

# Práctico 0

## Repaso de elementos de Octave

Jimena Ferreira (jimenaf@fing.edu.uy)  
Claudia Santiviago(csantiviago@fing.edu.uy)  
Ana Paula Nova (anova@fing.edu.uy)  
Dinámica y Control de Procesos  
Marzo 2021

# Anuncios del curso

## Programa del curso

- <https://eva.fing.edu.uy/mod/resource/view.php?id=26097>

## Práctico

- Al comienzo de cada semana se publicarán los videos de la resolución de los ejercicios correspondiente a esa semana.
- Se subirán algunos ejercicios, la realización del resto queda a cargo de cada estudiante. (Recomendamos hacerlos y realizar las consultas en las clases de consulta)
- Tendremos clase de consulta virtuales los días miércoles y jueves.

## Evaluaciones

- Se detalla en el eva del curso como serán las evaluaciones de este año.

# Qué es Octave?

- ❑ Es un lenguaje de alto nivel, en un ambiente interactivo para la computación numérica, visualización y programación.
  - Lenguaje estructurado, secuencial e interpretativo
  - Está basado en matrices

# ¿Para qué usar Octave?

- ❑ Cálculos numéricos:
  - Simulación
  - Estimación de parámetros y análisis estadístico
  - Optimización
  - Control
- ❑ Programación:
  - Algoritmos de resolución de un problema
  - Paquetes de resolución numérica
- ❑ Visualización

# Nociones básicas

Número: 2568

Vectores:  $(1\ 2\ 3\ 4)$  → `[1,2,3,4]` o `[1 2 3 4]` o `[1:4]` o `1:4`

Matrices:  $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{pmatrix}$  → `[1,2,3,4;5 6 7 8]` o `[1:4;5:8]`  
o `[[1;5], [2;6], [3;7], [4;8]]`

Creación de variables:

`Nombre_variable = Matriz`

`Nombre_variable = 'Palabra'`

¡¡Ojo con los nombres!!

Algunas operaciones: `+` `-` `*` `/` `.*` `./` `'`

Algunos comandos útiles: `%` `,` `;` `length(v)` `size(M)` `inv(M)`

`help rutina`

# Editor

Permite guardar sentencias o comandos en un archivo.

Por ejemplo, las operaciones del Ejercicio 1 pueden ser guardadas en un archivo.

Los archivos de Octave son guardados como **.m**



¡¡Cuidado que Matlab también los abre!!

# Scripts y funciones

## SCRIPTS

Es análogo a escribir una sentencia o comando en la ventana de comandos.

## FUNCIONES

Permite escribir sentencias y comandos. Tiene un

¡¡Ojo con los nombres!!

**Sintaxis:**

```
function [VS1,VS2,...,VSm] = nombre_funcion(VE1,VE2,...VEn)
```

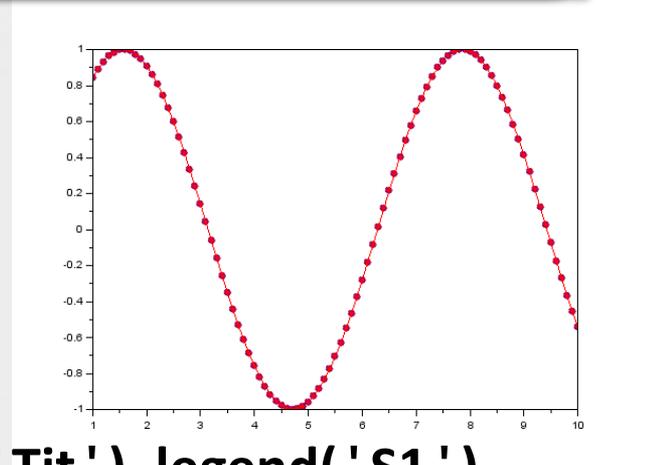
```
endfunction % es opcional si está al final del archivo
```

# Comandos de gráficos

## Principales comandos:

- Gráficos 2D: **plot(x,y)**, **fplot**
- Gráficos 3D: **mesh**, **surf**, **meshgrid**
- Ventana de grafico: **figure**
- Etiquetas: **xlabel(' x ')**, **ylabel(' y ')**, **title(' Tit ')**, **legend(' S1 ')**
- Subdivisión de ventana: **subplot(f,c,elem)**
- Mantener gráfico: **hold on**
- Formatos:
  - línea (**--** **:** **-** **.-**)
  - marcador (**o** **x** **.** **d** **\***)
  - color (**b** **r** **g** **k**)

Ejemplo: `plot(x, sin(x), '-*r')`



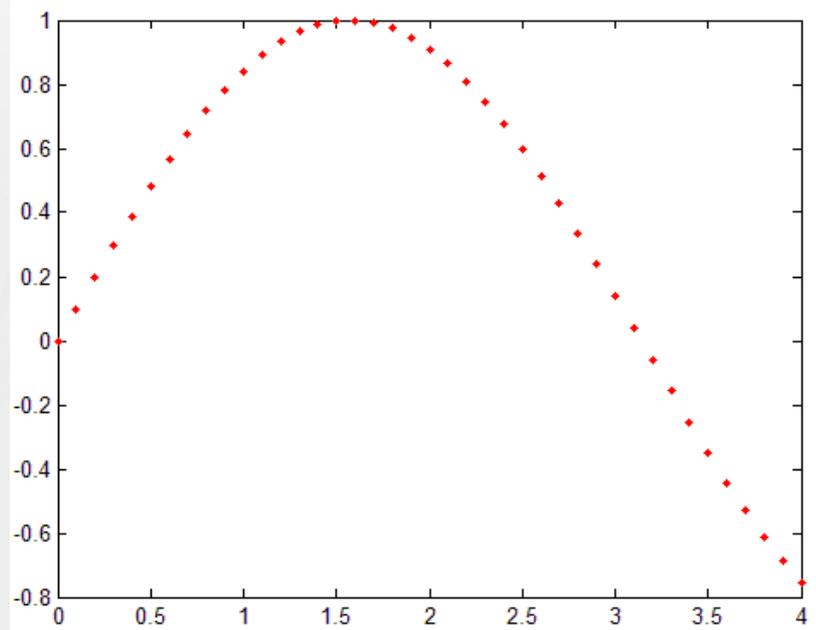
# Ejemplo plot

Funciones

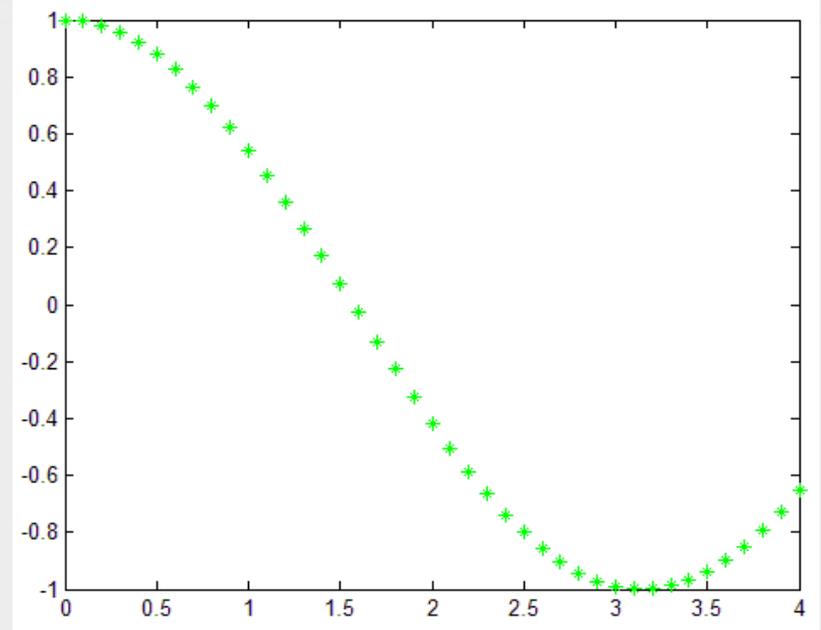
$$f_1(x) = \sin(x)$$

$$f_2(x) = \cos(x)$$

```
plot(x, sin(x), '.r')
```



```
plot(x, cos(x), '*g')
```



*Ejemplo en Octave usando plot y fplot*

## Ejercicio 3

Algunos comandos de Octave:

Operación	Comando en Octave
$\ln(x)$	<code>log(x)</code>
$\log(x)$	<code>log10(x)</code>
$e^x$	<code>exp(x)</code> o <code>e.^x</code>
$a \times 10^b$	<code>aEb</code> o <code>aeb</code>

# Lenguaje de programación

## Operadores lógicos

- & | ~

## Estructuras de selección

- If-end if-else-end if-elseif-end  
if-elseif-else-end ...

## Estructuras de repetición

- for-end
- while-end

# Lenguaje de programación

Condiciones lógicas: Condiciones que tiene como respuesta verdadero (1) o falso (0).

Por ejemplo: tenemos una variable x con valor 5

condición:  $x==3$

respuesta: F

condición:  $x<1$

respuesta: F

condición:  $x>= 3$

respuesta: T

condición:  $x\neq 3$

respuesta: T

Para usar varias condiciones lógicas a la vez se deben usar conectores  $\&$  (and) u  $|$  (or).

Por ejemplo:

condiciones:  $x>3 \ \& \ x<9$  (se busca que el x esté entre 3 y 9)

condiciones:  $x>9 \ | \ x==1$  (se busca que el x sea mayor que 9 o igual a 1)

# Lenguaje de programación

```
if condiciones
  sentencias
end
```

```
if condiciones 1
  sentencias 1
  if condiciones 2
    sentencias 2
  else
    sentencias 3
  end
else
  sentencias 4
end
```

```
if condiciones 1
  sentencias 1
else
  sentencias 2
end
```

```
if condiciones 1
  sentencias 1
elseif condiciones 2
  sentencias 2
end
```

```
if condiciones 1
  sentencias 1
elseif condiciones 2
  sentencias 2
else
  sentencias 3
end
```

# Lenguaje de programación

```
while condiciones
  sentencias
end
```

```
for var=inicio:paso:final
  sentencias
end
```

```
while condiciones
  sentencias 1
  if condiciones 2
    sentencias 2
  else
    sentencias 3
  end
  sentencias 4
end
```

```
for var=inicio:paso:final
  sentencias 1
  if condiciones 1
    sentencias 2
  else
    sentencias 3
  end
  sentencias 4
end
```

# Lenguaje de programación

## Ejemplos

```
function salida=Ej1(x)
if x <= 3
    salida=1;
else
    salida=0;
end
endfunction %opcional
```

```
function v=Ej2(v,k)

lv=length(v)
for i=1:lv
    if v(i)<k
        v(i)=k;
    end
end
endfunction %opcional
```

```
t = [0:10]
M = [0:0.2:2;0:0.1:1]
f = 1
valor = 0.7
[fil,col]=size(M)
i=1
while M(f,i)<valor & i<=col
    i=i+1;
end
if i==col+1
    salida = [];
    disp('No existe');
else
    salida = t(i);
    disp('t (min) =');
    disp(salida);
end
```

# Ejercicio 4