Logiciel de simulation "MATLAB"

Logiciel de simulation

Université de Tlemcen

Département GEE

Dr Saidi Farah

Janvier 2024

Table des matières

Objectifs

L'objectif de ce cours est de familiariser avec ***MATLAB***, un environnement de programmation et de modélisation puissant largement utilisé dans les domaines de l'ingénierie, des sciences et de la recherche.

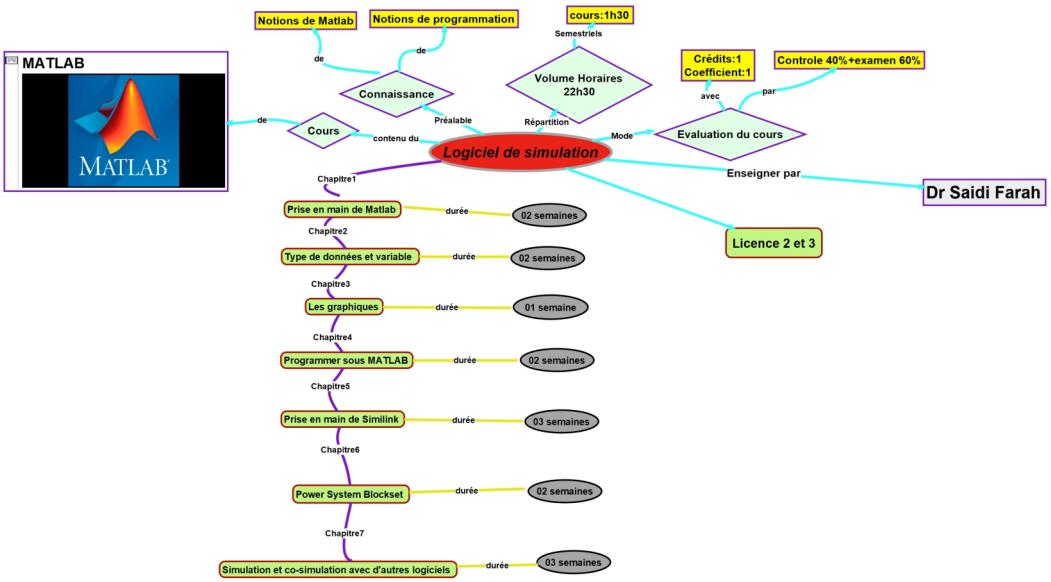
Les principaux objectifs du cours :

- Comprendre les bases de MATLAB.
- Maîtriser les techniques de programmation.
- Appliquer MATLAB à des projets pratiques .

Introduction

MATLAB est bien plus qu'un simple outil de calcul, c'est un environnement complet conçu pour répondre aux besoins variés des scientifiques, des ingénieurs et des chercheurs dans de nombreux domaines. Ce cours vous guidera à travers les bases de ***MATLAB*** jusqu'aux concepts avancés, vous permettant de maîtriser cet environnement et d'exploiter tout son potentiel.

Ce cours est divisé en plusieurs chapitres, chacun couvrant un aspect spécifique de ***MATLAB***. Nous commencerons par une introduction générale à ***MATLAB***, suivie par des chapitres sur la manipulation des données, la programmation, les graphiques, la modélisation et la simulation, et bien plus encore. Chaque chapitre sera accompagné d'exemples pratiques, d'exercices et de projets pour renforcer vos compétences et votre compréhension.



Carte de conception

I Prérequis recommandés

Les prérequis recommandés pour aborder efficacement ce premier chapitre sont:

1. ***Connaissance de base en mathématiques*** : Une compréhension générale des concepts mathématiques tels que les opérations arithmétiques, les fonctions, les équations, etc.
2. ***Familiarité avec la programmation*** : Bien qu'il ne soit pas nécessaire d'avoir une expérience préalable avec ***MATLAB***, une certaine familiarité avec les concepts de base de la programmation (variables, structures de contrôle, fonctions, etc.) peut faciliter la compréhension des exemples de code et des instructions fournies.
3. ***Logiciel MATLAB installé*** : L'installation et l'accès à un environnement « ***MATLAB*** » fonctionnel.
4. ***Curiosité et désir d'apprendre*** : La volonté d'explorer de nouveaux concepts et de pratiquer activement avec ***MATLAB*** est essentielle pour tirer le meilleur parti du chapitre et développer vos compétences en utilisant ce logiciel.

II Exercice : Test des prérequis

Qu'est-ce que MATLAB ?

- Un langage de programmation utilisé uniquement pour la simulation.
- Un environnement de développement intégré pour l'analyse numérique et la programmation technique.
- Un système d'exploitation open source.

III Exercice : Test des prérequis

Quelle est la principale utilisation de MATLAB ?

- ☐ Analyse des réseaux sociaux.
- ☐ Résolution de problèmes mathématiques et techniques.
- ☐ Édition de vidéos.

IV Chapitre 1

MATLAB, une contraction de "***Matrix LABoratory***", représente un environnement de calcul scientifique puissant, complet et convivial. Fondamentalement centré sur le traitement des matrices, ***MATLAB*** manipule les données, les variables et les images sous forme de matrices. Ce logiciel est développé par la société américaine " ***MathWorks***" et est accessible sur leur site web *www.mathworks.com*. ***MATLAB*** offre une gamme de fonctionnalités, notamment l'analyse de données, le développement d'algorithmes, les calculs mathématiques, les simulations et la modélisation de systèmes variés, ainsi que la représentation graphique des résultats obtenus.

Pour les ingénieurs, chercheurs et tout scientifique, ***MATLAB*** constitue un outil interactif intégrant calcul numérique et visualisation. Son environnement performant, ouvert et programmable permet des gains significatifs en productivité et en créativité.

1. Environnement de MATLAB

MATLAB représente un environnement complet, ouvert et extensible dédié au calcul et à la visualisation. Il offre une pléthore de fonctions mathématiques, scientifiques et techniques, totalisant plusieurs centaines voire milliers, en fonction des versions et des modules optionnels disponibles.

L'approche matricielle de ***MATLAB*** permet le traitement des données sans aucune contrainte de taille, permettant ainsi d'effectuer des calculs numériques et symboliques de manière fiable et rapide. Les fonctions graphiques de ***MATLAB*** facilitent la modification interactive des paramètres des graphiques pour les adapter selon nos besoins.

En outre, ***MATLAB*** est accompagné de ***Simulink***, un environnement de modélisation puissant basé sur les schémas-blocs, permettant la simulation de systèmes dynamiques linéaires et non linéaires.

Quelles sont les particularités de MATLAB ?

MATLAB offre une expérience de travail interactive, que ce soit en mode commande ou en mode programmation, tout en offrant des fonctionnalités de visualisation graphique. Il est reconnu comme l'un des meilleurs langages de programmation.

Par rapport à d'autres langages, ***MATLAB*** se distingue par :

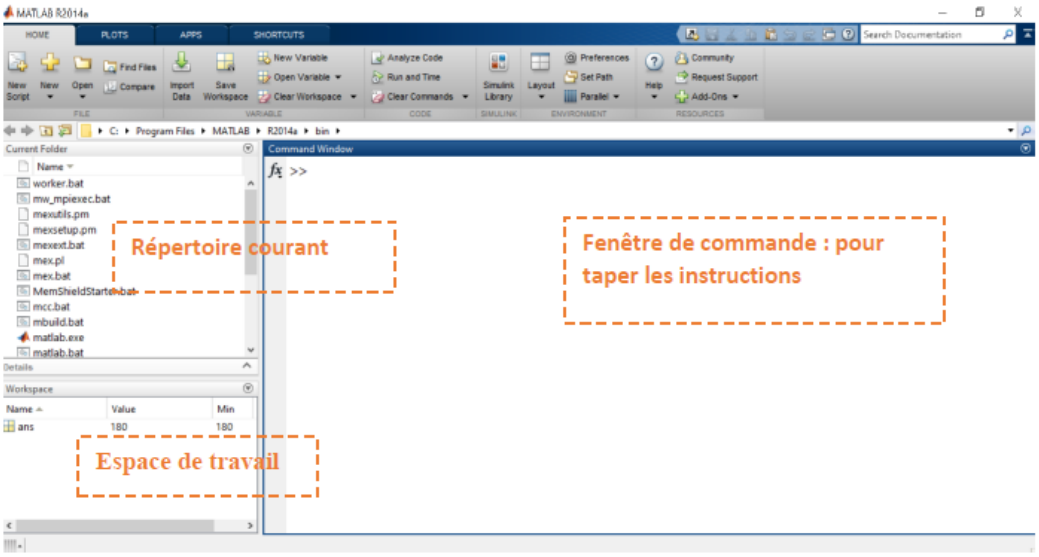
- Sa facilité de programmation.
- Sa richesse en fonctionnalités mathématiques et techniques.
- Sa capacité à traiter les données sous forme matricielle de manière efficace.
- Son environnement interactif favorisant une exploration aisée des données et une expérimentation rapide.

2. Démarrage du MATLAB

Pour démarrer « ***MATLAB*** » il suffit un double clik sur l'icone

MATLAB

L'interface ***MATLAB*** se compose d'une fenêtre principale divisée en sous-fenêtres :



Fenêtre de démarrage de MATLAB

- Console d'exécution (Command window):

Dans l'interface ***MATLAB,*** la fenêtre de Commande, identifiable par l'invite de commande "**>>**", constitue le point d'accès principal pour interagir avec le logiciel. C'est ici que les utilisateurs saisissent les instructions à exécuter.

- Le répertoire courant (Current folder):

Le répertoire courant permet à l'utilisateur de naviguer dans l'arborescence des fichiers de son système et d'accéder au contenu du répertoire actuellement sélectionné. C'est dans ce répertoire que les programmes de l'utilisateur doivent être placés pour être reconnus et exécutés par ***MATLAB.***

- L'espace de travail (Workspace):

L'espace de travail offre une vue d'ensemble des variables actuellement définies dans l'environnement ***MATLAB***, affichant leur nom, type et taille en mémoire.

3. Présentation et généralités

Modes de fonctionnement

- ***Mode interactif*** : Dans ce mode, MATLAB exécute instantanément les instructions au fur et à mesure qu'elles sont entrées par l'utilisateur.
- ***Mode exécutif*** : Ce mode est utilisé pour exécuter un programme MATLAB contenu dans un fichier avec une extension ***.m***. Chaque fichier ***.m*** contient une série d'instructions MATLAB qui seront exécutées en séquence par MATLAB.

Symboles et Notations

- Le symbole "***>>***" sert de pointeur, indiquant à l'utilisateur où entrer les commandes.
- La variable "***ans***" (pour "answer" ou "réponse" en français) stocke le résultat de la dernière opération effectuée.
- Il n'est pas possible de naviguer dans l'historique des commandes précédentes dans la fenêtre de Commande. Les nouvelles commandes sont simplement saisies après le dernier ***">>"***.

3.1. Les opérateurs

3.1.1. Les opérateurs arithmétiques

Les opérations arithmétiques sont les opérations mathématiques de base effectuées sur des nombres.

- ***Addition (+)*** : L'addition consiste à combiner deux nombres pour obtenir leur somme.
  Par exemple :
  >> 3 + 5
  >> ans= 8.
- ***Soustraction (-)*** : La soustraction consiste à retrancher un nombre d'un autre.
  Par exemple :
  >>7 - 4
  >>ans = 3.
- ***Multiplication (*)*** : La multiplication consiste à répéter une somme un nombre de fois égal à un autre nombre.
  Par exemple :
  >>2 * 6
  >>ans= 12.
- ***Division (/)*** : La division consiste à partager un nombre en parties égales.
  Par exemple :
  >> 10 / 2
  >>ans = 5.
- ***Exponentiation (^)*** : L'exponentiation consiste à élever un nombre à une puissance donnée.
  Par exemple :
  >>2^3
  >>ans= 8
- ***Modulo (%)*** : Le modulo retourne le reste de la division entière de deux nombres.
  Par exemple :
  >>10 % 3
  >>ans = 1 (car 10 divisé par 3 donne un reste de 1).

En utilisant ces opérations, il est possible d'effectuer une variété de calculs mathématiques dans MATLAB, que ce soit pour des tâches simples ou complexes.

3.1.2. Les caractères spéciaux

Les caractères spéciaux, dans le contexte de MATLAB, sont des symboles utilisés pour effectuer des opérations spécifiques ou pour représenter des valeurs particulières.

;	Supprime l'affichage du résultat dans la fenêtre de commande
,	Séparateur d'instructions
:	Crée une plage de valeurs (utilisé dans les vecteurs, matrices, etc.)
"	Délimiteur pour les chaines de caractères
[ ]	Concaténation de chaines de caractères
\n	Nouvelle ligne
\t	Tabulation

3.1.3. Opérateurs Logiques

&	ET logique (AND)
|	OU logique (OR)
~	NON logique (NOT)

3.1.4. Opérateurs de Comparaison

==	Égal à
~=	Différent de
<	Inférieur à
>	Supérieur à
<=	Inférieur ou égal à
>=	Supérieur ou égal à

3.1.5. Opérateurs d'Affectation

=	Affectation simple
*+=, -=, = /=	Affectations combinées

3.2. Commande MATLAB

MATLAB propose également une gamme de commandes intégrées qui permettent d'effectuer diverses opérations.

3.2.1. Commandes de Manipulation de Variables

clear	Supprime les variables de l'espace de travail.
whos	Affiche les détails des variables présentes dans l'espace de travail.
size	Retourne les dimensions d'une matrice.
length	Retourne la longueur d'un vecteur.
reshape	Modifie la forme d'une matrice sans changer ses éléments.

3.2.2. Commandes pour les Graphiques

plot	Trace des graphiques 2D
scatter	Trace un nuage de points
bar	Trace un graphique en barres
hist	Trace un histogramme
surf	Trace un graphique en 3D

3.2.3. Commandes de Contrôle de Flux

if, else, elseif	Structure conditionnelle pour exécuter des instructions en fonction d'une condition
for	Boucle itérative pour répéter des instructions un certain nombre de fois
while	Boucle itérative pour répéter des instructions tant qu'une condition est vraie
switch, case	Structure de contrôle pour sélectionner parmi plusieurs alternatives

3.2.4. Commandes pour les Mathématiques et Statistiques

sum	Calcule la somme des éléments d'un vecteur ou d'une matrice
mean	Calcule la moyenne des éléments d'un vecteur ou d'une matrice
std	Calcule l'écart-type des éléments d'un vecteur ou d'une matrice
max, min	Retournent respectivement le maximum et le minimum d'un vecteur ou d'une matrice

3.3. Fonctions mathématiques élémentaires

En MATLAB, il existe de nombreuses fonctions mathématiques intégrées qui permettent d'effectuer des calculs complexes.

3.3.1. Fonctions Trigonométriques :

sin(x)	Calcule le sinus de x (en radians)
cos(x)	Calcule le cosinus de x (en radians)
tan(x)	Calcule la tangente de x (en radians)
asin(x)	Calcule l'arc sinus de x (en radians)
acos(x)	Calcule l'arc cosinus de x (en radians)
atan(x)	Calcule l'arc tangente de x (en radians)

3.3.2. Fonctions Exponentielles et Logarithmiques :

exp(x)	Calcule l'exponentielle de x (e^x)
log(x)	Calcule le logarithme népérien de x (ln(x))
log10(x)	Calcule le logarithme décimal de x (log10(x))
sqrt(x)	Calcule la racine carrée de x

3.3.3. Fonctions de Puissance et de Valeur Absolue :

power(x,y)	Calcule x élevé à la puissance y (x^y)
abs(x)	Calcule la valeur absolue de x

3.3.4. Fonctions Statistiques :

mean(x)	Calcule la moyenne des éléments de x
std(x)	Calcule l'écart-type des éléments de x
max(x)	Retourne la valeur maximale de x
min(x)	Retourne la valeur minimale de x

✎ *Remarque*

En MATLAB, les noms de variable doivent respecter certaines règles pour garantir leur utilisation correcte dans les scripts et les fonctions. Voici quelques points importants à retenir :

- Les noms de variable doivent commencer par une lettre et peuvent être suivis de lettres, de chiffres ou de tirets bas (_).
- Les noms de variable sont sensibles à la casse, ce qui signifie qu'ils font la distinction entre les majuscules et les minuscules.
- Les caractères spéciaux, à l'exception du tiret bas (_), ne sont pas autorisés dans les noms de variable.

- Il est recommandé d'éviter d'utiliser des noms de variables qui correspondent déjà à des noms de fonctions intégrées, tels que min, max, exp, etc., pour éviter toute confusion.
- MATLAB générera une erreur si un mot-clé réservé est utilisé comme nom de variable. Les mots-clés réservés incluent for, end, if, while, function, return, entre autres.

* *

*

Dans ce cours d'introduction à MATLAB, nous avons couvert les principaux aspects de ce logiciel, notamment les opérateurs arithmétiques, les fonctions mathématiques, les variables et les commandes spécifiques. Nous avons également examiné les bonnes pratiques de programmation, telles que le choix de noms de variable appropriés et la prise en compte des mots-clés réservés.

Il est important de noter que la maîtrise de MATLAB nécessite de la pratique et de l'expérience. En explorant les exemples, en résolvant des problèmes et en développant des projets, vous pourrez approfondir vos compétences et exploiter tout le potentiel de ce puissant logiciel.

V Activité d'apprentissage

1. Exercice

Quelle est la fonction utilisée pour générer des nombres aléatoires en MATLAB ?

- random
- rand
- run

2. Exercice

Quelle est la fonction utilisée pour afficher un message à l'écran dans MATLAB ?

- show
- disp
- msg

VI Types de données et variables

1. Les types de données

Les types de données jouent un rôle crucial dans MATLAB, car ils déterminent la manière dont les données sont stockées, manipulées et interprétées par le logiciel.

1.1. Nombres

- ***Entiers*** : Les nombres entiers sont des nombres entiers sans partie décimale. Ils peuvent être signés (positifs ou négatifs) ou non signés (positifs uniquement).
- ***Réels*** : Les nombres réels sont des nombres avec une partie décimale. Ils peuvent être stockés avec une précision simple (float) ou double (double), selon les besoins.

Exemple

- ***Entiers*** : 1, -5, 100.
- ***Réels*** : 3.14, -0.5, 10.75.

1.2. Caractères et Chaînes de Caractères

- ***Caractères*** : Les caractères individuels sont représentés par des valeurs ASCII.
- ***Chaînes de Caractères*** : Les chaînes de caractères sont des séquences de caractères. Elles sont délimitées par des apostrophes simples (' ') ou des guillemets doubles (" ").

Exemple

- Caractères : 'a', 'Z', '@'.
- Chaînes de caractères : 'Bonjour', "MATLAB".

1.3. Logique

Booléens : Les valeurs booléennes sont utilisées pour représenter la vérité ou la fausseté (true ou false). Elles sont souvent utilisées dans les expressions conditionnelles.

Exemple

true (ou 1), false (ou 0).

1.4. Tableaux et Matrices

- ***Vecteurs*** : Les vecteurs sont des tableaux unidimensionnels contenant des éléments de même type.
- ***Matrices*** : Les matrices sont des tableaux bidimensionnels contenant des éléments de même type.

Exemple

- Vecteur : [1, 2, 3], [0.5, 0.75, 1.0].
- Matrice : [1, 2; 3, 4], [0.5, 0.75; 1.0, 1.25].

1.5. Structures de Données

Les structures permettent de regrouper différents types de données dans une seule entité. Elles sont composées de champs (variables) et de leurs valeurs associées.

Exemple

struct('nom', 'John', 'age', 30).

1.6. Cellules

Les cellules sont des conteneurs pouvant stocker différents types de données dans des cellules individuelles. Elles sont souvent utilisées pour stocker des données hétérogènes.

Exemple

{'John', 30, [1, 2, 3]}.

2. Les variables

Les variables sont des symboles qui représentent des valeurs ou des données stockées en mémoire. Elles peuvent être utilisées pour stocker une variété de types de données, y compris les nombres complexes, les variables booléennes, les chaînes de caractères, les vecteurs, les matrices et les polynômes.

2.1. Nombre Complexe

- Un nombre complexe est un nombre qui a à la fois une partie réelle et une partie imaginaire.
- En MATLAB, les nombres complexes sont représentés sous la forme a + bi, où a est la partie réelle, b est la partie imaginaire et i est l'unité imaginaire (√(-1)).

Exemple

z = 3 + 4i représente un nombre complexe avec une partie réelle de 3 et une partie imaginaire de 4.

2.1.1. Parties Réelle et Imaginaire

Pour accéder à la partie réelle et à la partie imaginaire d'un nombre complexe, vous pouvez utiliser les fonctions ***real*** et ***imag***.

Exemple

- re = real(z); % Obtient la partie réelle de z
- im = imag(z); % Obtient la partie imaginaire de z

2.1.2. Conjugaison

La conjugaison d'un nombre complexe est obtenue en changeant le signe de sa partie imaginaire. Vous pouvez utiliser la fonction ***conj*** pour cela.

Exemple

z_conj = conj(z); % Calcule le conjugué de z

2.1.3. Module et Argument

Le module d'un nombre complexe correspond à sa distance à l'origine dans le plan complexe, tandis que l'argument est l'angle qu'il forme avec l'axe réel positif. Vous pouvez utiliser les fonctions ***abs*** et ***angle*** pour cela.

Exemple

- module_z = abs(z); % Calcule le module de z
- argument_z = angle(z); % Calcule l'argument de z en radians

2.2. Variables Booléennes

Les variables booléennes sont souvent utilisées pour représenter des conditions logiques dans les expressions conditionnelles ou les boucles.

2.2.1. Utilisation dans des Instructions de Contrôle

Les variables booléennes sont couramment utilisées dans des instructions de contrôle telles que les boucles ***for*** et ***while***, ainsi que les structures conditionnelles ***if***, ***else*** et ***elseif***.

Exemple

Resultat=true ;

if Resultat

disp("La variable est vraie !");

else

disp("La variable est fausse !");

end

2.3. Vecteurs

Un vecteur est un tableau unidimensionnel de valeurs numériques ou de caractères, stockées de manière consécutive en mémoire.

2.3.1. Création de Vecteurs

Vous pouvez créer des vecteurs en utilisant des crochets [ ] pour délimiter les éléments du vecteur. Les éléments sont séparés par des virgules (,).

Exemple

v = [1, 2, 3, 4, 5]; % Crée un vecteur avec les valeurs de 1 à 5

2.3.2. Accès aux Éléments

Vous pouvez accéder aux éléments individuels d'un vecteur en utilisant leurs indices. Les indices commencent à 1 pour le premier élément et se terminent à la longueur du vecteur.

Exemple

premier_element = v(1); % Accède au premier élément du vecteur v

deuxieme_element = v(2); % Accède au deuxième élément du vecteur v

2.3.3. Modification d'Éléments

Vous pouvez modifier les éléments d'un vecteur en affectant de nouvelles valeurs à des indices spécifiques.

Exemple

v(3) = 10; % Modifie le troisième élément du vecteur v pour qu'il devienne 10

2.3.4. Longueur du Vecteur

La fonction ***length*** peut être utilisée pour déterminer la longueur d'un vecteur, c'est-à-dire le nombre d'éléments qu'il contient.

Exemple

longueur = length(v); % Obtient la longueur du vecteur v

2.3.5. Opérations Vectorielles

MATLAB prend en charge les opérations vectorielles telles que l'addition, la soustraction, la multiplication et la division entre des vecteurs, ainsi que des opérations vectorielles avec des scalaires.

Exemple

w = v + 1; % Ajoute 1 à chaque élément du vecteur v

x = v .* 2; % Multiplie chaque élément du vecteur v par 2

2.4. Matrices

Une matrice est un tableau bidimensionnel de valeurs numériques ou de caractères, organisées en lignes et en colonnes.

2.4.1. Création de Matrices

Vous pouvez créer des matrices en utilisant des crochets ***[ ]*** pour délimiter les lignes de la matrice et des points-virgules ***;*** pour séparer les lignes.

Exemple

```
A = [1, 2, 3; 4, 5, 6; 7, 8, 9]; % Crée une matrice 3x3
```

2.4.2. Accès aux Éléments

Vous pouvez accéder aux éléments individuels d'une matrice en utilisant leurs indices de ligne et de colonne.

Exemple

```
element_1_2 = A(1, 2); % Accède à l'élément situé à la première ligne et deuxième colonne de la matrice A
```

2.4.3. Modification d'Éléments

Vous pouvez modifier les éléments d'une matrice en affectant de nouvelles valeurs à des indices spécifiques.

Exemple

```
A(2, 3) = 10; % Modifie l'élément situé à la deuxième ligne et troisième colonne de la matrice A pour qu'il devienne 10
```

2.4.4. Taille de la Matrice

Les fonctions ***size*** et ***length*** peuvent être utilisées pour déterminer la taille d'une matrice, c'est-à-dire le nombre de lignes et de colonnes qu'elle contient.

Exemple

```
[nb_lignes, nb_colonnes] = size(A); % Obtient le nombre de lignes et de colonnes de la matrice A
```

2.4.5. Opérations Matricielles

MATLAB prend en charge les opérations matricielles telles que l'addition, la soustraction, la multiplication et la division entre des matrices, ainsi que des opérations matricielles avec des scalaires.

Exemple

```
M1=[1 2 4;1 3 5;6 2 3] ; %matrice carrée de 3x3

V1=[4 ;5 ;6] ; %vecteur colonne de 3 éléments

M=M1*V1 % Multiplie chaque ligne de la matrice M1 par le vecteur V1

add=M1+2 %joute 2 à chaque élément de la matrice M1

sous=M1-1 %soustraire 1 à chaque élément de la matrice M1
```

* *

*

En conclusion, ce chapitre nous a permis de découvrir les bases de la manipulation des variables en MATLAB, notamment les nombres complexes, les vecteurs et les matrices.

Ces connaissances de base nous préparent à explorer des applications plus avancées de MATLAB dans les domaines de l'analyse numérique, de la modélisation et de la simulation, ainsi que de l'apprentissage automatique et de l'analyse de données. En poursuivant notre exploration, nous continuerons à découvrir les capacités étendues de MATLAB et son potentiel pour résoudre une variété de problèmes complexes.

VII Activité d'apprentissage

1. Exercice

Si vous avez une variable x définie comme suit : x = [1, 2, 3, 4, 5], quelle est la valeur de x(3) ?

2. Exercice

Quelle est la valeur de la fonction sqrt(25) ?

VIII Exercice : TD N°1

Exercice 1 : Création de Variables

Créez différentes variables MATLAB pour stocker des nombres réels, des nombres complexes, des chaînes de caractères, des vecteurs et des matrices.

Exercice 2 : Manipulation des Vecteurs et des Matrices

Effectuez des opérations simples telles que l'addition, la soustraction, la multiplication et la transposition sur les vecteurs et les matrices suivants:

* a=[1 12 3 40]

* b=[21;3;5;-5;16]

* c=[10 4 -3; -7 5 12; 10 -12 14]

* d=[2 22 3;14 15 26; 5 20 11;3.3 25 21]

Exercice 3 : Utilisation de Fonctions Intégrées

Utilisez les fonctions max(), min(), sum() pour effectuer des opérations sur des vecteurs et des matrices.

* A=[-2 33 2 5 14 90]

* B=[56 42 5 72 27 10]

* C=[32 12 21 ;52 25 62 ;-3 -45 51]

* D=[0.1 0.3 0 10 ;0.125 -.6 -0.125 ;41 1.23 3.21]