

## TP Automatique des Systèmes Linéaires

---

# Introduction à Matlab

---

HESTIM CASABLANCA  
DÉPARTEMENT GÉNIE INDUSTRIEL

Animé par :  
Korota Arsène COULIBALY



**Nom :** COULIBALY

**Prénoms :** Korota Arsène

**DEUST** en Génie Electrique Génie Mécanique(FST de Tanger)

**Formation en Contrôle des systèmes Industriels**(ENSET Mohammedia).

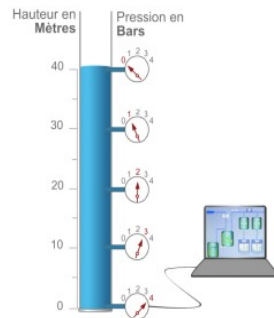
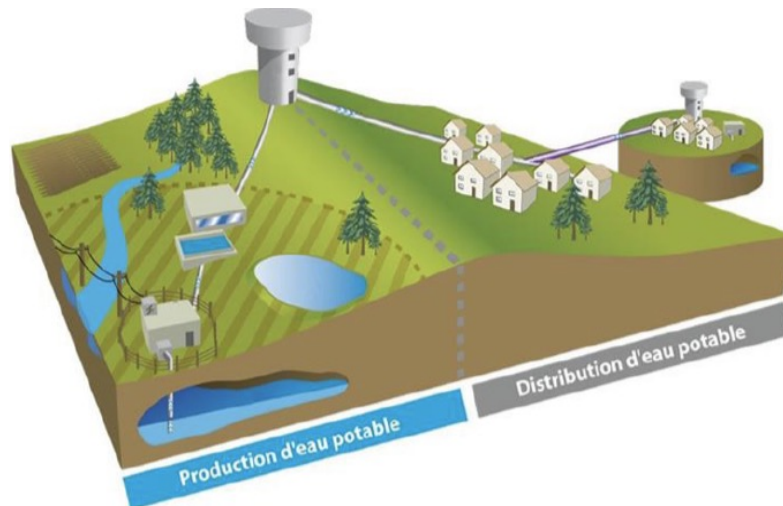
**Ingénieur** en Systèmes Embarqués et Informatique Industrielle(ENSA Fès)

**Thèse de Cotutelle en Industrie 4.0** (ENSAM de Casablanca et l'Université de Bordeaux)

# Notion de systèmes asservis

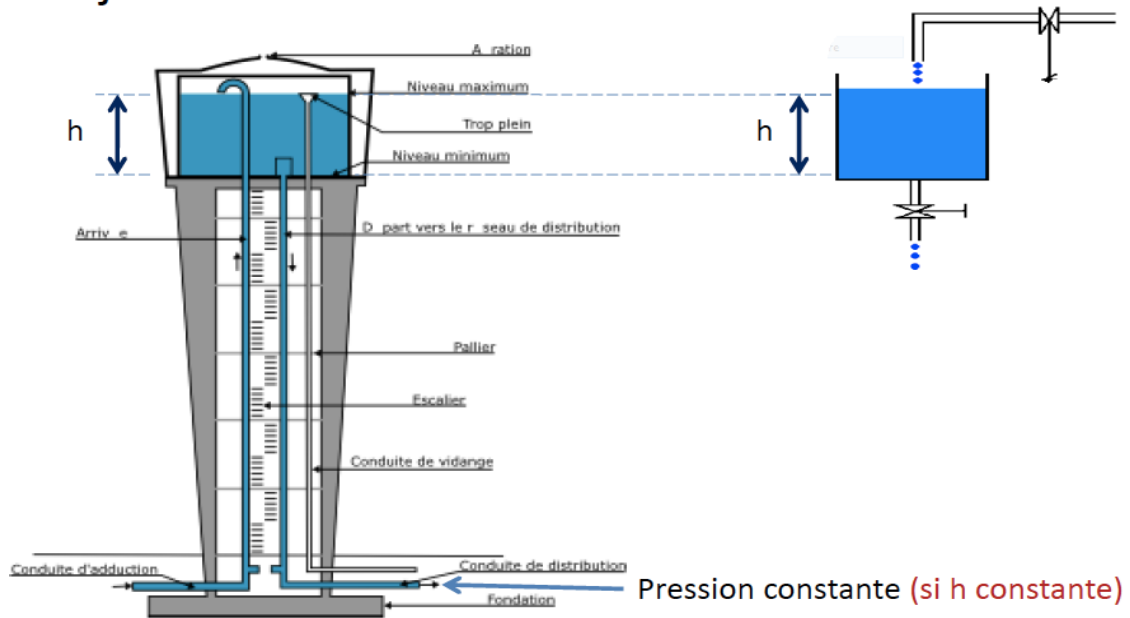
## Alimentation d'une ville en eau potable

Pression constante = niveau d'eau constant



## Notion de systèmes asservis

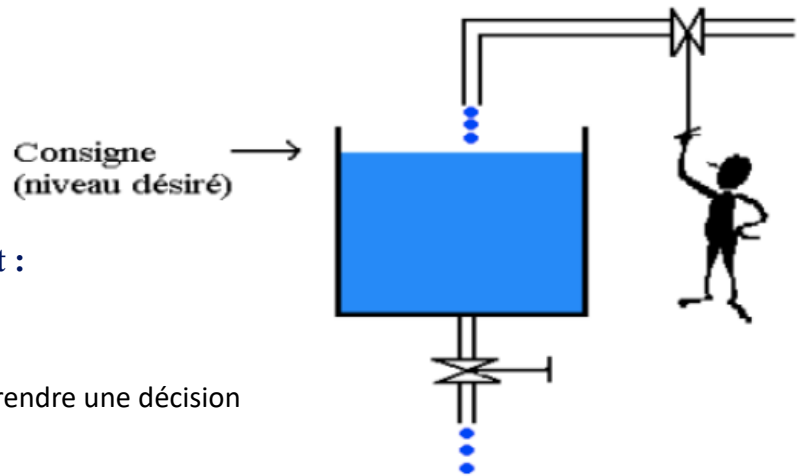
Objectif: hauteur constante



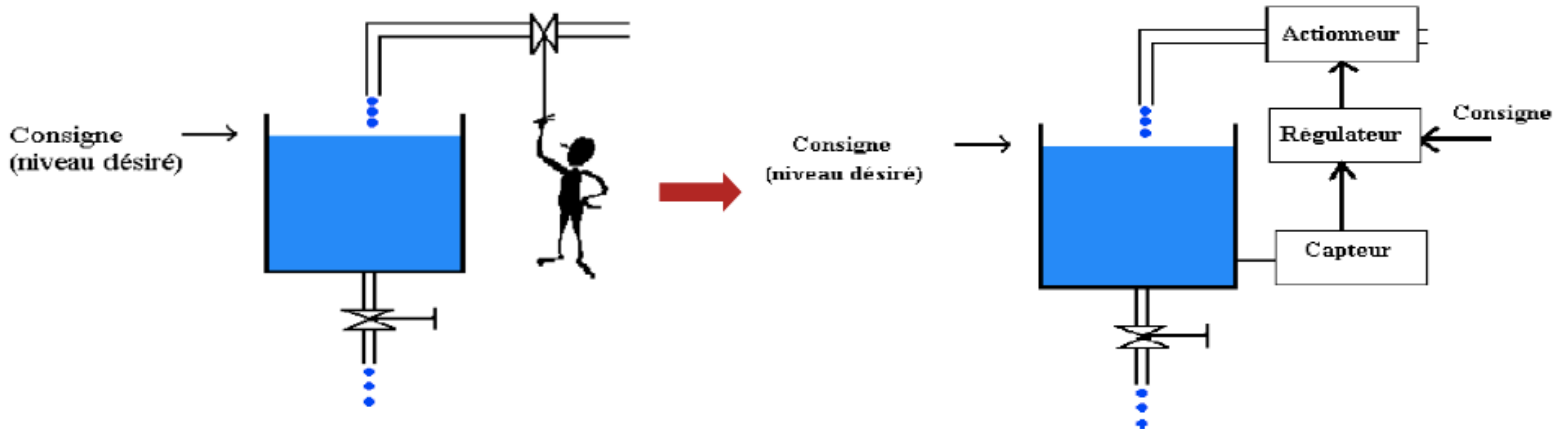
## Notion de systèmes asservis

Pour réguler le niveau manuellement, il faut :

1. **Observer/mesurer** le niveau
2. **Comparer** à la consigne et **raisonner** pour prendre une décision
3. **Agir** sur la vanne (ouvrir ou fermer)  
(douceMENT/brusquement)

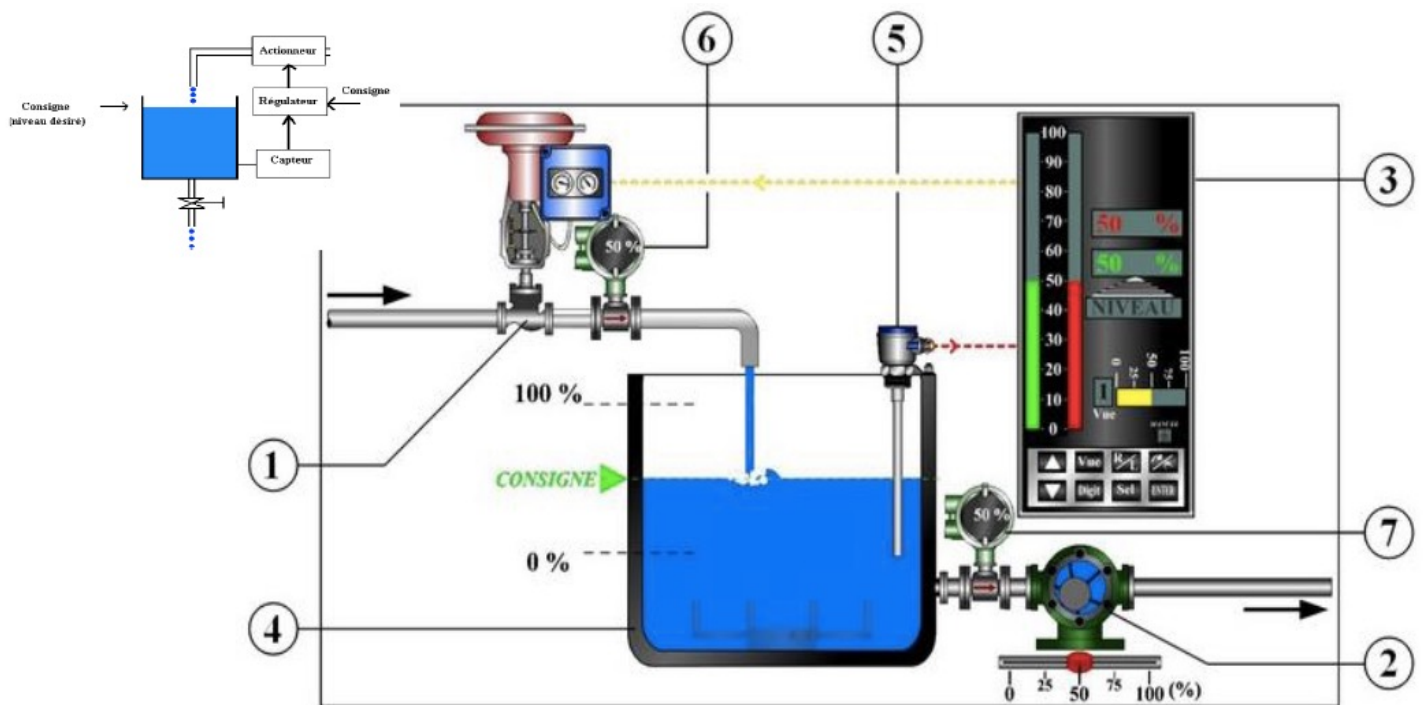


## Notion de systèmes asservis

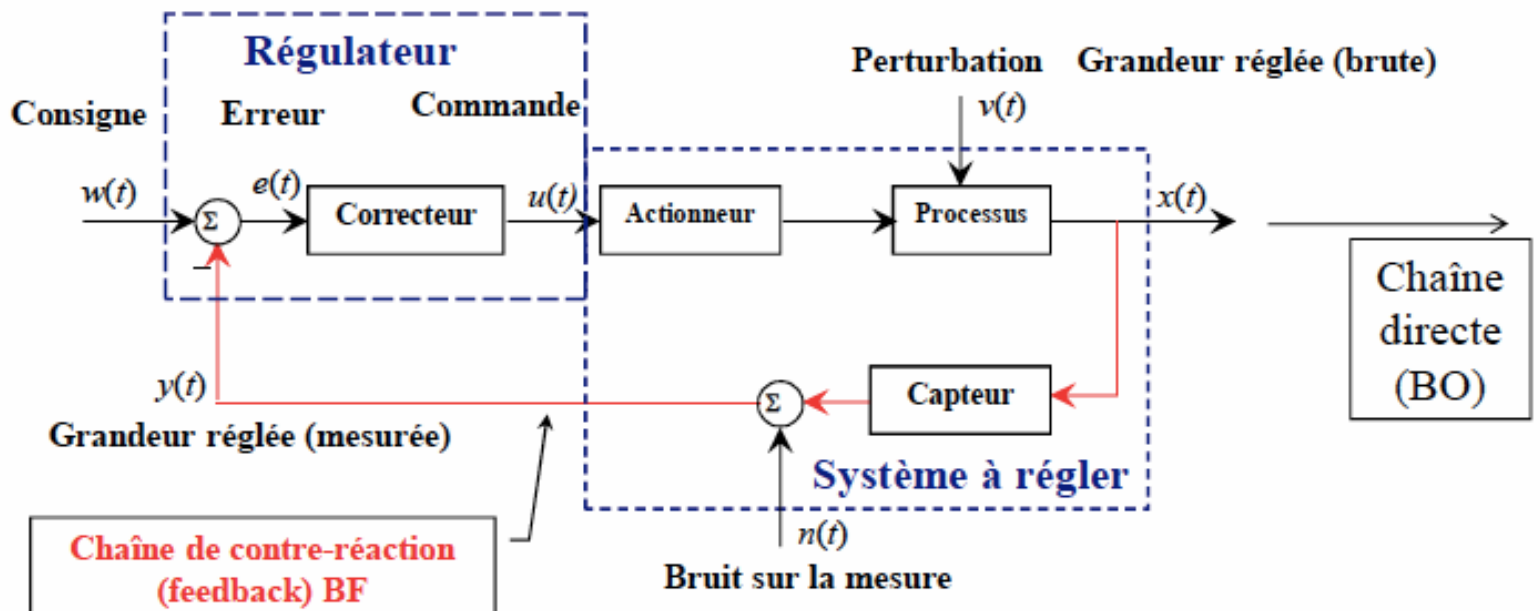


# Notion de systèmes asservis

En pratique



## Notion de systèmes asservis





Objectifs : Maîtriser l'outil **Matlab** ainsi que les critères de régulations les plus exigés en industrie à savoir la **stabilité**, la **précision**, la **rapidité**.

Volume horaire : 14h

Mode d'évaluation : Rapports de TP

# 1 Introduction

MATLAB est un logiciel de calcul numérique. Il propose un langage pour le calcul scientifique, l'analyse de données, leur visualisation, le développement d'algorithmes... La fenêtre principale de son interface est une fenêtre interactive type console pour l'exécution d'instructions. MATLAB, combiné à Simulink et certaines toolboxes, constitue un outil logiciel puissant, très utilisé par la communauté des Automaticiens. Sur MATLAB tout est matrice. Il est important de connaître les notions de bases du calcul matriciel pour mieux exploiter la puissance de MATLAB.

Voici une petite comparaison entre Matlab et C/C++ :

- Matlab est un langage de programmation haut niveau alors que le C/C++ est un langage de programmation bas niveau c'est-à-dire que la syntaxe de Matlab est très proche du langage humain.
- Matlab est un langage interprété alors que le C/C++ est un langage compilé.
- Le C/C++ est plus rapide que Matlab

Dès que l'on initie Matlab, nous sommes introduits dans l'interface de Matlab. Cette interface se compose de diverses sous-fenêtres, notamment la fenêtre de commande, destinée à l'entrée de commandes, la fenêtre de l'historique, où l'historique de toutes les variables employées est consigné, le Workspace, et le répertoire courant, entre autres éléments.

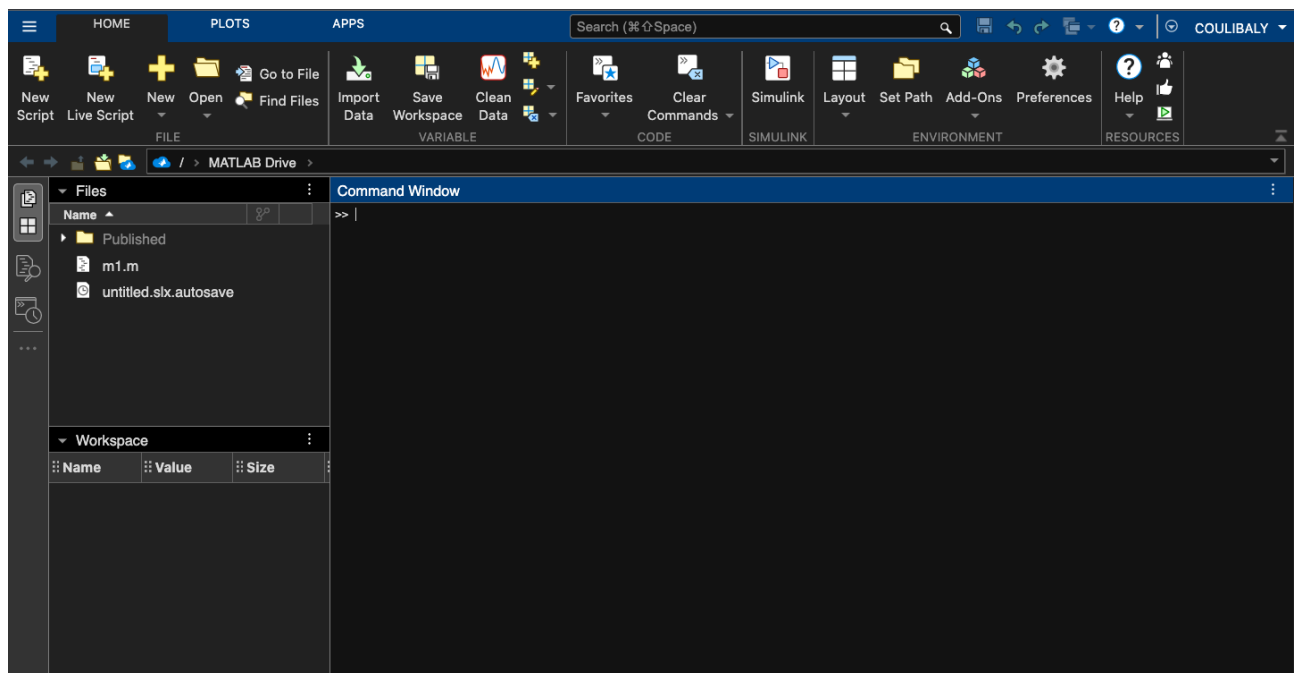


FIGURE 1 – Interface de Matlab

Sur Matlab il est possible de créer des **M-File**. Un m-file ou fichier script est fichier avec l'extension `.m` contenant un ensemble de commandes MATLAB. Ce fichier est exécuté directement sans compilation.

## 1.1 Vecteurs et Matrices

- **scalaire** : `» x = 5`
- **Vecteur** :
  - » `V1 = [1 2 3]` % Vecteur ligne
  - » `V2 = [1; 2; 3]` % Vecteur colonne
  - » `V3 = 1 : 7` % Vecteur ligne contenant 7 éléments : les nombres de 1 à 7.
- **Mtrice** : `A = [1 4 5; 2 3 2; 0 1 0]` % Matrice carrée de taille 3.

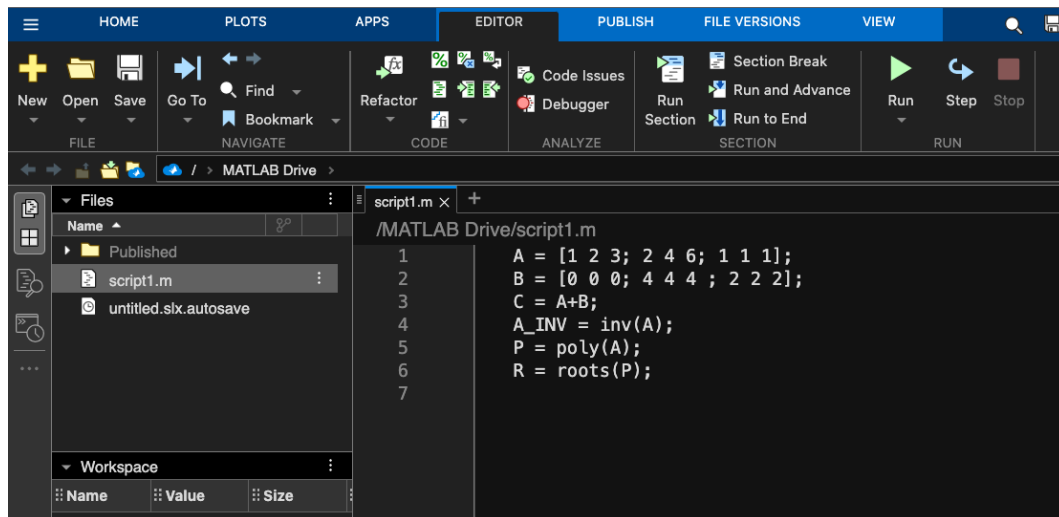


FIGURE 2 – Script matlab

**Remarque :** Le symbole % est utilisé pour faire des commentaires sur MATLAB.

## 1.2 Fonctions pour créer des matrices

Ces fonctions sont utilisées pour générer des matrices.

- **zeros(m,n)** % Génère une matrice ne contenant que des zéros.
- **ones(m,n)** % Génère une matrice de taille mxn ne contenant que des 1.
- **eye(n)** % matrice identité.

## 1.3 Opérations sur les vecteurs

- » **v1 + 3** % ajoute un scalaire au vecteur v1.
- » **v1 + v2** % addition élément par élément des vecteurs v1 et v2.
- » **v1.\*v2** % multiplication élément par élément.
- » **v1\*v2** % dot product
- » **v1'** ou **transpose(v1)** : % transposée du vecteur v1.
- » **v1./v2** % division élément par élément

## 1.4 Opérations sur les Matrices

- » **A + 3** % Ajout du scalaire 3 à chaque élément de la matrice A.
- » **A \* 3** % Multiplication de chaque élément de la matrice A par le scalaire 3.
- » **A \* v** % Multiplication de la matrice A par un vecteur v. Ceci est possible si le nombre de colonnes de A est égal au nombre de lignes de v.
- » **cos(A)** % Calcul du cosinus de la matrice A.
- » **sin(A)** % Calcul du sinus de la matrice A.
- » **tan(A)** % Calcul de la tangente de la matrice A.
- » **exp(A)** % Calcul de l'exponentielle la matrice A.
- » **A + B** % Addition de deux matrices. L'addition est possible si  $\dim(A) = \dim(B)$ .
- » **A \* B** % Multiplication de la matrice A par la matrice B. Ceci est possible si et seulement si le nombre de colonnes de A est égale au nombre de lignes de B.
- » **A.^3** % Puissance élément par élément de A » **A'** ou **transpose(A)** % Transposée de la matrice A.
- » **A<sup>-1</sup>** ou **inv(A)** % Inverse de la matrice A » **A/B** % Équivalent à  $A * \text{inv}(B)$ .

- » **A backslash B** % Equivalent à  $\text{inv}(A)*B$ .
- » **det(A)** % retourne le déterminant de A.

## 1.5 Technique d'indexing et de slicing

Cette technique permet de un élément ou une sous matrice de la matrice. **Sur Matlab l'index commence toujours par 1 et non 0**

- » **A(1,1)** % Extrait l'élément se trouvant à la première ligne et première colonne.
- » **A(:,2)** % Extrait toute la deuxième colonne.
- » **A(2,1:3)** % Extrait les éléments de 1 à 3 de la deuxième ligne.

## 1.6 Exercices

### 1.7 Les bases du langage Matlab

- **Variables :**
  - n = 25 % Integer
  - a = 6.2 % Real Number
  - univ = "HESTIM" % Text
- **Data type :**
  - double(n) % Converti le type de la variable n en double.
  - int8(a), int16(a), int32(a), int64(a) % Conversion en 8bit, 16bits, 32bits, 64bits
- **Les nombres complexes**
  - » i % Nombre imaginaire i
  - » j % Nombre imaginaire j
  - » x = 1+i % Nombre complexe
  - » real(x) % Partie réelle de x
  - » imag(x) % Partie imaginaire de x
  - » angle(x) % argument x
  - » conj(x) % Conjugué de x
  - » abs(x) % Amplitude ou le module de x.
- **Les expressions mathématiques**
  - » **abs(a)** % Valeur absolue de a
  - » **cos(a) ; sin(a) ; tan(a)** % Fonctions trigonométriques
  - » **acos(a) ; asin(a) ; atan(a)**
  - » **sqrt(a)** % racine carré de a
  - » **exp(a)** % exponentielle de a
  - » **log(a)** % logarithme naturel de a

## 1.8 Tracé des graphes

Le scrip suivant trace sur une même figure les fonctions trigonométriques sinus et cosinus sur le même repère.

```

x = 0:pi/100:4*pi;
y1 = sin(x);
y2 = cos(x);
plot(x,y1,'black',x,y2,'red--','LineWidth',2)
xlabel('X')
ylabel('Y')
axis([0 4*pi -1 1])
title('Tracé des fonctions Sinus et du Cosinus ', 'FontSize',12)
legend('sin(x)', 'cos(x)')

```

FIGURE 3 – Script de tracé des fonctions sinus et cosinus

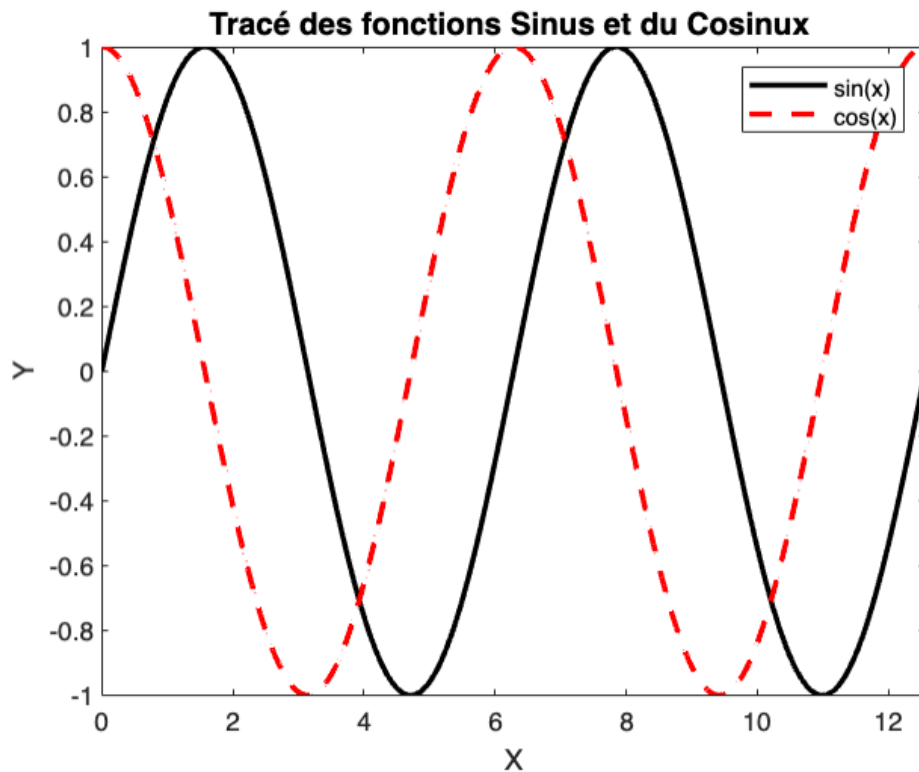


FIGURE 4 – graphes des fonctions sinus et cosinus