

# Formatos y Operadores

## Formatos numéricos

format \_\_\_\_\_

|          |  |
|----------|--|
| short    | coma fija con 4 decimales (defecto)                      |
| long     | coma fija con 15 decimales                               |
| hex      | cifras hexadecimales                                     |
| bank     | números con dos cifras decimales                         |
| short e  | notación científica con 4 decimales                      |
| short g  | notación científica o decimal, dependiendo del valor     |
| long e   | notación científica con 15 decimales                     |
| long g   | notación científica o decimal, dependiendo del valor     |
| rational | expresa los números racionales como cocientes de enteros |

El formato numérico se refiere siempre a la cantidad de cifras con las que MATLAB muestra los resultados, en todos los casos realiza los cálculos con doble precisión, es decir con unas 16 cifras decimales equivalentes.

## Operadores relacionales

|    |                   |
|----|-------------------|
| <  | menor que         |
| >  | mayor que         |
| <= | menor o igual que |
| >= | mayor o igual que |
| == | igual que         |
| ~= | distinto que      |

## Operadores lógicos

|          |  |
|----------|--|
| &        | and (función equivalente: and(A,B)). Se evalúan siempre ambos operandos, y el resultado es true sólo si ambos son true.      |
| &&       | and breve: si el primer operando es false ya no se evalúa el segundo, pues el resultado final ya no puede ser más que false. |
|          | or (función equivalente: or(A,B)). Se evalúan siempre ambos operandos, y el resultado es false sólo si ambos son false.      |
|          | or breve: si el primer operando es true ya no se evalúa el segundo, pues el resultado final no puede ser más que true.       |
| ~        | negación lógica (función equivalente: not(A))  |
| xor(A,B) | realiza un "or exclusivo", es decir, devuelve 0 en el caso en que ambos sean 1 ó ambos sean 0.                               |

El carácter (~) se puede obtener escribiendo el número 126 manteniendo Alt presionada.

## **Operadores aritméticos (para números, vectores o matrices)**

- + adición o suma
- sustracción o resta
- \* multiplicación
- ' traspuesta
- ^ potenciación
- \ división-izquierda
- / división-derecha
- .\* producto elemento a elemento de un vector o matriz
- ./ y .\ división elemento a elemento de un vector o matriz
- .^ elevar a una potencia elemento a elemento de un vector o matriz

# Funciones

## Funciones matemáticas

|          |   |
|----------|---|
| sin(x)   | Seno  |
| cos(x)   | Coseno  |
| tan(x)   | Tangente  |
| asin(x)  | Arco seno   |
| acos(x)  | Arco coseno   |
| atan(x)  | Arco tangente (devuelve un ángulo entre $-\pi/2$ y $+\pi/2$ )                                 |
| atan2(x) | Arco tangente (devuelve un ángulo entre $-\pi$ y $+\pi$ ); se le pasan 2 argumentos, ver help |
| sinh(x)  | Seno hiperbólico  |
| cosh(x)  | Coseno hiperbólico  |
| tanh(x)  | Tangente hiperbólica  |
| asinh(x) | Arco seno hiperbólico   |
| acosh(x) | Arco coseno hiperbólico   |
| atanh(x) | Arco tangente hiperbólica   |
| log(x)   | Logaritmo natural   |
| log10(x) | Logaritmo decimal   |
| exp(x)   | Función exponencial   |
| sqrt(x)  | Raíz cuadrada   |

## Funciones numéricas

|          |  |
|----------|--|
| sign(x)  | Devuelve -1 si $<0$ , 0 si $=0$ y 1 si $>0$ .                      |
| rem(x,y) | Resto de la división (2 argumentos que no tienen que ser enteros). |
| mod(x,y) | Similar a rem  |
| round(x) | Redondeo hacia el entero más próximo                               |
| fix(x)   | Redondea hacia el entero más próximo a 0                           |
| floor(x) | Valor entero más próximo hacia $-\infty$                           |
| ceil(x)  | Valor entero más próximo hacia $+\infty$                           |
| gcd(x)   | Máximo común divisor   |
| lcm(x)   | Mínimo común múltiplo  |
| real(x)  | Parte real   |
| imag(x)  | Parte imaginaria   |
| abs(x)   | Valor absoluto   |

## Funciones que actúan sobre vectores

|                            |   |
|----------------------------|---|
| <code>max(x)</code>        | Devuelve el valor máximo y la posición que ocupa  |
| <code>min(x)</code>        | Devuelve el valor mínimo y la posición que ocupa  |
| <code>sum(x)</code>        | Suma de los elementos de un vector  |
| <code>cumsum(x)</code>     | devuelve el vector suma acumulativa de los elementos de un vector (cada elemento del resultado es una suma de elementos del original)   |
| <code>mean(x)</code>       | valor medio de los elementos de un vector   |
| <code>std(x)</code>        | desviación típica   |
| <code>prod(x)</code>       | producto de los elementos de un vector  |
| <code>cumprod(x)</code>    | devuelve el vector producto acumulativo de los elementos de un vector   |
| <code>[y,i]=sort(x)</code> | ordenación de menor a mayor de los elementos de un vector x. Devuelve el vector ordenado y, y un vector i con las posiciones iniciales en x de los elementos en el vector ordenado y. |

## Funciones que actúan sobre vectores y matrices

|                            |  |
|----------------------------|--|
| <code>[m,n]=size(A)</code> | Devuelve el número de filas y de columnas de la matriz A. Si la matriz es cuadrada basta recoger el primer valor de retorno. |
| <code>n=length(x)</code>   | Calcula el número de elementos de un vector x.   |
| <code>A=diag(x)</code>     | Forma una matriz diagonal A cuyos elementos diagonales son los elementos de un vector ya existente x.                        |
| <code>x=diag(A)</code>     | Forma un vector x a partir de los elementos de la diagonal de una matriz ya existente A.                                     |
| <code>triu(A)</code>       | Forma una matriz triangular superior a partir de una matriz A.   |
| <code>tril(A)</code>       | Ídem con una matriz triangular inferior.   |

# Gráficos

## Funciones gráficas 2-D elementales

MATLAB dispone de algunas funciones básicas para crear gráficos 2-D. Estas funciones se diferencian principalmente por el tipo de escala que utilizan en los ejes de abscisas y de ordenadas.

|            |   |
|------------|---|
| plot()     | Crea un gráfico a partir de vectores y/o columnas de matrices, con escalas lineales sobre ambos ejes. |
| loglog()   | Ídem con escala logarítmica en ambos ejes.  |
| semilogx() | Ídem con escala lineal en el eje de ordenadas y logarítmica en el eje de abscisas.                    |
| semilogy() | Ídem con escala lineal en el eje de abscisas y logarítmica en el eje de ordenadas.                    |

### Función PLOT (ver comando: >> help plot)

La forma de utilizar la función plot() es con dos vectores como argumentos y una cadena de caracteres que permite modificar algunas características del trazo, el primero corresponde a los valores de abscisas mientras que el segundo corresponde a las ordenadas.

Sintaxis

```
plot(x,y,'.r-')
```

Ejemplos:

– graficar la función “coseno” desde 0 hasta  $2\pi$

```
>> x=0:pi/25:2*pi;
>> y=cos(x);
>> plot(x,y)
```

– graficar las funciones “seno” y “coseno” desde 0 hasta  $2\pi$

```
>> x=0:pi/25:2*pi;
>> y1=sin(x);
>> y2=cos(x);
>> plot(x,y1)
>> hold on
>> plot(x,y2)
```

Las siguientes tablas indican las distintas posibilidades para la cadena de caracteres que indican el color, el tipo de línea y el marcador.

| Símbolo | Color             |
|---------|-------------------|
| y       | amarillo (yellow) |
| m       | fucsia (magenta)  |
| c       | turquesa (cyan)   |
| r       | rojo (red)        |
| k       | negro (black)     |
| g       | verde (green)     |
| b       | azul (blue)       |
| w       | blanco (white)    |

| Símbolo | Marcadores (markers)               |
|---------|------------------------------------|
| .       | puntos                             |
| o       | círculos                           |
| x       | marcas en x                        |
| +       | marcas en +                        |
| *       | marcas en *                        |
| s       | marcas cuadradas (square)          |
| d       | marcas en diamante (diamond)       |
| ^       | triángulo apuntando arriba         |
| v       | triángulo apuntando abajo          |
| >       | triángulo apuntando a la derecha   |
| <       | triángulo apuntando a la izquierda |
| p       | estrella de 5 puntas               |
| h       | estrella se seis puntas            |

| Símbolo | Estilo de línea      |
|---------|----------------------|
| -       | líneas continuas     |
| :       | líneas a puntos      |
| -.      | líneas a barra-punto |
| --      | líneas a trazos      |

## Ejes

|   |  |
|---|--|
| <code>axis([xmin, xmax, ymin, ymax])</code> | Define los valores máximo y mínimo según cada eje  |
| <code>axis('auto x')</code>                 | utiliza el escalado automático sólo en dirección x   |
| <code>axis('auto xz')</code>                | utiliza el escalado automático sólo en direcciones x, z                                      |
| <code>axis(axis)</code>                     | mantiene los ejes en sus actuales valores si agrega nuevas gráficas con <code>hold on</code> |
| <code>axis('equal')</code>                  | el escalado es igual en ambos ejes   |
| <code>axis('square')</code>                 | la ventana será cuadrada   |
| <code>axis('normal')</code>                 | elimina las restricciones introducidas por 'equal' y 'square'                                |
| <code>axis('off')</code>                    | elimina las etiquetas, los números y los ejes  |
| <code>axis('on')</code>                     | restituye las etiquetas, los números y los ejes  |

## Funciones de gráfica bidimensionales

|                      |   |
|----------------------|---|
| <code>bar()</code>   | crea diagramas de barras  |
| <code>barh()</code>  | diagramas de barras horizontales                                    |
| <code>bar3()</code>  | diagramas de barras con aspecto 3-D                                 |
| <code>bar3h()</code> | diagramas de barras horizontales con aspecto 3-D                    |
| <code>pie()</code>   | gráficos con forma de "torta"                                       |
| <code>pie3()</code>  | gráficos con forma de "torta" y aspecto 3-D                         |
| <code>area()</code>  | similar <code>plot()</code> , pero rellenando en ordenadas de 0 a y |
| <code>hist()</code>  | dibuja histogramas de un vector                                     |

# Matrices y Vectores

## Matrices particulares

|            |  |
|------------|--|
| eye(4)     | forma la matriz unidad de tamaño (4×4)   |
| zeros(3,5) | forma una matriz de ceros de tamaño (3×5)  |
| zeros(4)   | forma una matriz de ceros de tamaño (4×4)  |
| ones(3)    | forma una matriz de unos de tamaño (3×3)   |
| ones(2,4)  | forma una matriz de unos de tamaño (2×4)   |
| rand(3)    | forma una matriz de números aleatorios entre 0 y 1, con distribución uniforme de tamaño (3×3)                          |
| rand(2,5)  | forma una matriz de números aleatorios entre 0 y 1, con distribución uniforme de tamaño (2×5)                          |
| randn(4)   | forma una matriz de números aleatorios de tamaño (4×4), con distribución normal, de valor medio 0 y varianza 1         |
| randn(2,5) | forma una matriz de números aleatorios de tamaño (2×5), con distribución normal, de valor medio 0 y varianza 1         |
| magic(4)   | crea una matriz (4×4) con los números 1, 2, ... 4*4, con la propiedad de que todas las filas y columnas suman lo mismo |
| hilb(5)    | crea una matriz de Hilbert de tamaño (5×5).  |
| invhilb(5) | crea directamente la inversa de la matriz de Hilbert   |

La matriz de Hilbert es una matriz cuyos elementos (i,j) responden a la expresión  $(1/(i+j-1))$ . Esta es una matriz especialmente difícil de manejar por los grandes errores numéricos a los que conduce.

## Vectores

|                   |  |
|-------------------|--|
| linspace(x1,x2,n) | genera un vector con n valores igualmente espaciados entre x1 y x2                     |
| logspace(d1,d2,n) | genera un vector con n valores espaciados logarítmicamente entre $10^{d1}$ y $10^{d2}$ |

# Ejemplos

**Considérese el siguiente sistema de ecuaciones lineales**

$$Ax = b$$

Resolución con MATLAB:

```
x = inv(A)*b
```

```
x = A\b
```

## Operador (:)

Se define un vector x con el siguiente comando:

```
>> x=1:10
      x=
      1  2  3  4  5  6  7  8  9 10
```

Con un paso determinado:

```
>> x=1:2:10
      x =
      1  3  5  7  9
```

```
>> x=10:-1:1
      x =
      10  9  8  7  6  5  4  3  2  1
```