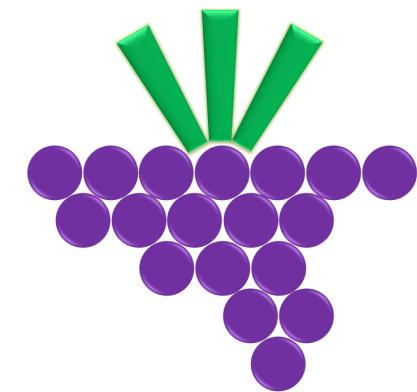




# para ciencias aplicadas e ingeniería



## Unidad 2-A



San Rafael, Argentina, Mayo-Junio de 2021



**UNCUYO**  
UNIVERSIDAD  
NACIONAL DE CUYO



FACULTAD DE  
**CIENCIAS APLICADAS**  
A LA INDUSTRIA

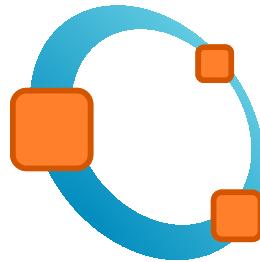


**MoCCAi**  
MODELADO COMPUTACIONAL EN CIENCIAS APLICADAS E INGENIERÍA



# Introducción a Octave

## para ciencias aplicadas e ingeniería



**Daniel Millán**



**Nicolas Muzi**



Eduardo Rodríguez



Nora Moyano



Rodrigo Bautista



Brian Villegas



Brando Martinelli

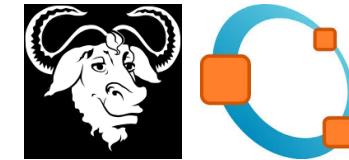


Heber Duran

San Rafael, Argentina, Mayo-Junio de 2021



# Unidad 2-A



## A. Operaciones con vectores/ matrices

1. Vectores/matrices como arreglos de números.
2. Operaciones con vectores/matrices.
3. Tipos de matrices predefinidos.
4. Operador (:). Matriz vacía [ ]. Borrado filas/columnas.
5. Operadores relacionales. Operadores lógicos.

## B. Trazado de gráficos

6. Función *plot()*.
  - Estilos de línea y marcadores.
  - Añadir curvas a un gráfico ya existente.
  - Control de los ejes: *axis()*.
7. Control de ventanas gráficas: *figure()*.
8. Otras funciones gráficas 2D.
9. Resumen funciones para gráficas 3D.





# 1. Vectores/matrices

- Los vectores se pueden definir directamente introduciendo los elementos que lo componen, separados por comas o espacios, por ejemplo:

```
>> u=[1,2,3] % vector fila
```

- Observamos que **u** es un vector fila.
- En el caso de desear un vector columna utilizamos ;
- Para transformar un vector fila en columna se debe transponer dicho vector mediante la orden '**'** (comilla).

**Ejemplo:** ¿Matemáticamente es posible calcular la suma de **u** y **v**?

¿Qué sucede cuando realizo esta suma en Octave?

- Octave genera por defecto vectores fila:

```
>> w(1)=1; w(2)=4; w(3)=9;  
>> w
```





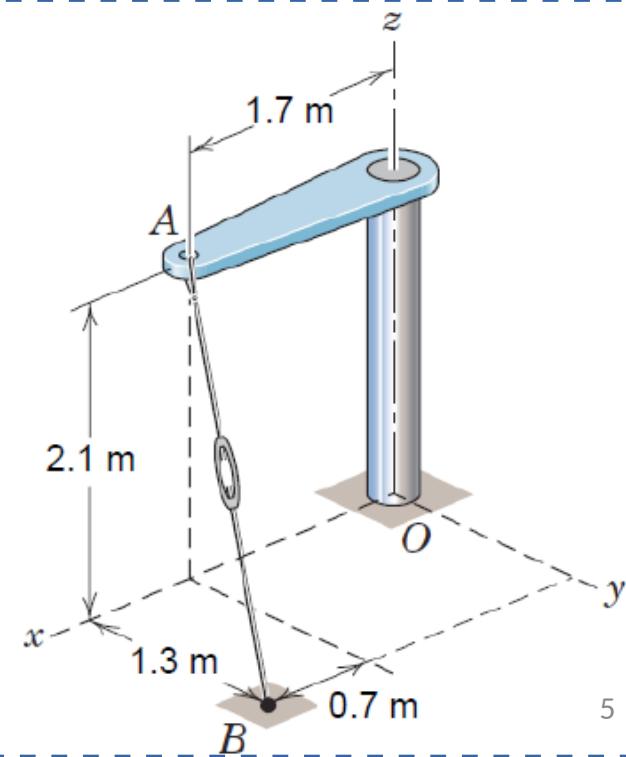
# 1. Vectores/matrices

- Producto escalar, vectorial y tensorial

```
>> u=[1,2,3]; v=[1,1,1];
>> dot(u,v) %producto escalar o punto (u.v)
>> cross(u,v) %producto vectorial o cruz (u×v)
>> u'*v %producto tensorial o abierto (u⊗v)
```

**Ejemplo:** El tensor de la figura, se ajusta hasta que la tensión del cable AB es de 2.5kN.

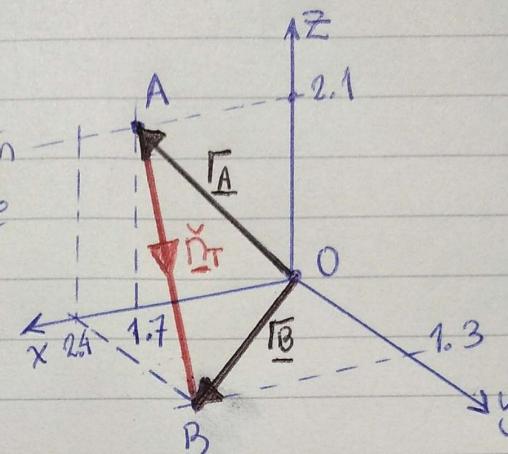
Determinar, con **Octave**, el momento  $\mathbf{M} = \mathbf{r} \times \mathbf{T}$  respecto al punto O de la tensión del cable, que actúa en el punto A, y la magnitud de ese momento [¿Unidades?].



# 1. Vectores/matrices

Brazo tensionado

$$T = 2.5 \text{ [kN]} \quad \text{tensión en el cable}$$



$$\underline{r}_A = (1.7, 0, 2.1) \text{ [m]}$$

$$\underline{r}_B = (2.4, 1.3, 0) \text{ [m]}$$

vector posición pto A

vector posición pto B

$$\underline{M}_0 = \underline{r}_A \times \underline{T} \quad [\text{kN}] \quad \text{momento en } O$$

$$\underline{T} = T \underline{\hat{n}}_T \quad [\text{kN}] \quad \text{vector fuerza}$$

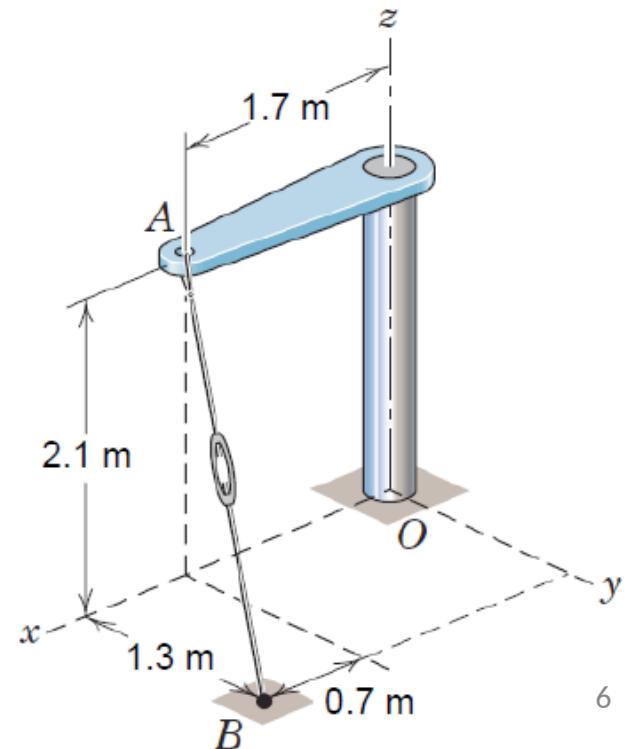
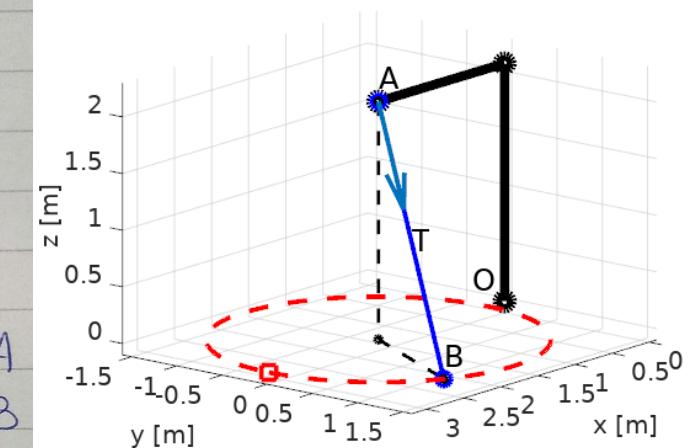
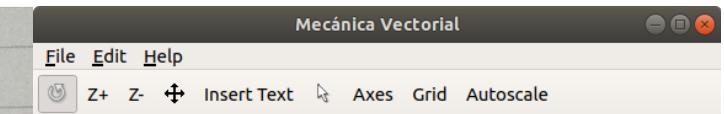
$$\underline{n}_T = \frac{\underline{r}_B - \underline{r}_A}{\|\underline{r}_B - \underline{r}_A\|}$$

versor : vector unitario  
en la dirección de la fuerza  
aplicada (adimensional)

Octave

$$\left\{ \begin{array}{l} \underline{M}_0 = T * \text{cross}(\underline{r}_A, \underline{n}_T) \\ m\underline{M}_0 = \text{norm}(\underline{M}_0), \\ \|\underline{M}_0\| = \sqrt{\underline{M}_0 \cdot \underline{M}_0} \end{array} \right.$$

módulo o magnitud o norma en  $L_2$





# 1. Vectores/matrices

- Las matrices se crean introduciendo sus elementos de la siguiente manera

$$>> A=[1 \ 4 \ -3; \ 2 \ 1 \ 5; \ -2 \ 5 \ 3] \rightarrow A = \begin{bmatrix} 1 & 4 & -3 \\ 2 & 1 & 5 \\ -2 & 5 & 3 \end{bmatrix}$$

- Adicionalmente es posible crear matrices mediante vectores filas/columnas

```
>> f1=[1 4 -3]; f2=[2 1 5]; f3=[-2 5 3];
>> Af=[f1;f2;f3]
```

```
>> c1=[1;2;-2]; c2=[4;1;5]; c3=[-3;5;3];
>> Ac=[c1,c2,c3]
```

**Ejemplo:** Calcular

- la transpuesta de  $A := A^T$ ,
- comprobar la inversa:  $A^{-1} = \text{inv}(A)$ .





## 2. Operaciones con vectores/matrices

- Operadores aritméticos:
  - + adición o suma
  - sustracción o resta
  - \* multiplicación
  - ' traspuesta
  - ^ potenciación
  - \ división-izquierda
  - / división-derecha

**Ejemplo:** comprobar las operadores anteriores empleando las matrices

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad A^{-1} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \quad B^{-1} = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}$$





## 2. Operaciones con vectores/matrices

- Resolución de sistemas lineales

$$\mathbf{A} \mathbf{x} = \mathbf{b}$$

donde  $\mathbf{A}$  es una matriz invertible de  $N \times N$  (filas x columnas),  $\mathbf{x}$  y  $\mathbf{b}$  son vectores columna de  $N \times 1$  no nulos.

- La resolución de este sistema de ecuaciones se puede realizar de 2 formas diferentes

$$\mathbf{x} = \text{inv}(\mathbf{A})^* \mathbf{b}$$

$$\mathbf{x} = \mathbf{A} \backslash \mathbf{b}$$

**Ejemplo:** Dada la matriz  $\mathbf{A} = [1 \ 4 \ -3; \ 2 \ 1 \ 5; \ 3.1 \ 5.1 \ 3.1]$  y el vector  $\mathbf{b} = [1; 2; 3]$ , obtenga el vector solución  $\mathbf{x}$  mediante **2 formas diferentes** y compruebe los resultados obtenidos mediante el cálculo de la norma del vector resto.

$$\text{resto} = \|\mathbf{A} \mathbf{x}_{\text{sol}} - \mathbf{b}\|$$





## 2. Operaciones con vectores/matrices

- Operadores aritméticos elemento a elemento (\*, ^, \ y /). Para ello basta precederlos por un punto (.)

**Ejercicio:** funcionamiento de los operadores aritméticos

```
>> [1 2 3 4].^2  
>> [1 2 3 4].*[1 -1 1 -1]
```

```
>> [1 -1;1 -1].^3  
>> [1 2;3 4]*[1 -1;1 -1]  
>> [1 2;3 4].*[1 -1;1 -1]
```

```
>> [1 2;3 4]./[1 -1;1 -1]  
>> [1 2;3 4].\*[1 -1;1 -1]
```





### 3. Tipos de matrices predefinidos

- **eye(4)** forma la matriz unidad de tamaño (4×4)
- **zeros(3,5)** forma una matriz de ceros de tamaño (3×5)
- **zeros(4)** ídem de tamaño (4×4)
- **ones(2,3)** forma una matriz de unos de tamaño (2×3)
- **linspace(0,1,7)** genera un vector con 7 valores igualmente espaciados entre 0 y 1
- **rand(3)** crea una matriz de números aleatorios entre 0 y 1, con distribución uniforme, de tamaño (3×3)
- **randn(4)** matriz de números aleatorios de 4×4, con distribución normal, de valor medio 0 y varianza 1.
- **magic(4)** crea una matriz (4×4) con los números 1, 2, ... 4\*4, tal que todas las filas y columnas suman lo mismo.
- **hilb(5)** matriz de Hilbert de 5×5. El elemento (i,j) está dado por (1/(i+j-1)). Esta matriz produce grandes errores numéricos al resolver sistemas lineales  $\mathbf{A} \mathbf{x} = \mathbf{b}$ .





# 3. Tipos de matrices predefinidos

- Formación de un matriz a partir de Otras ya definidas
  - recibiendo alguna de sus propiedades (por ejemplo el tamaño),
  - por composición de varias submatrices más pequeñas,
  - modificándola de alguna forma.

**Ejemplo:** Dada la matriz **A=floor(5\*rand(3))** y el vector **x=[1;2;3]** comprobamos las siguientes sentencias

- **[m,n]=size(A)** devuelve el número de filas y de columnas de la matriz A.  
**n=length(x)** calcula el número de elementos de un vector x.
  - **zeros(size(A))** forma una matriz de **ceros** del mismo tamaño que A  
**ones(size(A))** ídem pero crea una matriz de **unos**
  - **B=diag(x)** forma una matriz diagonal **B** cuyos elementos diagonales son los elementos del vector x.
  - **x=diag(B)** forma un vector x a partir de los elementos de la diagonal de B.
  - **diag(diag(A))** crea una matriz diagonal a partir de la diagonal de la matriz.
- ```
>> B=diag(diag(A))  
>> C=[A, B; zeros(2,3), ones(2,3)]
```





# 4. Operador (:). Matriz vacía [ ].

## Borrado filas/columnas.

- El operador ":" es muy importante y puede usarse de varias formas.

**¡Probar y jugar, es la mejor forma de aprender!**

**Ejercicio:** comprobar el funcionamiento de las expresiones siguientes

Arreglo de números o vector

```
>> x=1:10
>> x=1:2:10
>> x=1:1.5:10
>> x=10:-1:1

>> x=[0.0:pi/50:2*pi]';
>> y=sin(x); z=cos(x);
>> [x y z]
```

Matrices

```
>> A=magic(5)
>> A(2,3)
>> A(5,1:4)
>> A(3,:)
>> A(end,:)
>> A(3:5,:)
>> A([1 2 5],:)
>> A(:) % vector columna
>> B=eye(size(A));
>> B([2 4 5],:)=A(1:3,:)
```





# 4. Operador (:). Matriz vacía [ ].

## Borrado filas/columnas.

- Una matriz definida sin ningún elemento entre los corchetes es una matriz que existe, pero que está vacía, o lo que es lo mismo que tiene *dimensión cero*.
- Las funciones ***exist()*** e ***isempty()*** permiten chequear si una variable existe y si está vacía.

**Ejemplo:** verificamos el funcionamiento de las órdenes anteriores

```
>> A=magic(3)
>> B=[]
>> exist("B"), exist("C")
>> isempty(A), isempty(B)
>> A(:,3)=[ ]
```

- En el último ejemplo se ha eliminado la 3<sup>a</sup> columna de A asignándole la matriz vacía.



# 5. Operadores relacionales.

## Operadores lógicos.

- Operadores relacionales, se aplican a vectores y matrices:
  - < menor que
  - > mayor que
  - <= menor o igual que
  - >= mayor o igual que
  - == igual que
  - $\neq$  distinto que ( $\neq$  ambas órdenes son idénticas)
- La comparación se realiza elemento a elemento, y el resultado es otra matriz de unos y ceros del mismo tamaño.

**Ejercicio:** comprobar el funcionamiento de los operadores empleando

```
>> A=[1 2;0 3]; B=[4 2;1 5];
>> A==B
>> A~=B
```





# 5. Operadores relacionales.

## Operadores lógicos.

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

**Ejemplo:** comprobar el funcionamiento de los operadores empleando

**>> A=[1 1;0 1]; B=[1 1;1 1]; % <== valores lógicos**

**>> A&B**

**>> A&&B**



**FIN.**

San Rafael, Argentina 2021