

Chapitre 1 : Présentation et généralités

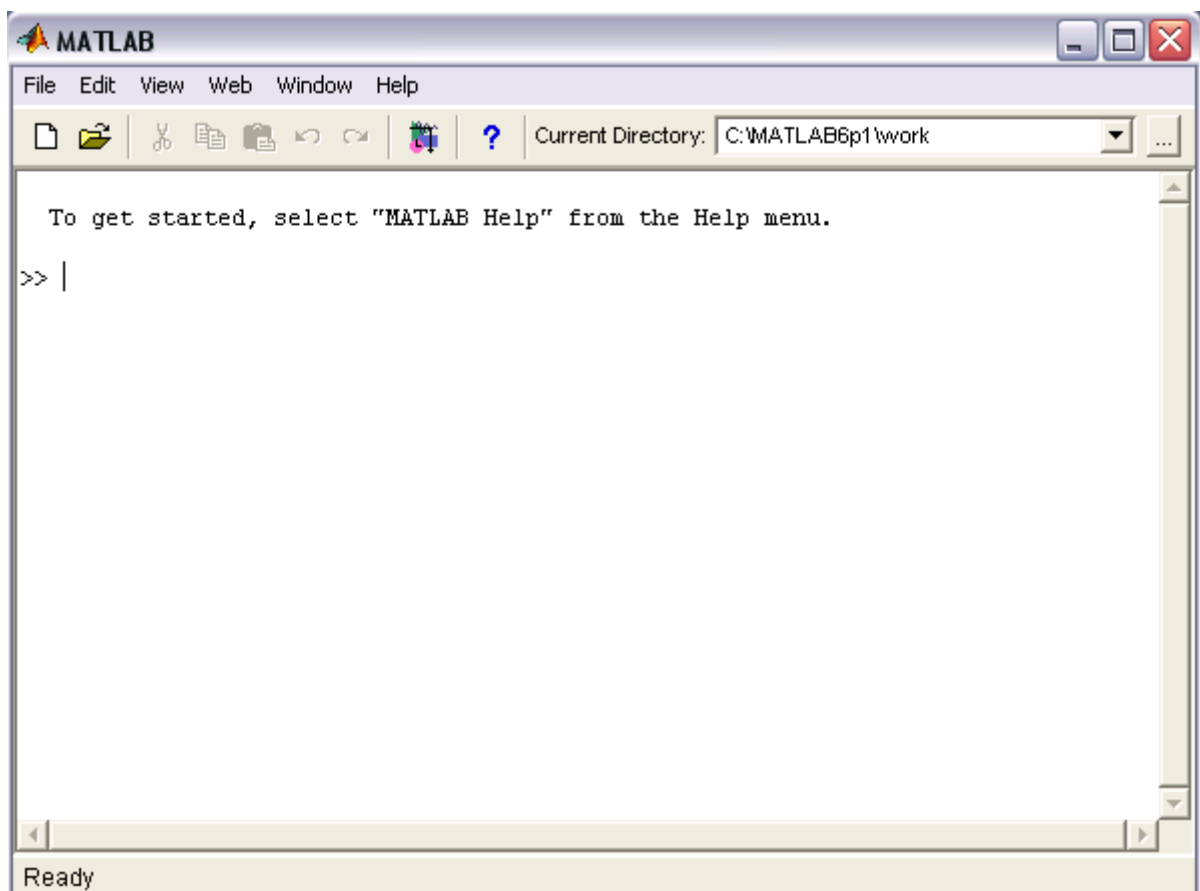
1- Introduction :

Matlab est un logiciel de calcul numérique produit par MathWorks (voir le site web <http://www.mathworks.com/>). Il est disponible sur plusieurs plateformes. Matlab est un langage simple et très efficace, optimisé pour le traitement des matrices, d'où son nom. Pour le calcul numérique, Matlab est beaucoup plus concis que les “vieux” langages (C, Pascal, Fortran, Basic). Un exemple: plus besoin de programmer des boucles pour modifier un à un les éléments d'une matrice. On peut traiter la matrice comme une simple variable. Matlab contient également une interface graphique puissante, ainsi qu'une grande variété d'algorithmes scientifiques.

On peut enrichir Matlab en ajoutant des “boîtes à outils” (toolbox) qui sont des ensembles de fonctions supplémentaires, profilées pour des applications particulières (traitement de signaux, analyses statistiques, optimisation, etc.).

1.1 Une session Matlab

Pour lancer Matlab commencer par ouvrir une fenêtre de commande Matlab



Le prompt Matlab (») indique que Matlab attend des instructions. Voici un exemple de session Matlab

```
» A=2 (après retour chariot voici ce q'on va obtenir)
A=
    2
```

Remarque : quand une instruction comporte une variable = à une expression alors l'affichage du résultat est la même variable = au résultat. Si l'instruction est seulement le calcul d'une instruction alors l'affichage du résultat est **ans** = au résultat.

```
exemple : » B=2*A+1  ⬅
           B =
           5

           >> 3*A  ⬅
           ans =
           6
```

Chaque ligne d'instruction doit se terminer par un retour chariot (validation). La commande pour quitter Matlab est **quit.**

1.2 L'espace de travail

Matlab permet de définir des données variables. Les variables sont définies au fur et à mesure que l'on donne leurs noms (identification) et leurs valeurs numériques ou leurs expressions mathématiques. Matlab ne nécessite pas de déclaration de type ou de dimension pour une variable (tableau...).

Voici quelques commandes pour faciliter la programmation :

who : fournit la liste des variables définies dans l'espace de travail (workspace).

whos : donne plus d'informations sur les variables.

clear : efface les variables du workspace. Il est possible de ne détruire qu'une partie des variables en tapant **clear** liste de noms de variables.

clc : efface l'écran.

Exemple :

```
>> x=2;y=x*x;z=y/4;
>> A=[1 5; 5 8];B=A*A;
>> t='bonjour';

>> who
Your variables are:
A B t x y z

>> whos
```

```
>> clear x y t
>> who
Your variables are:
A B z

>> clear
>> who
>>
```

Name	Size	Bytes	Class
A	2x2	32	double array
B	2x2	32	double array
t	1x7	14	char array
x	1x1	8	double array
y	1x1	8	double array
z	1x1	8	double array

Grand total is 18 elements using 102 bytes

1.3 Obtenir de l'aide

Pour obtenir de l'aide on utilise la fonction help suivie du nom de la fonction.

Exemple :

```
>> help clear
```

```
CLEAR Clear variables and functions from memory.
CLEAR removes all variables from the workspace.
CLEAR VARIABLES does the same thing.
CLEAR GLOBAL removes all global variables.
CLEAR FUNCTIONS removes all compiled M- and MEX-functions.
```

D'autres exemples seront traités en TP

1.4 Syntaxe d'une ligne d'instruction

- Si une instruction est suivie d'un point virgule (;) le résultat de cette instruction n'est pas affiché.
- Pour re-afficher un résultat contenu dans une variable il suffit de taper le nom de la variable.
- Le résultat de la dernière instruction exécutée peut être rappelé par la commande ans.

Exemple:

```
>> an = 2012 ↵
an=
    2012
>> jour = 30 ; ↵
>>
>> jour ↵
jour =
    30
```

```
>> B = 3*16 ; ↵
>>
>> B ↵
B =
    48
>> 2*6 ↵
ans=
    12
>> B+an ↵
ans=
   2060
```

- Plusieurs instructions Matlab peuvent figurer sur une même ligne. Il faut les séparer par une virgule ou par un point virgule.

- Si une instruction est précédée du symbole % l'instruction est ignoré par Matlab il l'a considère comme commentaire

Exemple :

```
>> x=5 ; y=0 ; z=1 ;
```

```
>> a=3, b=-2, c=1
```

```
a =
```

```
3
```

```
b =
```

```
-2
```

```
c =
```

```
1
```

```
>> % Calcul du discriminant Delta
```

```
>> D = b*b - 4*a*c ;
```

- Si une commande est trop longue pour tenir sur une ligne, il est possible de poursuivre sur la ligne suivante en terminant la ligne par 3 points (. . .).

```
>> cout_moyen = cout ... % commande sur deux lignes  
/ nombre;
```

```
>> t=x+2*y... ⏏
```

```
+3*z-1 ⏏
```

```
t =
```

```
7
```

```
>>
```