

Módulo 1: Introducción a MATLAB

- Descripción de MATLAB.
- Historia de MATLAB.
- Documentación de MATLAB.
- Caracteres especiales.
- Cadenas de caracteres.
- Arreglo numérico.
- Operador punto.

Operadores aritméticos básicos

```
a = 2 + 2 % Suma
```

```
ans = 4
```

```
b = 6 - 9 % Resta
```

```
ans = -3
```

```
c = 5*3 % Multiplicación
```

```
ans = 15
```

```
d = 8/2 % División
```

```
ans = 4
```

```
e = 5^3 % Exponenciación
```

```
ans = 125
```

Definición de arreglos numéricos

$$A = [1 \ 6 \ 9 \ 4]$$

$$B = \begin{bmatrix} 3 \\ 5 \\ 7 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0.2 & 9 \\ 2 & 1 & 0.4 \\ -4 & 3 & 8 \end{bmatrix}$$

$$V = [1 \ 5 \ 2]$$

```
A = [1, 6, 9, 4] % Vector Fila (separar elementos con , )
```

```
A = 1x4
     1     6     9     4
```

```
B = [3; 5; 7] % Vector Columna (separar elementos con ; )
```

```
B = 3x1
     3
     5
     7
```

```
C = [1, 0.2, 9; 2, 1, 0.7; -4, 3, 8]
```

```
C = 3x3
     1.0000     0.2000     9.0000
     2.0000     1.0000     0.7000
    -4.0000     3.0000     8.0000
```

```
% Matriz (separa columnas por , y separa filas por ; )
```

```
V = [1, 5, 2]
```

```
V = 1x3
     1     5     2
```

Operaciones de álgebra lineal con arreglos

Verificar que las dimensiones de las matrices operadas sean consistentes:

$$D_{3 \times 1} = C_{3 \times 3} B_{3 \times 1} = \begin{bmatrix} 1 & 0.2 & 9 \\ 2 & 1 & 0.4 \\ -4 & 3 & 8 \end{bmatrix}_{3 \times 3} \begin{bmatrix} 3 \\ 5 \\ 7 \end{bmatrix}_{3 \times 1}$$

```
D = C*B
```

```
D = 3x1
    67.0000
    15.9000
    59.0000
```

$$E_{3 \times 4} = B_{3 \times 1} A_{1 \times 4} \begin{bmatrix} 3 \\ 5 \\ 7 \end{bmatrix} \begin{bmatrix} 1 & 6 & 9 & 4 \end{bmatrix}$$

$$E = B * A$$

$$E = 3 \times 4$$

$$\begin{bmatrix} 3 & 18 & 27 & 12 \\ 5 & 30 & 45 & 20 \\ 7 & 42 & 63 & 28 \end{bmatrix}$$

$$F_{1 \times 1} = V_{1 \times 3} B_{3 \times 1} = \begin{bmatrix} 1 & 5 & 2 \end{bmatrix} \begin{bmatrix} 3 \\ 5 \\ 7 \end{bmatrix}$$

$$F = V * B$$

$$F = 42$$

Operador Transpuesta

Vector fila se vuelve vector columna:

$$G = V^T = \begin{bmatrix} 1 \\ 5 \\ 2 \end{bmatrix}_{3 \times 1}$$

$$V$$

$$V = 1 \times 3$$

$$\begin{bmatrix} 1 & 5 & 2 \end{bmatrix}$$

$$G = V' \text{ \% Transpuesta con operador '}$$

$$G = 3 \times 1$$

$$\begin{bmatrix} 1 \\ 5 \\ 2 \end{bmatrix}$$

Operación que no tiene sentido:

$$H_{\epsilon?} = G_{3 \times 1} B_{3 \times 1} = \begin{bmatrix} 1 \\ 5 \\ 2 \end{bmatrix}_{3 \times 1} \begin{bmatrix} 3 \\ 5 \\ 7 \end{bmatrix}_{3 \times 1}$$

Las dimensiones no son consistentes!

```
H = G*B
```

```
Error using *
Incorrect dimensions for matrix multiplication. Check that the number of columns in the first
matrix matches the number of rows in the second matrix. To perform elementwise multiplication, use
'.*'.
```

Una operación que sí tiene sentido sería, por ejemplo, transponer G antes de multiplicar:

$$H_{1 \times 1} = G_{1 \times 3}^T B_{3 \times 1} = [1 \ 5 \ 2]_{1 \times 3} \begin{bmatrix} 3 \\ 5 \\ 7 \end{bmatrix}_{3 \times 1}$$

```
H = G' * B
```

```
H = 42
```

Operador Punto (.)

Realizar operaciones elemento a elemento. Por ejemplo:

$$J = G_{3 \times 1} \cdot * B_{3 \times 1} = \begin{bmatrix} 1 \\ 5 \\ 2 \end{bmatrix}_{3 \times 1} \cdot * \begin{bmatrix} 3 \\ 5 \\ 7 \end{bmatrix}_{3 \times 1} = \begin{bmatrix} 3 \\ 25 \\ 14 \end{bmatrix}$$

```
J = G .* B
```

```
J = 3x1
     3
    25
    14
```

Operación que no tiene sentido:

$$K = V^2 = V_{1 \times 3} V_{1 \times 3}$$

```
K = V^2
```

Error using ^
Incorrect dimensions for raising a matrix to a power. Check that the matrix is square and the power is a scalar. To perform elementwise matrix powers, use '.*'.

Pero se puede elevar al cuadrado cada elemento de V, usando el operador punto:

$$K = V.^2 = [(1)^2 \quad (5)^2 \quad (2)^2] = [1 \quad 25 \quad 4]$$

```
K = V.^2
```

```
K = 1x3
     1     25     4
```

División elemento a elemento:

$$L = G_{3 \times 1} ./ B_{3 \times 1} = \begin{bmatrix} 1 \\ 5 \\ 2 \end{bmatrix}_{3 \times 1} ./ \begin{bmatrix} 3 \\ 5 \\ 7 \end{bmatrix}_{3 \times 1} = \begin{bmatrix} 1/3 \\ 1 \\ 2/7 \end{bmatrix}$$

```
L = G ./ B
```

```
L = 3x1
     0.3333
     1.0000
     0.2857
```

Rangos

Inicialización de rangos de valores en un vector.

```
serie_1 = -10 : 2 : 10
```

```
serie_1 = 1x11
    -10     -8     -6     -4     -2      0      2      4      6      8     10
```

```
% Generar un arreglo con todos los números entre -10 y 10 saltando de a 2
```

```
serie_2 = linspace( -10, 10, 100 )
```

```
serie_2 = 1×100  
-10.0000    -9.7980    -9.5960    -9.3939    -9.1919    -8.9899    -8.7879    -8.5859 ...
```

```
% Generar un arreglo con 100 datos igualmente espaciados entre -10 y 10
```

Uso de palabras / strings / cadenas de caracteres

```
% Variables de tipo string se definen con ''  
pal_1 = 'hola'
```

```
pal_1 =  
'hola'
```

```
pal_2 = 'mundo'
```

```
pal_2 =  
'mundo'
```

```
pal_3 = ' '
```

```
pal_3 =  
' '
```

```
pal_4 = '123' % esto es un conjunto de caracteres, no es el número 123!
```

```
pal_4 =  
'123'
```

Concatenación de palabras

Unir varias palabras en una sola:

```
join_words = [pal_1, pal_3, pal_2, ' soy ', pal_4]
```

```
join_words =  
'hola mundo soy 123'
```

```
new_number = 410 % Esto es un número!
```

```
new_number = 410
```

No se pueden concatenar palabras con números!, el resultado no es lo esperado:

```
join_words = [pal_1, pal_3, pal_2, ' soy ', new_number]
```

```
join_words =  
'hola mundo soy t'
```

Se debe convertir el número a palabra antes:

```
pal_5 = num2str( new_number ) % esta es la palabra '410'
```

```
pal_5 =  
'410'
```

Ahora sí se pueden concatenar:

```
join_words = [pal_1, pal_3, pal_2, ' soy ', pal_5]
```

```
join_words =  
'hola mundo soy 410'
```