


TP Informatique 03

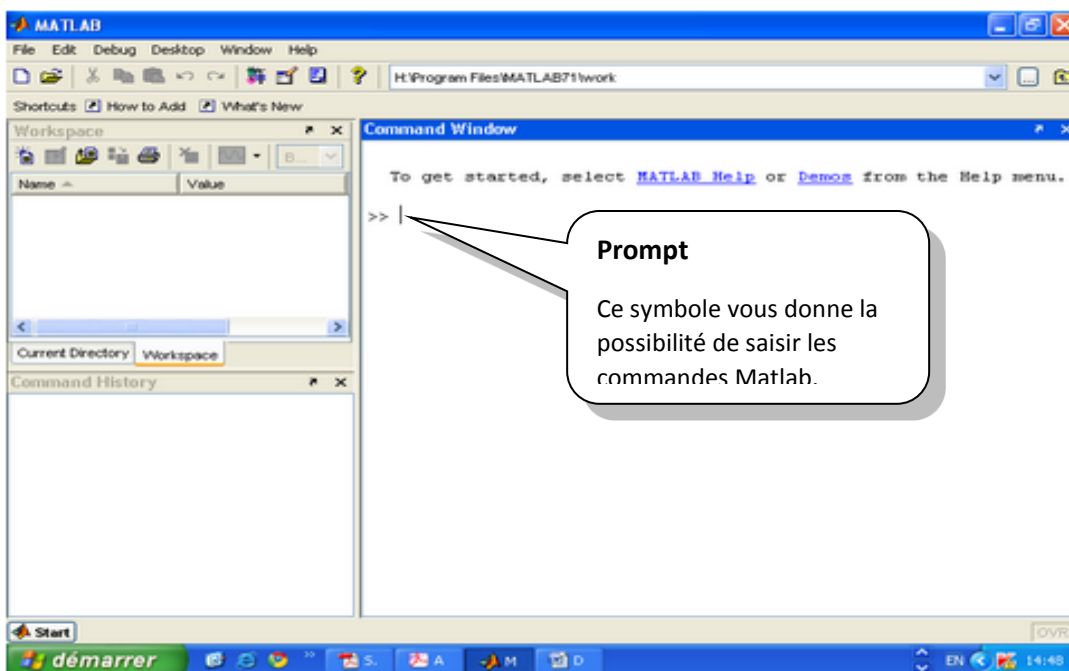
TP N°01: Initiations à Matlab

1. Introduction

« Matlab » est un logiciel conçu pour les calculs numériques. Le nom « Matlab » est l'abréviation des deux mots « Matrix Laboratory ». Comme ce nom indique, Matlab est basé sur le traitement des matrices. Autrement dit, Matlab manipule les données, les variables, les image, comme étant des matrices. Ce logiciel est développé par la société américaine « math works » (le site www.mathworks.com). Le logiciel Matlab permet d'analyser les données, de développer les algorithmes, de faire les calculs mathématiques, de faire des simulations et des modélisations des différents systèmes et de représenter graphiquement les résultats obtenus.

2. Démarrage du « Matlab »

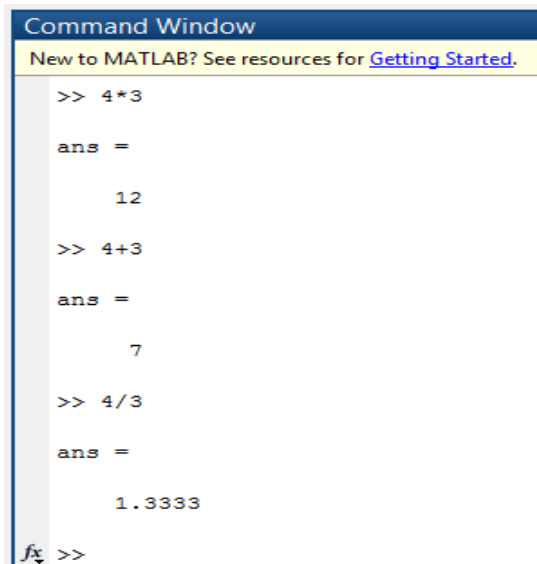
Pour démarrer « Matlab » il suffit un double clic sur l'icône Matlab . Une fenêtre appelée « fenêtre de commandes » (anglais : command Windows) apparaît (figure 1). Cette fenêtre est composée des champs *Fenêtre de commande*, *Espace de travail*, *Fenêtre d'historique de commandes*.



3. Opérations arithmétiques

Matlab permet de faire les différentes opérations arithmétiques telles que l'addition, la soustraction, la multiplication et la division (+, -, *, /). Nous allons nous intéresser actuellement aux des matrices 1x1, c'est à dire des nombres.

Exemples :



```
Command Window
New to MATLAB? See resources for Getting Started.

>> 4*3
ans =
    12

>> 4+3
ans =
     7

>> 4/3
ans =
    1.3333

fx >>
```

Remarque :

Matlab fait le calcul et affiche le résultat qui est automatiquement affecté à la variable spécifique « **ans** » qui est, ce qu'on appelle, une variable temporaire.

On appelle l'écriture « $2+3$ » *une commande* ou *une expression Matlab*.

4. Manipulation des variables

On va maintenant travailler avec des variables définies par l'utilisateur :

```
>> x = 2 ; y = 1.5 ;
```

Somme = x+y

Somme = 3.5000 Le résultat est affecté et stocké dans la variable **Somme**

Différence = x-y

Différence = 0.5000

Produit = x*y

Produit = 3

Division = x/y

Division = 1.3333

De même : » **Somme=Somme*3**

Effectue le produit du contenu de la variable **Somme** et du nombre **3** et sauvegarde le résultat dans **Somme**.

tapez la commande suivante :

```
» b=3 ;
```

Le rôle du point-virgule à la fin de chaque ligne de commande est de masquer le résultat de l'instruction de la ligne de commande dans le command window (ECHO ECRAN).

5. Les vecteurs

On définit un vecteur ligne en donnant la liste de ses éléments entre crochets ([]) par exemple

```
vec = [1 2 4 7 9 2.3]
```

vec =

1.0000 2.0000 4.0000 7.0000 9.0000 2.3000

Les éléments sont séparés au choix par des espaces ou par des virgules.

On définit un vecteur colonne en donnant la liste de ses éléments séparés par des points virgules (;)

par exemple

```
>> col = [1; 2; 4; 7]
```

```
col =
```

```
1
```

```
2
```

```
4
```

```
7
```

On peut concaténer deux vecteurs :

```
>> vec1 = [1 3 5];
```

```
>> vec2 = [9 10 11];
```

```
>> vec = [vec1 vec2]
```

```
vec =
```

```
1 3 5 9 10 11
```

Et on peut également prendre la transposée pour passer d'une ligne à une colonne ou réciproquement :

```
>> vec1 = [1 3 5];
```

```
>> vec = vec1'
```

```
vec =
```

```
1
```

```
3
```

```
5
```

Il n'est pas nécessaire de définir la taille d'un vecteur (c'est automatique), par contre la commande `length()` permet de retourner cette quantité.

```
>> length(vec)
```

```
ans = 3
```

Une autre méthode pour générer des vecteurs dont les éléments sont à intervalle régulier consiste à utiliser `[a : s : b]`. On crée alors un vecteur entre *a* et *b* avec un espacement ou pas *s* :

```
>> vec = [1: 2 : 9]
```

```
vec =
```

```
1 3 5 7 9
```

En utilisant la commande `linspace(a, b, N)`, un vecteur de *N* éléments entre la valeur *a* et *b* sera générer.

```
>> vec = linspace(1, 5, 8)
```

```
vec =
```

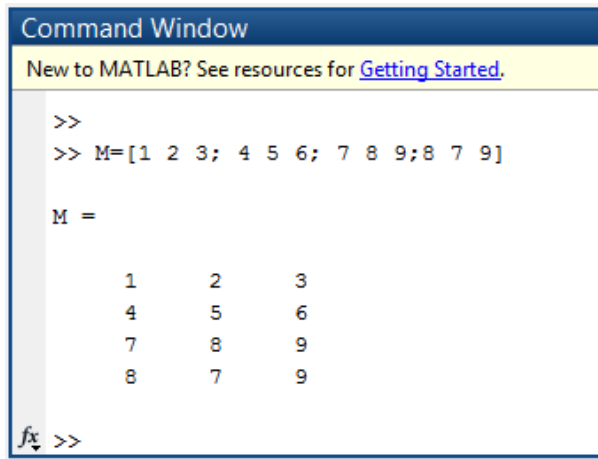
```
1.0000 1.5714 2.1429 2.7143 3.2857 3.8571 4.4286 5.0000
```

```
>> vec(3 : 6)
```

```
2.1429 2.7143 3.2857 3.8571
```

6. Les matrices

Afin de créer une matrice dans Matlab, il faut suivre les étapes suivantes :



```
Command Window
New to MATLAB? See resources for Getting Started.
>>
>> M=[1 2 3; 4 5 6; 7 8 9;8 7 9]

M =

     1     2     3
     4     5     6
     7     8     9
     8     7     9

fx >>
```

Une matrice est composée de m lignes et n colonnes. Si on souhaite connaître la valeur de m ou n, on utilise la commande size(A).

```
>> A = [1 2 5 ; 4 3 6]
A =
     1     2     5
     4     3     6
```

```
>> [m n] = size(A)
m = 2
n = 3
```

7. Les matrices particulières

Il y a des matrices particulières qu'il est très utile de connaître

1- **La matrice zéro** : on crée une matrice (r x c) de 0 en utilisant l'instruction zeros(r, c). Par exemple

```
>> A=zeros(2,3)
```

```
A =
     0     0     0
     0     0     0
```

2- **La matrice de UN** : on crée une matrice (r x c) de 1 en utilisant l'instruction ones(r, c). Par exemple

```
>> A=ones(2,3)
```

```
A =
     1     1     1
     1     1     1
```

3- **La matrice Identité** : on crée une matrice identité (r x r) en utilisant l'instruction eye(r). Par exemple

```
>> A=eye(3)
```

```
A =
     1     0     0
     0     1     0
     0     0     1
```

4- On peut créer une matrice de contenu aléatoire dans les valeurs ses des éléments sont comprises entre 0 et 1.

```
>> rand(3,3)
```

```
A =
     0.0114     0.1434     0.2577
     0.5645     0.1129     0.9742
     0.9878     0.7811     0.2192
```