

# Pre-Laboratorio 1 MA-33A 2007:

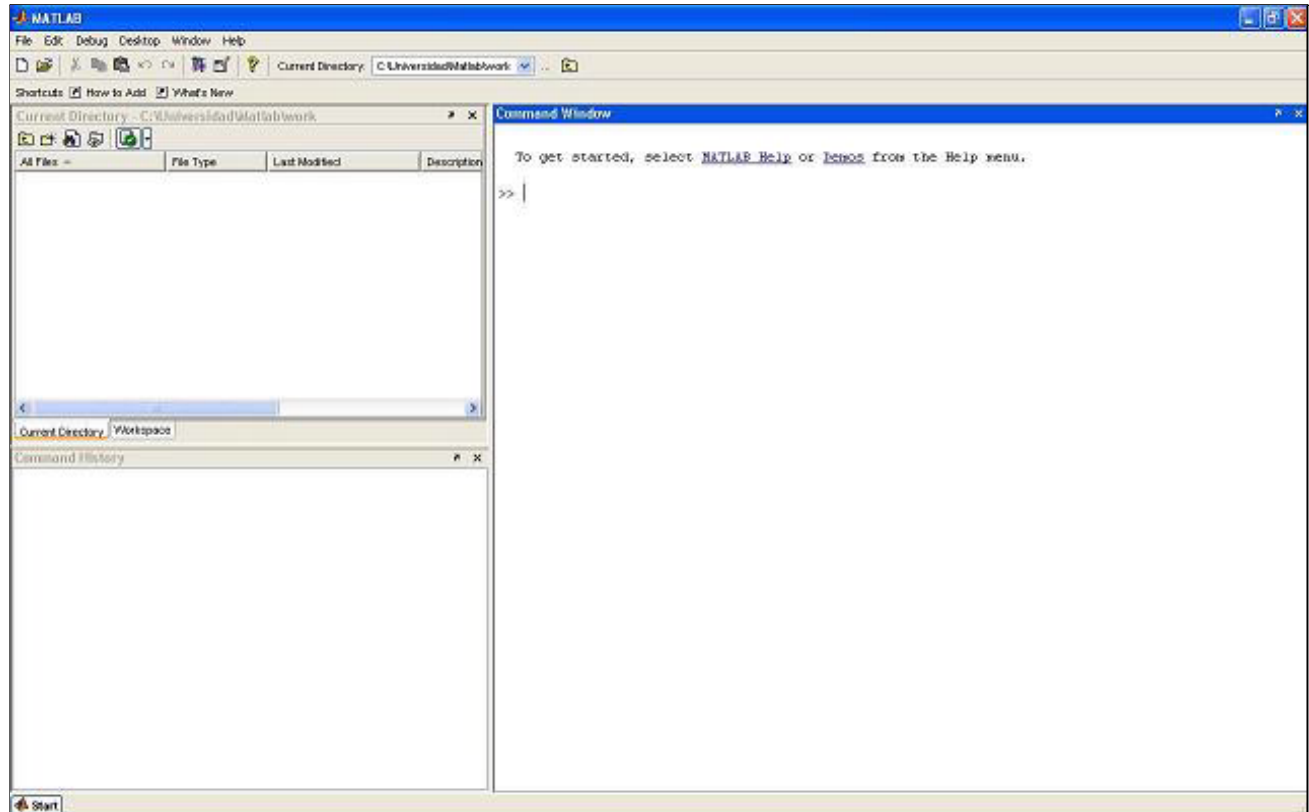
## Introducción a Matlab

Gonzalo Hernández - Gonzalo Rios

UChile - Departamento de Ingeniería Matemática

### 1 Interfaz de Matlab

Al ejecutar el programa Matlab, se abrirá una ventana con la siguiente interfaz gráfica:



Explicaremos cada una de las subventanas:

## 1.1 Command

La ventana llamada Command es la ventana principal de Matlab. En esta ventana es donde se definen las variables, se escriben pequeños programas, se ejecutan funciones y se muestran los resultados.

## 1.2 Current Directory

Muestra la dirección del directorio de trabajo, es decir, donde se guardan las funciones programadas para luego ocuparlas. Por defecto, la dirección del directorio es "C:\Matlab\work", pero se puede cambiar a cualquier otra especificando la dirección. Aquí aparecerá una lista de las funciones que hayamos guardado en esa carpeta.

## 1.3 Command History

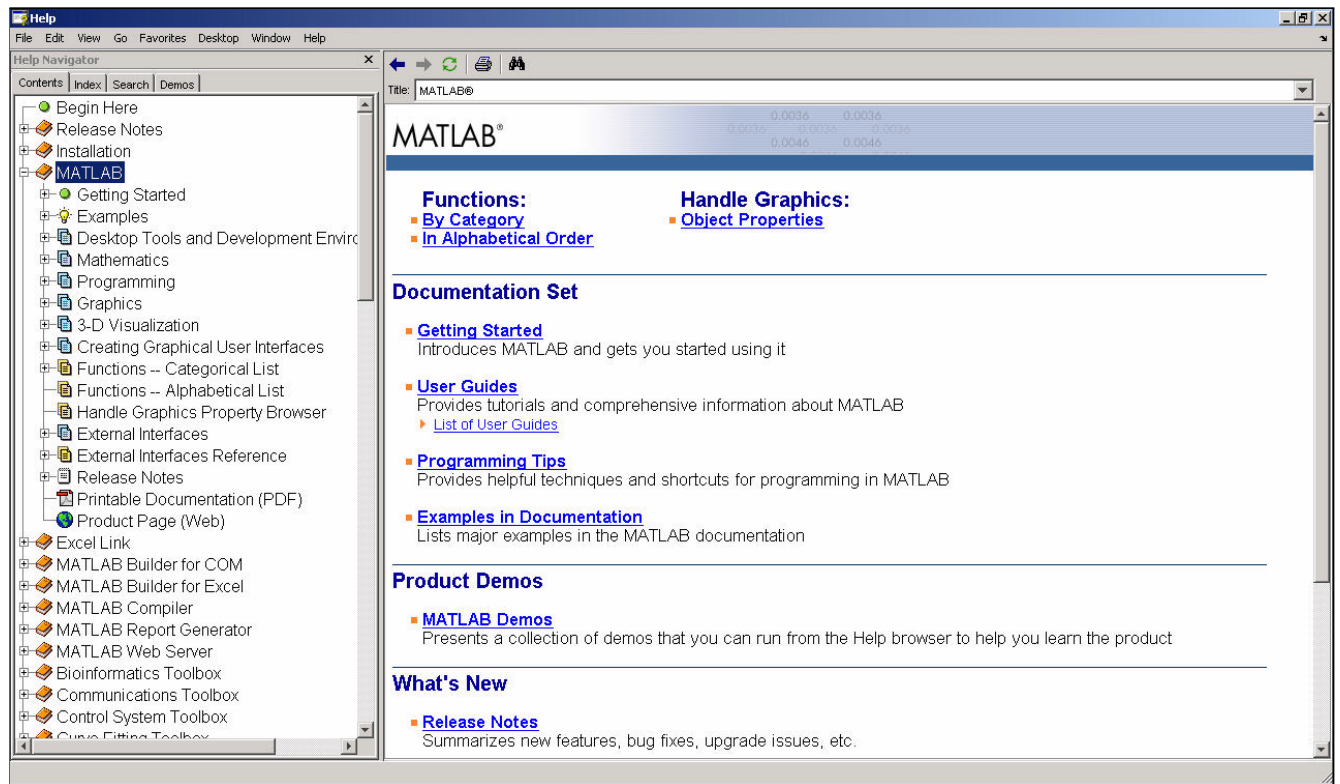
En esta ventana se registran todos los comandos introducidos en la ventana Command Windows en esta sesión y las sesiones anteriores. Para limpiar el historial, basta hacer click con el boton derecho del mouse y elegir "Clear Entire History".

## 1.4 Workspace

Aparece una lista con el valor y la información de todas las variables definidas. El comando **clear** borra todas las variables que se han creado.

## 1.5 Help

La ventana Help contiene información de ayuda sobre Matlab y sus funciones. Cualquier duda que se tenga, basta apretar la tecla **F1** y aparecerá la ventana Help.



## 2 Trabajando en Command

- 1) Luego de escribir un comando, al apretar la tecla **Enter** ese se ejecuta y muestra en pantalla todas las salidas que el comando generó.
- 2) Si al final de escribir un comando se termina con el **punto y coma (;)**, el comando se ejecuta pero no se muestra en pantalla ninguna salida.
- 3) Si al comienzo de una línea se escribe el símbolo **porcentaje (%)**, la línea es tomada como un comentario y no se ejecuta.
- 4) El comando **clc** limpia todos los comandos en la ventana Command, pero las variables predefinidas no son modificadas.

## 3 Operaciones Aritméticas

Operación	Simbolo	Ejemplo
Suma	+	$5+3=8$
Resta	-	$5-3=2$
Multiplicación	*	$5*3=15$
División por la derecha	/	$5/3$
División por la izquierda	\	$5\backslash 3=3/5$
Exponenciación	^	$5^3=5^3=125$

### 3.1 Orden de precedencia

Al evaluar una expresión con varios tipos de operaciones aritméticas, unas tiene mayor prioridad que otras:

Precedencia	Operación
Primera	Paréntesis ( )
Segunda	Exponenciación ^
Tercera	Multiplicación * y división /
Cuarta	Suma + y Resta -

## 4 Formatos Numéricos

En Matlab se puede cambiar el formato numérico de las expresiones. Por defecto, Matlab tiene *format short*, que se puede cambiar por otro, como por ejemplo:

Comando	Descripción	Ejemplo: $>> 290/7$
format short	Reales de 4 decimales con redondeo	41.4286
format long	Reales de 14 decimales con redondeo	41.42857142857143
format short e	Notación científica de 4 decimales	4.1429e+001
format long e	Notación científica de 15 decimales	4.142857142857143e+001
format short g	Punto flotante de 5 dígitos con redondeo	41.429
format long g	Punto flotante de 15 dígitos con redondeo	41.4285714285714
format bank	Reales de 2 decimales con redondeo.	41.43
format rat	Cuociente de enteros pequeños	290/7

## 5 Variables

Matlab trabaja principalmente con variables para poder hacer cálculos y guardar resultados. Veamos como trabajar:

- 1) Las variables, a diferencia de otros lenguajes como Java, no es necesario declararlas, solo es necesario asignar un valor, llamado expresión.
- 2) Una expresión puede ser simplemente un número o una expresión aritmética usando números y otras variables previamente asignadas, por ejemplo:

$$2^3 + 46 + (5 * 10) + x$$

- 3) Para asignar una variables se usa el signo igual (=) de la siguiente forma:

$$\text{Nombre\_variable} = \text{Expresion}$$

### 5.1 Reglas sobre los nombres de las variables

- 1) Los nombres de las variables pueden tener hasta 63 caracteres alfanuméricos o la barra abajo (\_), pero siempre deben empezar con una letra. Matlab es sensible, es decir, las minúsculas son diferentes a las mayúsculas.
- 2) Existen nombres prohibidos para las variables, como son el caso de for, if, while, entre otros.
- 3) Si se usa como nombre de variable el nombre de una función predefinida, como cos,sin,exp, entre otros, las funciones predefinidas no funcionarán, pero si las variables. Evitar usar estos nombres para evitar confusión al programar.
- 4) Existen variables predefinidas, que se pueden redefinir en algunos casos especiales. Algunas de estas variables son:
  - (a) **pi** : El número  $\pi$
  - (b) **eps** : La diferencia más pequeña entre dos números:  $2^{-52}$
  - (c) **inf** : Usado para representar el infinito
  - (d) **i** : La unidad imaginaria:  $\sqrt{-1}$
  - (e) **j** : La unidad imaginaria:  $\sqrt{-1}$
  - (f) **NaN** : Representa el resultado de una operación no válida, como por ejemplo  $0/0$ ,  $\text{inf}*0$

### 5.2 Manejo de Variables

- 1) Como se mencionó previamente, en la pantalla Workspace aparece una lista de todas las variables definidas y su información.
- 2) Para liberar memoria usada por Matlab, se puede eliminar variables definidas con algunos comandos básicos. Algunos de éstos son:
  - (a) **clear** : Remueve todas las variables de la memoria
  - (b) **clear x y z** : Remueve solo las variables x, y y z de la memoria
- 3) Para obtener información sobre las variables, pueden ser de gran utilidad los comandos:
  - (a) **who** : Muestra una lista con todas las variables en la memoria
  - (b) **whos** : Muestra una lista de todas las variables en la memoria con la información sobre sus bytes y clases.

## 6 Lógica

A veces es necesario usar lógica para algunos programas, como veremos más adelante, por lo cual necesitamos saber si una sentencia es verdadera o falsa.

### 6.1 Representación del True y False

En Matlab, a diferencia de Java, no tiene las variables True y False. En vez de eso, los representa con números:

True	1 ó distinto de 0
False	0

Obs: Cuando Matlab entrega el valor True, siempre entrega el valor 1, pero reconoce cualquier número distinto de 0 como True.

### 6.2 Operadores Relacionales

Los operadores relacionales en Matlab son:

Operador	Descripción
<	Menor
>	Mayor
<=	Menor o Igual
>=	Mayor o Igual
==	Igual
~=	Distinto

Estos operadores se pueden aplicar a dos números o a dos vectores de igual dimensión, que compara elemento por elemento. Ejemplo:

```
>> x=[0 1 2 3 4]
```

```
x =
```

```
0     1     2     3     4
```

```
>> y=[4 3 2 1 0]
```

```
y =
```

```
4     3     2     1     0
```

```
>> x<=y
```

```
ans =
```

```
1     1     1     0     0
```

### 6.2.1 Operadores Lógicos

Los operadores lógicos en Matlab son:

Operador	Nombre	Ejemplo	Descripción
&	y	x&y	Devuelve True si x e y son True. En otro caso, devuelve False.
	ó	x y	Devuelve True si x o y son True. En otro caso, devuelve False.
~	No	~x	Devuelve True si x es False. Devuelve False si x es True.

### 6.2.2 Orden de precedencia

La precedencia en Matlab de los operadores aritméticos, relacionales y lógicos, de mayor a menor, es la siguiente:

Nº	Operación
1	Parentesis
2	Exponenciación
3	No lógico
4	Multiplicación, División
5	Suma, Resta
6	Operadores relacionales
7	Y lógico
8	O lógico

## 7 Bifurcaciones

En Matlab, al igual que otros lenguajes, se puede programar bifurcaciones, es decir, se ejecuta un conjunto de comandos si solo si una cierta expresión condicional es verdadera.

### 7.1 Estructura de un if-end

```
if (expresión condicional)
    ..... //Comandos
    ..... //de
    ..... //Matlab
end
```

Ejemplo:

```
>> x=12;
>> y=-3;
>> if (x~=13)|(y<0)
    z=x*y
end

z =

    -36

>> x=13;
>> y=2;
>> if (x~=13)|(y<0)
    z=x*y
end
>>
```

## 7.2 Estructura de un if-else-end

También se puede hacer una bifurcación. En este caso, se ejecuta un conjunto de comandos si la expresión condicional es verdadera, o ejecuta otro conjunto de comandos si la expresión condicional es falsa.

```
if (expresión condicional)
    ..... //Comandos
    ..... //de
    ..... //Matlab
else
    ..... //Comandos
    ..... //de
    ..... //Matlab
end
```

Ejemplo:

```
>> x=10;
>> y=5;
>> if (x+y)/3 < 5
z=x*y
else
z=x-y
end
```

z =

5

```
>> x=4;
>> y=5;
>> if (x+y)/3 < 5
z=x*y
else
z=x-y
end
```

z =

20

## 8 Ciclos

Como en muchos otros lenguajes de programación, entre ellos incluido Java, en Matlab existen los ciclos o loop, claro que con una sintaxis diferente de Java.

### 8.1 Estructura de un while-end

```
while (expresión condicional)
    ..... //Comandos
    ..... //de
    ..... //Matlab
end
```

#### 8.1.1 Observaciones

- 1) El valor de verdad de la expresión del while se revisa solo al comienzo de cada ciclo.
- 2) En el caso que el valor de verdad de la expresión es falsa ejecutando un ciclo, se terminará de ejecutar todas las instrucciones.

Ejemplo:

```
>> x=0;
n=0;
while x<4
x=x*x+1
n=n+1
end
```

x =

1

n =

1

x =

2

n =

2

x =

5



n =

3

## 8.2 Estructura de un for-end

```
for k = r : s : t
    ..... //Comandos
    ..... //de
    ..... //Matlab
end
```

**k** : nombre de la variable indicadora del ciclo.

**r** : valor de la variable **k** en la primera iteración.

**s** : Valor que se le incrementa a la variable **k** al final de cada iteración

**t** : Valor de la variable **k** en la última iteración

### 8.2.1 Observaciones

- 1) El incremento **s** puede ser negativo, por ejemplo **k=25:-5:10** produce cuatro iteraciones con **k=25,20,15,10**.
- 2) Si el incremento **s** es omitido, por defecto vale 1, es decir **k = r : t**  $\iff$  **k = r : 1 : t**
- 3) Si **r = t**, el ciclo es ejecutado una vez.
- 4) Si los valores de **r**, **s** y **t** no hacen posible que **k** sea igual a **t**, entonces:
  - (a) Si **s** es positivo, la última iteración es cuando **k** vale el mayor número menor que **t**, por ejemplo **k=8:10:50** produce cinco iteraciones con **k=10,18,28,38,48**.
  - (b) Si **s** es negativo, la última iteración es cuando **k** vale el menor número mayor que **t**, por ejemplo **k=10:-3:2** produce tres iteraciones con **k=10,7,4**.

Ejemplo

```
>> for k=1:3:9
    x=k^2
end
```

x =

1

x =

16

x =

49