

**Institut de maintenance et de Sécurité Industrielle  
(IMSI - Oran)**

# **Informatique 3**

SUPPORT DE COURS

Chapitre 3

par  
MOUFOK Souad

# Chapitre 3

## La programmation avec Matlab

### 3.1 Introduction

Jusqu'ici, nous n'avons utilisé que la fenêtre « Command Window » pour créer et manipuler les variables, ainsi que l'utilisation des différentes fonctions prédéfinies.

Cette partie de travail, ne permet pas d'exécuter des programmes avec beaucoup de lignes de commandes pour résoudre des problèmes plus complexes, qui demandent des commandes plus structurées ou plus nombreuses.

Ce présent chapitre, traite la partie programmation Matlab, qui explique comment utiliser Matlab comme un véritable langage de programmation en utilisant une nouvelle fenêtre appelée « Script » et afficher l'exécution dans la fenêtre « Command Window ».

### 3.2 Fichier Script

#### 3.2.1 Définition

Il est possible d'enregistrer une séquence d'instructions dans un seul fichier appelé « Script ». Un script ou « M.File » est un fichier texte qui regroupe plusieurs commandes Matlab, identiques à celles que l'on peut employer directement dans la fenêtre de commandes de MATLAB, enregistré sous Matlab avec l'extension « .m » et qui joue le rôle de programme principal.

#### 3.2.2 Utilisation de la fenêtre script ou « M.File »

- Ouverture de la fenêtre : on crée des fichiers scripts en utilisant une fenêtre qui s'ouvre de plusieurs façons.

- à partir de la barre d'outils en cliquant sur l'icône  ,
- en tapant la commande `>> edit`,
- à partir du menu fichier en cliquant sur File→New→ M-file (figure cidessous).

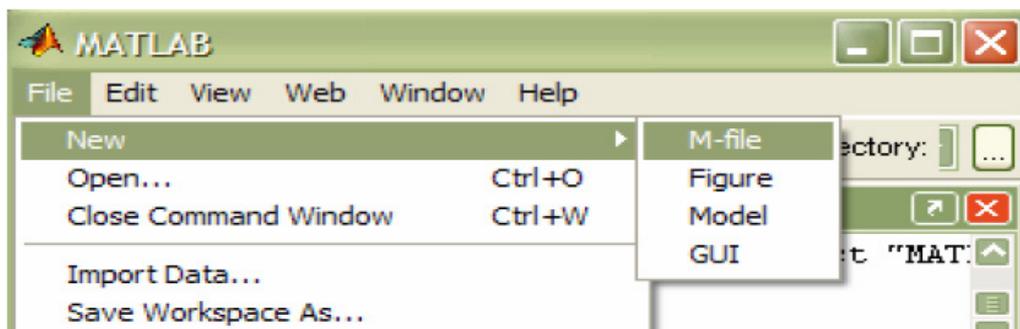
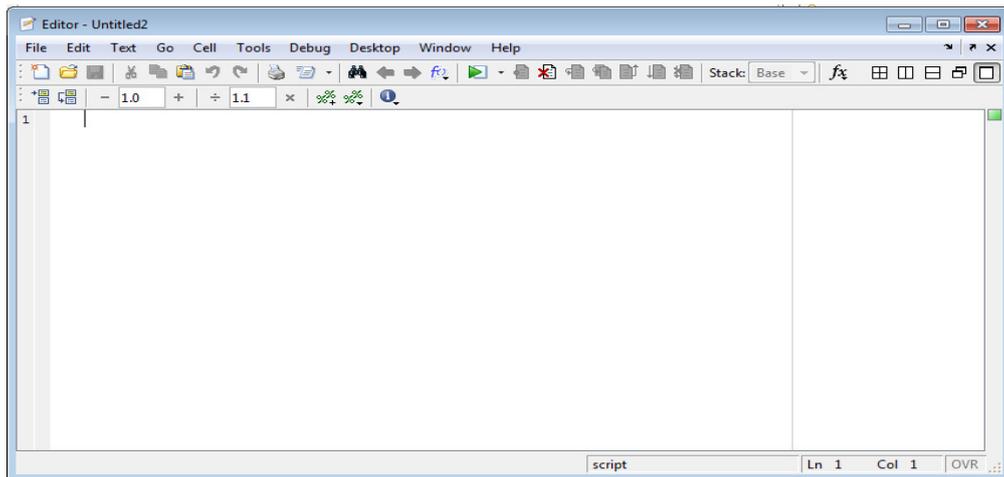


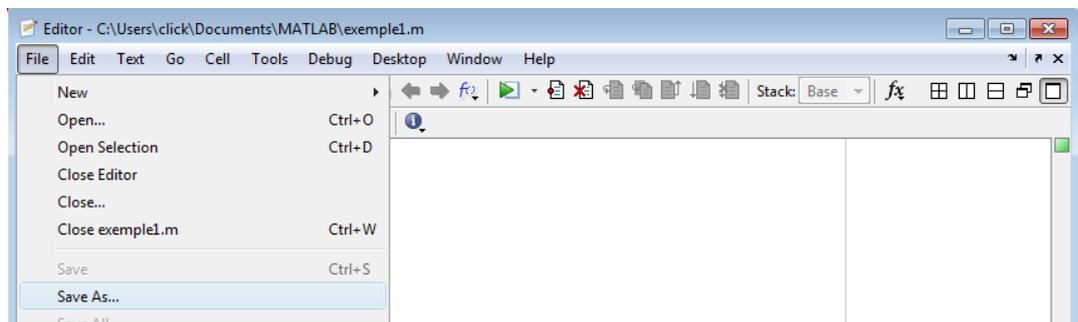
Figure 3.1 Ouverture d'un nouveau fichier « M.file ».



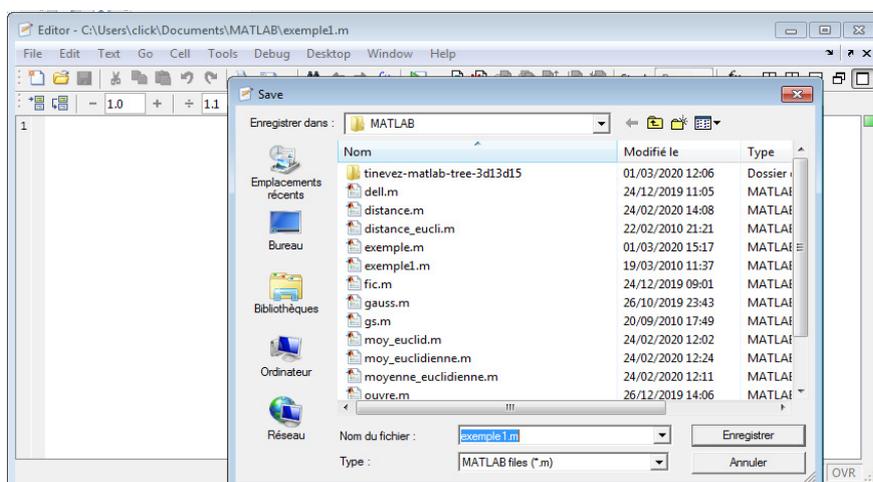
**Figure 3.2** Fenêtre d'un fichier « script » ou « M.file ».

b. Enregistrement du fichier script : après l'ouverture de chaque fichier « M.file », on doit l'enregistrer sous Matlab de la façon suivante.

- à partir de la barre d'outils du fichier script, en cliquant sur l'icône  ,
- à partir du menu fichier en cliquant sur M-file→File→Save As (figure 3.3).
- la figure 3.4 représente la fenêtre d'enregistrement, pour enregistrer le script sous le nom « exemple.m ».



**Figure 3.3** Enregistrement d'un fichier script.



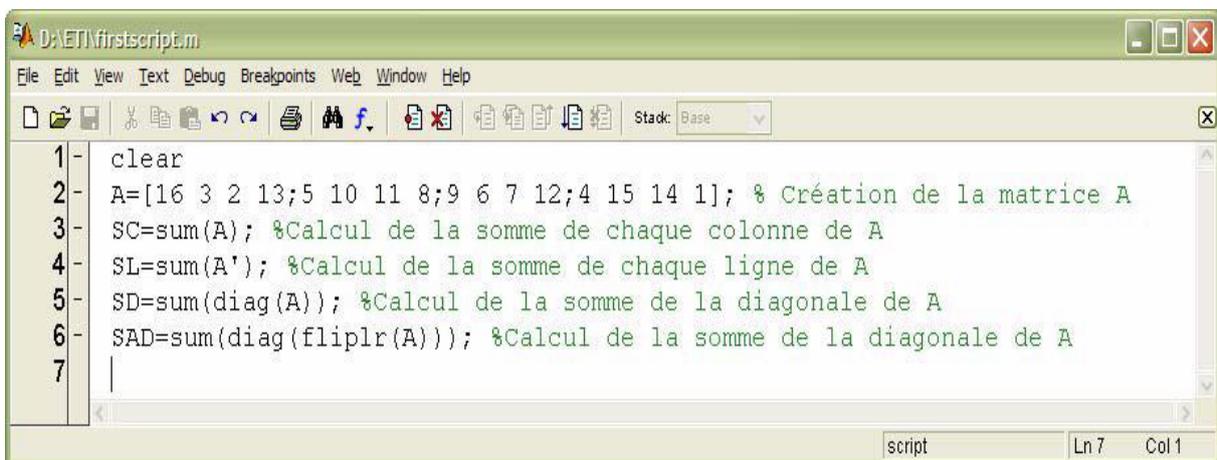
**Figure 3.4** Une fenêtre d'enregistrement.

c. Exécution d'un fichier script : Après l'enregistrement du fichier script, on peut l'exécuter de la façon suivante

- à partir de la barre d'outils du fichier script, en cliquant sur l'icône ,
- Si le script est écrit dans le fichier de nom **exemple.m** on l'exécute dans la fenêtre MATLAB en tapant son nom dans la fenêtre de commande (>> **exemple.m** et valider).

### Exemple 3.1:

1. Ouvrez l'éditeur de texte (ou de script) de Matlab. Reproduire alors le script présenté dans la fenêtre 3.5 ci-dessous.
2. Enregistrer le dans votre répertoire sous le nom « exemple.m ».
3. Exécuter le et afficher les résultats obtenus, c.-à-d. les valeurs des SC, SL, SD et SAD.



```
D:\ETI\firstscript.m
File Edit View Text Debug Breakpoints Web Window Help
[Icons] Stack: Base
1- clear
2- A=[16 3 2 13;5 10 11 8;9 6 7 12;4 15 14 1]; % Création de la matrice A
3- SC=sum(A); %Calcul de la somme de chaque colonne de A
4- SL=sum(A'); %Calcul de la somme de chaque ligne de A
5- SD=sum(diag(A)); %Calcul de la somme de la diagonale de A
6- SAD=sum(diag(fliplr(A))); %Calcul de la somme de la diagonale de A
7- |
script Ln 7 Col 1
```

Figure 3.5 Exemple 3.1.

### Remarque :

- Il est important de commencer un programme par l'instruction **clear**. Cette instruction effacera toutes les variables se trouvant dans l'espace. Ainsi, toutes les variables seront créées par le présent programme.
- Il est important de commenter abondamment un programme. Ceci permet de comprendre le programme lorsqu'on a besoin de le réutiliser après une longue période. Dans Matlab, une ligne commentaire commence par « % ».

## 3.3 Entrée et Sorties

### 3.3.1 Lecture des données (les entrées)

La commande « input » permet de demander à l'utilisateur d'un programme de fournir des données.

#### Syntaxe :

Variable=input('une phrase indicative') ;

Variable : une valeur déposée par l'utilisateur sera mise dans cette variable.

Input : une commande matlab permet de lire une valeur donnée par l'utilisateur.

### Exemple 3.2:

```
>> A=input('entrer une valeur');
    entrer une valeur 5
    |
>> B=input('entrer un vecteur');
    entrer un vecteur [1 2 3]
>> B

B =

     1     2     3
```

### 3.3.2 Ecriture des données (les sorties)

La commande « disp » permet d'afficher un tableau de valeurs numériques ou de caractères. L'autre façon d'afficher un tableau est de taper son nom. La commande « disp » se contente d'afficher le tableau sans écrire le nom de la variable ce qui peut améliorer certaines présentations.

Syntaxe :

Disp(objet)

- disp : commande matlab permet d'afficher à l'écran un objet.
- Objet : peut être un nombre, un vecteur, une matrice, une chaîne de caractère ou une expression.

### Exemple 3.3:

```
>> A %affichage de la valeur de A

A =

     5

>> disp(A)
     5

>> disp(B)
     1     2     3
```

## 3.4 les expressions logiques

### 3.4.1 les opérations de comparaison

L'opération de comparaison	signification
==	L'égalité
~=	L'inégalité
<, >, <=, >=	Inférieure, supérieure, inférieur égale, supérieur égale.

### 3.4.2 les opérations logiques

L'opération de comparaison	signification
&	Le <u>et</u> logique
	Le <u>ou</u> logique
~	La <u>négarion</u> logique

### 3.4.3 Fonctionnement des opérations logiques

- Toute valeur égale à 0 sera considérée comme fausse.
- Toute valeur différente de 0 sera considérée comme vraie.

a	b	a&b	a   b	~a
1 (vraie)	1 (vraie)	1	1	0
1 (vraie)	0 (faux)	0	1	0
0 (faux)	1 (vraie)	0	1	1
0 (faux)	0 (faux)	0	0	1

#### Exemple 3.4 :

```
>> x=10;
>> y=20;
>> x>y % affiche la valeur 0 (faux)
```

```
ans =
```

```
0
```

---

```
>> x==y
```

```
ans =
```

```
0
```

```
>> (x>9)&(y>10) % vraie et vraie le résultat 1 (vraie)
```

```
ans =
```

```
1
```

### 3.4.4 comparaison des matrices et des vecteurs

La fonction	Description
isequal	Teste si deux (ou plusieurs) matrices sont égales (ayants les mêmes éléments partout). Renvoie 1 si c'est vraie et 0 sinon.
isempty	Teste si une matrice est vide ou non. Renvoie 1 si la matrice est vide et 0 sion.

### Exemple 3.5:

```
>> A=[1 2 3];
>> B=[1 2 3];
>> isequal(A,B)
```

```
ans =
```

```
1
```

---

```
>> C=[1 3 3];
>> A==C %comparer les éléments des deux vecteurs
```

```
ans =
```

```
1 0 1
```

---

```
>> D=[];
>> isempty(D)
```

```
ans =
```

```
1
```

## 3.5 Instructions de contrôle

### 3.5.1 Les instructions « if », « else » et « elseif »

Syntaxes:

Instruction "if"	Instruction "else"	Instruction "elseif"
<b>if</b> (condition)  Instructions.....  <b>end</b>	<b>if</b> (condition)  Instructions.....  <b>else</b> Instructions ....  <b>end</b>	<b>if</b> (condition 1) Instructions..... <b>elseif</b> (condition 2) Instructions..... <b>elseif</b> (condition n) Instructions ..... <b>else</b> Instructions ....  <b>end</b>

### Exemple 3.6:

Ecrire un programme sous matlab qui vérifie si un nombre donnée par l'utilisateur est divisible par la valeur 7 ou non. (Utiliser la fonction « mod »)

Solution :

```

1 - a=input('entrer une vleur:');
2 - if mod(a,7)==0
3 -     disp('a est divisible par 7');
4 - else
5 -     disp('a n'est pas divisible par 7');
6 - end
7 - |
    
```

Figure 3.6 Solution de l'exemple 3.6

Exécution :

```

entrer une vleur:6
a n'est pas divisible par 7
    
```

### 3.6 les boucles

#### 3.6.1 la boucle for

Une boucle est une structure qui permet d'exécuter un certain nombre de fois un même bloc d'instructions.

- On peut créer une boucle en utilisant **for ... end**.
- On peut aussi réaliser des boucles **for** imbriquées.

Syntaxe :

Création d'une seule boucle for	Création de boucles for imbriquées
<pre> <b>for</b> (variable= expression_vecteur)     groupes d'instructions <b>end</b>                     </pre>	<pre> <b>for</b> (variable1= expression_vecteur)     <b>for</b> (variable2= expression_vecteur)         groupes d'instructions     <b>end</b> <b>end</b>                     </pre>

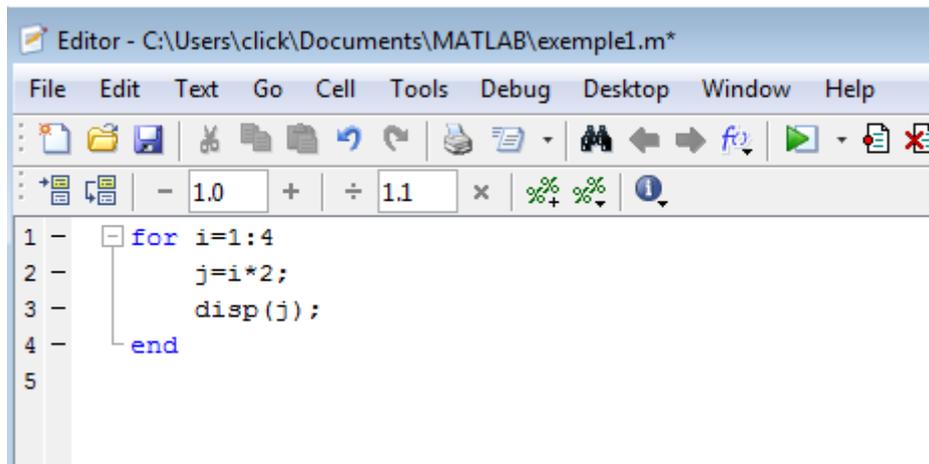
Remarque :

- Expression\_vecteur : correspond à la définition d'un vecteur utilisé de la façon suivante (début : pas : fin).
- Variable : on l'appelle aussi indice. Il parcourt tous les éléments du vecteur défini par « expression\_vecteur », où pour chaque indice on exécute un groupe d'instructions.

### Exemple 3.7:

Ecrire un programme matlab qui permet de calculer l'expression suivante «  $j=i*2$  », sachant que la variable  $i$  représente l'indice allant de la valeur 1 jusqu'à la valeur 4, ensuite afficher la valeur de la variable  $j$ .

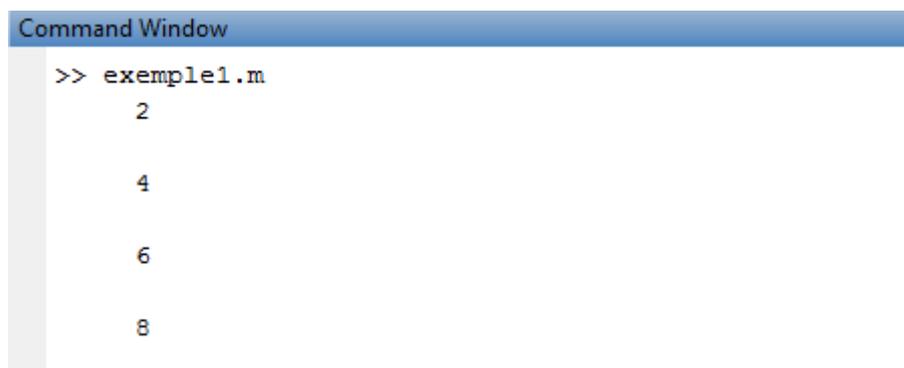
### Solution :



```
Editor - C:\Users\click\Documents\MATLAB\exemple1.m*
File Edit Text Go Cell Tools Debug Desktop Window Help
+ - 1.0 + ÷ 1.1 × % %
1 - for i=1:4
2 -     j=i*2;
3 -     disp(j);
4 - end
5
```

**Figure 3.7** Solution de l'exemple 3.7

### Résultat d'exécution :



```
Command Window
>> exemple1.m
2
4
6
8
```

**Figure 3.8** Résultat d'exécution de l'exemple 3.7

### Exemple 3.8:

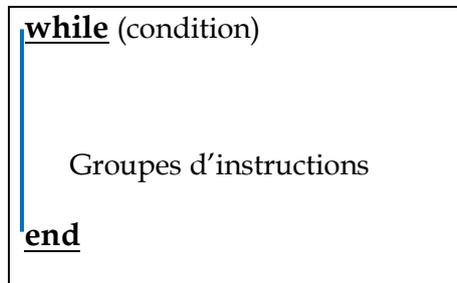
Ecrire un programme matlab qui permet de calculer la somme des éléments d'une matrice donnée par l'utilisateur.

### Solution :



On peut créer une boucle en utilisant **while ... end**.

Syntaxe :



Exemple 3.9:

Ecrire un programme qui calcule le n factoriel

Solution :

```
Editor - C:\Users\click\Documents\MATLAB\exemple1.m
File Edit Text Go Cell Tools Debug Desktop Window Help
- 1.0 + 1.1 x %>% %>%
1 - n=input('entrer la valeur de n:');
2 - m=1;
3 - i=1;
4 - while (i<=n)
5 -     m=m*i;
6 -     i=i+1;
7 - end
8
9 - disp(m);
```

**Figure 3.11** Solution de l'exemple 3.9

Résultat d'exécution :

```
Command Window
>> exemple1.m
entrer la valeur de n:4
24
```

**Figure 3.12** Résultat d'exécution de l'exemple 3.9

Table d'exécution (exemple 3.9) pas à pas :

<b>n</b>	<b>i</b>	<b>I&lt;=n</b>	<b>m</b>	<b>disp(m)</b>
4	1	vraie	1*1=1	
	2	vraie	1*2=2	
	3	vraie	2*3=6	
	4	vraie	6*4=24	
	5	faux		
				24

## Références

[1] : Abdellah MECHAQRANE, (2008) Introduction à Matlab et simulation. Université Sidi Mohammed Ben Abdallah, Faculté Des Sciences et Techniques Fès, Département Génie Electrique.

[2] : John Chaussard, (2017), Introduction à Matlab, Ecole Sup Galilée - Coursus Ingénieur - 1ère année.

[3] : Yassine Ariba - Jérôme Cadieux, Manuel Matlab, Départements GEI & Mécanique, Icam de Toulouse.

[4] : Matlab help