

# Introducción a MATLAB/ OCTAVE

Fundamentos Físicos  
de la Informática, 2006



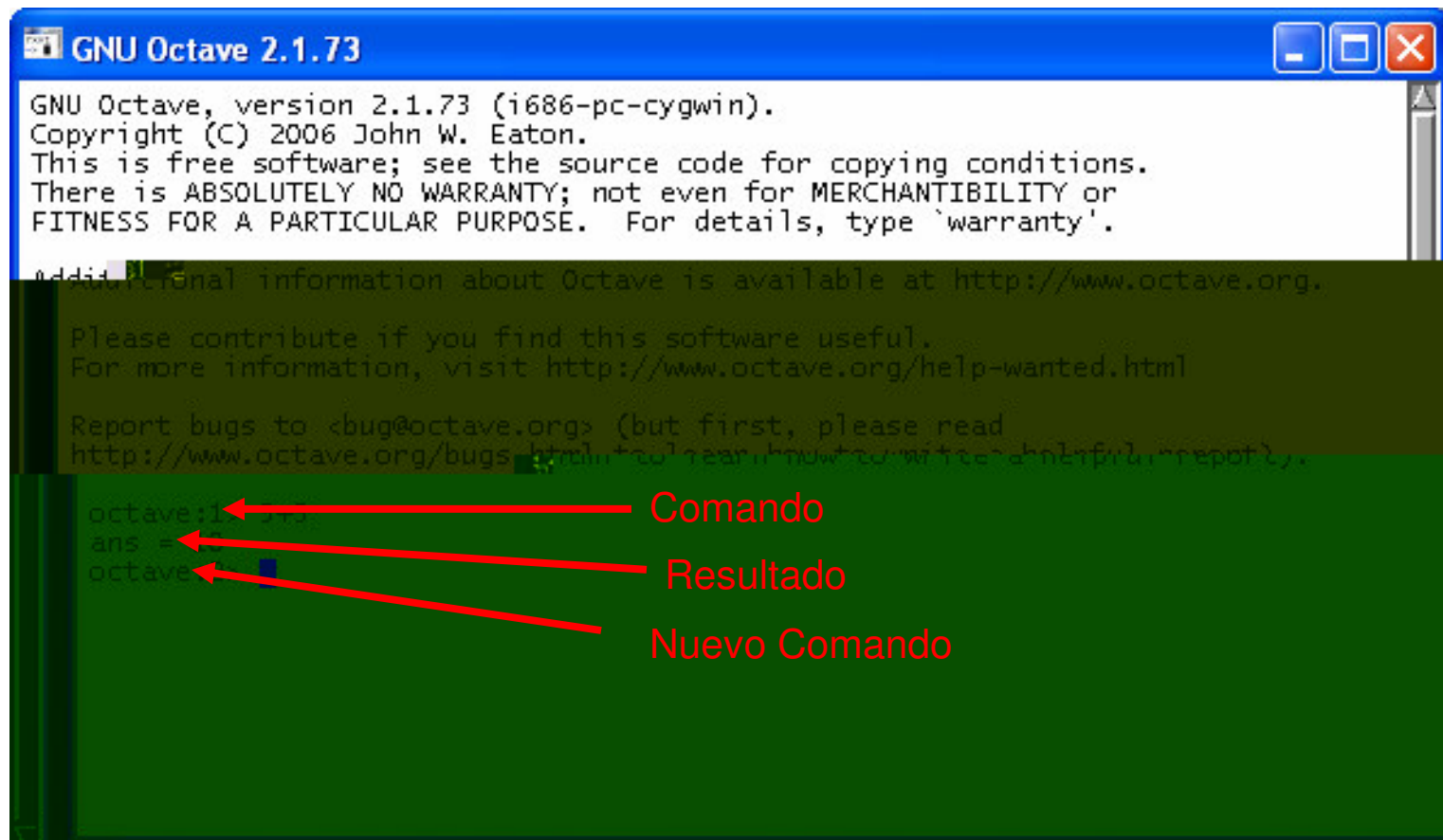
# Matlab/ Octave

- Matlab es un lenguaje de programación orientado al cálculo numérico, principalmente matricial
- Octave es un programa “clone” gratuito de Matlab

Bajar en:

<http://www.gnu.org/software/octave/download.html>

# Línea de comandos



```
GNU Octave, version 2.1.73 (i686-pc-cygwin).  
Copyright (C) 2006 John W. Eaton.  
This is free software; see the source code for copying conditions.  
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or  
FITNESS FOR A PARTICULAR PURPOSE. For details, type `warranty'.  
  
Additional information about Octave is available at http://www.octave.org.  
  
Please contribute if you find this software useful.  
For more information, visit http://www.octave.org/help-wanted.html  
  
Report bugs to <bug@octave.org> (but first, please read  
http://www.octave.org/bugs.html to learn how to write a helpful report).  
  
octave:1  
ans =  
octave:
```

Comando

Resultado

Nuevo Comando

# Comandos

- Los comandos básicos son parecidos a los de cualquier calculadora

+	adición
-	substracción
*	multiplicación
/	división
^	potencia
sqrt()	raiz cuadrada

log10()	logaritmo base 10
log()	logaritmo neperiano
exp()	exponencial
pi	constante pi
sin, cos	seno, coseno
tan	tangente

# Ejercicio 1

- Efectuar las operaciones:

$$25757 + 3221$$

$$47586 - 232$$

$$23 \times 54$$

$$245 / 33$$

$$5^3$$

$$3 \times (23 + 14.7 - 4/6) / 3.5$$

$$5^{-3}$$

$$\text{sqrt}(5)$$

$$\log(1000)$$

$$\ln(e^{10})$$

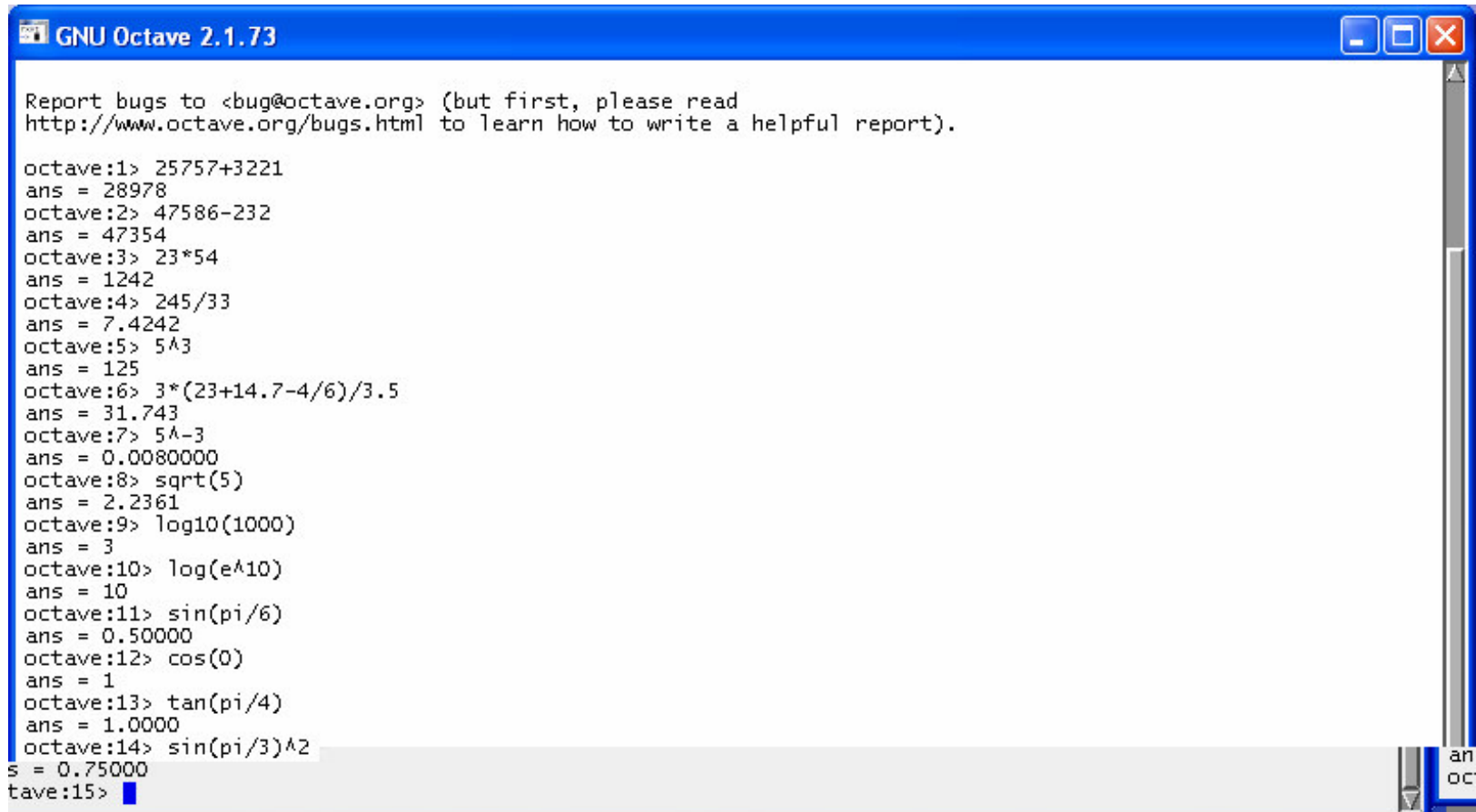
$$\sin(\pi/6)$$

$$\cos(0)$$

$$\tan(\pi/4)$$

$$\sin(\pi/3)^2$$

# Ejercicio 1 - Solución



```
GNU Octave 2.1.73
Report bugs to <bug@octave.org> (but first, please read
http://www.octave.org/bugs.html to learn how to write a helpful report).

octave:1> 25757+3221
ans = 28978
octave:2> 47586-232
ans = 47354
octave:3> 23*54
ans = 1242
octave:4> 245/33
ans = 7.4242
octave:5> 5^3
ans = 125
octave:6> 3*(23+14.7-4/6)/3.5
ans = 31.743
octave:7> 5^-3
ans = 0.0080000
octave:8> sqrt(5)
ans = 2.2361
octave:9> log10(1000)
ans = 3
octave:10> log(e^10)
ans = 10
octave:11> sin(pi/6)
ans = 0.50000
octave:12> cos(0)
ans = 1
octave:13> tan(pi/4)
ans = 1.0000
octave:14> sin(pi/3)^2
ans = 0.75000
octave:15>
```

# Caracteres especiales

- “ y ” - sirven para introducir texto
- % - símbolo de comentario, ignora la línea
- ... - continuación en la línea siguiente
- ; - separación de comandos. Si lo ponemos en el final del comando no se muestra el resultado

## Caracteres especiales - ejemplo

```
octave:1> "hola"  
ans = hola  
octave:2> 'hola'  
ans = hola  
octave:3> % Que efecto tiene esta línea?  
octave:3> sin( $\pi/6$ );  
octave:4> 'ahora las lineas...  
> se juntan'  
ans = ahora las lineas se juntan  
octave:5>
```



# Matrices y Vectores

- Una matriz es una tabla rectangular de números

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 7 \\ 4 & 9 & 2 \\ 6 & 0 & 5 \end{bmatrix} \quad \text{Matriz } 4 \times 3$$

- Un vector es una matriz de una dimensión

$$R = [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9] \quad \text{Matriz } 1 \times 9 \text{ o} \\ \text{vector fila con} \\ \text{9 elementos}$$

# Introducir una matriz

- Se introduce fila a fila
- Se separan los números con espacios o comas
- Se separan las columnas con punto y coma
- Se agrupa toda la matriz entre corchetes

# Introducir una matriz - ejemplo

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 7 \\ 4 & 9 & 2 \\ 6 & 0 & 5 \end{bmatrix} \longrightarrow A=[1 \ 2 \ 3;1 \ 2 \ 7;4 \ 9 \ 2;6 \ 0 \ 5]$$

```
octave:1> A=[1 2 3;1 2 7;4 9 2;6 0 5]
```

```
A =
```

```
1 2 3
```

```
1 2 7
```

```
4 9 2
```

```
6 0 5
```

# Introducir vectores

- Introducir un vector línea:

```
octave:2> B=[1 2 3]
```

```
B =
```

```
1 2 3
```

- Introducir un vector columna:

```
octave:3> C=[4;5;6]
```

```
C =
```

```
4
```

```
5
```

```
6
```

# Generar vectores automáticamente con incremento constante

$C=[1\ 3\ 5\ 7\ 9]$



$C=1:2:9$

$D=[2\ 5\ 8\ 11\ 14]$



$D=2:3:14$

Limite inferior

Incremento

Limite superior

## Ejercicio 2

- Generar automáticamente un vector línea de 8 elementos, cuyo primer elemento es el 6 y el octavo el 41.

## Ejercicio 2 - solución

octave:1> F=6:5:41

F =

6 11 16 21 26 31 36 41

# Operaciones entre matrices

- $+$ ,  $-$ ,  $*$  y  $^{\wedge}$  denotan la adición, substracción, multiplicación y potencia de matrices

– Ejemplos:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad B = \begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix}$$

$$A + B = \begin{bmatrix} 1+4 & 2+3 \\ 3+2 & 4+1 \end{bmatrix} = \begin{bmatrix} 5 & 5 \\ 5 & 5 \end{bmatrix}$$



# Operaciones entre matrices

$$A * B = \begin{bmatrix} (1 \times 4) + (2 \times 2) & (1 \times 3) + (2 \times 1) \\ (3 \times 4) + (4 \times 2) & (3 \times 3) + (4 \times 1) \end{bmatrix} = \begin{bmatrix} 8 & 5 \\ 20 & 13 \end{bmatrix}$$

$$A^2 = A * A = \begin{bmatrix} (1 \times 1) + (2 \times 3) & (1 \times 2) + (2 \times 4) \\ (3 \times 1) + (4 \times 3) & (3 \times 2) + (4 \times 4) \end{bmatrix} = \begin{bmatrix} 7 & 10 \\ 15 & 22 \end{bmatrix}$$

# Operaciones entre matrices

- Que pasa si ponemos un punto antes de las operaciones, por ejemplo, \* y ^?
- Si tenemos 2 matrices, A y B:

$$A^*B=A.*B \quad ?$$
$$A^2=A.^2 \quad ?$$

## Ejercicio 3

- Considerando las matrices  $A$  y  $B$ :

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad B = \begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix}$$

- 1) Calcular  $A \cdot B$  y  $A \cdot B$
- 2) Calcular  $A^2$  y  $A \cdot A$
- 3) Comentar los resultados

## Ejercicio 3 - solución

$$A * B = \begin{bmatrix} (1 \times 4) + (2 \times 2) & (1 \times 3) + (2 \times 1) \\ (3 \times 4) + (4 \times 2) & (3 \times 3) + (4 \times 1) \end{bmatrix} = \begin{bmatrix} 8 & 5 \\ 20 & 13 \end{bmatrix}$$

$$A .* B = \begin{bmatrix} (1 \times 4) & (2 \times 3) \\ (3 \times 2) & (4 \times 1) \end{bmatrix} = \begin{bmatrix} 4 & 6 \\ 6 & 4 \end{bmatrix}$$

$$A * B \neq A .* B$$

```

GNU Octave 2.1.73
octave:1> A=[1 2;3 4]
A =
   1   2
   3   4

octave:2> B=[4 3;2 1]
B =
   4   3
   2   1

octave:3> A*B
ans =
   8   5
  20  13

octave:4> A.*B
ans =
   4   6
   6   4

octave:5>

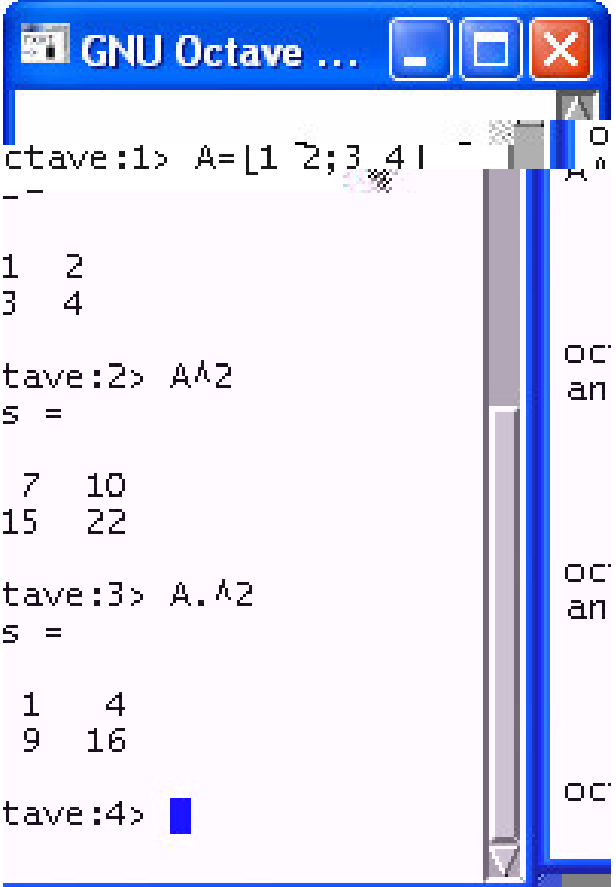
```

## Ejercicio 3 - solución

$$A^2 = A * A = \begin{bmatrix} (1 \times 1) + (2 \times 3) & (1 \times 2) + (2 \times 4) \\ (3 \times 1) + (4 \times 3) & (3 \times 2) + (4 \times 4) \end{bmatrix} = \begin{bmatrix} 7 & 10 \\ 15 & 22 \end{bmatrix}$$

$$A.^2 = \begin{bmatrix} 1^2 & 2^2 \\ 3^2 & 4^2 \end{bmatrix} = \begin{bmatrix} 1 & 4 \\ 9 & 16 \end{bmatrix}$$

$$A^2 \neq A.^2$$



```

GNU Octave ...
octave:1> A=[1 2;3 4]
--
1 2
3 4
octave:2> A^2
s =
7 10
15 22
octave:3> A.^2
s =
1 4
9 16
octave:4>

```

## Ejercicio 3 - conclusión

- $*$  y  $^$  son operadores aritméticos matriciales, los cálculos se efectúan según las normas de operaciones de matrices
- $.*$  y  $.^$  son operadores aritméticos escalares, los cálculos se efectúan entre elementos en posiciones correspondientes de las matrices
- Nota: lo mismo se verifica para el operador división

# Matrices y Vectores – indexar

- **V(k)**: k-ésimo elemento del vector V

$$D=[2 \ 5 \ 8 \ 11 \ 14]$$

$$D(3)=8$$

$$D(5)=14$$

- **A(k,l)**: elemento en la posición kl de la matriz A

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 7 \\ 4 & 9 & 2 \\ 6 & 0 & 5 \end{bmatrix}$$

$$A(1,3)=3$$

$$A(3,2)=9$$

# Matrices y Vectores – indexar

- **V(m:n)**: parte del vector que va desde el m-ésimo hasta el n-ésimo elemento

$$D=[2 \ 5 \ 8 \ 11 \ 14]$$

$$D(1:3)=[2 \ 5 \ 8]$$

$$D(3:5)=[8 \ 11 \ 14]$$

- **A(k,:)**: k-ésima fila de la matriz A
- **A(:,l)**: l-ésima columna de la matriz A

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 7 \\ 4 & 9 & 2 \\ 6 & 0 & 5 \end{bmatrix}$$

$$A(2,:)=[1 \ 2 \ 7]$$

$$A(4,:)=[6 \ 0 \ 5]$$



## Matrices y Vectores – extraer dimensión

- **length(V)**: número de elementos del vector V

$$D=[2 \ 5 \ 8 \ 11 \ 14] \quad \text{length}(D)=5$$

- **[filas,columnas]=size(A)**: número de filas y columnas de la matriz A

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 7 \\ 4 & 9 & 2 \\ 6 & 0 & 5 \end{bmatrix}$$

$$[\text{filas},\text{columnas}]=\text{size}(A)$$

$$\text{filas}=4$$

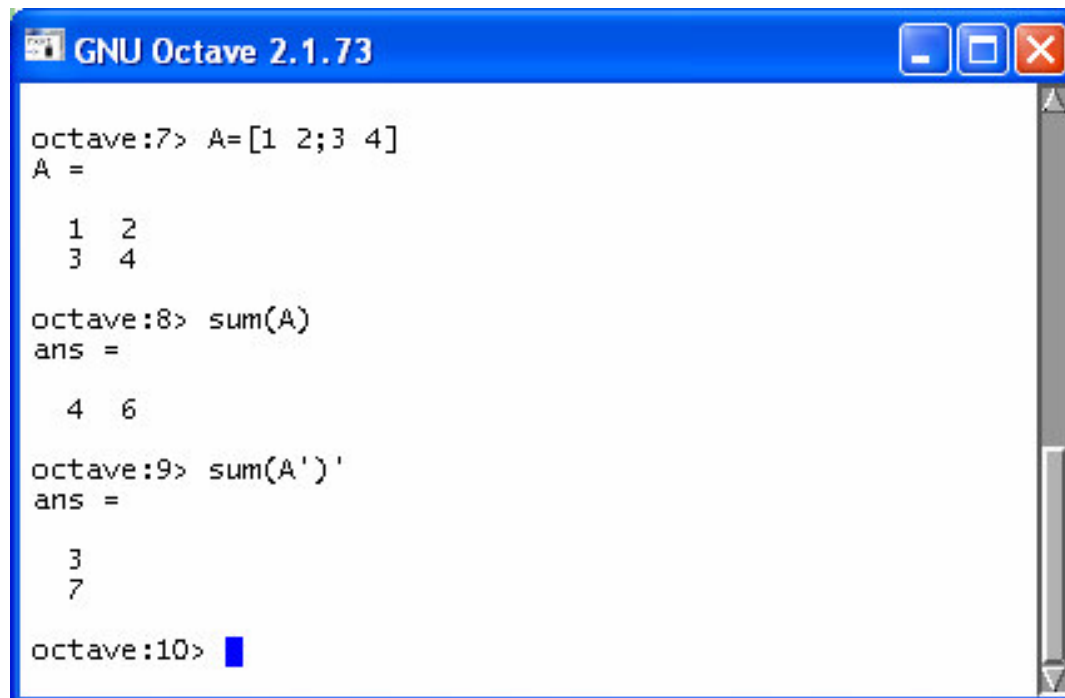
$$\text{columnas}=3$$

## Matrices y Vectores – redimensionar

- **reshape(A,m,n)**: transforma A en una matriz  $m \times n$
- **diag(A)**: genera un vector columna que contiene los elementos de la diagonal de la **matriz A**
- **diag(V)**: genera una matriz con los elementos del **vector V** en la diagonal principal

## Matrices - suma de filas y columnas

- Suma de los elementos por columna: **sum(A)**
- Suma de los elementos por fila: **sum(A')'**



```
GNU Octave 2.1.73
octave:7> A=[1 2;3 4]
A =
   1  2
   3  4
octave:8> sum(A)
ans =
   4  6
octave:9> sum(A')'
ans =
   3
   7
octave:10> █
```

## Ejercicio 4

- Dado el vector  $v=[3\ 1\ 5\ 7\ 9\ 2\ 6\ 4]$ , ejecuta e explica lo que hace cada línea:
  - `length(v)`
  - `v(3)`
  - `v(1:8)`
  - `v(6:-2:1)`
  - `v([1 6 2 1 3])`
  - `sum(v)`
  - `reshape(v,2,4)`
  - `diag(v)`

## Ejercicio 4 - solución

```
octave:19> v=[3 1 5 7 9 2 6 4]
v =
```

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

```
octave:20> length(v)
ans = 8
```

```
octave:21> v(3)
ans = 5
```

```
octave:22> v(1:8)
ans =
```

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

```
octave:23> v(6:-2:1)
ans =
```

```
2 7 1
```

```
octave:24> v([1 6 2 1 3])
ans =
```

```
3 2 1 3 5
```

```
octave:25> sum(v)
ans = 37
```

```
octave:26> reshape(v,2,4)
ans =
```

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

```
octave:27> diag(v)
ans =
```

```
3 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0
0 0 5 0 0 0 0 0
0 0 0 7 0 0 0 0
0 0 0 0 9 0 0 0
0 0 0 0 0 2 0 0
0 0 0 0 0 0 6 0
0 0 0 0 0 0 0 4
```

# Generación de matrices

- **eye(m,n)** :genera una matriz de dimensión  $m \times n$  con unos en la diagonal principal y ceros en las restantes posiciones
- **zeros(m,n)** :genera una matriz nula de dimensión  $m \times n$
- **ones(m,n)** :genera una matriz de dimensión  $m \times n$  con unos en todas las posiciones

## Ejercicio 5

- Utilizar los comandos apropiados para generar las siguientes matrices:

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad E = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

## Ejercicio 5 - solución

```
GNU Octave 2.7.7
octave:10> C=eye(3,3)
C =

    1    0    0
    0    1    0
    0    0    1

octave:11> D=zeros(2,2)
D =

    0    0
    0    0

octave:12> E=ones(2,2)
E =

    1    1
    1    1

octave:13> █
```



# Visualización gráfica - plots

## plot (x,y,'parámetros')

Parámetros (*help plot*):

- Estilo puntos: **+**, **\***, **o**, **x**
- Línea con puntos: **-+**, **-\***, **-o**, **-x**
- Color: **r**, **g**, **b** y **m** para *red*, *green*, *blue* y *magenta*

# Visualización gráfica-etiquetas



# Visualización gráfica - ejemplo

```
octave:6> x=1:10
```

```
X =
```

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

```
octave:7> y=x.^2
```

```
y =
```

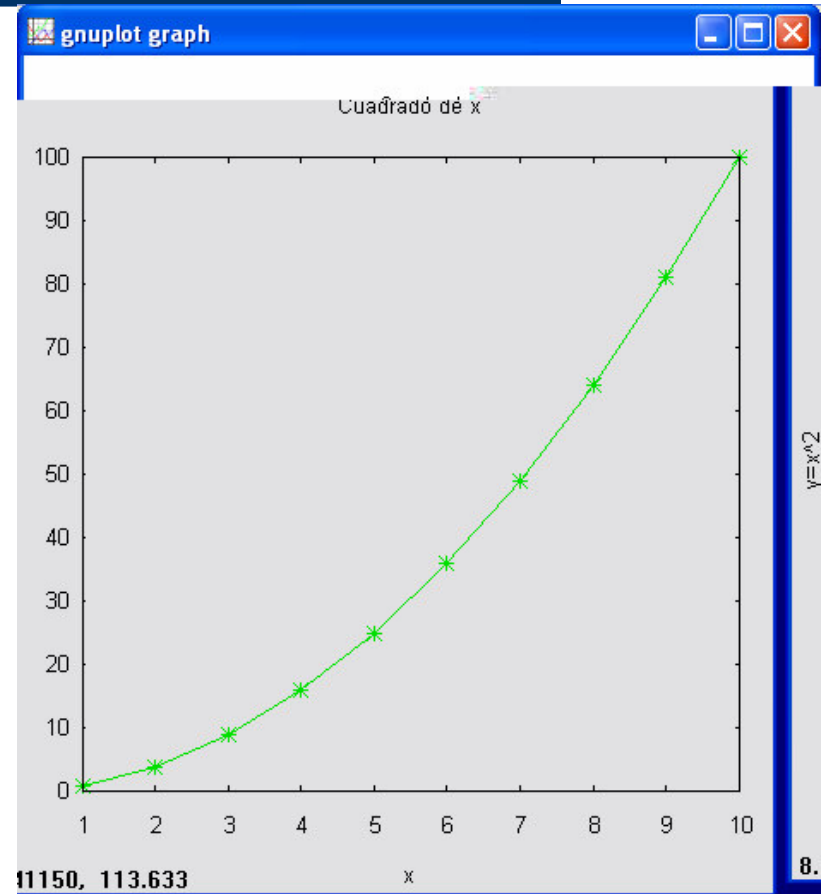
```
1 4 9 16 25 36 49 64 81 100
```

```
octave:8> plot(x,y,'*g')
```

```
octave:9> xlabel('x')
```

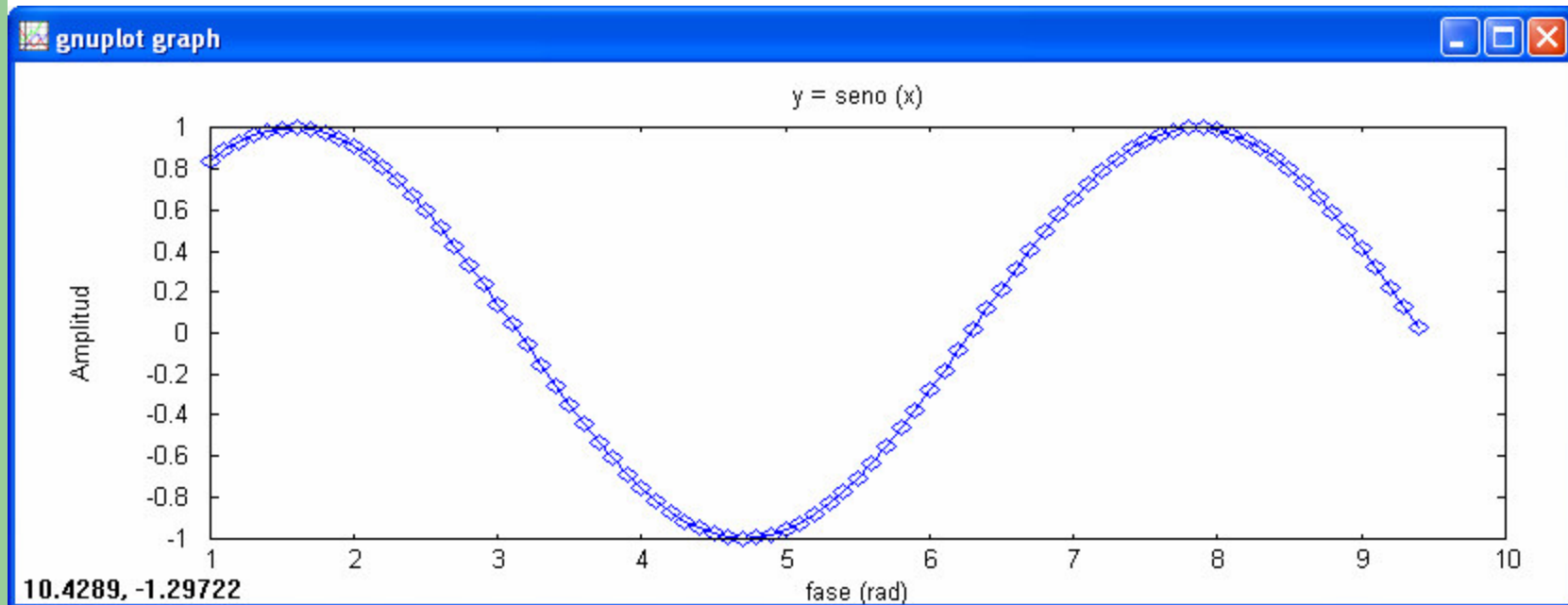
```
octave:10> ylabel('y=x^2')
```

```
octave:11> title('Cuadrado de x')
```



## Ejercicio 6

- Representar gráficamente  $y=\sin(x)$ , desde  $x = 0$  hasta  $3\pi$  con un incremento de 0.1.
- Etiquetar los ejes, dar título y definir el estilo y color de la línea



## Ejercicio 6 - solución

```
octave:13> x=0:0.1:3*pi;  
octave:14> y=sin(x);  
octave:15> plot(x,y,'b-o')  
octave:16> xlabel('fase (rad)')  
octave:17> ylabel('Amplitud')  
octave:18> title('y=seno(x)')
```

## Ejemplos de funciones estadísticas

- Media y desviación estandar:

Considerando un vector  $x$ :

Media:  $\text{mean}(x) = \bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$  **mean(x)**

Desviación estandar:  $\text{std}(x) = \sigma(x) = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N - 1}}$  **std(x)**

## Ejercicio 7

- Utilizar “help” para ver la descripción de las funciones **mean** y **std**
- Generar un vector que va de 1 a 100 con un incremento de 0.5
- Calcular la media y la desviación estandar

## Ejercicio 7 - solución

```
octave:1> x=1:0.5:100;  
octave:2> mean(x)  
ans = 50.500  
octave:3> std(x)  
ans = 28.795
```



# Script

- Un script es un sistema para automatizar la ejecución de comandos
- Es el equivalente a un programa
- Es un archivo con extensión .m que contiene todas las sentencias que escribiríamos línea por línea en el intérprete

# Script - ejemplo

a=465 y b=34

Calcular a+b, a-b, a\*b y a/b

Escribiendo línea por línea en el intérprete:

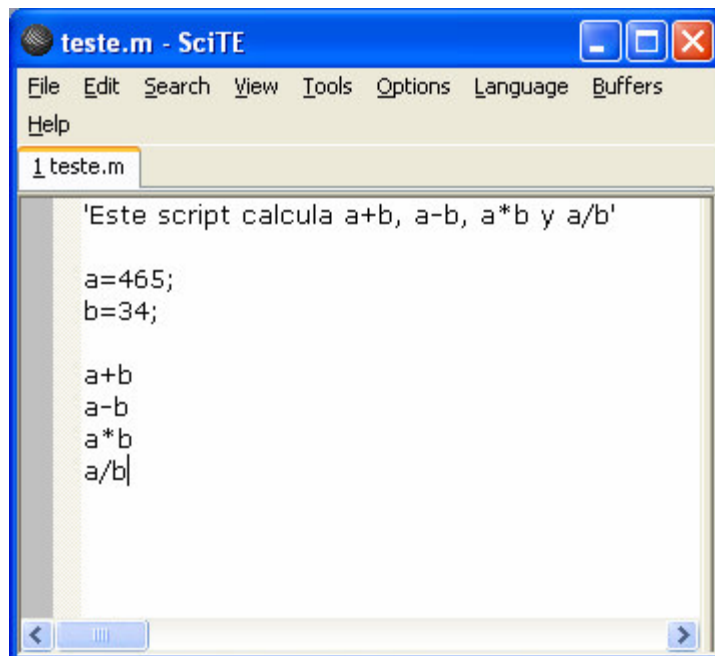


```
GNU Octave 2.1.73
octave:15> a=465;
octave:16> b=34;
octave:17> a+b
ans = 499
octave:18> a-b
ans = 431
octave:19> a*b
ans = 15810
octave:20> a/b
ans = 13.676
octave:21> █
```

# Script - ejemplo

Escribiendo un script llamado **teste.m**:

**Script**

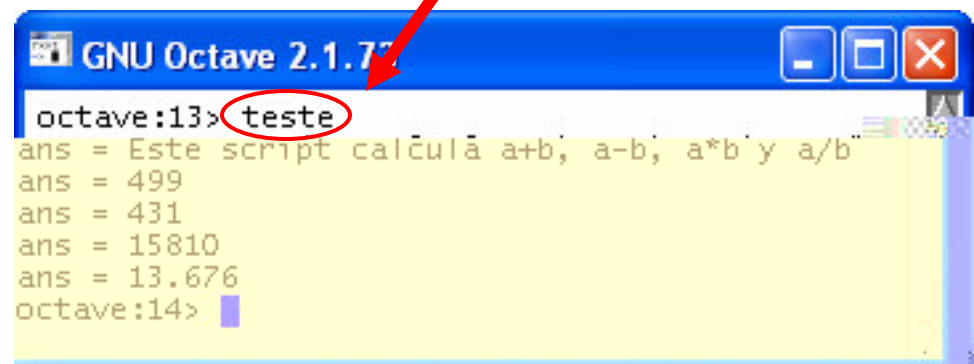


```
teste.m - SciTE
File Edit Search View Tools Options Language Buffers
Help
_1 teste.m
'Este script calcula a+b, a-b, a*b y a/b'

a=465;
b=34;

a+b
a-b
a*b
a/b
```

Ejecutamos el script escribiendo su nombre en el intérprete



```
GNU Octave 2.1.77
octave:13> teste
ans = Este script calcula a+b, a-b, a*b y a/b
ans = 499
ans = 431
ans = 15810
ans = 13.676
octave:14>
```

# Crear un script

- Ejecutar “edit”
- Escribir el script
- Guardar en c:/OCTAVEFILES/teste.m
- Ejecutar “cd c:”
- Ejecutar “cd OCTAVEFILES”
- Ejecutar “teste”

## Ejercicio 8

- Crear un script llamado **matrices.m** que haga la adición, substracción y multiplicación de las siguientes matrices:

$$B = \begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix} \quad E = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Utiliza los operadores aritméticos matriciales

# Ejercicio 8 - solución

```
matrices.m - SciTE
File Edit Search View Tools Options Language Buffers Help
1 matrices.m
%Suma, substracion y multiplicacion de dos matrices
B=[4 3;2 1];
E=[1 1;1 1];
B+E
B-E
B*E
```

```
GNU Octave 2.1.73
ans =
  1.75000  1.75000
  0.75000  0.75000
octave:1> matrices
ans =
  5  4
  3  2
ans =
  3  2
  1  0
ans =
  7  7
  3  3
octave:7>
```

# Condicional if

- Es una sentencia que nos permite implementar una condicion

```
a=465;  
b=34;
```

```
if a==b  
    'a es igual a b'  
else  
    'a es diferente de b'  
end
```

# Operadores lógicos

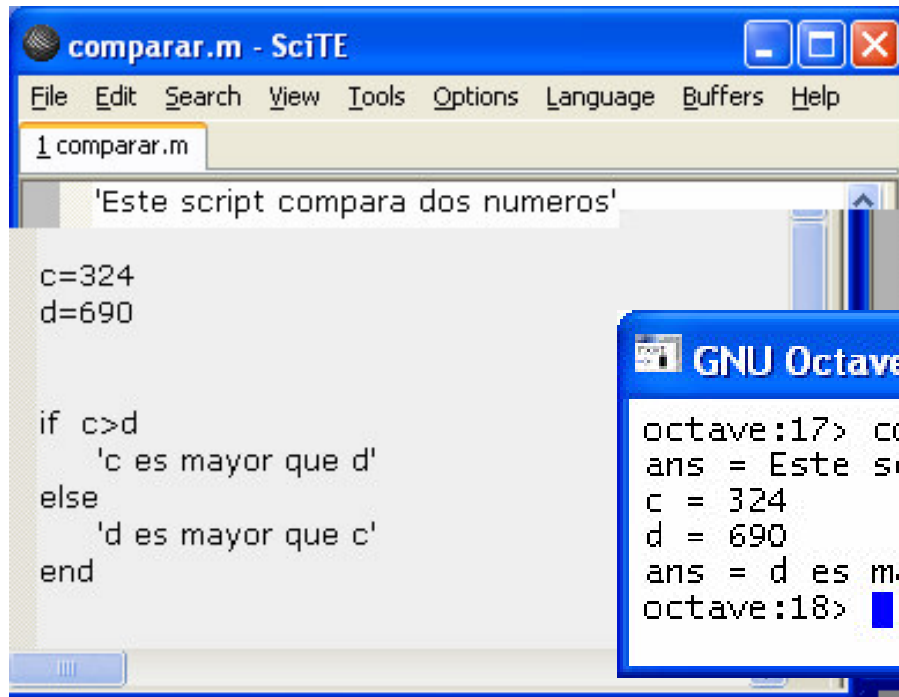
$\wedge$	menor
$\vee$	mayor
$==$	igual
$\leq$	menor o igual
$\geq$	mayor o igual
$\neq$	diferente
$\&$	and
$ $	or
$\sim$	not



## Ejercicio 9

- Crear un script llamado **compara.m** que haga la comparación entre dos números y diga cual es el mayor

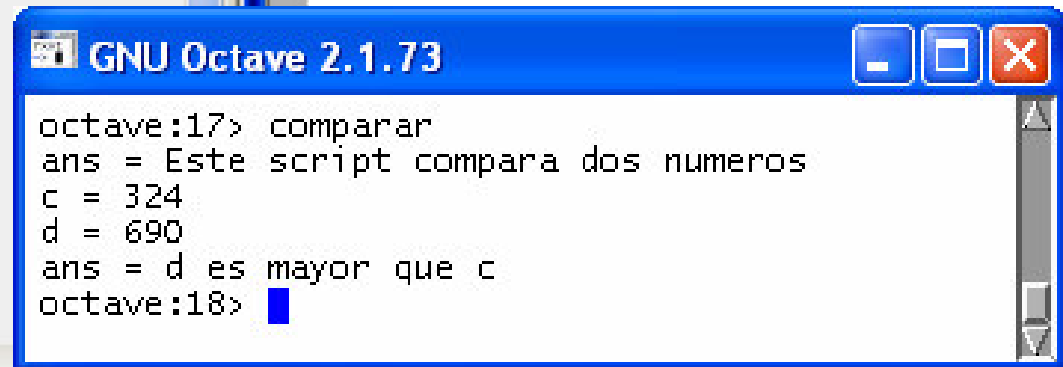
## Ejercicio 9 - solución



```
comparar.m - SciTE
File Edit Search View Tools Options Language Buffers Help
comparar.m
'Este script compara dos numeros'

c=324
d=690

if c>d
    'c es mayor que d'
else
    'd es mayor que c'
end
```



```
GNU Octave 2.1.73
octave:17> comparar
ans = Este script compara dos numeros
c = 324
d = 690
ans = d es mayor que c
octave:18> █
```

# Ciclo for

- Ejecuta las líneas de comando en el intervalo deseado:

```
for i=1:n  
    y=sqrt(i)  
end
```

## Ejercicio 10

- Utilizando el ciclo **for**, calcula el seno y el coseno entre 0 y  $2\pi$  con incrementos de 0.5

## Ejercicio 10 - solución

```
for i=0:0.5:2*pi
    k=sin(i)
    l=cos(i)
end
```

## Ejercicio 11

- Considera la siguiente onda:

$$y(t)=e^{-\alpha t} \sin(2\pi ft)$$

en donde  $f=10\text{Hz}$  y  $t \in [0,2]\text{s}$ .

- a) Representa gráficamente la función y etiqueta los ejes
- b) Representa la misma onda sin amortiguamiento

# Ejercicio 11 - solución

```
onda.m - SciTE
File Edit Search View Tools Options Language Buffers Help
1 onda.m

% Onda con y sin amortiguamiento

% Constantes:
alfa=2;      % amortiguamiento
f=10;       % frecuencia

t=0:0.005:2;|
ya=exp(-alfa*t).*sin(2*pi*f*t);      % Con amortiguamiento
yb=sin(2*pi*f*t);                    % Sin amortiguamiento

subplot (2,1,1);
plot(t,ya,'b-*')
xlabel('tiempo')
ylabel('amplitud')
title('Onda con amortiguamiento')

subplot (2,1,2);
plot(t,yb,'m-*')
xlabel('tiempo')
ylabel('amplitud')
title('Onda sin amortiguamiento')
```

