

TP 01 : Introduction à la programmation MATLAB

1. Objectifs :

À l'issue de cette séance de travaux pratiques, l'étudiant devra être capable de :

- Comprendre l'environnement de travail MATLAB et ses principales composantes.
- Définir, manipuler et gérer des matrices.
- Réaliser des opérations mathématiques simples sur les matrices.
- Rappel sur les boucles, les fonctions mathématiques prédéfinies sous MATLAB.
- Résolution des équations mathématiques.

2. Introduction :

MATLAB (*Matrix Laboratory*), est un langage de programmation et un environnement de calcul numérique développé par MathWorks. Il est principalement utilisé pour le calcul matriciel, la visualisation de données, l'implémentation d'algorithmes et l'interfaçage avec d'autres langages de programmation.

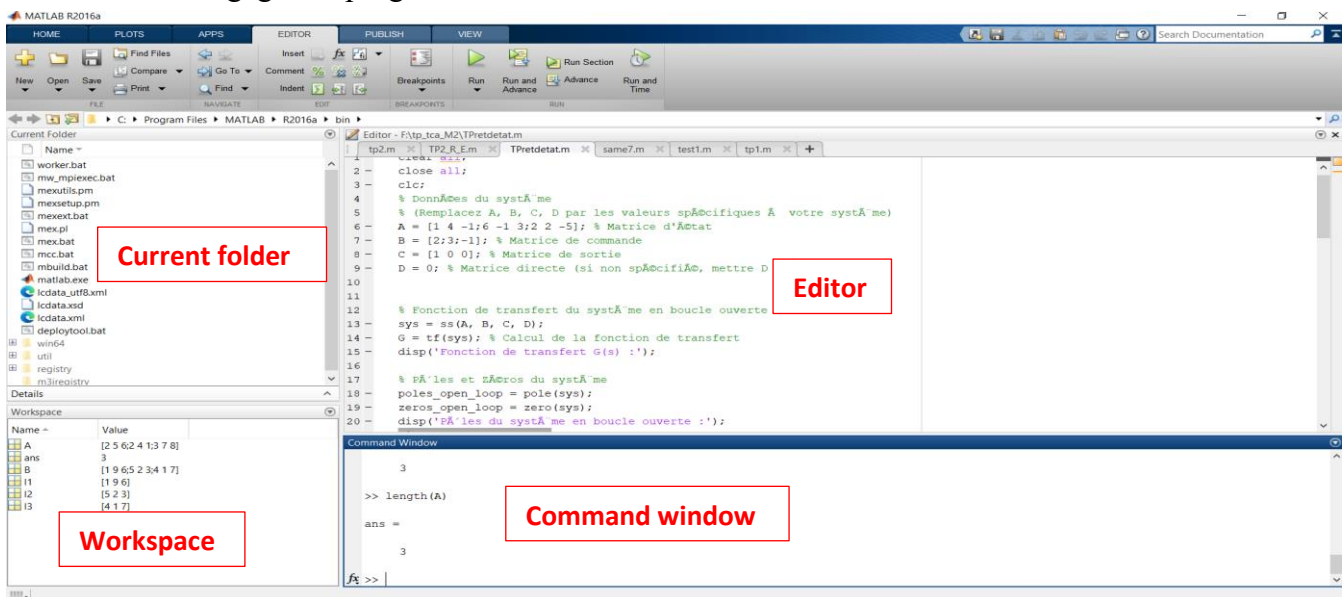


Figure 1. Les parties de l'interface MATLAB

Command Window : permet de saisir et d'exécuter directement les commandes MATLAB.

Workspace : affiche les variables en mémoire et leurs caractéristiques.

Editor (Script Editor) : permet d'écrire, modifier et exécuter des scripts et fonctions MATLAB.

Current Folder : affiche les fichiers et dossiers du répertoire de travail.

3. Opérations arithmétiques sur les matrices :

Soient trois matrices A, B et C :

$$A = \begin{pmatrix} 2 & 5 & 6 \\ 2 & 4 & 1 \\ 7 & 3 & 8 \end{pmatrix}; B = \begin{pmatrix} 1 & 9 & 6 \\ 5 & 2 & 3 \\ 4 & 1 & 7 \end{pmatrix}; C = \begin{pmatrix} 4 & 1 \\ 6 & 3 \end{pmatrix}; D = \begin{pmatrix} 2 & 3 \\ 1 & 4 \\ 3 & 1 \end{pmatrix}$$

Initialisation des matrices :

A = [2 5 6; 2 4 1; 3 7 8]

>> A =

2 5 6

2 4 1

3 7 8

l1 = [1 9 6] >> l1 = 1 9 6 (vecteur ligne)

l2 = [5 2 3] >> l2 = 5 2 3 (vecteur ligne)

l3 = [4 1 7] >> l3 = 4 1 7 (vecteur ligne)

B = [l1 ; l2 ; l3]

B =

1 9 6

5 2 3

4 1 7

Effectuer les opérations suivantes :

Addition et soustraction : A+B ; A+C ; A-B ; A-C.

L'opérateur « + » permet d'effectuer une addition matricielle. Toutefois, cette opération n'est possible que si les matrices ont les **mêmes dimensions**. Par exemple les matrices A et C sont de tailles incompatibles, l'addition n'est pas définie.

Dans ce cas, MATLAB affiche le message d'erreur suivant :

??? Error using ==> +

Matrix dimensions must agree.

Multiplication:

Produit matriciel : Le nombre de colonnes de A doit être égal au nombre de lignes de B.

$$(A*B) ; (A*C); (A*D)$$

Produit terme à terme : Les matrices doivent avoir la même taille. $(A .* B) ; (A .* D)$

Division élément par élément : $(A ./ B)$

Puissance : $(A.^2)$ puissance matricielle, $(A.^2)$ puissance élément par élément.

Transposée : A', B', C' et D' .

4. Affichage des valeurs

Pour voir le contenu d'une variable, tapez simplement son nom.

si vous tapez: `Aff = A + B`

Pour éviter l'affichage du contenu de la variable, mettez un point-virgule à la fin de toute instruction d'affectation. Par exemple: `Aff= A + B;`

5. Utilisation des sous matrices :

On souhaite souvent ne référencer qu'une partie d'une matrice. MATLAB fournit des moyens simples et puissants de le faire.

Pour référencer une partie d'une matrice, donnez le nom de la matrice suivi de parenthèses avec des expressions indiquant la partie souhaitée. Le cas le plus simple se présente lorsqu'un seul élément est souhaité. Par exemple, en utilisant C dans la section précédente:

$$C(1,2) \text{ vaut } 1 ; \quad C(2,1) \text{ vaut } 6$$

Dans tous les cas, la première expression entre parenthèses indique la ligne (ou les lignes), tandis que la deuxième expression indique la ou les colonnes. C. à. d. pour indiquer un élément on donne le numéro de la ligne et de la colonne (**la position**).

Lorsqu'on veut rechercher plus d'un élément d'une matrice, à titre d'exemple, pour indiquer «toutes les lignes», utilisez le signe deux points (:) pour la première expression. Pour indiquer «toutes les colonnes», utilisez deux points (:) pour la deuxième expression:

`B(1, :) = 1   9   6   >>` la première ligne

`B(:, 2) = 9`

`2       >>` la deuxième colonne

`1`

6. La taille d'une matrice ou un vecteur :

La taille d'une matrice ou d'un vecteur correspond au nombre de lignes et de colonnes qu'il contient. Dans MATLAB, la fonction **size** permet de déterminer les dimensions d'une matrice, tandis que la fonction **length** est utilisée pour connaître la taille d'un vecteur. Cette information est essentielle pour vérifier la compatibilité des opérations matricielles.

```
>> size(A)
```

```
ans =
```

```
3 3
```

```
>> length(l1)
```

```
ans =
```

```
3
```

7. Les boucles et les fonctions mathématiques :

☞ L'instruction de test **if** : permet d'exécuter un bloc d'instructions uniquement si une condition est vraie. Elle s'écrit comme suit :

```
if condition1
    Opération1 ;
elseif condition 2
    Opération2 ;
else
    Opération3 ;
end
```

Exemple 1:

```
R= input('R= ');
if R==10 ;
    Surface=pi*R^2 ;
elseif R< 10
    Surface=pi*R^3 ;
else
    Surface = pi*R ;
end
surface
```

☞ L'instruction **for** : permet de répéter un ensemble d'instructions pour un nombre déterminé de fois.

```
for i = valeur initiale : incrément : valeur finale
    Opération ;
end
```

Exemple 2:

```
for R=1 :0.01 :5 ;
    V=4/3*pi*R.^3 ;
end
disp([R,V])
```

☞ L'instruction **while** : exécute des instructions tant qu'une condition est vérifiée.

while condition

Opération ;

end

Exemple 3:

```
r = input('r= ');  
while r<5  
    v=4/3*pi*r.^3 ;  
end  
disp([r,v])
```

Pour avoir une aide instantanée sur une commande MATLAB, on peut taper la commande **help** suivie du nom de la commande, dans la fenêtre de la commande de Matlab (Command window)

- Fonctions mathématiques prédéfinies

Fonctions de base : sum, prod, mean, max, min

Fonctions mathématiques : sqrt, abs, exp, log, sin, cos, tan

Fonctions matricielles : det, inv, rank, eig

8. Résolution d'une équation du deuxième ordre $f(x)=0$:

Pour résoudre une équation quadratique de la forme :

$$x^2 + 3x + 4 = 0$$

1^{ère} méthode :

Coefficients = [1 3 4] ;

X = roots(Coefficients) ;

2^{ème} méthode :

a=1 ; b=3 ; c=4 ;

delta = b^2 - 4*a*c;

x1 = (-b + sqrt(delta))/(2*a);

x2 = (-b - sqrt(delta))/(2*a);