TP : Optimisation

TP 1 : Présentation du logiciel Matlab.

1.1 Introduction :

Le mot **MATLAB** est l'abréviation du mot composé **MATrix LABoratory**. Le logiciel MATLAB a été développé par la société The MathWorks, il présente un environnement de développement dans le domaine du calcul matriciel numérique.
L'intérêt de Matlab tient, d'une part, à sa simplicité d'utilisation : pas de compilation, déclaration implicite des variables utilisées, et d'autre part, à sa richesse fonctionnelle : arithmétique matricielle et nombreuses fonctions de haut niveau dans de nombreux domaines (analyse numérique, graphique, ...). C'est un logiciel multiplate-forme, c'est-à-dire qu'on peut l'utiliser sous des différents environnements (systèmes d'exploitation), tels que : Linux, Unix, Mac OS et Windows. MATLAB est un logiciel propriétaire (non libre). On peut l'utiliser dans des différents domaines tels que les sciences physiques, le traitement de signal, l'analyse et le traitement d'images, l'analyse numérique, l'algèbre en général, l'automatique, l'informatique industrielle, la commande des systèmes, les réseaux de neurones, la logique floue, le calcul symbolique, l'aéronautique, etc. Le site web officiel de ce logiciel est : www.mathworks.com .

Le logiciel MATLAB permet de manipuler des tableaux et des matrices, d'afficher des courbes et des données, de mettre en œuvre des algorithmes, de tracer des graphiques en deux et en trois dimensions, de résoudre des équations et des polynômes, etc. Donc les travaux pratiques de MATLAB ont pour objectif, la maîtrise de cet outil qui était très puissant et très utilisable pour le calcul numérique et la visualisation graphique. Le logiciel MATLAB est construit autour du langage MATLAB. Une interface en ligne de commande, qui est un des éléments du bureau MATLAB, permet d'exécuter des commandes simples.

Ce premier TP(TP1) a pour but de vous familiariser avec l'usage de Matlab. Matlab est un langage de programmation de haut niveau doublé d'un environnement de travail. Un premier exemple introductif montre rapidement le principe de fonctionnement du logiciel. Nous présentons ensuite un ensemble de fonctions de base nécessaire pour débuter en Matlab.

Le logo du logiciel MATLAB est illustré sur la figure 1.1.

Figure 1.1 : Le Logo du logiciel MATLAB.

1.2 Interface principale du Logiciel MATLAB :

Pour lancer le programme, tapez **matlab** dans une fenêtre de commandes. Une fenêtre logo fait une brève apparition, puis dans la fenêtre de commandes, le symbole « >> » apparaît : c'est l'invité de MATLAB qui attend vos commandes.

Le logiciel propose un véritable environnement de travail sous la forme d'un espace de travail (Workspace), composé de multiples fenêtres comme illustré sur la figure 1.2.

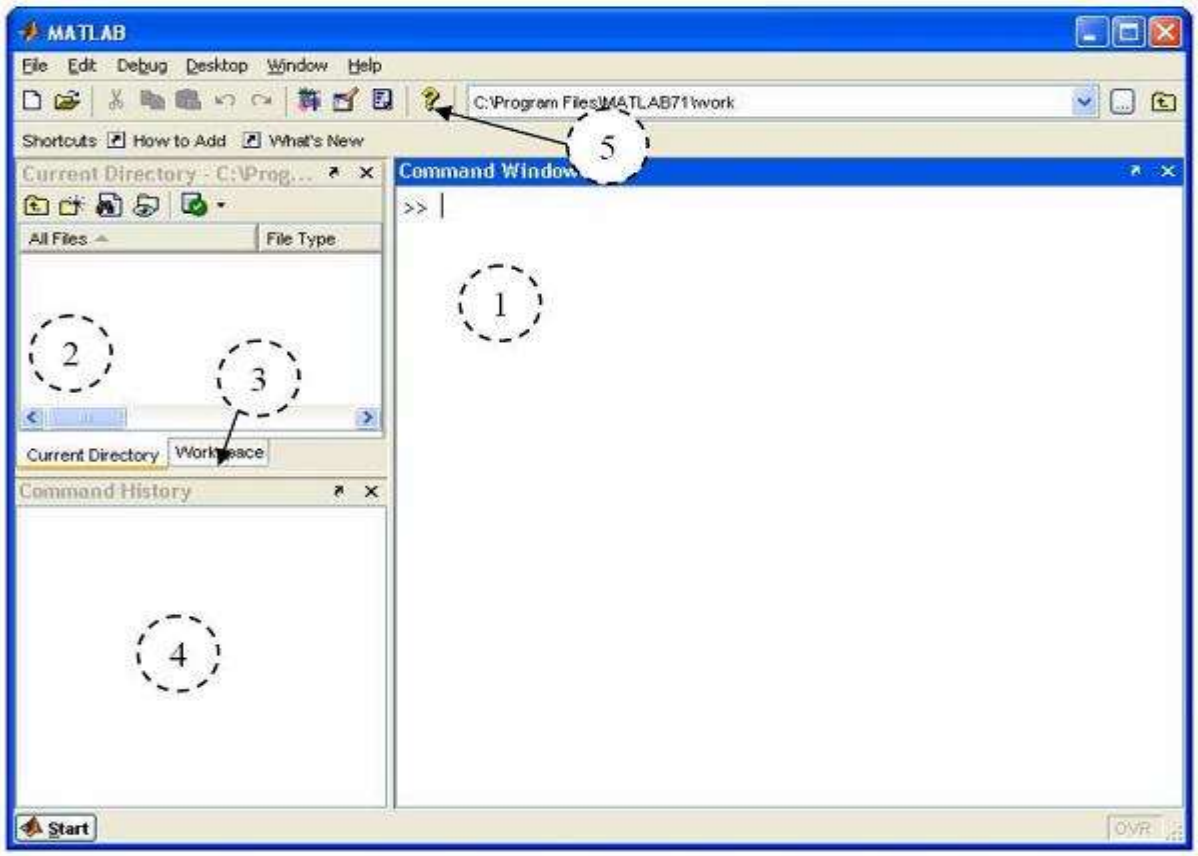

Figure 1.2 : Interface principale du logiciel MATLAB.

On peut distinguer cinq groupes réparties comme suit :

1. **Command window (console d'exécution)** : Il s'agit de la fenêtre principale de l'interface. A l'aide de la commande **« >> »,** l'utilisateur peut entrer les instructions à exécuter.

2. **Current directory (répertoire courant)** : permet de naviguer et de visualiser le contenu du répertoire courant de l'utilisateur. Les programmes de l'utilisateur doivent être situés dans ce répertoire pour être visible et donc exécutable.

3. **Workspace (espace de travail) :** permet de visualiser les variables définies, leur type, la taille occupée en mémoire…

4. **Command history** : historique des commandes que l'utilisateur a exécutées. Il est possible de faire glisser ces commandes vers la fenêtre de commande.

5. **Choix du répertoire courant.** c'est le dossier où doit se situer vos programmes (fichiers *.m). Vous pouvez mettre vos programmes dans un autre dossier mais dans ce cas il faut l'inclure dans File >> Set Path

Notons que la **console d'exécution** (**Command Window**) est la fenêtre centrale de l'interface, c'est à partir de là que l'utilisateur pourra lancer les commandes interprétées par Matlab.
Le principe est simple et intuitif, le tout est de connaitre les fonctions appropriées et de respecter leur syntaxe.

Vous pourrez quitter la session avec la commande quit.

1.3 Les variables

Dans Matlab, les variables et les scalaires sont manipulés comme des matrices (qui sont des tableaux de données à deux entrées, par exemple, une matrice avec m lignes et n colonnes est dite « de taille (m,n) »).
Par exemple, un scalaire (qui est une variable d'un seul élément) serait une matrice de 1 x 1.
Les variables sont définies avec l'opérateur d'affectaion **« = ».**

```
>> x = 35
x =
35
>> x = 'bonjour'
x =
 bonjour
```

1.4 Des Commandes Principales

- **help**

Pour plus d'informations sur une commande, on peut utiliser la commande help suivie du nom de la commande demandée :
help <command>
Exemple : help help

- **quit**

Cette commande est utilisée pour quitter MATLAB, à la fin de notre travail.

- **clc**

Pour effacer la fenêtre.

- **Clear/clear all**

Elle réinitialise l'environnement (le "workspace") en détruisant toutes les variables actives en mémoire.

- **Whos/ who**

L'ensemble des variables actives peut être consulté grâce aux commandes whos et who .

1.5 Des opérations sur les scalaires

+ : l'addition
>> x = 1+9

- :la soustraction
>> x = 6-2

*** : la multiplication**
>> x = 35*3

/ : la division
>> x = 35/5

^ : la puissance (l'exposant)
>> x = 5^2

1.6 La ligne de commande

On peut définir plusieurs variables dans une même ligne de commande. La séparation entre les variable est avec le virgule **« , ».**

> >x=50**,** y=9**,** z=x+y**,**

1.7 Le point vergule **";"**

Il faut terminer l'opération par un point-virgule **";"** sinon, toutes les étapes du calcul seront affichées sur l'écran.

Exemple:
Dans la fenêtre de commande, tapez:
>>a = 4*5;
> > x=5 ; y=5 ; z=x+y
z= 10

1.8 Constantes prédéfinies

Il existe des symboles auxquels sont associés des valeurs prédéfinies. En voici quelques-uns :

Symbole	Signification	Valeur
pi	Nombre π	3.141592...
i ou j	Nombre complexe	$\sqrt{-1}$
realmax	Plus grand nombre flottant codable	1.7977e+308
realmin	Plus petit nombre flottant codable	2.2251e-308

1.9 Quelques fonctions utiles

Ci-joint quelques fonctions :

log(x) : Logarithme néperien de x.
log10(x) : Logarithme en base 10 de x.
exp(x) :Exponentielle de x.
sqrt(x) : Racine carrée de x.
sign(x) : Fonction valant 1 si x est positif ou nul et -1 sinon.

cos(x) : Cosinus.
acos(x) : Cosinus inverse (arccos).
sin(x) : Sinus. asin(x) :
Sinus inverse (arcsin).
tan(x): Tangente. atan(x):
Tangente inverse (arctan).
cosh(x): Cosinus hyperbolique (ch).
acosh(x): Cosinus hyperbolique inverse (argch).
sinh(x): Sinus hyperbolique (sh).
asinh(x): Sinus hyperbolique inverse (argsh).
tanh(x): Tangente hyperbolique (th).
atanh(x): Tangente hyperbolique inverse (argth).

1.10 Vecteurs et matrices

Avec Matlab, on travaille essentiellement avec un type d'objet : les matrices . Une variable scalaire est une matrice de dimension 1 x 1 et un vecteur est une matrice de dimension 1 x n ou n x 1. Il est capital d'être à l'aise avec ces notions pour comprendre au mieux la philosophie de Matlab et l'exploiter efficacement.

1.10.1 Définition d'un vecteur

Un vecteur n'est rien d'autre qu'un tableau de valeurs. Il existe plusieurs façons de créer un vecteur et la plus simple d'entre elles est de l'écrire explicitement.

```
>> v = [1 2 3 4]
v =
     1     2     3     4
```

L'ensemble des composantes est donné entre crochets et les valeurs sont séparées par un espace (ou une virgule « **,** »). Nous avons ici défini un vecteur ligne. Un vecteur colonne est créé en utilisant un point-virgule « **;** » comme délimiteur.

```
>> v = [1 ; 2 ; 3 ; 4]
v =

     1
     2
     3
     4
```

Bien que simple, cette méthode n'est pas pratique pour définir des vecteurs de taille importante. Une seconde méthode utilise l'opérateur deux-points « : ». Il permet de discrétiser un intervalle avec un pas constant.

```
>> v = 0:0.2:1
v =
         0    0.2    0.4    0.6    0.8    1
```

Cette instruction crée un vecteur contenant des valeurs allant de 0 à 1 avec un pas de 0.2. La syntaxe est la suivante : ***vecteur = valeur_initial:incrément:valeur_finale.*** Par défaut, le pas est égal à 1. Par exemple :

```
>> v = 0:5
v =
         0    1     2     3     4     5
```

On peut accéder aux différents éléments d'un tableau en spécifiant un (ou des) indice(s) entre parenthèses.

```
>> v = [6 4 -1 3 7 0.3];
>> v(3)
ans =
      -1

>> v(2:4)
ans =
      4    -1     3
```

v(3) retourne le 3ième élément du vecteur v.
L'argument 2:4 permet de sélectionner un bloc d'éléments (ici du second au quatrième).

1.10.2 Quelques fonctions utiles

Nous présentons dans ce paragraphe un ensemble de fonctions usuelles liées à l'utilisation des tableaux.

`length(v)`	renvoie la taille du tableau.
`max(v)`	renvoie la valeur maximale du tableau.
`min(v)`	renvoie la valeur minimale du tableau.
`mean(v)`	renvoie la valeur moyenne des éléments du tableau.
`sum(v)`	calcul la somme des éléments du tableau.
`prod(v)`	calcul le produit des éléments du tableau.
`sort(v)`	range les éléments du tableau dans l'ordre croissant.

Toutes les fonctions mathématiques vues auparavant sont applicables aux variables de type vecteur. Dans ce cas, la fonction est opérée sur chacun des éléments du vecteur.

```
>> v = [0 pi/4 pi/2 pi 2*pi]
 v =
         0    0.7854    1.5708    3.1416    6.2832
>> cos(v)
ans =
    1.0000    0.7071    0.0000   -1.0000    1.0000
```

1.10.3 Définition d'une matrice

La définition d'une matrice est délimitée pas des crochets d'une ligne sont séparés par un espace et les différentes lign virgules « ; ». Ainsi pour définir une variable matricielle

$$M = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

on écrira :

```
>> M = [1 2 3 ; 4 5 6 ; 7 8 9];
```

ou

```
>> M = [1,2,3 ; 4,5,6 ; 7,8,9];
```

L'accès à un élément d'une matrice s'opère en spécifiant des indices entre parenthèses à la suite de son nom. L'élément situé la i ième ligne et la j ième colonne est obtenu par la commande M(i,j). Par exemple, la valeur est récupérée en tapant

```
>> M(2,3)
ans =
     6
```

On peut également modifier directement un des éléments en lui affectant une nouvelle valeur.

```
>> M(2,3)=13;
>> M
M =
     1     2     3
     4     5    13
     7     8     9
```

1.10.4 Matrices particulières

- La matrice nulle :

```
>> Z = zeros(2,3)
Z =
     0     0     0
     0     0     0
```

- La matrice pleine de 1 :

```
>> U = ones(4,3)
U =
     1     1     1
     1     1     1
     1     1     1
     1     1     1
```

- La matrice identité :

```
>> I  = eye(3)
I =
     1     0     0
     0     1     0
     0     0     1
```

- La matrice diagonale :

```
>> D = diag([2,4,0,7])
D =
     2     0     0     0
     0     4     0     0
     0     0     0     0
     0     0     0     7
```

1.10.5 Quelques fonctions utiles

Nous présentons dans ce paragraphe un ensemble de fonctions usuelles liées à l'utilisation des matrices.

« size(M) » renvoie les dimensions de la matrice.

« max(M) » renvoie un vecteur-ligne contenant les valeurs maximales associées à chaque colonne.

« min(M) » renvoie un vecteur-ligne contenant les valeurs minimales associées à chaque colonne.

« rank(M) » renvoie le rang de la matrice.

« det(M) » renvoie le déterminant de la matrice.

« diag(M) » extrait la diagonale de la matrice.

« triu(M) » extrait la matrice-triangle supérieure de M. tril donne la matrice-triangle inférieure.

« eig(M) » renvoie un vecteur contenant les valeurs propres de la matrice.

1.10.6 Opérations sur les matrices

Un des atouts remarquables de Matlab est la possibilité d'effectuer les opérations arithmétiques traditionnelles de façon naturelle sans avoir à les programmer. Les opérateurs standards sont donc directement applicables aux matrices. Si la commande entrée ne respecte pas les règles de calcul matriciel (compatibilité des opérandes), le logiciel renverra une erreur.

\+ addition

\- soustraction

* produit

/ division à droite

\ division à gauche

^ puissance

‘ transposition

inv( ) inversion

```
>> A = [2 1;6 9];
>> B = [1 0;-4 3];

>> A + B
ans =
     3     1
     2    12

>> A * B
ans =
    -2     3
   -30    27

>> A'
ans =
     2     6
     1     9

>> inv(B)
ans =
    1.0000         0
    1.3333    0.3333
```

```
>> A / B
ans =
    3.3333    0.3333
   18.0000    3.000

>> A^2
ans =
    10    11
    66    87
```

Le tableau suivant résume quelques-uns des opérateurs évoqués précédemment.

Syntaxe Matlab	Ecriture mathématique	Composante ij
A	A	A_{ij}
B	B	B_{ij}
A+B	A+B	$A_{ij} + B_{ij}$
A-B	A-B	$A_{ij} - B_{ij}$
A.*B		$A_{ij} B_{ij}$
A.^B		$A_{ij}^{B_{ij}}$
A.^s		A_{ij}^{s}
A*B	AB	$\sum_k A_{ik} B_{kj}$
A/B	AB^{-1}	
A\B	$A^{-1}B$	
A'	A^T	A_{ji}