



**Universidad de Chile**  
Facultad de Cs. Físicas y Matemáticas  
Departamento de Ciencias de la Computación

**CC3501 - Computación Gráfica, Visualización  
y Modelación para Ingenieros**  
Prof. María Cecilia Rivara

# Introducción a Matlab

## Auxiliar 1

Universidad de Chile  
Facultad de Cs. Física y Matemáticas  
Departamento de Ciencias de la Computación  
CC3501 - Computación Gráfica, Visualización y Modelación para Ingenieros



# Contenido

- 1 **Introducción**
  - Características
  - Espacio de Trabajo
  - Editor
- 2 **Programación**
  - Variables
  - Matrices
  - Condicionales
  - Control de flujo
  - Funciones
- 3 **Gráficos**
  - Gráficos XY
  - Graficos 3D
- 4 **CC**

# Características

## Matlab: Laboratorio de Matrices

MATLAB (abreviatura de MATrix LABoratory, "laboratorio de matrices") es un software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows y Apple Mac OS X.

- Poderoso lenguaje de programación, orientado a aplicaciones científicas.
- Simple uso de arreglos y matrices. En Matlab todo es una matriz.
- Gran cantidad de funciones para aplicaciones generales.
- Poderosos Tolboxes para aplicaciones específicas.

# Espacio de Trabajo

## Interprete integrado

The screenshot displays the MATLAB 7.8.0 (R2009a) interface. The main window is divided into several panes:

- Editor:** Shows a script file named `rect.m` with the following code:
 

```
1 function out=rect(t)
2   out=u(t+0.5)-u(t-0.5);
3   end
```
- Variable Editor:** Displays a variable `a` of type `<1x3 double>` with the following values:
 

	1	2	3	4	5
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
- Workspace:** Shows a table of variables:
 

Name	Value
a	[2,3,4]
b	<20x20 double>
- Command Window:** Displays the following commands and their outputs:
 

```
>> a=[2,3,4];
>> b=zeros(20);
f1 >>
```
- Command History:** Lists the commands entered in the Command Window, including:
 

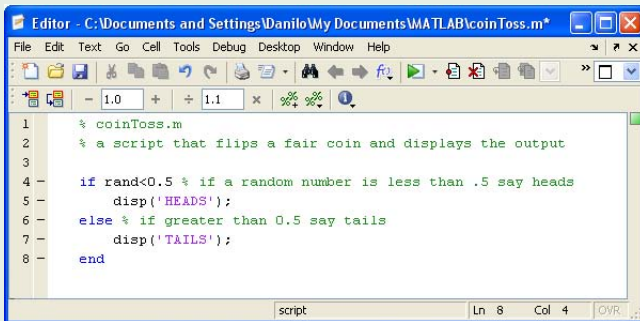
```
[A,n] = relajacion(10,1C
[A,n] = metodorelajaci
df()
1
[A,n] = relaxTarea(5,5,
[A,n] = relaxTarea(5,5,C
[A,n] = relajacion(5,5,C
17-07-10 08:51 PM -->
[A,n] = relajacion(5,5,C
[A,n] = relajacion(400,
[A,n] = relajacion(400,4
12-08-10 01:33 AM -->
12-08-10 11:24 PM -->
```

The bottom of the window shows the **Start** button and the **Over** status bar.

# Editor

- Permite guardar y editar fácilmente scripts de Matlab.
- Reconocimiento de sintaxis.
- Revisión y corrección de sintaxis.
- Siempre debería utilizar el editor.

## Editor integrado



The screenshot shows the MATLAB integrated editor window. The title bar reads "Editor - C:\Documents and Settings\Danilo\My Documents\MATLAB\coinToss.m\*". The menu bar includes File, Edit, Text, Go, Cell, Tools, Debug, Desktop, Window, and Help. The toolbar contains various icons for file operations, editing, and execution. Below the toolbar is a numeric keypad with fields for values like 1.0 and 1.1, and operators like +, -, \*, /, and %%. The main text area contains the following MATLAB script:

```
1 % coinToss.m
2 % a script that flips a fair coin and displays the output
3
4 if rand<0.5 % if a random number is less than .5 say heads
5     disp('HEADS');
6 else % if greater than 0.5 say tails
7     disp('TAILS');
8 end
```

The status bar at the bottom indicates "script", "Ln 8", "Col 4", and "OVR".

# Variables

- Tipado dinámico, no requiere especificar tipo de variable.
- Nombres son sensibles a mayúsculas.
- Punto y coma evita el despliegue del resultado.

64-bit Double [default]

1

```
>> a = 64.5;
```

# Variables

- Tipado dinámico, no requiere especificar tipo de variable.
- Nombres son sensibles a mayúsculas.
- Punto y coma evita el despliegue del resultado.

## 64-bit Double [default]

```
1 >> a = 64.5;
```

## 8-bit Char

```
1 >> b = 'a';
```

# Variables

- Tipado dinámico, no requiere especificar tipo de variable.
- Nombres son sensibles a mayúsculas.
- Punto y coma evita el despliegue del resultado.

## 64-bit Double [default]

```
1 >> a = 64.5;
```

## 8-bit Char

```
1 >> b = 'a';
```

## String

```
1 >> s = 'This_is_a_string';
```

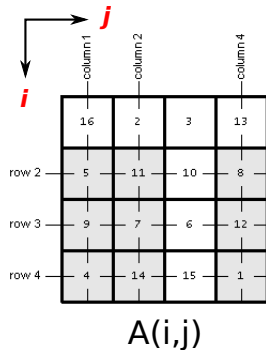


# Matrices

## Vectores

```
A=[16 2 3 13]; B=[5; 9; 4];
```

- ¡Índices comienzan desde 1!
- Sistema de referencia según figura:



	column 1	column 2		column 4
	16	2	3	13
row 2	5	11	10	8
row 3	9	7	6	12
row 4	4	14	15	1

$A(i,j)$

# Matrices

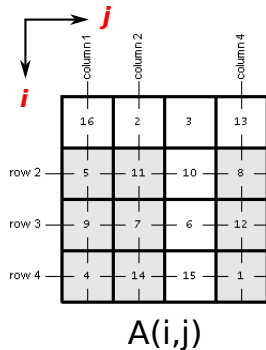
## Vectores

$A = [16 \ 2 \ 3 \ 13]; B = [5; \ 9; \ 4];$

## Matriz

$C = [5 \ 11 \ 10 \ 8; \ 9 \ 7 \ 6 \ 12]$

- ¡Índices comienzan desde 1!
- Sistema de referencia según figura:



	column 1	column 2		column 4
	16	2	3	13
row 2	5	11	10	8
row 3	9	7	6	12
row 4	4	14	15	1

$A(i,j)$

# Matrices

## Vectores

```
A=[16 2 3 13]; B=[5; 9; 4];
```

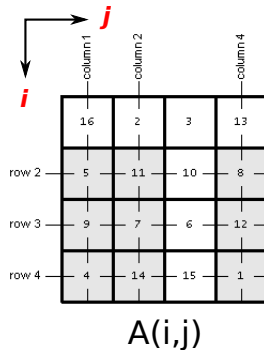
## Matriz

```
C=[5 11 10 8; 9 7 6 12]
```

## Concatenar Matriz

```
D=[A;C]
```

- ¡Índices comienzan desde 1!
- Sistema de referencia según figura:



	column 1	column 2	column 3	column 4
row 1	16	2	3	13
row 2	5	11	10	8
row 3	9	7	6	12
row 4	4	14	15	1

$A(i,j)$

# Matrices

## Vectores

```
A=[16 2 3 13]; B=[5; 9; 4];
```

## Matriz

```
C=[5 11 10 8; 9 7 6 12]
```

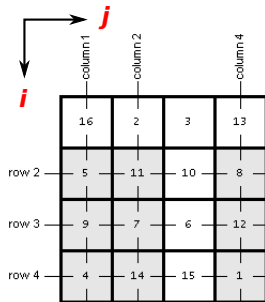
## Concatenar Matriz

```
D=[A;C]
```

## Indexar Matriz

```
>> [i,j]=size(D) → i=3, j=4  
>> D(2,3) → 10  
>> D(2,:) → [5 11 10 8]  
>> D(1:2,3) → [3; 10]
```

- ¡Índices comienzan desde 1!
- Sistema de referencia según figura:



	column1	column2	column3	column4
row 1	16	2	3	13
row 2	5	11	10	8
row 3	9	7	6	12
row 4	4	14	15	1

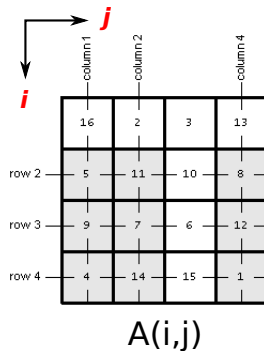
$A(i,j)$

# Matrices

## Matriz de ceros

```
A=zeros(5,5);
```

- ¡Índices comienzan desde 1!
- Sistema de referencia según figura:



	column 1	column 2		column 4
	16	2	3	13
row 2	5	11	10	8
row 3	9	7	6	12
row 4	4	14	15	1

$A(i,j)$

# Matrices

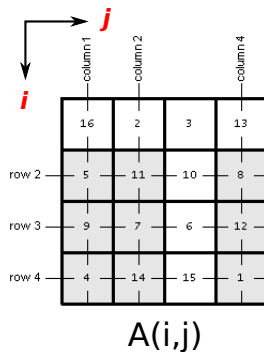
## Matriz de ceros

```
A=zeros(5,5);
```

## Matriz de unos

```
A=ones(5,5);
```

- ¡Índices comienzan desde 1!
- Sistema de referencia según figura:



	column 1	column 2	column 3	column 4
row 1	16	2	3	13
row 2	5	11	10	8
row 3	9	7	6	12
row 4	4	14	15	1

$A(i,j)$

# Matrices

## Matriz de ceros

```
A=zeros(5,5);
```

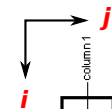
## Matriz de unos

```
A=ones(5,5);
```

## Matriz random

```
A=rand(5,5);
```

- ¡Índices comienzan desde 1!
- Sistema de referencia según figura:



	column 1	column 2	column 3	column 4
row 1	16	2	3	13
row 2	5	11	10	8
row 3	9	7	6	12
row 4	4	14	15	1

$A(i,j)$

# Matrices

## Matriz de ceros

```
A=zeros(5,5);
```

## Matriz de unos

```
A=ones(5,5);
```

## Matriz random

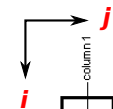
```
A=rand(5,5);
```

## Matriz dispersa

Estructura de datos útil para matrices con muchos ceros. Ahora espacio en memoria.

```
A=sparse(5,5);
```

- ¡Índices comienzan desde 1!
- Sistema de referencia según figura:



	column 1	column 2	column 3	column 4
row 1	16	2	3	13
row 2	5	11	10	8
row 3	9	7	6	12
row 4	4	14	15	1

$A(i,j)$



# Condicionales - Control de flujo

## Sentencia IF-ELSE

```
1  if (cond)
2  instruccion
3  elseif (cond)
4  instruccion
5  else
6  instruccion
7  end
```

Operador	Significado
<	Menor que
>	Mayor que
<=	Menor o igual
>=	Mayor o igual
&&	Y lógico
	O lógico

# Condicionales - Control de flujo

## Sentencia IF-ELSE

```
1  if (cond)
2      instruccion
3  elseif (cond)
4      instruccion
5  else
6      instruccion
7  end
```

Operador	Significado
<	Menor que
>	Mayor que
<=	Menor o igual
>=	Mayor o igual
&&	Y lógico
	O lógico

## Sentencia FOR

```
1  for i=start:end
2      instruccion
3  end
```

```
1  for i=start:step:end
2      instruccion
3  end
```

# Funciones

1 **function** [x,y,z] = foo(a,b,c)

- Palabra reservada: **function**
- Nombre de función **foo**. Se guarda en archivo **foo.m**
- Múltiples entradas: **a, b, c**
- Múltiples salidas: **x, y, z**. No requiere sentencia return.

```
C:\MATLAB6p5\work\stats.m
File Edit View Text Debug Breakpoints Web Window Help
% stats: computes the average, standard deviation, and range
% of a given vector of data
%
% [avg,sd,range]=stats(x)
% avg - the average (arithmetic mean) of x
% sd - the standard deviation of x
% range - a 2x1 vector containing the min and max values in x
% x - a vector of values
function [avg,sd,range]=stats(x)
avg=mean(x);
sd=std(x);
range=[min(x); max(x)];
```

Help file

Function declaration

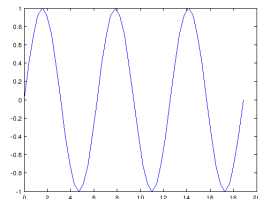
Outputs

Inputs

# Gráficos: Gráficos XY

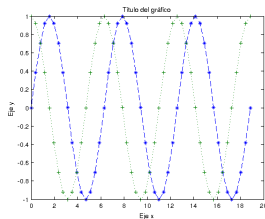
## Función plot

```
1 x = 0:pi/8:6*pi;  
2 y = sin(x);  
3 plot(x,y)
```



## Detallar gráficos

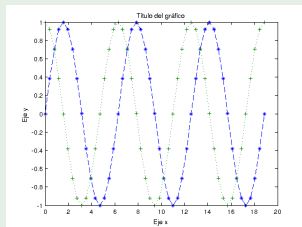
```
1 x = 0:pi/8:6*pi;  
2 y1 = sin(x); y2 = cos(x);  
3 plot(x,y1,'*--',x,y2,':+')  
4 title('Título del gráfico');  
5 xlabel('Eje_x');  
6 ylabel('Eje_y');
```



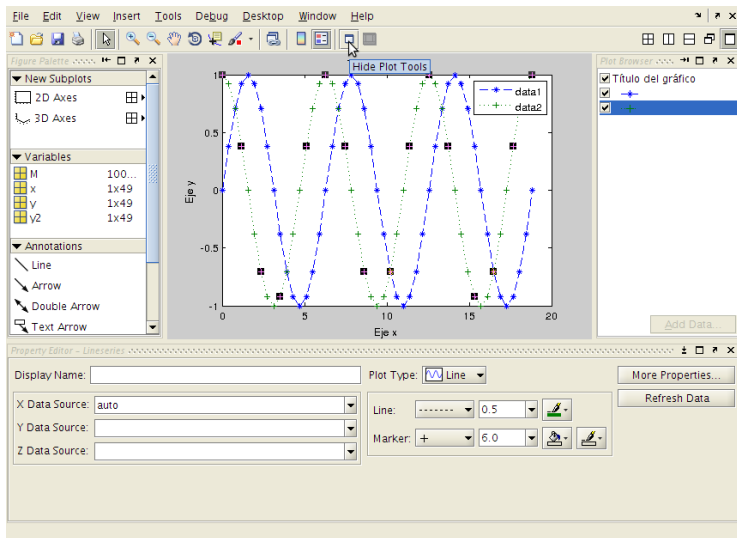
# Gráficos: Gráficos XY

## Detallar gráficos

```
1 x = 0:pi/8:6*pi;  
2 y1 = sin(x); y2 = cos(x);  
3 plot(x,y1,'*--',x,y2,':+')  
4 title('Títutlo');  
5 xlabel('Eje_x');  
6 ylabel('Eje_y');
```



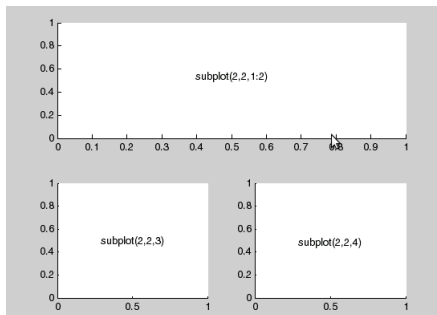
# Gráficos: Editor



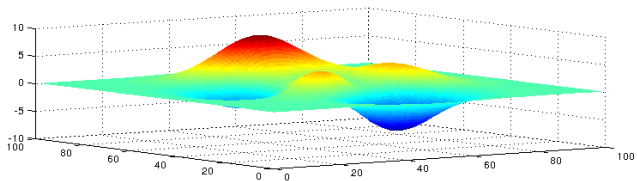
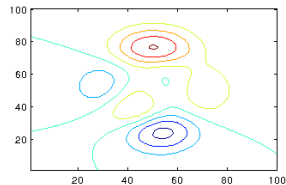
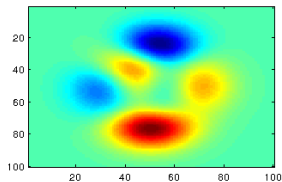
# Gráficos: 2D-3D

## Funciones útiles

- `imagesc`: Mapa de colores de una matriz en 2D.
- `contour`: Curva de nivel de una superficie en 2D.
- `surf`: Superficie 3D
- `subplot`: Varios gráficos en una misma figura.



# Gráficos: 2D-3D







**Introducción a Matlab por Carlos González Cortés** se encuentra bajo una Licencia Creative Commons Atribución-NoComercial-LicenciarIgual 3.0 Unported.