Faculté des sciences

Département de mathématiques

Master 1: MF/MSS

Module : Apprentissage logiciel/LCS

1. Introduction

- Qu'est-ce que MATLAB ?

C'est langage de développement informatique dédié aux applications scientifiques, c'est un logiciel de calcul scientifique.

C'est un langage interprété qui ne demande pas une compilation.

C'est un langage développé par la société mathworks, l'appellation MATLAB = MATrix LABoratory.

C'est un laboratoire pour manipuler des matrices. Il se base sur la notion des matrices, car chaque problème pour MATLAB est traduit sous forme de matrices ou bien des tableaux.

- Avantages et inconvénients : comme tout langage de développement, on trouve des avantages et des inconvénients
  - ➢ Avantages :
    - Facilité d'utilisation, prise en main rapide ;
    - Existence de toolboxes utiles pour l'ingénieur (bibliothèque qui facilite les travaux de modélisation, optimisation, …) ;
    - Possibilité de l'interfacer avec d'autres langages (C, C++, …) ;
    - Permet de faire du calcul parallèle.
  - ➢ Inconvénients :
    - Limitation en mémoire ;
    - Ce n'est pas un programme open source (payant).
- Les domaines d'application :

- Le calcul de probabilités ou des statistiques ;
- Le calcul intégral ou la dérivation ;
- Le traitement du signal ;
- L'optimisation ;
- Le traitement d'image ;
- L'automatisme ;
- IOT (Internet Of Things)…

Environnement MATLAB

1- Editeur (Workspace) : Ecrire le code ou le script ;
2- Commande Window : voir l'exécution ;
3- Commande history : voir l'historique des variables et du traitement ;
4- Current folder: Donne l'herborescence des fichiers MATLAB.

Particularités du langage MATLAB

MATLAB est un langage interprété, c'est-à-dire :

- Les commandes tapées dans la fenêtre de commande sont exécutées immédiatement ;
- Les variables sont déclarées en mémoire dès leurs apparitions dans une expression ;
- Les variables peuvent être réassignées ;
- Les variables peuvent être effacées sélectivement du workspace.

L'extension du fichier de données est **.m**

2. Un peu de pratique

Ligne de commandes : il existe deux types de commandes

- **Expression** : formule permettant de calculer immédiatement un résultat

```
>> 5*3
ans =  15
>> pi
ans =  3.1416
```

Remarque : tous les éléments de l'expression doivent être connus au moment de son évaluation par l'interpréteur.

- **Instruction :** ensemble structuré d'expressions ou d'affectations

```
x=pi;
y=x/2;
z=sin(y);
```

Help(nom de la fonction) : permet de donner une documentation sur la fonction recherché.

Pour effacer l'écran : **clc**

- **Input (entrées)/output (sorties)**

- **Input (' ') :** Introduire une ou des variables par l'intermédiaire du clavier

```
X=input('text')
Chaine_caractère = input('text', 's')
```

- **disp (' ') :** permet d'afficher un tableau de valeurs numériques ou de caractères

```
disp(variable)
disp(['un message :', num2str(variable)])
```

Num2str : permet de modifier le type d'un numéro à une chaine de caractère

Exemple : écrire un programme qui demande 2 variables a et b et qui calcul leur somme « S ».

D'abord on doit enregistrer le fichier que l'on nomme **somme.m**, après l'enregistrement on écrit le programme sur workspace :

```
a=input('donner la valeur de a:');

b=input('donner la valeur de b:');

S=a+b;

disp(S) (sinon utiliser disp(['la somme est:',num2str(S)]); pour l'affichage)
```

On appelle le programme par le nom du fichier (dans window de commande), on écrit le nom du fichier **somme** et on clic sur entrée.

- **sprintf(' ') :**

Sprintf(format, variable)

➔ **Modèle d'édition de caractères :** format = %Ls

% est le symbole de début de format

s le symbole précisant que la donnée est de type chaine de caractères

L est un entier donnant la longueur total du champ (en nombre de caractère)

Exemple :

```
>> sprintf('%s','bienvenu dans ce cours')
ans = bienvenu dans ce cours
```

➔ **Modèle d'édition des réels :** format = %L.Dt
% est le symbole de début de format
L est un entier donnant la longueur total du champ (en nombre de caractères, point-virgule comprise)
D est le nombre de décimales à afficher
t spécifie le type de notation utilisée

Les principales valeurs possibles pour **t** sont les suivantes :

d : pour les entiers

e : pour une notation à virgule flottante où la partie exposant est délimitée par un e minuscule (exp : 3.1415e+00)

E : même notation mais E remplace e (exp : 3.1415E+00)

f : pour une notation à virgule fixe (exp : 3.1415)

g : la notation la plus compacte entre la notation à virgule flottante et la notation à virgule fixe est utilisée.

Exemple :

```
>> sprintf('sin(%8.6f)=%4.2f',x,y)
ans = sin(1.047198)=0.87
```

3. Script et fonction

Il est utile d'utiliser deux types de fichiers (qui portent l'extension **.m**)

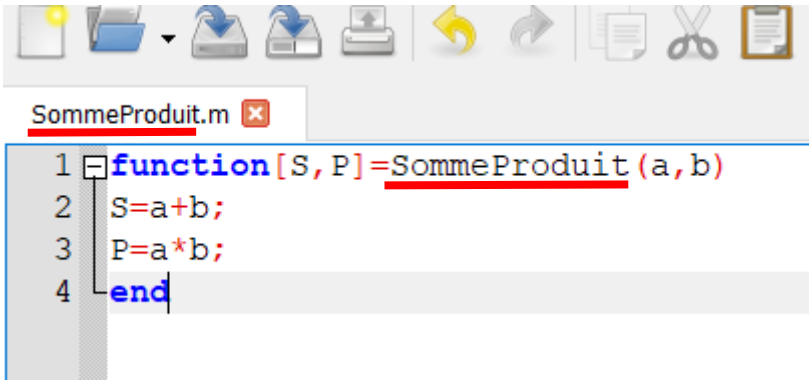
➔ **Script M-files** : ni entrée, ni sortie et utilise les variables de l'espace de travail.
➔ **Fonction M-files** : contient une fonction qui accepte des arguments en entrée et renvoie des arguments en sortie, et les variables internes sont locales à la fonction.

- **Syntaxe générale d'une fonction** :

```
Function [Y1,…,YN] = Nom_fonction (X1,… XM)
        Les arguments              les variables d'entrées
% le corps de la fonction (instructions + variables locales)
end
```

Exemple de script : comme celui de calcul de la somme a+b, si on veut l'appeler, on écrit le nom du fichier dans command window et puis entrée.

Exemple de fonction : on ouvre un nouveau script, écrire la fonction qui calcul la somme et le produit de deux nombres.



Le plus important c'est l'enregistrement du fichier doit porter le même nom de la fonction

4. Variables scalaires

Format numérique : commande format pour afficher des nombres en virgule fixe ou flottante.

Prenons l'exemple :

```
>> pi^4
ans =  97.409
>> format short, pi^4 %fixe, 5 chiffres
ans =  97.409
>> format short e, pi^4 %flottante, 5 chiffres
ans =    9.7409e+01
>> format long, pi^4 %fixe, 15 chiffres
ans =  97.409091034002430
>> format long e, pi^4 %flottante, 15 chiffres
ans =    9.740909103400243e+01
```