end{bmatrix}$$

C' = es la transpuesta de la matriz C

$|C|$ = es el determinante de la matriz C

I = es la matriz identidad de 3×3

$.*$ = producto de elemento por elemento

C^{-1} = es la inversa de la matriz C

2. Grafique las siguientes ecuaciones

a) $c = 1 - e^{-\omega_n t} (1 + \omega_n t)$

para un tiempo t de 0 a 3, considerando una $\omega_n = 4$

b) $y = \sin(2t) + \cos(4t)$

para un tiempo t de 0 a 5

c) $y = 30t^2 e^{-4t}$

para un tiempo t de 0 a 4

d) $y = \sin(4t)\cos(2t)$

para un tiempo t de 0 a 5

Nota: no olvide utilizar el operador $.*$ en el producto de los vectores

3. Conclusiones