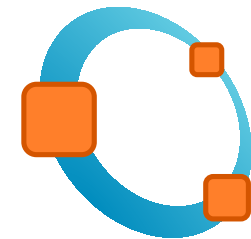
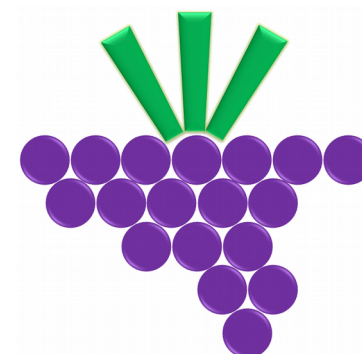




# Catalizate en Octave



## Clase 2



Daniel Millán, Nicolás Muzi,  
Petronel Schoeman, Gabriel Rosa, Juan Cruz Luffi

San Rafael, Argentina 3-4 Octubre 2019



**UNCUYO**  
UNIVERSIDAD  
NACIONAL DE CUYO



FACULTAD DE  
**CIENCIAS APLICADAS  
A LA INDUSTRIA**



# Guión de órdenes (*script*)

1. Archivos que contienen un guión de órdenes (*script*).
2. Creación y edición de un *script*.
3. Ejecución de *scripts*.
4. Resolución de problemas aritméticos simples empleando *scripts*.

# 1. Archivos *\*.m* que poseen un guión de órdenes (*scripts*).

- Los archivos con extensión *.m* son ficheros de texto sin formato (ficheros ASCII) que constituyen el centro de la programación en Octave.
- Estos archivos se crean y modifican con un editor de textos cualquiera.
- Octave 4 en adelante provee su propio editor de textos.
- Existen dos tipos de archivos *\*.m*, los ***ficheros de guiones de comandos*** (llamados *scripts* en inglés) y las ***funciones***.
- Los *scripts* contienen un conjunto de comandos (programa) que se ejecutan secuencialmente cuando se teclea el nombre del archivo en la ventana de órdenes o se incluye dicho nombre en otro fichero *\*.m*.

## 2. Creación y edición de *scripts*.

- Un archivo *script* se puede crear desde cualquier editor de texto, en particular Octave trae incorporado uno.
- El nombre no debe contener caracteres extraños al idioma inglés como: **acentos**, **ñ**, **ç**, **ü**...
- En un script se alojan guiones de órdenes que se ejecutan de forma secuencial (de arriba hacia abajo).

**Ejercicio:** crear un archivo script con el editor que se llame “hola\_pepe.m” y alojarlo en el Escritorio de su PC. El script debe contener las siguientes líneas:

```
%texto que se muestra en la terminal  
disp('hola mundo soy pepe')
```



### 3. Ejecución de *scripts*.

- El *script* se puede llamar con **F5** desde el editor o desde la línea de órdenes.
- El *script* crea variables que pertenecen al ***espacio de trabajo base*** de Octave, y permanecen en él cuando se termina la ejecución de dicho programa.
- Un *script* puede llamar a otros *scripts*.

**Ejercicio:** ejecute “hola\_pepe.m” desde el editor y desde la terminal. ¿Qué observa?

## 4. Resolución de problemas aritméticos empleando *scripts*

- **Ejercicio:** determine el valor de la siguiente expresión algebraica por medio del uso de un *script*

$$\frac{2}{(0.1)^{\frac{1}{2}}} - \frac{1}{(\cos 0.4)^{-1}} \cdot \frac{1}{(\tan 45^{\circ})^{\frac{1}{3}}}$$

# Ejercicio

- **Encuentro de dos amigos**

*sic. “El primer desafío serio que planteo en este libro es uno de los **rompecabezas** más bonitos de matemáticas recreativas y procede de Rusia. Su solución exige reflexión, concentración, creatividad, lógica, perspicacia y atención por los detalles.”*

- **Dos matemáticos rusos se encuentran en un avión:**
  - Iván: “Si no recuerdo mal, tienes 3 hijos, ¿qué edad tienen ahora?”
  - Ígor: “Sí, tengo tres hijos. El producto de sus edades es 36 y la suma de sus edades es la fecha de hoy”.
  - Iván: “Lo siento Ígor, pero eso no me dice la edad de tus hijos”.
  - Ígor: “Oh, olvidé decírtelo. Mi hijo menor es pelirrojo”.
  - Iván: “Ah, ahora está claro. Ya sé exactamente la edad de tus hijos”.
- **¿Cómo ha calculado Iván la edad de los hijos de Ígor?**



# Trazado de gráficos

1. Mejorando la presentación.
2. Funciones gráficas 2D elementales.





# 1. Mejorando la presentación.

- La capacidad de proporcionar visualizaciones de calidad a partir de datos de salida es clave en el análisis de datos.
- Sin visualización, las simulaciones numéricas son difíciles y a veces imposibles de interpretar.
- Cuando las computadoras se introdujeron en el dominio técnico-científico, la generación de publicaciones con imágenes de calidad para realizar análisis detallados de los resultados numéricos de problemas complejos fue uno de los mayores desafíos para los ingenieros y científicos de todo el mundo.
- Octave proporciona esta funcionalidad. Sus características de trazado permite elegir entre varios tipos de gráficos en 2D y 3D, decorar figuras con: títulos, nombres de ejes, cuadrículas, etiquetas para datos, ecuaciones, etc.
- La visualización de los resultados de experimentos basados en la simulación permiten una comprensión intuitiva.

## 2. Funciones gráficas 2D elementales.

- Dibujo simplificado de funciones: ***ezplot***, ***ezcontour***.
- Las ***ezALGO*** poseen varias formas de uso, siendo la más útil por su simplicidad cuando se emplean cadenas de caracteres: ***ezplot("x+2")*** ó ***ezplot('x+2')***
- Por defecto ***ezALGO*** grafica entre  $-2\pi < x | y < 2\pi$ .

**Ejercicio:** comprobar el funcionamiento de ***ezcontour***

a) ***ezcontour("cos(x/2)\*sin(y/2)")***

b) ***ezcontour("cos(x/2)+sin(y/2)")***

## 2. Funciones gráficas 2D elementales.

- Funciones orientadas a *embellecer* la figura:
  - **title**('título') añade un título al dibujo
  - **xlabel**('tal') añade una etiqueta al eje de abscisas. Con **xlabel off** desaparece
  - **ylabel**('cual') añade una etiqueta al eje de ordenadas. Con **ylabel off** desaparece
  - **text**(x,y,'texto') introduce 'texto' en el lugar especificado por las coordenadas **x** e **y**. Si **x** e **y** son vectores, el texto se repite por cada par de elementos.
  - **legend**() define rótulos para las distintas líneas o ejes utilizados en la figura. Para más detalle, consultar el **Help/doc**
  - **grid** activa la inclusión de una cuadrícula en el dibujo. Con **grid off** desaparece la cuadrícula.
  - Las propiedades se establecen sobre los ejes actuales, a los que se accede con la función **gca** (*get current axis*).

## 2. Funciones gráficas 2D elementales.

- Dado gráfico es posible definir el valor de retorno opcional de su “controlador de gráficos”.

**H = ezALGO (...)**

**Ejercicio:** comprobar las funciones anteriores en la curva plana que describe la “Espiral de Arquímedes”.

```
h=ezplot('t*cos(t)','t*sin(t)',[0,4*pi])
title('Espiral Arquimides')
xlabel('x(t)')
ylabel('y(t)')
legend('Espiral')
text(1,1,'Hola Mundo','fontsize',20,'color','red')
grid on
set(gca,'fontsize',20,'color',[0.2,0,0.2])
set(h,'linewidth',3,'color','green')
```