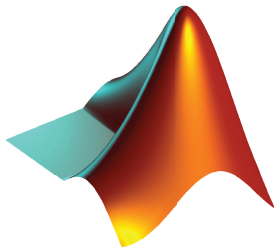


Bienvenidos a MATLAB



EMat

Escuela de
Matemática

Filánder Sequeira Chavarría

Última actualización: 8 de agosto de 2020

Organización de la presentación

- 1 Interfaz gráfica
- 2 Tipos de variables
- 3 Operadores y expresiones
- 4 Matrices

MATLAB (MATrix LABoratory)

- Es un programa para realizar cálculos numéricos con vectores y matrices, reales o complejos.
- Consiste además, de un conjunto de paquetes para diferentes ramas de la matemática e ingeniería. Por ejemplo, para Ecuaciones Diferenciales, Optimización, Estadística, etc.
- Es una de las aplicaciones computacionales más útiles que existen para la implementación de métodos numéricos.
- Dispone de un código básico y de varias librerías especializadas (toolboxes).
- Esta disponible tanto en Windows como en Unix.
- No es gratuito, pero la Universidad de Costa Rica les brinda acceso en sus computadoras personales mediante una licencia de estudiantes.

MATLAB (MATrix LABoratory)

- Es un programa para realizar cálculos numéricos con vectores y matrices, reales o complejos.
- Consiste además, de un conjunto de paquetes para diferentes ramas de la matemática e ingeniería. Por ejemplo, para Ecuaciones Diferenciales, Optimización, Estadística, etc.
- Es una de las aplicaciones computacionales más útiles que existen para la implementación de métodos numéricos.
- Dispone de un código básico y de varias librerías especializadas (toolboxes).
- Esta disponible tanto en Windows como en Unix.
- No es gratuito, pero la Universidad de Costa Rica les brinda acceso en sus computadoras personales mediante una licencia de estudiantes.

MATLAB (MATrix LABoratory)

- Es un programa para realizar cálculos numéricos con vectores y matrices, reales o complejos.
- Consiste además, de un conjunto de paquetes para diferentes ramas de la matemática e ingeniería. Por ejemplo, para Ecuaciones Diferenciales, Optimización, Estadística, etc.
- Es una de las aplicaciones computacionales más útiles que existen para la implementación de métodos numéricos.
- Dispone de un código básico y de varias librerías especializadas (toolboxes).
- Esta disponible tanto en Windows como en Unix.
- No es gratuito, pero la Universidad de Costa Rica les brinda acceso en sus computadoras personales mediante una licencia de estudiantes.

MATLAB (MATrix LABoratory)

- Es un programa para realizar cálculos numéricos con vectores y matrices, reales o complejos.
- Consiste además, de un conjunto de paquetes para diferentes ramas de la matemática e ingeniería. Por ejemplo, para Ecuaciones Diferenciales, Optimización, Estadística, etc.
- Es una de las aplicaciones computacionales más útiles que existen para la implementación de métodos numéricos.
- Dispone de un código básico y de varias librerías especializadas (toolboxes).
- Esta disponible tanto en Windows como en Unix.
- No es gratuito, pero la Universidad de Costa Rica les brinda acceso en sus computadoras personales mediante una licencia de estudiantes.

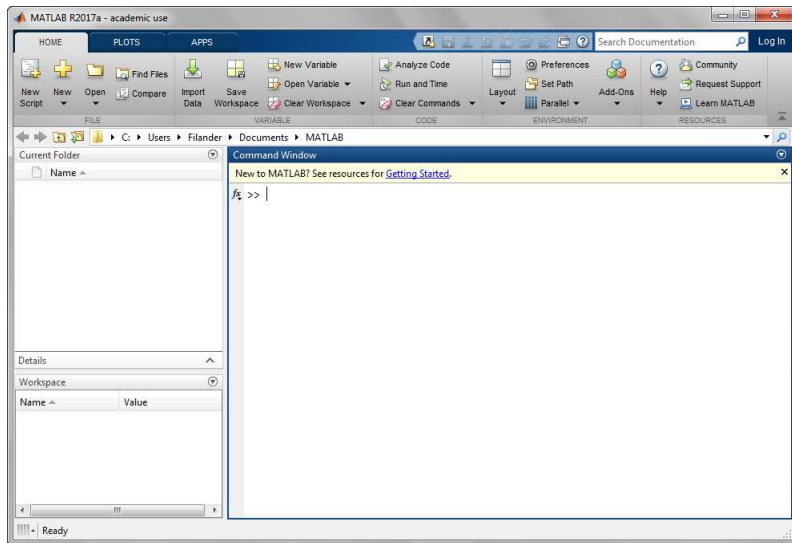
MATLAB (MATrix LABoratory)

- Es un programa para realizar cálculos numéricos con vectores y matrices, reales o complejos.
- Consiste además, de un conjunto de paquetes para diferentes ramas de la matemática e ingeniería. Por ejemplo, para Ecuaciones Diferenciales, Optimización, Estadística, etc.
- Es una de las aplicaciones computacionales más útiles que existen para la implementación de métodos numéricos.
- Dispone de un código básico y de varias librerías especializadas (toolboxes).
- Esta disponible tanto en Windows como en Unix.
- No es gratuito, pero la Universidad de Costa Rica les brinda acceso en sus computadoras personales mediante una licencia de estudiantes.

MATLAB (MATrix LABoratory)

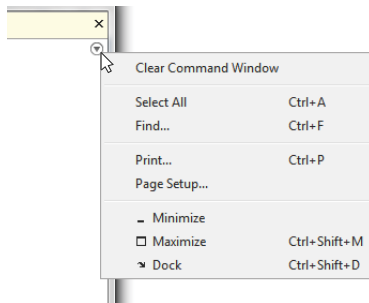
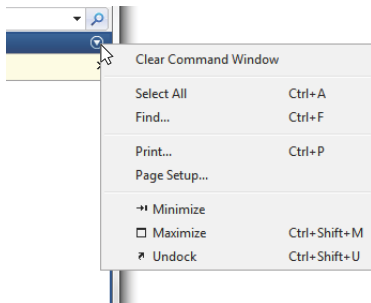
- Es un programa para realizar cálculos numéricos con vectores y matrices, reales o complejos.
- Consiste además, de un conjunto de paquetes para diferentes ramas de la matemática e ingeniería. Por ejemplo, para Ecuaciones Diferenciales, Optimización, Estadística, etc.
- Es una de las aplicaciones computacionales más útiles que existen para la implementación de métodos numéricos.
- Dispone de un código básico y de varias librerías especializadas (toolboxes).
- Esta disponible tanto en Windows como en Unix.
- No es gratuito, pero la Universidad de Costa Rica les brinda acceso en sus computadoras personales mediante una licencia de estudiantes.

Interfaz gráfica de MATLAB

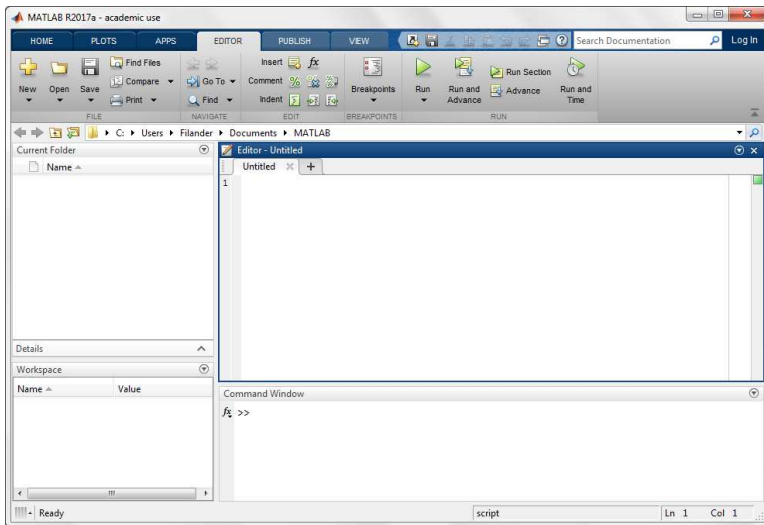


Interfaz gráfica de MATLAB

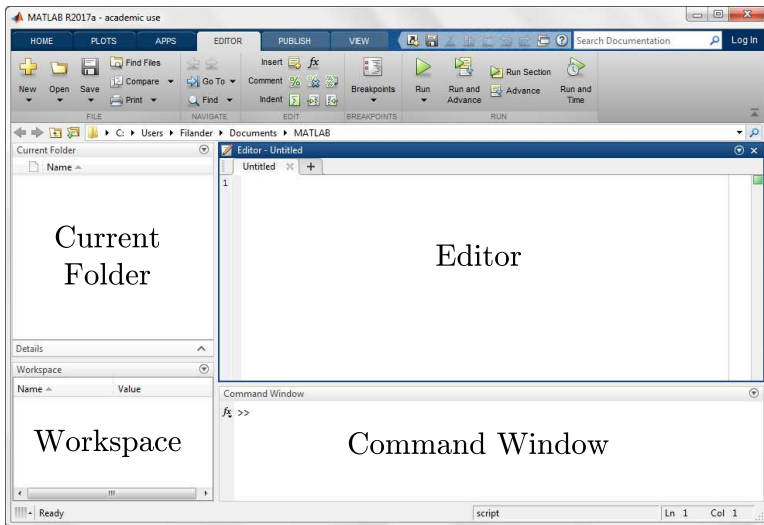
Esta interfaz puede modificarse al gusto del usuario. En particular, se puede tener cada una de las interfaces en ventanas separadas.



Interfaz gráfica de MATLAB

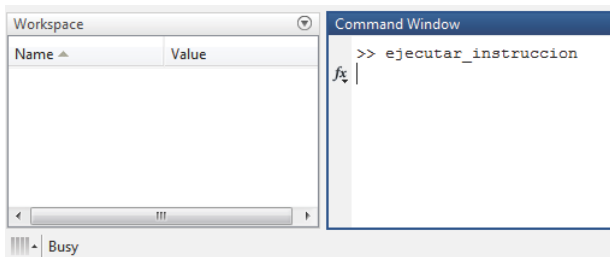


Interfaz gráfica de MATLAB



Command Window

Es donde se ejecutan los comandos de MATLAB. En esta interfaz aparece el *prompt* (`>>`), el cual indica que el programa está preparado para recibir instrucciones. Si no lo está, puede que este “ocupado” realizando alguna instrucción.



Command Window

Por ejemplo, ejecute la instrucción: `I = eye(6)` produce el resultado:

`I =`

1	0	0	0	0	0
0	1	0	0	0	0
0	0	1	0	0	0
0	0	0	1	0	0
0	0	0	0	1	0
0	0	0	0	0	1

- MATLAB es sensible a mayúsculas, es decir, no es lo mismo escribir la instrucción: `eye` que `EYE`, `Eye`, `eYe`, etc.
- Lo mismo para las variables, si nombran variables como `I` o `i`, MATLAB entenderá que son variables distintas.

Command Window

- Siempre que se oprima ENTER, lo que se ha escrito en la línea se interpretará por el compilador de MATLAB y tratará de ejecutarse.

Así, si se escribe: `hola` y luego se oprime ENTER, aparecerá un mensaje de error, dado que MATLAB no sabe como interpretar esa instrucción.

- En cambio la instrucción `eye(6)` si está definida, y se encarga de crear una matriz identidad de tamaño 6×6 .
- Más aún, la instrucción `I = eye(6)`, despliega la matriz identidad en la ventana de comandos. Para que esta no aparezca y solo la cree, basta con agregar un punto y coma (;) al final de la instrucción, es decir, `I = eye(6);`

Command Window

- Siempre que se oprima ENTER, lo que se ha escrito en la línea se interpretará por el compilador de MATLAB y tratará de ejecutarse.

Así, si se escribe: `hola` y luego se oprime ENTER, aparecerá un mensaje de error, dado que MATLAB no sabe como interpretar esa instrucción.

- En cambio la instrucción `eye(6)` si está definida, y se encarga de crear una matriz identidad de tamaño 6×6 .
- Más aún, la instrucción `I = eye(6)`, despliega la matriz identidad en la ventana de comandos. Para que esta no aparezca y solo la cree, basta con agregar un punto y coma (;) al final de la instrucción, es decir, `I = eye(6);`

Command Window



- Siempre que se oprima ENTER, lo que se ha escrito en la línea se interpretará por el compilador de MATLAB y tratará de ejecutarse.

Así, si se escribe: `hola` y luego se oprime ENTER, aparecerá un mensaje de error, dado que MATLAB no sabe como interpretar esa instrucción.

- En cambio la instrucción `eye(6)` si está definida, y se encarga de crear una matriz identidad de tamaño 6×6 .
- Más aún, la instrucción `I = eye(6)`, despliega la matriz identidad en la ventana de comandos. Para que esta no aparezca y solo la cree, basta con agregar un punto y coma (;) al final de la instrucción, es decir, `I = eye(6);`



Command Window

Para trabajar con esta ventana, es importante saber que:

- Se puede recorrer el historial de comandos ejecutados, opri-
miendo las teclas:  y 
- Solo se pueden agregar instrucciones lineales, salvo que se usen comas simples (,) para separar varias instrucciones.
- El comando `clc` borra lo que se muestra en la ventana de comandos.
- Cuando se usa una instrucciones como `eye(6)` sin decir como se llama la variable que recibe el resultado, se crea una variable con el nombre de `ans`, que almacena el resultado.
- Cuando se cierra MATLAB se pierden las ejecuciones y variables realizadas.



Command Window

Para trabajar con esta ventana, es importante saber que:

- Se puede recorrer el historial de comandos ejecutados, oprimiendo las teclas:  y 
- Solo se pueden agregar instrucciones lineales, salvo que se usen comas simples (,) para separar varias instrucciones.
- El comando `clc` borra lo que se muestra en la ventana de comandos.
- Cuando se usa una instrucciones como `eye(6)` sin decir como se llama la variable que recibe el resultado, se crea una variable con el nombre de `ans`, que almacena el resultado.
- Cuando se cierra MATLAB se pierden las ejecuciones y variables realizadas.



Command Window

Para trabajar con esta ventana, es importante saber que:

- Se puede recorrer el historial de comandos ejecutados, oprimiendo las teclas:  y .
- Solo se pueden agregar instrucciones lineales, salvo que se usen comas simples (,) para separar varias instrucciones.
- El comando `clc` borra lo que se muestra en la ventana de comandos.
- Cuando se usa una instrucciones como `eye(6)` sin decir como se llama la variable que recibe el resultado, se crea una variable con el nombre de `ans`, que almacena el resultado.
- Cuando se cierra MATLAB se pierden las ejecuciones y variables realizadas.



Command Window

Para trabajar con esta ventana, es importante saber que:

- Se puede recorrer el historial de comandos ejecutados, oprimiendo las teclas:  y 
- Solo se pueden agregar instrucciones lineales, salvo que se usen comas simples (,) para separar varias instrucciones.
- El comando `clc` borra lo que se muestra en la ventana de comandos.
- Cuando se usa una instrucciones como `eye(6)` sin decir como se llama la variable que recibe el resultado, se crea una variable con el nombre de `ans`, que almacena el resultado.
- Cuando se cierra MATLAB se pierden las ejecuciones y variables realizadas.

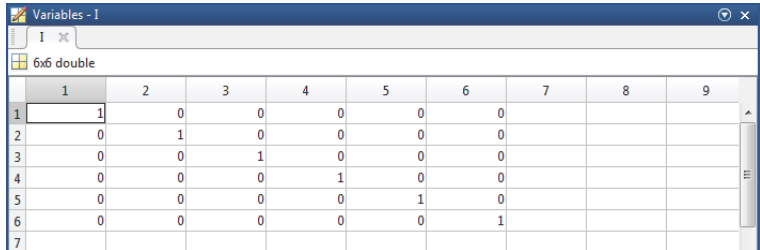
Command Window

Para trabajar con esta ventana, es importante saber que:

- Se puede recorrer el historial de comandos ejecutados, oprimiendo las teclas:  y 
- Solo se pueden agregar instrucciones lineales, salvo que se usen comas simples (,) para separar varias instrucciones.
- El comando `clc` borra lo que se muestra en la ventana de comandos.
- Cuando se usa una instrucciones como `eye(6)` sin decir como se llama la variable que recibe el resultado, se crea una variable con el nombre de `ans`, que almacena el resultado.
- Cuando se cierra MATLAB se pierden las ejecuciones y variables realizadas.

Workspace

Contiene la información sobre todas las variables que se hayan definido en esta sesión y permite ver y modificar las matrices con las que se esté trabajando.

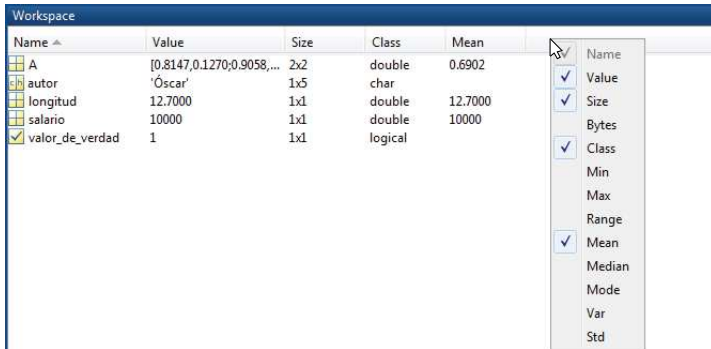


The screenshot shows a MATLAB window titled 'Variables - I'. Inside, there is a tab labeled 'I' and a label '6x6 double'. Below this is a table representing a 6x6 matrix. The columns are numbered 1 through 9 (with the 9th column being empty), and the rows are numbered 1 through 7 (with the 7th row being empty). The matrix contains 1s on the main diagonal (from row 1, column 1 to row 6, column 6) and 0s in all other positions.

	1	2	3	4	5	6	7	8	9
1	1	0	0	0	0	0			
2	0	1	0	0	0	0			
3	0	0	1	0	0	0			
4	0	0	0	1	0	0			
5	0	0	0	0	1	0			
6	0	0	0	0	0	1			
7									

Workspace

Esta ventana también brinda información sobre las variables que existen en la sesión actual de trabajo.



The screenshot shows the 'Workspace' window in R Studio. It contains a table with the following data:

Name	Value	Size	Class	Mean
A	[0.8147, 0.1270; 0.9058, ...]	2x2	double	0.6902
autor	'Óscar'	1x5	char	
longitud	12.7000	1x1	double	12.7000
salario	10000	1x1	double	10000
valor_de_verdad	1	1x1	logical	

A context menu is open over the table, listing various statistics and properties that can be calculated for the selected variable. The items in the menu are:

- Name
- Value
- Size
- Bytes
- Class
- Min
- Max
- Range
- Mean
- Median
- Mode
- Var
- Std

Ejemplo

Digitar las instrucciones:

```
A = rand(6), B = rand(6), B*A
```

La instrucción `rand(n)` crea una matriz de $n \times n$, con entradas generadas al azar entre 0 y 1.

Precisión finita

- Aunque aparezcan sólo 4 cifras, han sido calculados con 16 cifras de precisión (doble precisión).
- Para que aparezcan todos las 16 cifras, se escribe una vez la instrucción:

```
format long
```

o bien,

```
format long e
```

para la notación científica.

- El formato

```
format rat
```

reescribe el valor numérico en forma de una fracción.

Precisión finita

- Aunque aparezcan sólo 4 cifras, han sido calculados con 16 cifras de precisión (doble precisión).
- Para que aparezcan todos las 16 cifras, se escribe una vez la instrucción:

```
format long
```

o bien,

```
format long e
```

para la notación científica.

- El formato

```
format rat
```

reescribe el valor numérico en forma de una fracción.

Precisión finita

- Aunque aparezcan sólo 4 cifras, han sido calculados con 16 cifras de precisión (doble precisión).
- Para que aparezcan todos las 16 cifras, se escribe una vez la instrucción:

```
format long
```

o bien,

```
format long e
```

para la notación científica.

- El formato

```
format rat
```

reescribe el valor numérico en forma de una fracción.

Precisión finita

- Nótese que al trabajar con al menos 16 dígitos, la computadora no trabaja con todos los números reales (en particular no existen irracionales).

Por ejemplo, ¿qué se obtiene al hacer la operación

$$1 + 10^{-16} ?$$

- **Importante:** La instrucción `clc` borra lo que se muestra en la ventana de comandos, pero no borra las variables que se han creado. Para hacerlo, se el comando

```
clear all
```

Precisión finita


- Nótese que al trabajar con al menos 16 dígitos, la computadora no trabaja con todos los números reales (en particular no existen irracionales).

Por ejemplo, ¿qué se obtiene al hacer la operación

$$1 + 10^{-16} ?$$

- **Importante:** La instrucción `clc` borra lo que se muestra en la ventana de comandos, pero no borra las variables que se han creado. Para hacerlo, se el comando

```
clear all
```


- Para preservar fragmentos de código en MATLAB, se usa el Editor.
- Aquí se pueden escribir códigos de MATLAB, pero se puede dar ENTER sin que se ejecuten las cosas.
- Para que se ejecuten, basta con oprimir la tecla: 

- Para preservar fragmentos de código en MATLAB, se usa el Editor.
- Aquí se pueden escribir códigos de MATLAB, pero se puede dar ENTER sin que se ejecuten las cosas.
- Para que se ejecuten, basta con oprimir la tecla:



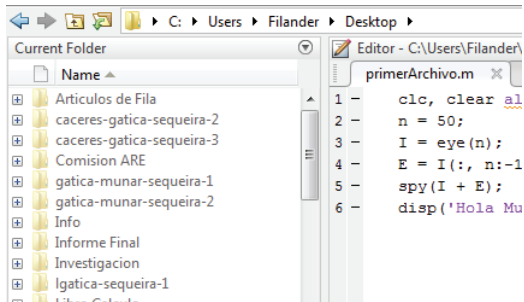
- Para preservar fragmentos de código en MATLAB, se usa el Editor.
- Aquí se pueden escribir códigos de MATLAB, pero se puede dar ENTER sin que se ejecuten las cosas.
- Para que se ejecuten, basta con oprimir la tecla:



- Para preservar fragmentos de código en MATLAB, se usa el Editor.
- Aquí se pueden escribir códigos de MATLAB, pero se puede dar ENTER sin que se ejecuten las cosas.
- Para que se ejecuten, basta con oprimir la tecla: 

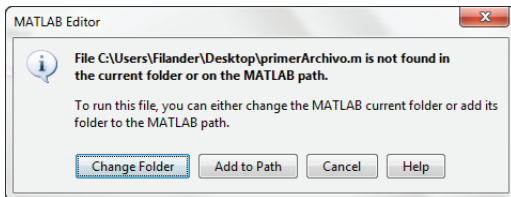
Directorio actual de trabajo

- Muestra el directorio (o carpeta), en el que MATLAB buscará los archivos que se soliciten.



Directorio actual de trabajo

- Cuando se ejecute un programa del editor, cuyo archivo no se encuentra en el directorio actual de trabajo se despliega el siguiente mensaje.



La ayuda

- MATLAB dispone de una excelente ayuda (`help`) con la que se puede encontrar la información que se desee, en inglés.



- Desde la ventana de comandos también es posible obtener información de los comandos, por medio de la instrucción: `help`, por ejemplo:

```
help eye
```

La ayuda

- MATLAB dispone de una excelente ayuda (`help`) con la que se puede encontrar la información que se desee, en inglés.



- Desde la ventana de comandos también es posible obtener información de los comandos, por medio de la instrucción: `help`, por ejemplo:

```
help eye
```

La ayuda

```
>> help eye
eye Identity matrix.
  eye(N) is the N-by-N identity matrix.

  eye(M,N) or eye([M,N]) is an M-by-N matrix with 1's on
  the diagonal and zeros elsewhere.

  eye(SIZE(A)) is the same size as A.

  eye with no arguments is the scalar 1.

eye(..., CLASSNAME) is a matrix with ones of class specified by
CLASSNAME on the diagonal and zeros elsewhere.

eye(..., 'like', Y) is an identity matrix with the same data type, sparsity,
and complexity (real or complex) as the numeric variable Y.

Note: The size inputs M and N should be nonnegative integers.
Negative integers are treated as 0.

Example:
  x = eye(2,3,'int8');

See also speye, ones, zeros, rand, randn.
```

[Reference page for eye](#)
[Other functions named eye](#)

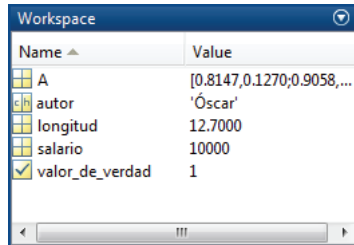
Organización de la presentación

- 1 Interfaz gráfica
- 2 Tipos de variables
- 3 Operadores y expresiones
- 4 Matrices

Variables

En MATLAB no es necesario definir el tipo de dato como ocurre en otros lenguajes de programación. Por ejemplo:

```
salario = 10000;  
longitud = 12.7;  
autor = 'Óscar';  
valor_de_verdad = true; % Revisar: help true  
A = rand(2);
```



Nombres de la variables

- Iniciar los nombres con letras y no con números. Por ejemplo, `x1` es un nombre correcto, pero `4ever` no lo es.
- Utilizar nombres significativos. Por ejemplo, si se va a almacenar una fecha, se pueden utilizar variables con nombres como: `dia`, `mes` y `anho`, o bien en inglés: `day`, `month` y `year`, ya que indican claramente que están guardando.
- No incluir: espacios, signos de admiración e interrogación, delimitadores y operaciones (`()`, `[]`, `{}`, `+`, `-`, `*`, `/`), ni caracteres especiales o acentos (`á`, `é`, `í`, `ó`, `ú`, `ñ`, `ç`, `&`, `$`, `#`, `@`), entre otros. El guión bajo (`_`) si es permitido.

Nombres de la variables

- Iniciar los nombres con letras y no con números. Por ejemplo, `x1` es un nombre correcto, pero `4ever` no lo es.
- Utilizar nombres significativos. Por ejemplo, si se va a almacenar una fecha, se pueden utilizar variables con nombres como: `dia`, `mes` y `anho`, o bien en inglés: `day`, `month` y `year`, ya que indican claramente que están guardando.
- No incluir: espacios, signos de admiración e interrogación, delimitadores y operaciones (`()`, `[]`, `{}`, `+`, `-`, `*`, `/`), ni caracteres especiales o acentos (`á`, `é`, `í`, `ó`, `ú`, `ñ`, `ç`, `&`, `$`, `#`, `@`), entre otros. El guión bajo (`_`) si es permitido.

Nombres de la variables

- Iniciar los nombres con letras y no con números. Por ejemplo, `x1` es un nombre correcto, pero `4ever` no lo es.
- Utilizar nombres significativos. Por ejemplo, si se va a almacenar una fecha, se pueden utilizar variables con nombres como: `dia`, `mes` y `anho`, o bien en inglés: `day`, `month` y `year`, ya que indican claramente que están guardando.
- No incluir: espacios, signos de admiración e interrogación, delimitadores y operaciones (`()`, `[]`, `{}`, `+`, `-`, `*`, `/`), ni caracteres especiales o acentos (`á`, `é`, `í`, `ó`, `ú`, `ñ`, `ç`, `&`, `$`, `#`, `@`), entre otros. El guión bajo (`_`) si es permitido.

Nombres de la variables

- Recordar que hay diferencia entre letras mayúsculas y minúsculas. Es decir, la variable x es distinta a la variable X .
- Por conveniencia, para los nombres de los vectores y escalares se utilizan letras minúsculas, mientras que los nombres de las matrices serán con letras mayúsculas.

Nombres de la variables

- Recordar que hay diferencia entre letras mayúsculas y minúsculas. Es decir, la variable x es distinta a la variable X .
- Por conveniencia, para los nombres de los vectores y escalares se utilizan letras minúsculas, mientras que los nombres de las matrices serán con letras mayúsculas.

Notación científica

Los valores muy grandes o muy pequeños pueden escribirse usando notación científica.

- $1.4\text{e}-4 \rightarrow 0.00014$

- $1.0\text{e}7 \rightarrow 10000000$

$$-3.5\text{e}-8 \equiv -3.5 \times 10^{-8}$$

Notación científica

Los valores muy grandes o muy pequeños pueden escribirse usando notación científica.

- $1.4e-4 \rightarrow 0.00014$

- $1.0e7 \rightarrow 10000000$

$$-3.5e-8 \equiv -3.5 \times 10^{-8}$$

Operaciones aritméticas

Símbolo	Nombre de la operación
*	Multiplicación
^	Potencia
/	División
+	Suma
-	Resta

Operaciones aritméticas

- Precedencia de los operadores $*$ y $/$ se realizan antes que $+$ y $-$.
- Los paréntesis se usan para agrupar cálculos y forzar a que las operaciones se realicen primero. Pero sólo se pueden emplear los redondos: $()$.
- Operadores de igual precedencia se ejecutan en el orden de aparición de izquierda a derecha.

Operaciones aritméticas

- Precedencia de los operadores $*$ y $/$ se realizan antes que $+$ y $-$.
- Los paréntesis se usan para agrupar cálculos y forzar a que las operaciones se realicen primero. Pero sólo se pueden emplear los redondos: $()$.
- Operadores de igual precedencia se ejecutan en el orden de aparición de izquierda a derecha.

Operaciones aritméticas

- Precedencia de los operadores $*$ y $/$ se realizan antes que $+$ y $-$.
- Los paréntesis se usan para agrupar cálculos y forzar a que las operaciones se realicen primero. Pero sólo se pueden emplear los redondos: $()$.
- Operadores de igual precedencia se ejecutan en el orden de aparición de izquierda a derecha.

Cadena de caracteres (String)

Se trabajan con palabras en lugar de números:

```
nombre = 'Esteban';  
apellido = 'Segura';
```

Más aún, para mostrar una cadena de caracteres en la ventana de comandos, se pueden utilizar las instrucciones:

```
disp('HOLA')
```

o bien

```
error('HOLA')
```

Solo que este último se utiliza para enviar mensajes de error.

Cadena de caracteres (String)

Se trabajan con palabras en lugar de números:

```
nombre = 'Esteban';  
apellido = 'Segura';
```

Más aún, para mostrar una cadena de caracteres en la ventana de comandos, se pueden utilizar las instrucciones:

```
disp('HOLA')
```

o bien

```
error('HOLA')
```

Solo que este último se utiliza para enviar mensajes de error.

- Los comentarios corresponden a fragmentos de textos, que se escriben junto al código (usualmente en el Editor), pero que no corresponden a instrucciones a ejecutar, sino a pequeños mensajes utilizados para describir que hacen instrucciones dentro del código.
- Para iniciar un comentario en una línea se usa el símbolo de por ciento (%). Por ejemplo:

```
% Autor: Filánder Sequeira Chavarría  
% Fecha: 10 de Marzo de 2018
```


Organización de la presentación

- 1 Interfaz gráfica
- 2 Tipos de variables
- 3 Operadores y expresiones
- 4 Matrices

Operador de asignación

El operador básico de asignación es el signo de igualdad (=). Por ejemplo:

$$a = c;$$

Indica que “se asigna a la variable *a*, el valor de la variable *c*”. Es decir:



Ejercicio

Ejercicio

Escriba un programa que intercambie el contenido de dos variables dadas. Más precisamente, si se ingresan las variables:

$$a = 10;$$

$$b = -4;$$

y se realiza el programa de intercambio, se debe obtener como resultado: $a = -4$ y $b = 10$.

¿Cuántas variables auxiliares se necesitarían para intercambiar los valores en tres o más variables?

Ejercicio

Ejercicio

Escriba un programa que intercambie el contenido de dos variables dadas. Más precisamente, si se ingresan las variables:

$$a = 10;$$

$$b = -4;$$

y se realiza el programa de intercambio, se debe obtener como resultado: $a = -4$ y $b = 10$.

¿Cuántas variables auxiliares se necesitarían para intercambiar los valores en tres o más variables?

Ejercicio

Ejercicio (para la casa)

Escriba un programa que intercambie el contenido de cuatro variables: a , b , c y d en orden inverso. Por ejemplo, si

$$a = 10, \quad b = -4, \quad c = 19 \quad \text{y} \quad d = 86,$$

entonces al intercambiar los valores se obtiene el resultado:

$$a = 86, \quad b = 19, \quad c = -4 \quad \text{y} \quad d = 10.$$

Uso de paréntesis

En MATLAB es posible introducir una fórmula como la siguiente:

$$3(5 - 10)^2 + 12(15 + 3)$$

de la siguiente manera:

$$3 * (5 - 10) ^ 2 + 12 * (15 + 3) ;$$

Uso de paréntesis

En MATLAB es posible introducir una fórmula como la siguiente:

$$3(5 - 10)^2 + 12(15 + 3)$$

de la siguiente manera:

$$3 * (5 - 10) ^2 + 12 * (15 + 3) ;$$

¡La multiplicación no es implícita!

Ejercicio

Ejercicio

Ingresa la siguiente fórmula en MATLAB:

$$\frac{x^3 + \frac{1}{3}x^2 - \frac{5}{4}x + \pi}{\left(\frac{5}{6}\right)^2 (x - 2) - \frac{1}{x}}$$

para ello, cree una variable llamada x y para π utilice la instrucción: `pi`.

¿Qué se obtiene cuando $x = 0$?

Ejercicio

Ejercicio

Ingresa la siguiente fórmula en MATLAB:

$$\frac{x^3 + \frac{1}{3}x^2 - \frac{5}{4}x + \pi}{\left(\frac{5}{6}\right)^2 (x - 2) - \frac{1}{x}}$$

para ello, cree una variable llamada x y para π utilice la instrucción: `pi`.

¿Qué se obtiene cuando $x = 0$?

El resultado no es un número real

Si el resultado de una operación aritmética es mayor que el número más grande, en valor absoluto, posible (`realmax`), MATLAB dará `Inf` o `-Inf` como resultados. Si el resultado no puede ser determinado, este será `NaN` (*not a number*).

```
1/0      % resultado es Inf
0/0      % resultado es NaN
```

Organización de la presentación

- 1 Interfaz gráfica
- 2 Tipos de variables
- 3 Operadores y expresiones
- 4 Matrices

Definición de matrices

- Los vectores son considerados matrices con una fila o una columna.
- Es posible asignar a una variable una matriz. En Matemática es usual usar letras mayúsculas para nombrar matrices y minúsculas para los vectores.
- Las matrices se introducen por filas, a pesar que estas se almacenan por columnas. Los elementos en la fila se separan por espacios o comas, mientras que las filas se separan por punto y coma o cambios de líneas.

Definición de matrices

- Los vectores son considerados matrices con una fila o una columna.
- Es posible asignar a una variable una matriz. En Matemática es usual usar letras mayúsculas para nombrar matrices y minúsculas para los vectores.
- Las matrices se introducen por filas, a pesar que estas se almacenan por columnas. Los elementos en la fila se separan por espacios o comas, mientras que las filas se separan por punto y coma o cambios de líneas.

Definición de matrices

- Los vectores son considerados matrices con una fila o una columna.
- Es posible asignar a una variable una matriz. En Matemática es usual usar letras mayúsculas para nombrar matrices y minúsculas para los vectores.
- Las matrices se introducen por filas, a pesar que estas se almacenan por columnas. Los elementos en la fila se separan por espacios o comas, mientras que las filas se separan por punto y coma o cambios de líneas.

Ejemplo

Escribir lo siguiente en un archivo del Editor:

```
A = [1 2 3; 4 5 6; 7 8 9]
```

```
B = [1 4 7  
     2 5 8  
     3 6 9]
```

```
x = [0 0 0 0 0]
```

```
y = [0; 0; 0; 0; 0]
```

Ejemplo

Los resultados obtenidos al ejecutar corresponden a:

A =

1	2	3
4	5	6
7	8	9

B =

1	4	7
2	5	8
3	6	9

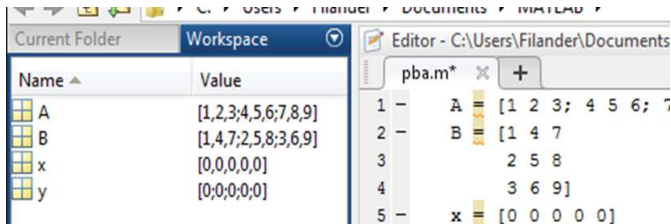
x =

0	0	0	0	0
---	---	---	---	---

y =

0
0
0
0
0

Ejemplo



The image shows a MATLAB interface. On the left, the 'Workspace' pane displays four variables: A, B, x, and y. Each variable is represented by a small grid icon. The 'Value' column shows the contents of each variable. On the right, the 'Editor' pane shows a file named 'pba.m*'. The code in the editor consists of five lines, each defining a variable and its value using square brackets.

Name	Value
A	[1,2,3;4,5,6;7,8,9]
B	[1,4,7;2,5,8;3,6,9]
x	[0,0,0,0,0]
y	[0;0;0;0;0]

```
1 - A = [1 2 3; 4 5 6; 7 8 9]
2 - B = [1 4 7
3       2 5 8
4       3 6 9]
5 - x = [0 0 0 0 0]
```

Importante

Para efectos de operaciones, siempre se debe tener en cuenta la diferencia entre vectores filas y vectores columna. Por ejemplo:

```
x = [10 20 30];      % vector fila
y = [11; 12; 13];    % vector columna
x * y                % Resultado: 740
y * x                % Resultado:  110    220    330
                     %              120    240    360
                     %              130    260    390
```

MATLAB tiene una operación poco natural que no diferencia entre vectores filas y vectores columna. Por ejemplo:

```
x = [10 20 30];      % vector fila  
y = [11; 12; 13];    % vector columna  
x + y
```

¿Cuál es el resultado obtenido?

Cuidado

MATLAB tiene una operación poco natural que no diferencia entre vectores filas y vectores columna. Por ejemplo:

```
x = [10 20 30];      % vector fila  
y = [11; 12; 13];    % vector columna  
x + y
```

¿Cuál es el resultado obtenido?

```
ans =  
  
    21    31    41  
    22    32    42  
    23    33    43
```

Acceder a una matriz o vector

Para obtener o modificar una entrada en una matriz o vector, se utilizan los paréntesis redondos, junto con los índices de la posición que se desea.

```
A(i, j) % para una matriz de nxn  
x(k)    % para un vector de nx1 o de 1xn
```

donde, $1 \leq i, j, k \leq n$.

Acceder a una matriz o vector

Para obtener o modificar una entrada en una matriz o vector, se utilizan los paréntesis redondos, junto con los índices de la posición que se desea.

```
A(i, j) % para una matriz de nxn  
x(k)    % para un vector de nx1 o de 1xn
```

donde, $1 \leq i, j, k \leq n$.

$$A = \begin{pmatrix} A(1,1) & A(1,2) & A(1,3) & \cdots & A(1,n) \\ A(2,1) & A(2,2) & A(2,3) & & \vdots \\ A(3,1) & A(3,2) & A(3,3) & & \vdots \\ \vdots & & & \ddots & \vdots \\ A(m,1) & \cdots & \cdots & \cdots & A(m,n) \end{pmatrix}$$

Ejemplo

Al ejecutar las instrucciones:

```
A = [ 1  4 -3  
      2  1  5  
     -2  5  3];  
A(1, 3) % Mostrar el valor en (1,3)
```

se obtiene el resultado:

```
ans =  
  
    -3
```

Importante

MATLAB permite crear vectores de manera dinámica, es decir, sin definir previamente su dimensión. A pesar de ello, se debe tener en cuenta que, por defecto, estos son vectores filas. Por ejemplo, la instrucción

```
x(1) = 1  
x(2) = 2
```

produce:

```
x =  
    1  
  
x =  
    1    2
```


Extraer bloques

Una matriz \mathbf{A} de tamaño $m \times n$, puede ser interpretada como un vector columna de m vectores fila de dimensión n , o bien, como un vector fila de n vectores columna de dimensión m .

$$\mathbf{A} = \begin{pmatrix} (A(1,1) & A(1,2) & A(1,3) & \cdots & A(1,n)) \\ (A(2,1) & A(2,2) & A(2,3) & \cdots & A(2,n)) \\ (A(3,1) & A(3,2) & A(3,3) & \cdots & A(3,n)) \\ \vdots \\ (A(m,1) & A(m,2) & A(m,3) & \cdots & A(m,n)) \end{pmatrix}$$

$$\mathbf{A} = \left(\begin{pmatrix} A(1,1) \\ A(2,1) \\ A(3,1) \\ \vdots \\ A(m,1) \end{pmatrix} \begin{pmatrix} A(1,2) \\ A(2,2) \\ A(3,2) \\ \vdots \\ A(m,2) \end{pmatrix} \begin{pmatrix} A(1,3) \\ A(2,3) \\ A(3,3) \\ \vdots \\ A(m,3) \end{pmatrix} \cdots \begin{pmatrix} A(1,n) \\ A(2,n) \\ A(3,n) \\ \vdots \\ A(m,n) \end{pmatrix} \right)$$

Extraer bloques

Para extraer el vector que representa la fila i , así como el vector que representa la columna j , se pueden efectuar, respectivamente, las siguientes instrucciones:

```
A(i, :) % Extrae la fila i de A  
A(:, j) % Extrae la columna j de A
```

En particular, en el comando $A(i, :)$, los dos puntos en la entrada correspondiente a la columna, le indica a MATLAB que debe recorrer todas las columnas. Análogamente, $A(:, j)$, establece que el recorrido es en todas las filas.

Extraer bloques

Por otro lado, es posible extraer bloques o submatrices de una matriz A , al considerar los índices de las filas y columnas que delimitan al bloque deseado. Esto, se realiza a través de la instrucción:

```
A(i1:i2, j1:j2) % Extrae un bloque de A
```

donde, se considera como primera fila del bloque $i1$ y como última fila $i2$, mientras que la primera columna corresponde a $j1$ y la última $j2$.

Extraer bloques

$$\begin{pmatrix} A(1,1) & A(1,2) & \dots & & \dots & \dots & \dots & \dots & A(1,n) \\ A(2,1) & A(2,2) & & & & & & & \vdots \\ \vdots & \vdots & \ddots & & & & & & \vdots \\ i1 \leftarrow \vdots & \vdots & & & A(i1,j1) & \dots & A(i1,j2) & & \vdots \\ \vdots & \vdots & & & \vdots & & \vdots & & \vdots \\ i2 \leftarrow \vdots & \vdots & & & A(i2,j1) & \dots & A(i2,j2) & & \vdots \\ \vdots & \vdots & & & & & & \ddots & \vdots \\ A(m,1) & \dots & \dots & & \dots & \dots & \dots & \dots & A(m,n) \end{pmatrix}$$

$$A(i1:i2, j1:j2)$$

Operaciones con matrices

Operador	Descripción
$+$	adición o suma
$-$	sustracción o resta
$*$	producto o multiplicación
$'$	conjugada transpuesta
$^$	potenciación
\backslash	resolución de un sistema lineal
$. *$	producto elemento a elemento
$. /$	división elemento a elemento
$. ^$	eleva a una potencia elemento a elemento

Ejemplo

Las instrucciones:

```
A = [1 2 3; 4 5 6; 7 8 9];  
A^2  
A.^2
```

producen el resultado:

ans =

30	36	42
66	81	96
102	126	150

ans =

1	4	9
16	25	36
49	64	81

Ejemplo

Las instrucciones:

```
A = [ 2 -1 0 0  
      -1 2 -1 0  
        0 -1 2 -1  
        0 0 -1 2];  
b = [1; 0; 0; 1];  
x = A \ b      % Resolver el sistema lineal
```

producen el resultado:

```
x =  
  
1.0000  
1.0000  
1.0000  
1.0000
```

Operaciones elemento a elemento

La operación:

```
[1 2 3 4]^2
```

no está definida y por ende produce el error:

Error using ^

One argument must be a square matrix and the other must be a scalar. Use POWER (.^) for elementwise power.

Operaciones elemento a elemento

La operación:

```
[1 2 3 4]^2
```

no está definida y por ende produce el error:

```
Error using ^  
One argument must be a square matrix and the other must  
be a scalar. Use POWER (.^) for elementwise power.
```

En cambio:

```
[1 2 3 4].^2
```

produce:

```
ans =  
  
     1     4     9    16
```

Concatenar matrices

Los paréntesis cuadrados `[]` permiten incrementar el orden de las matrices y de los vectores. Siempre que los tamaños coincidan.

```
A = [1 2  
      3 4];
```

```
B = [5 6  
      7 8];
```

```
C = [A B]
```

```
D = [A; B]
```

Ejercicio

Ejercicio

Considere las matrices

$$\mathbf{A} := \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ -1 & -9 \end{pmatrix} \quad \text{y} \quad \mathbf{B} := \begin{pmatrix} 1 & 7 & 3 & 62 \\ 2 & 5 & 3 & -1 \\ 0 & 2 & 4 & 2 \end{pmatrix}.$$

Luego, con ayuda de ellas, construya las matrices:

$$\mathbf{C} := \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{pmatrix} \quad \text{y} \quad \mathbf{D} := \begin{pmatrix} b_{12} & b_{13} \\ b_{22} & b_{23} \end{pmatrix}.$$

Matrices especiales

Instrucción	Descripción
<code>eye (n)</code>	matriz identidad de $n \times n$.
<code>zeros (m, n)</code>	matriz con entradas ceros de $m \times n$.
<code>ones (m, n)</code>	matriz con entradas unos de $m \times n$.
<code>linspace (a, b, n)</code>	vector con n valores igualmente espaciados entre a y b .
<code>rand (m, n)</code>	matriz de $m \times n$ de números aleatorios entre 0 y 1 con distribución uniforme.
<code>hilb (n)</code>	matriz de Hilbert de $n \times n$.
<code>invhilb (n)</code>	inversa de la matriz de Hilbert.

Extracción de matrices

Instrucción	Descripción
<code>[m,n] = size(A)</code>	número de filas (m) y de columnas (n) de A.
<code>n = length(A)</code>	máximo entre el número de filas y columnas de A.
<code>A = diag(x)</code>	matriz diagonal con los elementos del vector x.
<code>x = diag(A)</code>	vector con la diagonal de A.
<code>L = tril(A)</code>	matriz triangular inferior a partir de la matriz A, incluyendo la diagonal.
<code>U = triu(A)</code>	matriz triangular superior a partir de la matriz A, incluyendo la diagonal.

Extracción de matrices

El comando `diag(A)` puede incluir además un parámetro adicional $k \in \mathbb{Z}$, que especifique la diagonal de A a extraer.

$k = 0 \swarrow$	$k = 1 \swarrow$	$k = 2 \swarrow$	$k = 3 \swarrow$	$k = 4 \swarrow$	
$k = -1 \swarrow$	$A(1,1)$	$A(1,2)$	$A(1,3)$	$A(1,4)$	$A(1,5)$
$k = -2 \swarrow$	$A(2,1)$	$A(2,2)$	$A(2,3)$	$A(2,4)$	$A(2,5)$
$k = -3 \swarrow$	$A(3,1)$	$A(3,2)$	$A(3,3)$	$A(3,4)$	$A(3,5)$
$k = -4 \swarrow$	$A(4,1)$	$A(4,2)$	$A(4,3)$	$A(4,4)$	$A(4,5)$
	$A(5,1)$	$A(5,2)$	$A(5,3)$	$A(5,4)$	$A(5,5)$

`diag(A, k)`

Ejercicio

Ejercicio

Escriba un fragmento de código en MATLAB que cree, para todo $n \in \mathbb{N}$, la matriz tridiagonal:

$$\mathbf{T}_n := \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{pmatrix}.$$

Ejercicio

Ejercicio (para la casa)

Escriba un fragmento de código en MATLAB que cree, para todo $n \in \mathbb{N}$, las matrices triangulares:

$$\mathbf{L} := \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 1 & \cdots & 1 & 1 \end{pmatrix} \quad \text{y} \quad \mathbf{U} := \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 0 & 2 & \cdots & 2 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & n \end{pmatrix}$$

Más aún, construya la matriz $\mathbf{A} = \mathbf{LU}$, y verifique (si es el caso) que el determinante de la matriz \mathbf{A} es $n!$.