



TP 1

Présentation d'un environnement de programmation scientifique (Matlab)



Objectifs :

Découverte de l'environnement :

- Se familiariser avec l'interface utilisateur de Matlab.
- Comprendre les principales fonctionnalités : console, éditeur de script, espace de travail, navigateur de variables, etc.
- Apprendre à utiliser l'aide en ligne et les documents de référence intégrés.

Plan de TP :

1. Partie Théorique

1.1 Introduction à l'environnement

1.2 Démarrage de l'outil (Matlab, Scilab, etc.).

1.3 Présentation de l'interface.

1.4 Explication des menus principaux et des options de personnalisation.

2 Partie Pratique

2.1 Exécution d'instructions simples dans la fenêtre de command (opérations arithmétiques, fonctions mathématiques de base).

Partie Théorique

La partie théorique a pour but de donner les concepts de base sur l'environnement de programmation scientifique MATLAB, pour que les étudiants comprennent son utilité et ses fonctionnalités principales avant de passer à la pratique.

1.1 Introduction à l'environnement

- **Définition** : MATLAB (MATrix LABoratory) est un logiciel interactif utilisé pour le calcul scientifique, le traitement des données, la visualisation, la modélisation et le développement d'applications. Il est particulièrement performant dans le calcul numérique, l'algèbre linéaire et le traitement de données.
- **Utilisation courante** : MATLAB est largement utilisé en ingénierie, physique, mathématiques appliquées, finance, et dans d'autres domaines scientifiques.

1.2 Démarrage de l'outil Matlab

Une fois Matlab téléchargé et installé

Via l'icône de bureau (Windows)

- Double-cliquez sur l'icône sur votre bureau pour lancer l'application.

Une fois Matlab lancé, vous verrez l'interface principale

1.2 Présentation de l'interface

Une fois Matlab lancé, vous verrez l'interface principale Figure 1, qui contient généralement :

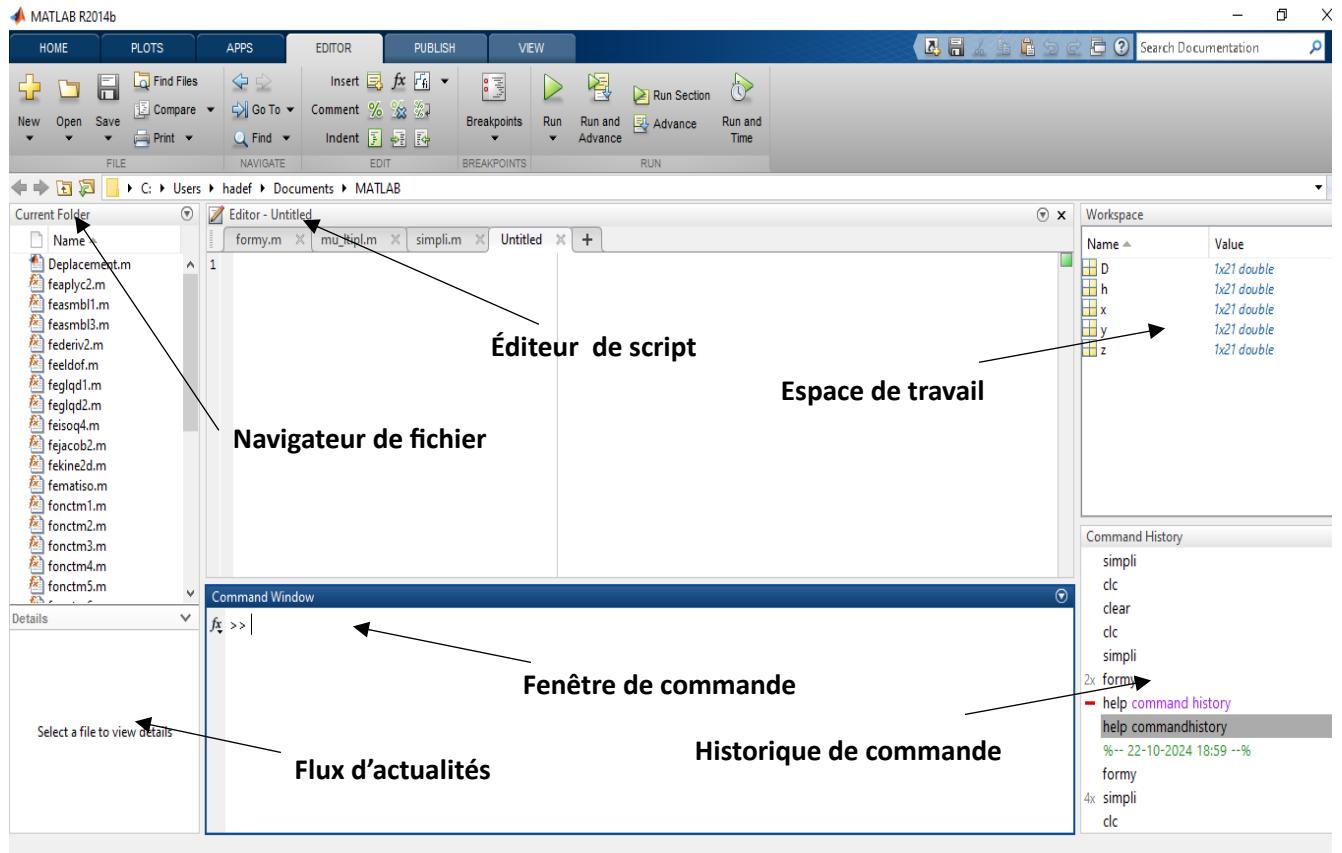


Figure 1 : La fenêtre principale de Matlab.

- **Console** : Il s'agit de la zone où vous tapez des commandes et voyez les résultats immédiatement. C'est un endroit pratique pour tester rapidement des expressions ou des commandes sans créer de script.
- **Éditeur de script** : C'est ici que vous écrivez des programmes plus longs ou des séquences d'instructions. L'éditeur permet de sauvegarder le code sous forme de fichiers avec extensions (.m).
- **Espace de travail (Workspace)** : C'est une fenêtre qui affiche toutes les variables que vous avez créées pendant une session, ainsi que leurs valeurs et types. Elle permet de garder un œil sur les données que vous manipulez.
- **Fenêtre de graphique** : Les outils Matlab permettent de créer des graphiques, des courbes et des visualisations de données. Cette fenêtre affiche ces résultats sous forme graphique.
- **Historique des commandes** : Cet espace contient une liste des commandes que vous avez récemment exécutées dans la console. Vous pouvez les consulter et réexécuter des commandes sans avoir à les taper à nouveau.

1.3 Explication des menus principaux et des options de personnalisation.

Une fois que vous avez démarré Matlab, il est important de comprendre les menus principaux et les différentes options de personnalisation disponibles pour faciliter l'utilisation de l'environnement. Ces outils offrent une interface graphique riche avec des menus qui permettent d'accéder à de nombreuses fonctionnalités.

1.3.1. Menus principaux dans Matlab :

Matlab a une interface similaire à celle de Matlab avec des menus principaux qui permettent de gérer les scripts, les variables et les graphiques.

A. Menu "File" (Fichier) :

- **New File** : Créer un nouveau fichier script.
- **Open** : Ouvrir un fichier Matlab existant (.m).
- **Save** : Sauvegarder le fichier actuel.
- **Quit** : Fermer Matlab.

B. Menu "Edit" (Édition) :

Utilisé pour les opérations standards d'édition de texte dans les scripts.

- **Cut, Copy, Paste** : Couper, copier et coller des éléments de code.
- **Find/Replace** : Rechercher ou remplacer du texte dans un script.

C. Menu "Execute" (Exécuter) :

Il regroupe les options pour exécuter des scripts et des commandes.

- **Load into Matlab** : Charger un fichier dans l'environnement Matlab.
- **Run File** : Exécuter un script en cours.

- **Stop Exécution** : Arrêter un script en cours d'exécution.

D. Menu "Graphics" (Graphiques) :

Comme dans Matlab, ce menu permet de générer des graphiques à partir de vos données.

- **2D Plot** : Tracer des courbes en deux dimensions.
- **3D Plot** : Créer des surfaces 3D pour visualiser des données à trois dimensions.

E. Menu "Applications" :

Matlab offre des modules spécialisés pour certaines analyses.

- **Signal Processing** : Outils pour l'analyse des signaux.
- **Optimisation** : Outils pour résoudre des problèmes d'optimisation.

F. Menu "Help" (Aide) :

Accès à l'aide en ligne de Matlab, documentation des commandes et des fonctions disponibles.

G. Menu "Préférences" (Préférences) :

Comme Matlab, permet de personnaliser l'environnement à travers le menu **Préférences** :

- **Apearance** : Configurer l'apparence de l'interface.
- **Langage** : Sélectionner la langue d'affichage (Français, Anglais, etc.).
- **Graphics** : Configurer les options par défaut pour les graphiques (couleurs, tailles des axes, etc.).

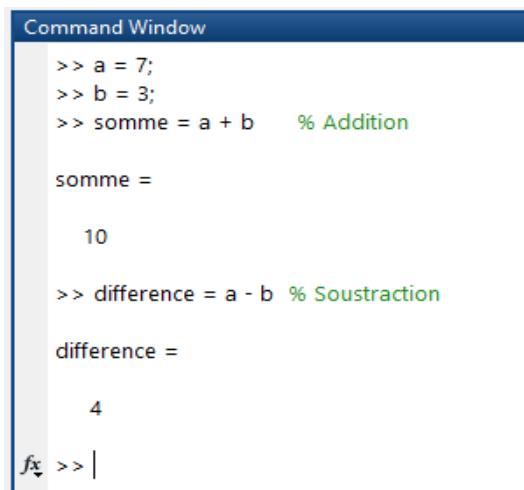
Partie Pratique

Cette partie pratique permet de manipuler directement les fonctionnalités de MATLAB introduites dans la partie théorique. Elle se compose de plusieurs exercices concrets pour découvrir pas à pas les commandes de base.

2.1 Exécution d'instructions simples dans la fenêtre de commande (opérations arithmétiques, fonctions mathématiques de base).

Exercice 1 : Création de variables et opérations de base

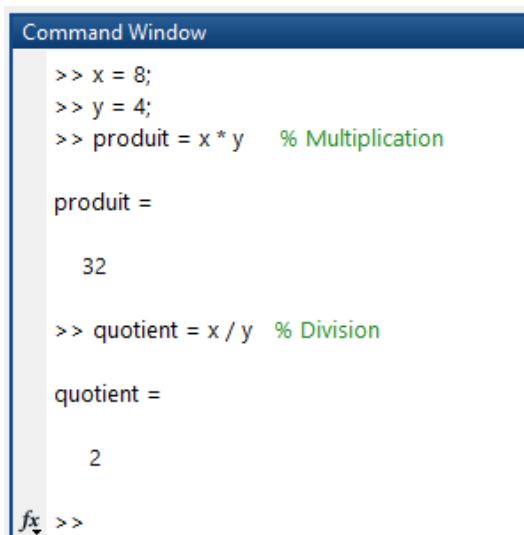
- **Objectif :** Manipuler les variables, effectuer des opérations de base et afficher des résultats.
- **Instructions :**
 - Créer deux variables, les additionner et les soustraire.



Command Window

```
>> a = 7;
>> b = 3;
>> somme = a + b      % Addition
somme =
10
>> difference = a - b % Soustraction
difference =
4
fx >> |
```

- Multiplier et diviser deux nombres.



Command Window

```
>> x = 8;
>> y = 4;
>> produit = x * y    % Multiplication
produit =
32
>> quotient = x / y   % Division
quotient =
2
fx >>
```

- ***clc*** : Efface l'écran (fenêtre) de MATLAB

Command Window

```
>> x = 8;
>> y = 4;
>> produit = x * y    % Multiplication

produit =
32

>> quotient = x / y   % Division

quotient =
2

fx >> clc
```

Command Window

```
fx >>
```

- ***clear ou (clear all)*** : Efface toutes les variables existantes en mémoire (workspace)

Command Window

```
>> x = 8;
>> y = 4;
>> produit = x * y    % Multiplication

produit =
32

>> quotient = x / y   % Division

quotient =
2

fx >> clear
```

Workspace

Name	Value
produit	32
quotient	2
x	8
y	4

Command Window

```
>> x = 8;
>> y = 4;
>> produit = x * y    % Multiplication

produit =
32

>> quotient = x / y   % Division

quotient =
2

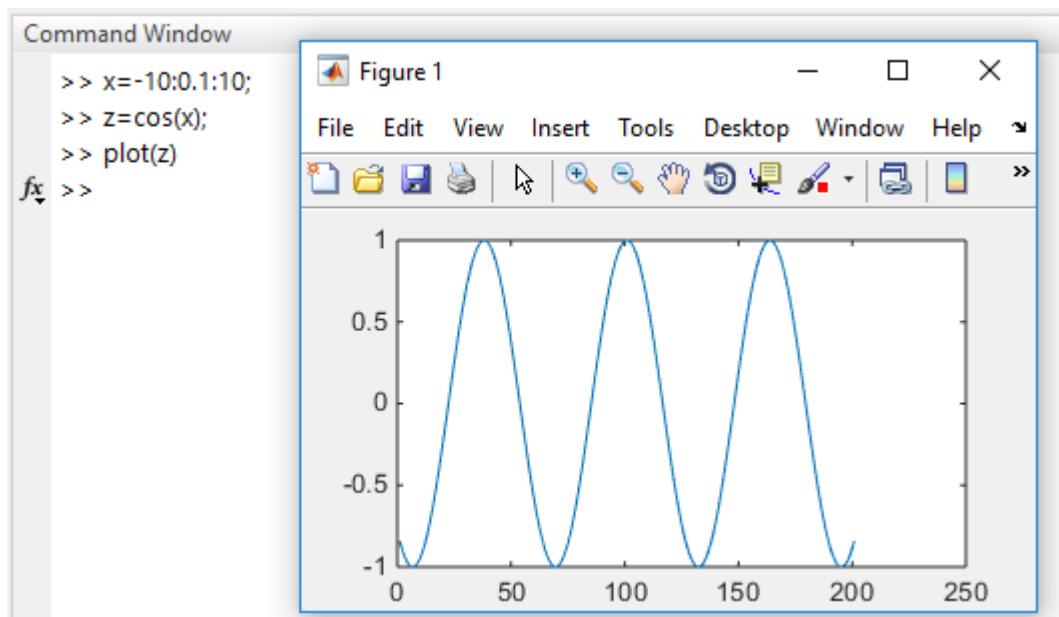
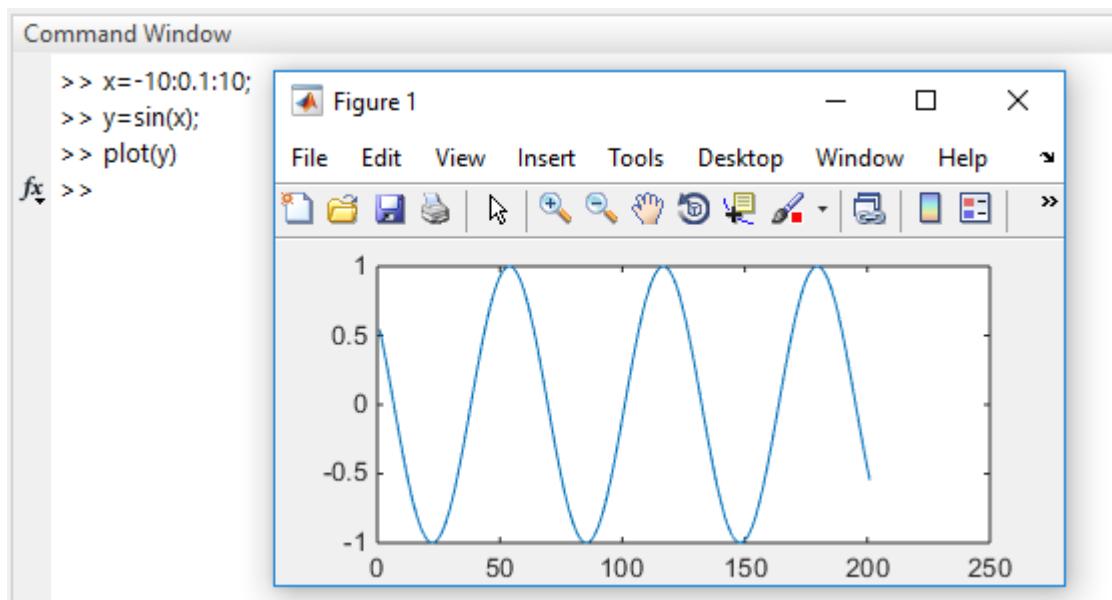
>> clear
fx >>
```

Workspace

Name	Value
------	-------

Exercice 2 : Calculer et tracer une fonction mathématique

- **Objectif :** Comprendre l'utilisation des vecteurs et des fonctions de tracé.
- **Instructions :**
 - Créer un vecteur x contenant des valeurs allant de -10 à 10 avec un pas de 0.1.
 - Calculer $y = \sin(x)$ et $z = \cos(x)$.
 - Utiliser la fonction plot pour afficher les courbes de y et z en fonction de x sur le même graphique.



Opération et caractères spéciaux		Fonctions élémentaires	
+	Plus	sin	Sinus
-	Moins	cos	Cosinus
*	Multiplication	tan	Tangente
/	Division	cot	Cotangente
.*	Multiplication élément par élément	exp	Exponentiel
^	Puissance	log	Logarithme
.^	Puissance élément par élément	sqrt	Racine carré
%	Ecrire un commentaire	abs	Absolue
;	semi-colonne	asin	Inverse sinus
:	colonne	acos	Inverse cosinus
'	transposé	atan	Inverse tangente
[]	crochet	acot	Inverse cotangente
()	parenthèse		
{}	accolade		

Conclusion

En conclusion, ce TP d'introduction à MATLAB permet de découvrir l'essentiel d'un environnement de programmation scientifique, utilisé largement pour les calculs numériques, la visualisation, et l'analyse de données. À travers des exercices pratiques, les étudiants apprennent à manipuler des variables, effectuer des opérations mathématiques de base, et à utiliser des fonctions intégrées pour résoudre des problèmes simples. Cette première étape leur donne une base solide pour explorer des applications plus complexes, où MATLAB devient un outil puissant pour le traitement de données, la modélisation et la simulation de systèmes scientifiques et techniques.