

# Informatique 3

*Univ-dbkm*  
*AZIZOU Fethi*



# Table des matières



|  |          |
|--|----------|
| <b>Objectifs</b>                                 | <b>3</b> |
| <b>I - Chapitre 3 : Programmer sous MATLAB</b>   | <b>4</b> |
| 1. Objectifs spécifiques .....                   | 4        |
| 2. Instructions de contrôle .....                | 4        |
| 2.1. Boucle FOR : parcours d'un intervalle ..... | 4        |
| 2.2. Boucle While : tant que . . . faire .....   | 5        |
| 2.3. L'instruction conditionnée IF .....         | 5        |
| 3. Ecrire des scripts M-files .....              | 5        |
| 4. Les fonctions .....                           | 6        |

# Objectifs

Apprendre à l'étudiant la programmation en utilisant des logiciels faciles d'accès (essentiellement : *Matlab*, *Scilab*, *Maple* ...). Cette matière sera un outil pour la réalisation des TP de méthodes numériques en S4



- borne\_inf et borne\_sup sont deux constantes réelles (appelées paramètres de la boucle).
- Séquence d'instructions est le traitement à effectuer pour les valeurs d'indices variant entre borne\_inf et borne\_sup avec un incrément de 1. On parle du corps de la boucle.

## 2.2. Boucle While : tant que . . . faire

Une seconde possibilité pour exécuter une séquence d'instructions de manière répétée consiste à effectuer une boucle tant qu'une condition reste vérifiée. On arrête de boucler dès que cette condition n'est plus satisfaite. Ce processus est mis en œuvre par la « boucle while ».

```
1 while  expression logique
2     séquence d'instructions
3 end
```

Où

- Expression logique est une expression dont le résultat peut être vrai ou faux.
- Séquence d'instructions est le traitement à effectuer tant que expression logique est vraie

## 2.3. L'instruction conditionnée IF

On a parfois besoin d'exécuter une séquence d'instructions seulement dans le cas où une condition donnée est vérifiée au préalable. Différentes formes d'instruction conditionnée existent sous Matlab. L'instruction conditionnée la plus simple a la forme suivante :

```
1 if  expression logique
2     séquence d'instructions
3 end
```

Où

- Expression logique est une expression dont le résultat peut être vrai ou faux ;
- Séquence d'instructions est le traitement à effectuer si expression logique est vraie.

Il existe une séquence conditionnée sous forme d'alternatives :

```
1 If  expression logique
2     séquence d'instructions 1
3 else
4     séquence d'instructions 2
5 end
```

Où

- Séquence d'instructions 1 est la séquence d'instructions à exécuter dans le cas où expression logique est vraie et séquence d'instructions 2 est la séquence d'instructions à exécuter dans le cas où expression logique est faux.

## 3. Ecrire des scripts M-files

Un fichier de script est un fichier externe contenant une suite d'instruction MATLAB. Les fichiers de script ont une extension de nom de fichier `.m`. Les M-files peuvent être des scripts qui exécutent simplement une suite d'instructions ou peuvent être des fonctions.

[cf. ]

### Exemple : Exemple

---

créer le script "test script" (soit vous tapez `>> edit test_script.m`, soit vous faites 'File'->'New'->'Script' puis 'Save As' en spécifiant "test script.m" comme nom) avec la suite d'instructions suivante :

```
1 clear all
2 a = 1
3 b = 2;
4 c = 3, d = 4;
5 e = a*b/(c+d),
6 scal = 11;
```

Sauvegardez puis exécutez le script (menu Debug->Save & Run ou Fleche verte ou F5). Observez la sortie sur la ligne de commande et conclure quant à l'utilisation des ; et ,

## 4. Les fonctions

Tout comme pour les scripts, il est possible de créer ses propres fonctions au sein d'un M-file. Contrairement aux scripts, les fonctions peuvent accepter des données en entrée et renvoyer des données en sortie. Chaque fonction a son propre espace en mémoire et ne voit pas les données du *workspace*. Seules les données qu'on aura spécifiées comme arguments d'entrée seront utilisables par la fonction.

On définit la fonction *fonc* de la manière suivante :

```
1 function[vars1,...,varsn]=fonc(vare1,...,varem)
2     séquence d'instructions
```

Le fichier doit impérativement commencer par le mot-clé `function`. Suit entre crochets les variables de sortie de la fonction, le symbole `=`, le nom de la fonction et enfin les variables d'entrée entre parenthèses. Si la fonction ne possède qu'une seule variable de sortie, les crochets sont inutiles. Il est impératif que la fonction ayant pour nom `fonc` soit enregistrée dans un fichier de nom `fonc.m` sans quoi cette fonction ne sera pas « visible » par matlab.

[cf. ]

### Exemple

---

```
1 function[mu,minv,maxv]=myMean( tab )
2 % MYMEAN(TAB) calcul de moyenne sur % un tableau 1D
3 % Pour un tableau 'tab' à 1 dimension %[mu,minv,maxv] = myMean( tab )
4 %retourne la moyenne 'mu', le minimum '%minv' le maximum 'maxv' des %éléments de
  'tab'.
5 mu = 0;
6 for i = 1:length(tab)
```

```
7 mu = mu + tab(i);
8 end
9 mu = mu / length(tab);
10 minv = min(tab);
11 maxv = max(tab);
```

5.

### Exercice 1 :

---

Créez un vecteur V contenant les entiers pairs de 0 à 42, puis écrivez un script qui parcourt le vecteur et remplace chacune des valeurs par son carré.

### Exercice 2 :

---

On définit  $n!$  comme étant le produit des entiers inférieurs à  $n$  et strictement positifs. Écrivez les instructions Matlab permettant de calculer  $\ln(57!)$ .

### Exercice 3 :

---

Écrivez une fonction de conversion de degrés Fahrenheit vers des degrés Celsius. (Rappel :  $C = 5/9(F - 32)$ )

### Exercice 4 :

---

Écrivez une fonction prenant 2 arguments d'entrée et 2 arguments de sortie qui permettra de déterminer la taille en cm et le poids en kg d'une personne à partir de sa taille en pouces (1 pouce = 2.54 cm) et son poids en pound (1 lb = 0.453 kg).