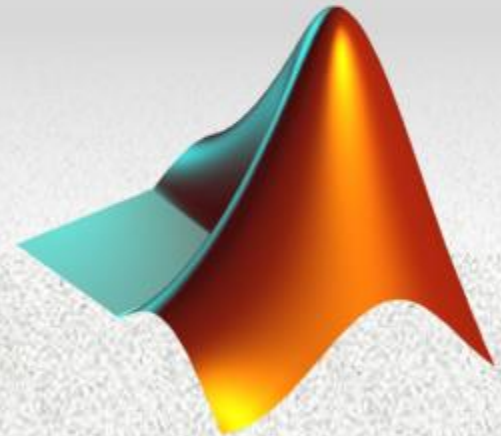




**UNIVERSIDAD
NACIONAL DE
INGENIERÍA**

MATLAB

R2017a



Maria Pimentel Herrera
uni.kernel@gmail.com

Numeración de los elementos de una matriz

MATLAB

	1	2	3	4
1	1	2	3	4
2	5	6	7	8
3	9	10	11	12

C/C++

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12

Disposición de los elementos de una matriz en la memoria

Column-major order

MATLAB

1	5	9	2	6	10	3	7	11	4	8	12
---	---	---	---	---	----	---	---	----	---	---	----

Row-major order

Lenguaje C/C++

1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	----	----	----

OPERADORES ARITMÉTICOS

Operador	Función	Definición
+	plus	Suma
-	minus	Resta
*	mtimes	Multiplicación
'	transpose	Traspuesta
^	mpower	Potenciación
\	mldivide	División-izquierda
/	mrdivide	División-derecha

M-File: s031.m

```
1 - clear all; %eliminando todas las variables definidas
2 - home; %ubica cursor en la parte superior de la pantalla
3   %Definiendo y guardando variable
4 - disp('Definiendo MATRICES')
5 - A=[1 2 3; 4 5 6; 7 8 9]
6 - B=[2 -4 6; 8 -1 0; 7 4 7]
7 - save var3|
```

```
clear all; %eliminando todas las  
variables definidas  
  
home; %ubica cursor en la parte  
superior de la pantalla  
  
%Definiendo y guardando variable  
disp('Definiendo MATRICES')  
  
A=[1 2 3; 4 5 6; 7 8 9]  
B=[2 -4 6; 8 -1 0; 7 4 7]  
  
save var3
```


M-File: s032.m

```
1      %OPERADORES ARITMETICOS
2 -    clear;
3 -    load var3;
4 -    clc;
5 -    disp('OPERACIONES CON MATRICES')
6 -    disp('suma con un escalar')
7 -    C=A+2, D=plus(A,2)
8 -    disp('suma de matrices')
9 -    C=A+B, D=plus(A,B)
10 -    disp('diferencia con un escalar')
11 -    C=A-1, D=minus(A,1)
12 -    disp('diferencia de matrices')
13 -    C=A-B, D=minus(A,B)
```

%OPERADORES ARITMETICOS

```
clear;  
load var3;  
clc;  
disp('operaciones con matrices')  
disp('suma con un escalar')  
C=A+2, D=plus(A,2)  
disp('suma de matrices')  
C=A+B, D=plus(A,B)  
disp('diferencia con un escalar')  
C=A-1, D=minus(A,1)  
disp('diferencia de matrices')  
C=A-B, D=minus(A,B)
```

M-File: s033.m

```
1      %OPERADORES ARITMETICOS 2
2 -    disp('multiplicación de matrices')
3 -    C=A*B, D=mtimes(A,B)
4 -    disp('transpuesta conjugada de matriz')
5 -    I=[2 3+i 5-2i; 4 i 0; 1+i 3 7], C=(I)', D=ctranspose(I)
6 -    disp('transpuesta no conjugada de matriz')
7 -    C=(I).', D=transpose(I)
8 -    disp('potenciación de matriz')
9 -    C=A^2 %C=A*A
10 -   D=mpower(A,2)
```


%OPERADORES ARITMETICOS 2

```
disp('multiplicación de matrices')
```

```
C=A*B, D=mtimes(A,B)
```

```
disp('transpuesta conjugada de  
matriz')
```

```
I=[2 3+i 5-2i; 4 i 0; 1+i 3 7],
```

```
C=(I)', D=ctranspose(I)
```

```
disp('transpuesta no conjugada de  
matriz')
```

```
C=(I).', D=transpose(I)
```

```
disp('potenciación de matriz')
```

```
C=A^2 %C=A*A
```

```
D=mpower(A,2)
```

M-File: s034.m

```
1      %OPERADORES ARITMETICOS 3
2      %división derecha
3 -    format rat;
4 -    A, C=A/3, D=rdivide(A,3)
5      %división elemento a elemento
6 -    A, B, C=A./B, D=rdivide(A,B)
7      %división izquierda
8 -    C=B.\A, D=ldivide(B,A)
9 -    format|
```

%OPERADORES ARITMETICOS 3

%división derecha

format rat;

A, C=A/3, D=rdivide(A,3)

%división elemento a elemento

A, B, C=A./B, D=rdivide(A,B)

%división izquierda

C=B.\A, D=ldivide(B,A)

format

M-File: s035.m

```
1      %OPERADORES ELEMENTO A ELEMENTO
2 -    A=[1 2 3 4 5], B=[1 -1 2 -1 1]
3 -    disp('multiplicación elemento a elemento')
4 -    C=A.*B, D=times(A,B)
5 -    disp('potenciación elemento a elemento')
6 -    C=A.^B, D=power(A,B)|
```

%OPERADORES ELEMENTO A ELEMENTO

A=[1 2 3 4 5], B=[1 -1 2 -1 1]

disp('multiplicación elemento a
elemento')

C=A.*B, D=times(A,B)

disp('potenciación elemento a
elemento')

C=A.^B, D=power(A,B)

Sistema de ecuaciones lineales

Operadores para resolver sistema de ecuaciones lineales

Operator	Linear Equation	Syntax	Equivalent Syntax Using \
Backslash (\)	$A * x = B$	$x = A \setminus B$	Not applicable
Slash (/)	$x * A = B$	$x = B / A$	$x = (A' \setminus B')'$

Solución de sistemas de ecuaciones lineales $Ax=b$

donde:

x y b son vectores columna (b conocido)

A es una matriz cuadrada invertible

$$A(n,n) * x(n,1) = b(n,1)$$

Solución

$$\text{inv}(A(n,n)) * A(n,n) * x(n,1) = \text{inv}(A(n,n)) * b(n,1)$$

$$x(n,1) = \text{inv}(A(n,n)) * b(n,1)$$

M-File: s036.m

```
A=[2 -1 1; 1 1 0; 0 1 -3]
b=[3; 3; -7]
Ainv=inv(A), M=A*Ainv
disp('1ra. forma')
x=inv(A)*b
disp('2da. forma')
x=A\b
% Operador mas inteligente y general:
% A no es necesario ser invertible o
% cuadrada
```

Solución de sistemas de ecuaciones lineales $\mathbf{x}\mathbf{A}=\mathbf{b}$

donde:

\mathbf{x} y \mathbf{b} son vectores fila (\mathbf{b} conocido)

\mathbf{A} es una matriz cuadrada invertible

$$\mathbf{x}(1,n) * \mathbf{A}(n,n) = \mathbf{b}(1,n)$$

Solución

$$\mathbf{x}(1,n) * \mathbf{A}(n,n) * \text{inv}(\mathbf{A}(n,n)) = \mathbf{b}(1,n) * \text{inv}(\mathbf{A}(n,n))$$

$$\mathbf{x}(1,n) = \mathbf{b}(1,n) * \text{inv}(\mathbf{A}(n,n))$$

M-File: s037.m

```
A=[4 -2 6; 2 8 2; 6 10 3], b=[8 4 0]
disp('1ra. forma')
x=b*inv(A)
disp('2da. forma')
x=b/A
format;
```


Definición de variables

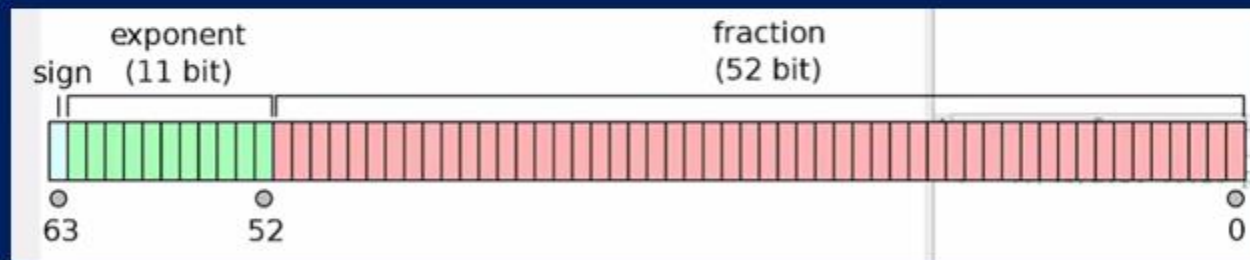
Double

- Matlab es un programa preparado para trabajar con vectores y matrices como caso particular también trabaja con variables escalares - matrices de dimensión 1
- Siempre trabaja con doble precisión, es decir guardando en 8 bytes = 64 bits
- Double = números reales de doble precisión

Double ➡ 8 bytes ➡ 64 bits

Double \Rightarrow 8 bytes \Rightarrow 64 bits

- **Double** = números reales de doble precisión
- 1er bit es para el signo 0=+, 1=-
- siguientes 11 bits son para la potenciación
- los restantes 52 bits es para la fracción



Error de calculo

1/0	%Inf o inf = infinito
0/0	%NaN no es un numero
inf/inf	%NaN
NaN+5	%NaN
Inf+5	%Inf

Operaciones útiles de coma flotante

- **eps** = devuelve la diferencia entre 1 y el número de coma flotante inmediatamente superior, es decir representa el número más pequeño que puede sumar a 1.

$$(1.0 + \text{eps}) > 1.0$$

- representa la exactitud relativa de la aritmética del computador da una idea de la precisión o del número de cifras almacenadas


```
eps    % 2.2204e-016
```

```
%devuelve el número más pequeño con  
el que se puede trabajar
```

```
realmin % 2.2251e-308
```

```
%devuelve el número más grande con  
el que se puede trabajar
```

```
realmax % 1.7977e+308
```

Integer, float y logical

```
% ahorrar memoria 8, 16, 32 y 64  
bits con signo o sin signo
```

```
a=int32(5) % entero de 32 bit
```

```
A=[4 5 6; 7 8 9] %Double
```

```
B=int32(A) % entero de 32 bit ocupa  
la mitad
```

Número de bytes	Número de bits	Tipo de entero (Con signo)	Tipo de entero (Sin signo)
1	8	int8	uint8
2	16	int16	uint16
4	32	int32	uint32
8	64	int64	uint64

```
% valor mas pequeño que puede formarse  
intmin('int8')      %-128  
intmin('int16')     %-32768  
intmin('int32')     %-2147483648  
intmin('int64')     %-9223372036854775808
```

```
% valor mas grande que puede formarse  
intmax('int8')      %127  
intmax('int16')     %32767  
intmax('int32')     %2147483647  
intmax('int64')     %9223372036854775807
```