

Programación en lenguaje MATLAB

Clase 1

Dr. Ing. Rodrigo Gonzalez

`rodrazalez@frm.utn.edu.ar`

Universidad Tecnológica Nacional,
Facultad Regional Mendoza.

1 Introducción

- Curso
- ¿Qué es MATLAB?
- ¿Qué es Simulink?
- MATLAB, ¿para qué?
- MATLAB, ¿por qué?

2 Interfaz

3 Operadores escalares

- Orden de cálculo

4 Funciones matemáticas elementales

5 Variables escalares

- Precisión de variables
- Tipos de variables

6 Editor

7 Matrices y vectores

- Vectores
- Matrices
- Direccionamiento de elementos
- Direccionamiento por rangos
- Otras funciones
- Arreglo de caracteres

Requisitos del curso:

- Álgebra lineal y análisis matemático.
- Sistemas de ecuaciones, derivadas, integrales.
- No es necesario haber programado en algún lenguaje previamente.

Además:

- El curso **no** está orientado a un área en particular (Química, Física, Finanzas, ...).
- Consta de 7 clases de 4 horas.
- Se aprueba con 80% de asistencia y un examen final práctico.



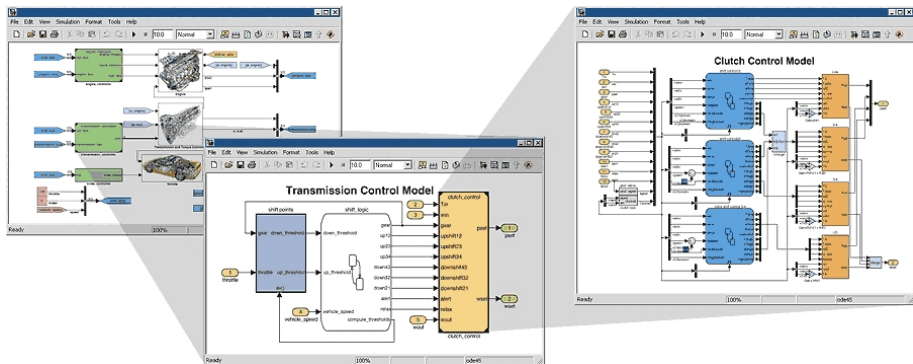
¿Qué es MATLAB?

- MATLAB: **Mat**rix **Lab**oratory.
- Desarrollado por la empresa MathWorks.
- Software propietario cerrado → \$\$\$\$.
- Opciones abiertas (gratis):
 - Octave, sintaxis compatible.
 - Freemat, sintaxis compatible.
- Es una entorno de programación completo (IDE).
- También es un lenguaje de alto nivel
 - Interpretado, como Java, Python.
 - Los comandos se ejecutan línea por línea.



¿Qué es Simulink?

- Una versión gráfica de MATLAB.
- La programación está orientada a conectar bloques.
- Posee muchas bibliotecas.
- Trabaja en conjunto con MATLAB, no son cosas separadas.



MATLAB, ¿para qué?

- Computación matemática.
- Modelado y simulación.
- Procesamiento y análisis de datos (Data Science).
- Desarrollo y verificación de algoritmos.
- Graficos en 2D y 3D.
- Desarrollo de software, GUI incluido.



MATLAB, ¿por qué?

- ¡Lo usa todo el mundo!

TIOBE Programming Community Index for April 2016

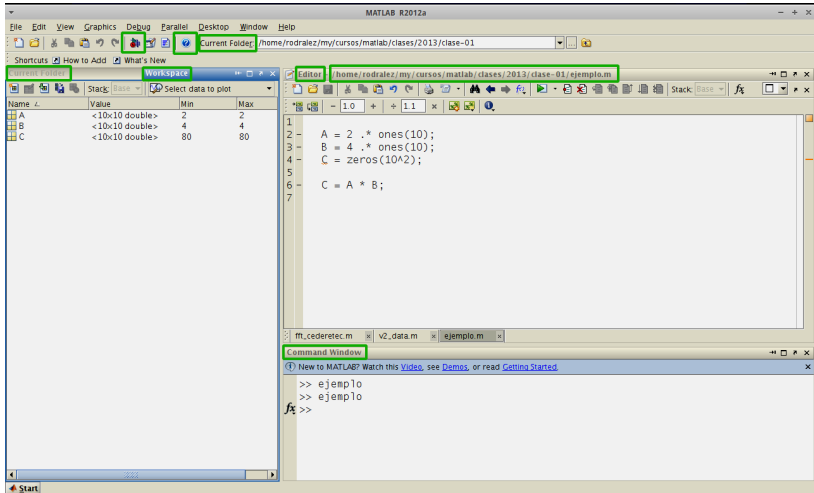
Apr 2017	Apr 2016	Change	Programming Language	Ratings	Change
1	1		Java	15.568%	-5.28%
2	2		C	6.966%	-6.94%
3	3		C++	4.554%	-1.36%
4	4		C#	3.579%	-0.22%
5	5		Python	3.457%	+0.13%
16	14	▼	Visual Basic	2.058%	+0.45%
17	16	▼	MATLAB	2.045%	+0.70%
18	44	⬆	Go	1.974%	+1.73%



MATLAB, ¿por qué?

- Popular en la industria y academia.
- Muchos libros con práctica en MATLAB.
- Optimizado para cálculo numérico.
- Optimizado para operar con matrices.
- Se puede integrar con otros lenguajes como C, C++, Fortran, Java™ y Microsoft® Excel®.
- Toolbox, toolbox y más toolbox.





La interfaz de MATLAB está en inglés.

- Current folder.
- Workspace window.
- Editor window.
- Command window.
- Command history window.
- Help window.



Command window

- Se comienza a escribir el comando a partir del *prompt* ».
- Sensible a mayúsculas (case sensitive).
- Muchos comandos, separados por comas.
- Si comando comienza con ' % ', se considera comentario.
- Si comando termina en ' ; ', no se muestra salida en consola.
- Si comando finaliza con ' . . . ', no se ejecuta y se esperan más comandos en siguiente línea.
- Números decimales se indican con punto (.), NO coma (,).
- Con flechas ARRIBA y ABAJO recorro los comandos ya ejecutados.
- Comando `clc` borra la pantalla.
- Comando `clear x` borra la variable `x` del Workspace.
- Comando `clear all` borra todas las variables del Workspace.
- Variable `ans` contiene el último resultado de lo que se ejecutó en consola.



Operadores escalares

- Suma '+'

```
» 2 + 3
```

- Resta '-'

```
» a = 7 - 11
```

```
» b = a + 5, b + b
```

- Multiplicación '*'

```
» c = 9.2 * 7.0;
```

```
» e = d * 0.98;
```

```
» clc
```

```
» clear all
```

Operadores escalares (2)

- División por derecha '/'

```
» d = 1.45 / 70.8
```

- División por izquierda '\'

```
» e = 70.8 \ 1.45
```

- Exponente '^'

```
» 23^2
```



Orden de cálculo

- 1 Paréntesis.
- 2 Exponente.
- 3 Multiplicación y división.
- 4 Suma y resta.
- 5 A igual orden, se ejecuta de izquierda a derecha.



Ejercicio 1

Ingrese por consola:

- 1 » $7+8/2$
- 2 » $(7+8)/2$
- 3 » $4 + 5/3 + 2$
- 4 » $5^3/2$
- 5 » $(5^3)/2$
- 6 » $5^{(3/2)}$
- 7 » $0.5784 - (0.8945) \dots$
- 8 » $^3 * 3 - 89$
- 9 » $4*3/6$

Funciones matemáticas elementales

- $\text{sqrt}(x) \rightarrow \sqrt{x}$
- $\text{exp}(x) \rightarrow e^x$
- $\text{abs}(x) \rightarrow |x|$
- $\text{log}(x) \rightarrow \ln x$
- $\text{log10}(x) \rightarrow \log_{10} x$
- $\text{factorial}(x) \rightarrow x!$
- $\sin(y), \cos(y), \tan(y) \dots$ ¡siempre en radianes!
- $\text{sind}(y), \text{cosd}(y), \text{tand}(y) \dots$ ¡siempre en grados!
- $\text{asin}(a), \text{acos}(a), \text{atan}(a) \dots$
- y más...



Variables escalares

- Operador asignación '='
- nombre_variable = valor o expresión.

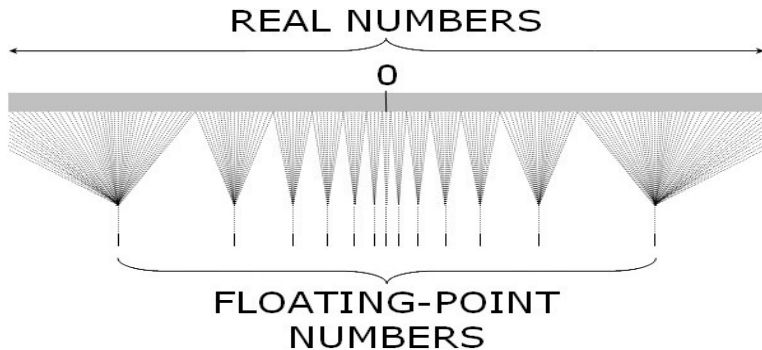
Ejercicio 2

```
1 >> x = 15 ;  
2 >> x = 5*x + 7  
3 >> x = 0.75;  
4 >> b = sin(x)^2 + cos(x)^2
```

Nota: ver nuevas variables en el Workspace.



Precisión de variables



Nota: ver columna 'class' en el Workspace. Agregar haciendo click derecho en la primera fila.

MATLAB soporta varios tipos de precisiones numéricas.

- Precisiones típicas:
 - Punto flotante de 64 bits (`double`), precisión por defecto.
 - Punto flotante de 32 bits (`single`).
- Precisiones más "exóticas": enteros (`int64`, `uint64`, `int32`, `uint32`, ...), punto fijo (fixed-point).

Precisión de variables

Rango en punto flotante de 64 bits (double)

- Negativo: $-1.79769\text{E}+308$ a $-2.22507\text{E}-308$
- Positivo: $2.22507\text{E}-308$ a $1.79769\text{E}+308$
- » `realmax, realmin`

Rango en punto flotante de 32 bits (single)

- Negativo: $-3.40282\text{E}+38$ a $-1.17549\text{E}-38$
- Positivo: $1.17549\text{E}-38$ a $3.40282\text{E}+38$
- » `realmax('single'), realmin('single')`

Números punto flotante especiales:

- Inf, -Inf. » `10/0, -8/0`
- NaN (Not a Number). » `0/0, Inf/Inf`

Función `eps()`

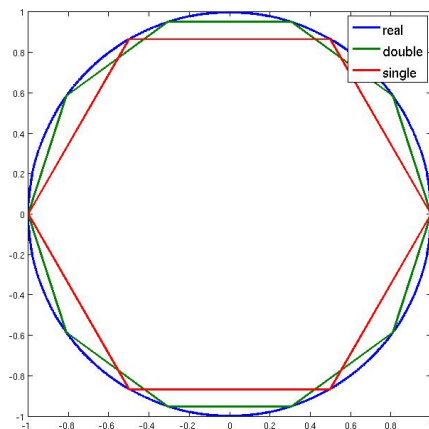
- Menor número entre un número y el siguiente número positivo.
- `eps`, sin argumento, menor número entre 1 y siguiente.

Ejercicio 3

```
1 » as = single(1)
2 » e = eps(1)
3 » 1.0 + 1e-32 % notacion cientifica
4 » x=1, tan(x) - sin(x)/cos(x)
5 » (2^52 + 1) - 2^52
6 » (2^53 + 1) - 2^53
```

Precisión de variables

En definitiva...



Tipos de variables

MATLAB soporta varios tipos de variables:

- Números (double, single, int64, ...)
- Caracteres (char) Ej: `a = 'd'`
- Cadena de caracteres (char) Ej: `s = 'Amo MATLAB'`
- Número complejos (complex) Ej: `a=3+4i`, `a=6-8j`
- Celdas (cell) Ej: `paciente={'Juan' , 43}`

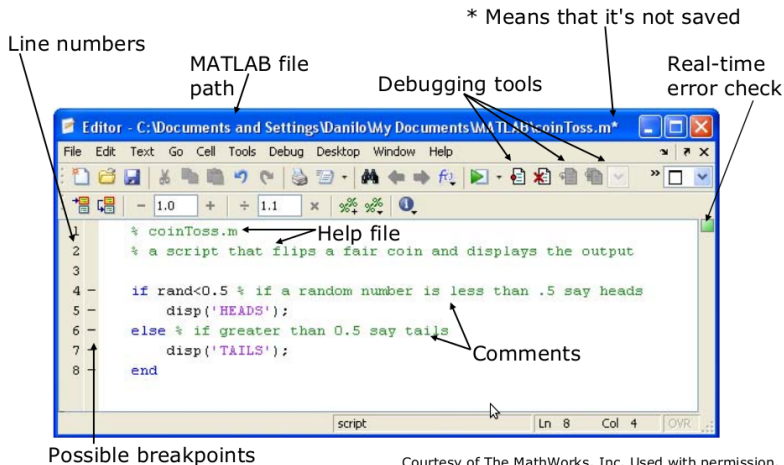


Nombre de variables, programas y funciones

A tener en cuenta:

- Pueden tener hasta 60 caracteres (depende de la versión).
- Pueden contener letras, números y guiones **bajos** ('_').
¡No guiones **medios**! ('—')
- Siempre debe comenzar con una letra Ej: q_1 , q_{-1}
- MATLAB es sensible a las mayúsculas $a \neq A$
- Evitar nombre de funciones de MATLAB.
- Evitar palabras reservadas de MATLAB como `ans`, `pi`, `eps`, `inf`, `i`, `j` o `NaN`.





Courtesy of The MathWorks, Inc. Used with permission.

Ejercicio 4

Cree un programa llamado `hola_mundo` que muestre por consola
Hola Mundo! Estoy aprendiendo MATLAB.

Tip: la función `disp()` muestra texto por consola.

Nota: los nombres de archivos siguen las mismas reglas de los nombres de variables.

Ejercicio 5

Una identidad trigonométrica está dada por:

$$\cos^2 \frac{x}{2} = \frac{\tan x + \sin x}{2 \tan x}$$

Verifique la igualdad calculando cada lado de la igualdad por separado. Puede utilizar $x = \frac{\pi}{5}$.



Más ejercicios:

4. Calculate:

$$a) \quad \cos^2\left(\frac{5\pi}{6}\right) \sin\left(\frac{7\pi}{8}\right)^2 + \frac{\tan\left(\frac{\pi}{6} \ln 8\right)}{\sqrt{7}}$$

$$b) \quad \cos\left(\frac{5\pi}{6}\right)^2 \sin^2\left(\frac{7\pi}{8}\right) + \frac{\tan\left(\frac{\pi \ln 8}{6}\right)}{7 \cdot \frac{5}{2}}$$

5. Define the variable x as $x = 13.5$, then evaluate:

$$a) \quad x^3 + 5x^2 - 26.7x - 52$$

$$b) \quad \frac{\sqrt{14x^3}}{e^{3x}}$$

$$c) \quad \log|x^2 - x^3|$$

6. Define the variables x and z as $x = 9.6$, and $z = 8.1$, then evaluate:

$$a) \quad xz^2 - \left(\frac{2z}{3x}\right)^{\frac{3}{5}}$$

$$b) \quad \frac{443z}{2x^3} + \frac{e^{-xz}}{(x+z)}$$

Vectores

- nombre_vector = [números o cadena de caracteres]
- Vector fila. Ej: [1, 1, 1, 1;] o [1 1 1 1]
- Vector columna. Ej: [1; 1; 1; 1;] o [1 1 1 1]'
- Operador transpuesta, '.

Ejercicio 6

Cree dos vectores columna a partir de los datos:

Año	1984	1986	1988	1990	1992	1996
Población (millones)	127	130	136	145	158	178



Vectores

Para crear vectores con valor inicial a , final c y espaciado constante b .

- `vect = a:b:c`

Ejercicio 7

Cree los vectores:

- 1 `>> t0 = 0:0.1:10`
- 2 `>> t1 = 1:2:13`
- 3 `>> t2 = 1.5:0.1:2.1`
- 4 `>> t3 = -3:7`
- 5 `>> t4 = 21:-2:1 % vector descendente`
- 6 `>> t5 = linspace(1, 10, 30) % inicio, fin,
cantidad`



Matrices

- nombre_mat = [números o caracteres]
- En MATLAB toda variable es una matriz.
- **Importante**: todas las filas deben tener el mismo número de elementos.

Ejercicio 8

Cree las matrices:

```
1 » A1 = [1, 2, 3; 4, 5, 6; 7, 8, 9;]  
2 » A2 = [1 2 3; 4 5 6; 7 8 9;]  
3 » A3 = [1 2 3 ...  
4 4 5 6 ...  
5 7 8 9;]
```



Ejercicio 9

```
1 » A = [1:5; 20:1:25; linspace(10, 10.6, 5);]  
2 » a=3, b=5, c=8;  
3 » B = [c; b*a; sin(b)^2; sqrt(14/a*b); 14 ]
```



Funciones elementales:

- `zeros(m,n)` , `zeros(n)`
- `ones(m,n)` , `ones(n)`
- `eye(n)` , `eye(m,n)`
- `inf(n)` , `nan(m)`
- Transpuesta A'

Ejercicio 10

- 1 » `Z1 = zeros(3,5)`
- 2 » `Z2 = zeros(5)`
- 3 » `A = 3 * ones(20,4)`
- 4 » `I = eye(6)`
- 5 » `At = A'` % doble click en `At` muestra su contenido

Direccionamiento de elementos

- `vect (n)`
- `A (m, n)`
- `A (n)`
- **Importante:** en MATLAB por defecto los elementos de una matriz se ordenan por columnas.

Ejercicio 11

Cree las matrices:

- 1 `>> A = [1:5; 6:10; 11:15]`
- 2 `>> A(3, 1)`
- 3 `>> A(3)`
- 4 `>> A(8)`



Direccionamiento por rangos

- $A(:, n)$ Todos los elementos de la columna n .
- $A(n, :)$ Todos los elementos de la fila n .
- $A(:, p:q)$ Todos los elementos entre columnas p y q .
- $A(m:n, :)$ Todos los elementos entre filas m y n .
- $A(m:n, p:q)$ Elementos entre filas m y n , y columnas p y q .
- $A(:, p:\text{end})$ Palabra reservada `end` indica último elemento del arreglo.

Ejercicio 12

- 1 `» A = [1:5; 6:10; 11:15; 16:20; 21:25]`
- 2 `» e = A(end)`
- 3 `» v1 = A(:, 1)`
- 4 `» v2 = A(3, :)`
- 5 `» D = A(3:5, 1:3)`
- 6 `» E = A(1:2:5, 2:2:5)`

Ejercicio 13

```
1 » A = [1:5; 6:10; 11:15]
2 » A(:,1) = zeros(3,1)
3 » A(3,:) = 21:25
4 » D = [A eye(3)] % matriz de matrices, columnas
5 » E = [A; ones(5)] % matriz de matrices, filas
6 » F(15,15) = A(3,4) % creacion matriz orden 15
7 » v(15) = A(1,5) % creacion vector orden 15
```



Agregar y borrar elementos

Ejercicio 14

```
1 >> A = [1:5; 6:10; 11:15];  
2 >> B = A;  
3 >> A(4,:) = 16:20 % agrega fila 4 en A  
4 >> B(2,:) = [] % elimina fila 2 de B
```



Otras funciones

- `n = length(v)` devuelve el número de elementos de vector `v`.
- `[m,n] = size(X)` devuelve dimensiones de `X` en un vector.
- `reshape(X,p,q)` modifica la dimensión de `X` de $m \times n$ a $p \times q$.
Se debe cumplir que $m * n = p * q$.
- `A = diag(v)` crea matriz con `v` en la diagonal.
- `v = diag(A)` extrae la diagonal de la matriz `A`.

Ejercicio 15

```
1 » I = eye(5,10);  
2 » [a,b] = size(I);  
3 » m = max(size(I));  
4 » v1 = diag(I);  
5 » II = diag(v1 * 5);  
6 » v2 = 1:20; R = reshape(v2, 4, 5);  
7 » N = nan(7); I = inf(7);
```

Ejercicio 16

- 1 Cree una matriz de 6×6 con ceros.
- 2 Copie un 5 en las filas y columnas centrales.
- 3 Elimine la última fila.
- 4 Modifique la dimensión de la matriz a 2×15 .



Arreglo de caracteres

- Un arreglo de caracteres también es una matriz y se utilizan las mismas herramientas.
- Una matriz de caracteres solo puede contener caracteres, no números. Ej: `{'d', 4}`
- A su vez, una matriz de números no puede contener caracteres.
- Esta situación se salva usando celdas.

Ej: `paciente = {'Juan', 43, 4246587}`

Ejercicio 17

- 1 » `nombre = 'Juan Gomez'`
- 2 » `size(nombre)`
- 3 » `nombre(:, 1:4)`
- 4 » `nombre(:, 6:14) = 'Gutierrez'`



Más ejercicios

- . Create the following matrix A:

$$\begin{bmatrix} 0.1 & 0.2 & 0.3 & 0.4 & 0.5 & 0.6 & 0.7 \\ 14 & 12 & 10 & 8 & 6 & 4 & 2 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 3 & 6 & 9 & 12 & 15 & 18 & 21 \end{bmatrix}$$

- a) Create a 3×4 matrix B from the 1st, 2nd, and 3rd rows, and the 1st through the 4th columns of the matrix A.
- b) Create a 2×7 matrix C from the 2nd, and 3rd rows, and all the columns of the matrix A.

- . The following matrix is defined in MATLAB: $M =$

$$\begin{bmatrix} 6 & 9 & 12 & 15 & 18 & 21 \\ 4 & 4 & 4 & 4 & 4 & 4 \\ 2 & 1 & 0 & -1 & -2 & -3 \\ -6 & -4 & -2 & 0 & 2 & 4 \end{bmatrix}$$

By hand (pencil and paper) write what will be displayed if the following commands are executed by MATLAB. Check your answers by executing the commands with MATLAB.

- a) $A=M([1,3], [2,4])$
- b) $B=M(:, [1,4:6])$
- c) $C=M([2,3], :)$

- . Using the zeros, ones, and eye commands create the following arrays:

a) $\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$

b) $\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

c) $\begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}$