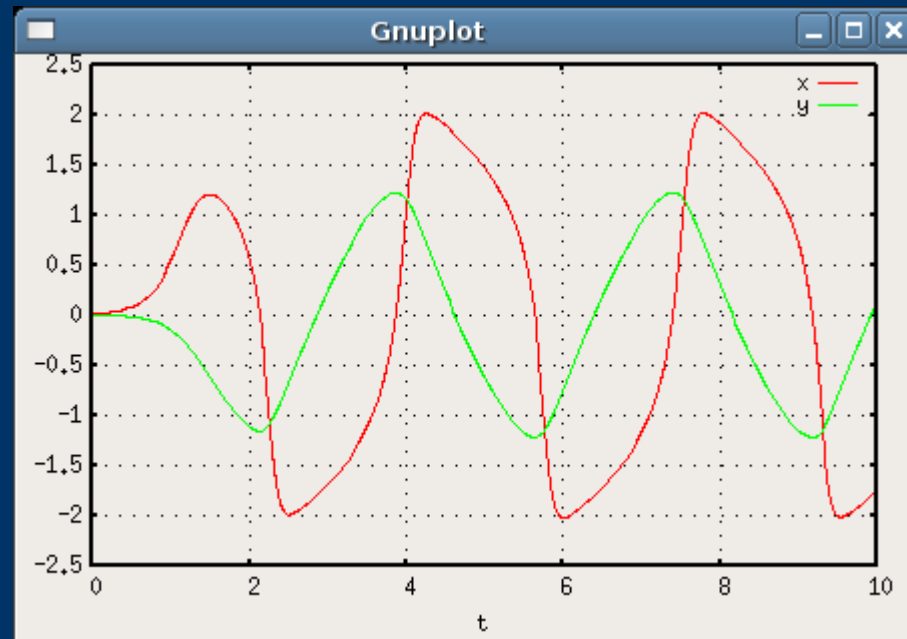


# OCTAVE



Laboratorio de Herramientas Computacionales  
I.S.C Bertha Georgina Flores Díaz

# ¿Qué es Octave?

- GNU Octave es un lenguaje de alto nivel destinado para el cálculo numérico. Provee una interfaz sencilla, orientada a la línea de comandos (consola), que permite la resolución de problemas numéricos, lineales y no lineales, además permite la ejecución de scripts.

- Octave nació alrededor del año 1988, y fue concebido originalmente para ser usado en un curso de diseño de reactores químicos para los alumnos de Ingeniería Química de la Universidad de Texas y la Universidad de Wisconsin-Madison.
- 
-

# *Iniciar y salir de Octave*

- Para ejecutar el programa escribimos en la consola el comando octave y presionamos la tecla [Enter]
- Para salir de Octave podemos usar el comando quit ó exit indistintamente.



# Cálculos básicos

- Un ejemplo sencillo de suma, multiplicación y división:

```
octave:1> 45 + 76 * 12.5 / 1.5
```

```
ans = 678.33
```

De igual manera también podemos calcular promedios:

```
octave:2> (12+15+20) / 3
```

```
ans = 15.667
```

Para calcular potencias utilizamos el símbolo “^” o el doble asterisco “\*\*”:

```
octave:3> 2^8
```

```
ans = 256
```

---

---

# Guardar y recuperar

- Al salir de Octave se perderán todas las variables que se han creado, para guardar una sesión y recuperarla mas tarde solo debe utilizar los comandos `save` y `load` respectivamente seguidos de un nombre asignado por el usuario.

```
octave:2> save nombre_sesion
```

- Esto almacenará la sesión en un archivo de nombre `nombre_sesion` dentro del directorio de trabajo. Para recuperar la sesión escribiremos lo siguiente en cualquier momento dentro del entorno Octave.

```
octave:1> load nombre_sesion var1 var2 ...
```

---

---

# *Variables*

- Las variables son identificadores que permiten almacenar datos, los cuales pueden cambiar durante la ejecución de un programa. Las variables nos permiten asignarle nombres a los valores para luego poder hacer referencia a estos.
  - No hay límite para la longitud en el nombre de una variable, pero estos deben estar constituidos por una secuencia de letras, underscores (símbolo de subrayado o guión bajo) o números y sólo pueden empezar con letras o underscores.
- 
-

# Ejemplo

- Por ejemplo, calculemos el área (A) de un trapecio con base mayor (B) de 10 cm , base menor (b) de 6 cm y la altura (h) de 4 cm.

```
octave:1> B=10
```

```
B = 10
```

```
octave:2> b=6
```

```
b = 6
```

```
octave:3> h=4
```

```
h = 4
```

```
octave:4> A=(h*(B+b))/2
```

```
A = 32
```

---

---

- También podemos introducir todos los comandos en una misma línea, para ello separamos cada comando con una coma ',', el ejemplo anterior queda así:

```
octave:5> B=10, b=6, h=4, A=(h*(B+b))/2
```

```
B = 10
```

```
b= 6
```

```
h= 4
```

```
A = 32
```

- Nota: ¿qué pasa si cambiamos la ',' por ';'?
- 
-

# *Vectores y matrices*

- En Octave disponemos de una gran variedad de formas para definir vectores y matrices, usualmente lo hacemos encerrando los elementos dentro de corchetes, los elementos separados por espacios o comas (,) definen un vector fila, los elementos separados por el retorno de carro o por punto y coma (;) definen una nuevo vector fila en la matriz.
- 
-

# Ejemplos

```
octave:1> a=[1,3,5]
a=
    1    3    5
```

```
octave:2> b=[5 7 9]
b=
    5    7    9
```

```
octave:3> c=[1;2;3]
e=
    1
    2
    3
```

```
octave:4> d=[2
> 1;0]
d=
    2
    1
    0
```

```
octave:5> e=[a ; b ;
    2, 4 6]
e =
    1    3    5
    5    7    9
    2    4    6
```

# Secuencias

- Una forma sencilla de producir una secuencia de números es utilizando la notación  $n:m$  donde  $n$  es el número inicial y  $m$  el final

```
octave:2> 1:10
```

```
ans =
```

```
1 2 3 4 5 6 7 8 9 10
```

```
octave:3> b=0:2:10
```

```
b=
```

```
0 2 4 6 8 10
```

---

---

# Funciones matemáticas

Función	Descripción
<b>sqrt ( x )</b>	Raíz cuadrada de $x$
<b>abs ( x )</b>	Valor absoluto de $x$
<b>log ( x )</b>	Logaritmo neperiano de $x$
<b>log2 ( x )</b>	Logaritmo en base a 2 de $x$
<b>log10 ( x )</b>	Logaritmo en base a 10 de $x$
<b>exp ( x )</b>	$e^x$
<b>pow2 ( x )</b>	Para cada elemento de $x$ , calcula $2^x$
<b>rem ( x , y )</b>	Resto entre la división de $x$ e $y$ (módulo)
<b>round ( x )</b>	Redondeo de $x$ al entero mas cercano

# *Sistema de ecuaciones*

- Para la resolución de sistemas de ecuaciones del tipo  $Ax = b$  utilizamos la notación  $a \backslash b$ , por ejemplo, para calcular el siguiente sistema de ecuaciones de primer grado con dos incógnitas,

$$6 \cdot x - 7 \cdot y = 5$$

$$8 \cdot x - 9 \cdot y = 7$$

- haremos lo siguiente: guardaremos los elementos  $x$  e  $y$  en una matriz  $a$  y la igualdad en un vector  $b$ , para finalmente ejecutar el comando  $a \backslash b$ .

# Ejemplo

```
octave:4> a=[6, -7;8,-9]
```

```
a=
```

```
6 -7
```

```
8 -9
```

```
octave:5> b=[5;7]
```

```
b=
```

```
5
```

```
7
```

```
octave:6> a\b
```

```
ans =
```

```
2
```

```
1
```

# Gráficos

- La forma mas simple para producir gráficas bidimensionales en Octave se hace a través del comando `plot(x[,y][,fmt])`, donde `x` y `y` opcionalmente representan los vectores coordenadas para cada punto.

```
octave:1> x = (rand(50,1));
```

```
octave:2> plot(x)
```

```
octave:3> plot(x, '@')
```