

Chapitre 1 : Présentation de Matlab

1.1. GENERALITES

L'environnement Matlab offre un langage sur lequel peut s'adapter l'outil d'analyse de différents domaines d'étude.

MATLAB (MATrix LABoratory) comprend de nombreuses fonctions (graphismes, opérateurs de calcul, solution d'équation, zéros d'une fonction, intégration, interpolation et bien d'autres fonctionnalités) ainsi qu'un langage de programmation. Les commodités de MATLAB sont en grande partie les possibilités qu'il offre pour réaliser des manipulations mathématiques complexes en écrivant peu d'instructions. Par ailleurs, il permet l'utilisation directe de plusieurs milliers de fonctions prédéfinies. L'utilisation des boîtes à outils (**toolboxes**) augmente sa popularité dans plusieurs disciplines comme la simulation, l'imagerie, ...etc.

Par le présent cours, on cherche à initier les étudiants à utiliser MATLAB et à écrire des programmes simples. Sur le web existe de nombreux cours sur Matlab. On peut aussi consulter la rubrique aide (en langue anglaise). Cette dernière contient les informations utiles pour améliorer son savoir-faire et approfondir ses connaissances.

On ouvre Matlab par un double clic sur l'icône . Il apparaît alors l'espace de travail.

Au démarrage, MATLAB affiche généralement 4 fenêtres constituées notamment de (fig. 1.1):

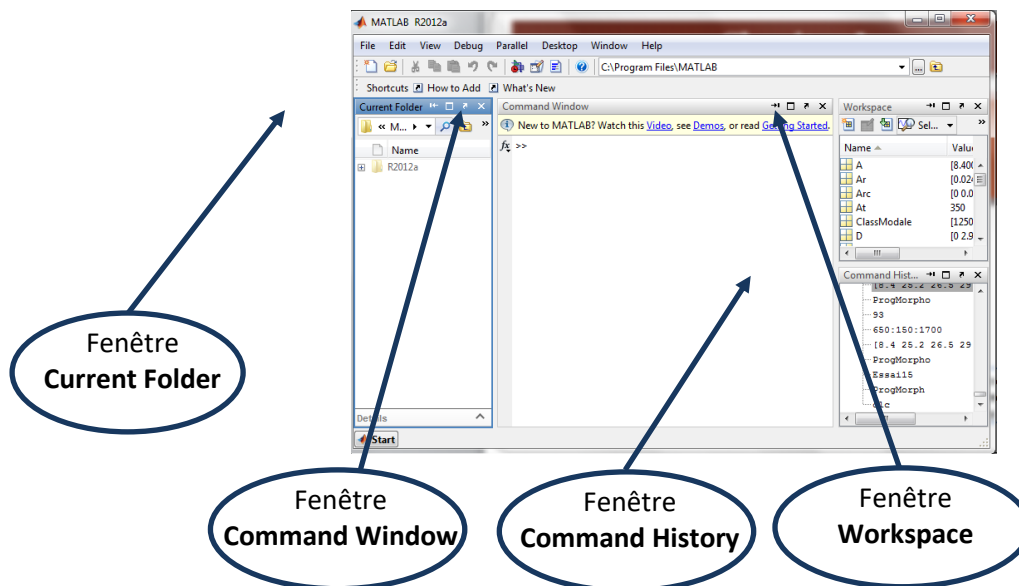


Fig.1.1 Environnement Matlab

- **Current Folder:** indique le répertoire courant ainsi que les fichiers existants.
- **Workspace:** indique toutes les variables existantes avec leurs types et valeurs.
- **Command History:** garde la trace de toutes les commandes entrées par l'utilisateur.

- **Command Window:** nous l'utilisons pour formuler nos expressions et interagir avec MATLAB. C'est la fenêtre la plus utilisée.

Sur l'espace « Command Window » apparaît le symbole « >> ». Le prompt Matlab (>>) indique que Matlab attend des instructions.

La commande '**quit**' pour fermer et quitter la session Matlab.

1.2. AIDE, DOCUMENTATION EN LIGNE

L'aide peut être obtenue directement dans la session en tapant help. La commande help toute seule vous indiquera les différents thèmes abordés dans la documentation.

Si vous tapez help 'commande', c'est-à-dire help + l'instruction recherchée, alors vous obtenez l'aide sur la commande précisée. Il existe des hyperliens de la documentation d'une fonction à une autre, ainsi que des exemples prêts à fonctionner. Vous trouvez aussi une documentation riche sur le web. Les liens utiles seront fournis au fur et à mesure.

1.3. OPERATIONS ELEMENTAIRES

Il est possible d'utiliser Matlab comme une calculatrice. Le moyen le plus simple est d'écrire les opérations directement dans la fenêtre de commande (*Command Window*) juste après le curseur (prompt) >>. Voici quelques exemples illustratifs :

a) Addition

```
>> 3+7 puis tapez entrée (qu'on note par la suite↵). Vous aurez le résultat :
ans=
    10
```

- ✓ *ans est pour lire la dernière réponse.*
- ✓ *La décimale est représentée par un point au lieu de la virgule (c'est la notation adoptée par les Anglo-Saxons).*

L'opération addition donnée par le symbole '+' peut être calculé par la fonction 'plus'. Pour plus de détails sur les fonctions voir chapitre 4.

```
plus(1,2)    ↵    ans= 3
```

b) Soustraction

```
>> 6 - 10    ↵    ans = -4.
```

c) Multiplication

```
>> 3*7       ↵    ans= 21
```

d) Division

```
>> 5/4       ↵    ans = 1.2500
```

```
>> 5\4       ↵    ans = 0.8000. Attention : cette opération permet de diviser 4 par 5.
```

e) Puissance

```
>> 2^5       ↵    ans = 32
```

*Les signes des opérations élémentaires sont utilisés de manière simple. Les signes +, -, *, / sont utilisés respectivement pour l'addition, la soustraction, la multiplication et la division.*

- ✓ *L'anti-slash, a\b, est utilisé pour diviser b par a.*
- ✓ *Ces opérations sont simples à utiliser, il faut juste faire attention aux parenthèses.*
- ✓ *L'instruction a^b donne la valeur de a à la puissance b.*

1.4. VARIABLE

1.4.1. Variables nommées

La réponse 'ans' est une variable créée automatiquement quand l'expression n'est pas désignée au paravent par un nom. Cependant, on peut nommer les variables par des lettres, des noms ou des lettres suivit de chiffre.

Le nom d'une variable ne doit contenir que des caractères alphanumériques ou le symbole '_' (underscore) (les symboles d'opération sont interdits). Le nom d'une variable doit commencer par une lettre de l'alphabet. Aussi, MATLAB distingue entre les lettres en minuscule et majuscule (**A** et **a** sont deux variables différentes).

Il est possible de conserver un résultat de calcul et de le stocker dans des variables.

On ne déclare pas les variables. Leur type (entier, réel, complexe) s'affectera automatiquement en fonction du calcul effectué.

1.4.2. Affectation des variables

Pour affecter une variable, on dit simplement à quoi elle est égale.

>> a=1.6	↔	a=	1.6000
>> a=12	↔	a=	12
>> b=0.7	↔	b=	0.7000
>> c=1.8	↔	c=	1.8

On peut introduire plusieurs instructions dans une même ligne. Il suffit de les séparer par une virgule pour dire que ce sont des instructions différentes.

>> A=2, B=4, C=45	↔	A=	2
		B=	4
		C=	45

On peut rééditer les variables pour revoir quelles valeurs leur sont affectées.

>> a,b,c,A,B,C	↔	a=	12
		b=	0.7000
		c=	1.8000
		A=	2
		B=	5
		C=	45

*Matlab mémorise la dernière valeur affectée à une variable (l'ancienne valeur est effacée).
Matlab distingue entre la lettre en majuscule et en minuscule.*

1.4.3. Opérations sur les variables

Les variables affectées peuvent être introduites dans de nouvelles expressions mathématiques.

>> u=5*a-b	↔	u=59.3000
------------	---	-----------

On peut, ensuite utiliser ces deux variables :

```
>> w=a/u    ↵    w=0.2024
```

- ✓ On sait que maintenant les valeurs de huit variables *a, b, c, A, B, C, u* et *w* sont en mémoire.
- ✓ Les variables stockées ne sont pas affichées en permanence à l'écran. Pour voir le contenu d'une variable, on tape son nom, exemple :

```
>>c    ↵    c= 1.8000
```

1.4.4. Lecture des erreurs

Lorsqu'une instruction est incorrecte, c'est-à-dire non conforme ou fausse, MATLAB lance un bip sonore et affiche un commentaire en rouge où il est expliqué le refus de l'instruction. . Voici des exemples sur les erreurs couramment commises.

- a) Introduction d'une variable à laquelle aucune valeur n'a été affectée.

```
>> X    ↵
```

Il apparaît en rouge le commentaire suivant qui précise que la variable X est inconnue.

Undefined function or variable 'X'.

- b) Erreur sur les parenthèses

```
>> (a+b))    ↵
```

Il apparaît le commentaire suivant :

Error: Unbalanced or unexpected parenthesis or bracket.

- c) Erreur sur le nom de la variable

```
>>A+=5    ↵
```

Error: The expression to the left of the equals sign is not a valid target for an assignment.

Le motif du refus est que les variables ne doivent pas contenir des opérateurs.

On peut faire des calculs en complexe, $\sqrt{-1}$ s'écrit indifféremment 'i' ou 'j', donc pour définir un complexe affecté par exemple à la lettre z, on tape :

```
>> z = a + b * i    ↵    z = 12.0000 + 0.7000i
```

On peut associer à une variable une chaîne de caractères. Exemple, affecter le nom solaire à la variable t. Pour cela, on affecte le nom entre apostrophes à la variable désignée.

```
>> t='solaire'    ↵    t=solaire.
```

1.4.5. Commandes 'who', 'whos' et 'clear'

La commande « who » est utilisée pour afficher les variables actives dans l'espace de travail. La fonction « whos » donne plus de détails : le nom, la taille du tableau (nombre de lignes et de colonnes) associé, l'espace mémoire utilisé (en Bytes) et la classe de données (principalement double 'array' s'il s'agit d'un tableau de valeurs réelles ou complexes et 'char' s'il s'agit d'un tableau de caractères).

```
>> who    ↵    Your variables are :    a b c A B C u w z t
```

Les variables sont présentées suivant l'ordre dont elles ont été introduites.

>> whos ↵

Name	Size	Bytes	Classe	Attributes
<i>a</i>	1x1	8	double	
<i>b</i>	1x1	8	double	
<i>c</i>	1x1	8	double	
<i>A</i>	1x1	8	double	
<i>B</i>	1x1	8	double	
<i>C</i>	1x1	8	double	
<i>u</i>	1x1	8	double	
<i>w</i>	1x1	8	double	
<i>z</i>	1x1	16	double	Complex
<i>t</i>	1x7	14	char	

A ce stade, l'ensemble des 13 variables actives totalisent 16 éléments utilisant 114 bytes.

Pour avoir le détail d'une variable, l'instruction est whos suivi du nom de la variable. Exemple :

>> whos z ↵

Name	Size	Bytes Class	Attributes
<i>z</i>	1x1	16 double	Complex

L'instruction 'who a*' affiche toutes les variables commençant par la lettre 'a'.

Les instructions 'clear' ou 'clear all' effacent les valeurs des variables stockées en mémoire. La différence entre 'clear' et 'clear all' sera abordée dans le chapitre 4.

L'instruction 'clear a' efface de la mémoire le contenu de la variable a.

Exemple:

>> clear all ↵

>> a+b ↵ *Undefined function or variable 'a'.*

Matlab commence par lire l'instruction et donne le statut de la première variable qui est non définie. Au-fait, les deux variables a et b ne sont pas définies.

- ✓ Les noms de variables doivent commencer par une lettre. Ils peuvent avoir une longueur quelconque.
- ✓ Une variable garde la dernière valeur qui lui a été affectée et reste dans la « command history » tant qu'il n'a pas été effacé par l'instruction « clear »
- ✓ L'instruction « clear all » efface toute les valeurs affectées aux variables utilisées au paravent.

1.5. FONCTIONS USUELLES

1.5.1. Variables et fonctions prédéfinies

>> pi ↔ (donne le nombre π)
>> eps ↔ (donne le nombre 2,22 e-16) (c'est la plus petite différence possible entre deux réels successifs, sinon les valeurs seront confondues, voir intérêt lors des calculs itératifs).

Pour appliquer une fonction à une valeur, il faut mettre cette dernière entre parenthèses. La commande d'une fonction s'écrit toujours en minuscules. Voici une liste **non exhaustive** de fonctions prédéfinies par MATLAB :

- ✓ fonctions trigonométriques et inverses : sin, cos, tan, tand, asin, acos, atan
- ✓ fonctions hyperboliques (on rajoute «h») : sinh, cosh, tanh, asinh, acosh, atanh
- ✓ racine, logarithme, logarithme base 10 et exponentielle : sqrt, log, log10, exp.
- ✓ fonctions erreur : erf, erfc
- ✓ abs(a) (donne la valeur absolue de a ou le module si la variable est complexe)
- ✓ angle(z) (donne la phase ou argument du nombre complexe z)
- ✓ around (t) est l'arrondi de l'élément t.

1.6. LES NOMBRES EN MATLAB

Par défaut, le résultat d'une opération de calcul comprenant des décimales est affiché avec 4 chiffres significatifs, soit quatre chiffres après la virgule.

Exemple :

>>1.7 ↔ 1.7000 ; >> pi ↔ 3.1416 ; >> sqrt(2) ↔ 1.4142

La commande 'format long' permet d'afficher 15 chiffres après la virgule. Après cette instruction, l'ensemble des nombres seront présentés avec une précision de 15 chiffres après la virgule.

Exemple :

>>1.7 ↔ 1.700000000000000 ; >> pi ↔ 3.141592653589793

Pour revenir à un affichage avec 4 chiffres significatifs, il suffit d'écrire l'instruction 'format short'.

Pour afficher uniquement 02 chiffres après la virgule, utiliser la commande format bank.

Pour afficher les nombres sous forme rationnelle, utiliser la commande format rat.

Exemple : >> a=0.02 ↔ a= 1/50.

Pour revenir à un affichage particulier, format long, short, bank ou rationnel, il suffit de réécrire le format d'affichage.

Voire aussi l'affichage après « format shortEng » et « format longEng »

1.7. HISTORIQUE DES COMMANDES

Les commandes que vous tapez sous Matlab peuvent être retrouvées et éditées grâce aux touches de direction ↑↓.

Pour reprendre une commande précédente, appuyez sur les touches ↑↓ pour remonter ou descendre, puis utilisez la touche «entrée» pour éditer une commande.

Pour relancer une commande, inutile de remettre le curseur à la fin, vous appuyez directement sur la touche «Entrée».

Vous pouvez retrouver toutes les commandes commençant par un groupe de lettres. Par exemple pour retrouver toutes les commandes commençant par «plot», tapez plot, puis appuyez sur les touches ↑↓.

1.8. EXERCICES

Exercice n°1 : L'objectif de cet exercice est de montrer l'importance de la position des parenthèses et des opérateurs.

a) Préciser l'ordre des opérations suivantes :

- | | | | |
|---------------|-----------------|-------------|---------------|
| 1. $2+3/6$ | • $(2+3)/5$ | • $6/3-5$ | • $2/(3-5)$ |
| 2. $(2+5)-7$ | • $0.1+3*(1+5)$ | • $0.1/10$ | • $.5*4.8/4$ |
| 3. $2*3^0.76$ | • $18/3^2$ | • $3^2/6$ | • $2*4^2/0.5$ |
| 4. $2*3/0.2$ | • $2/3*0.2$ | • $(2*3)^2$ | • $2*(3^2)$ |

b) Que se passe-t-il si vous tapez :

$S=3^0,76$; $1/0,56$

Exercice n°2 : Commenter les résultats affichés à la suite de chacune des instructions suivantes :

- | | | | |
|-------------------|----------------|----------------------------|-------------------|
| 1°) pi | 2°) Pi | 3°) $\cos(\pi/2)-\sqrt{4}$ | 4°) $\tand(45)$ |
| 5°) $\tan(\pi/4)$ | 6°) $\tan(45)$ | 7°) $\cosd(60)$ | 8°) $\sin(\pi/6)$ |

Exercice n°3 : Après l'instruction clear all, qu'affiche les opérations suivantes

1°) u	2°) yz=pi	3°) yadi=2	4°) a=3
5°) who	6°) who y	7°) who y*	8°) whos
9°) clear y	10°) clear a	11°) clear y*	12°) clear a*
13°) format long	14°) format short	15°) format bank	16°) format rat

Exercice n°4: Après clear all, introduire les instructions suivantes :

$a=1.5$; $a=12$; $b=0.5$; $c=0.1$;

Trouver la valeur des instructions justes, sinon précisez l'erreur.

- | | | |
|--|---------------------------------------|---------------------------------------|
| <input type="checkbox"/> $u=5*a-b$ | <input type="checkbox"/> $w=a/c$ | <input type="checkbox"/> $X= a+b/c$ |
| <input type="checkbox"/> $Y=(a+0.8)/(c*(8.1-c)^2)$ | <input type="checkbox"/> $b*C$ | <input type="checkbox"/> $(a+(b/c)^2$ |
| <input type="checkbox"/> $X\backslash Y$ | <input type="checkbox"/> X,Y | <input type="checkbox"/> X/Y |
| <input type="checkbox"/> $a=5.5$ | <input type="checkbox"/> $b=6$ | <input type="checkbox"/> $C=2^.7$ |
| <input type="checkbox"/> $b*C$ | <input type="checkbox"/> $(a+(b/c)^2$ | <input type="checkbox"/> $s=6X$ |

1.9. QUIZ1 : QUE DONNENT LES INSTRUCTIONS : « COCHER LA BONNE REPONSE »

		Réponses proposées			
N°	Instructions	A	B	C	D
1	>> 1/2	0.5000	0,5000	Erreur	0.5
2	>> .02	0.0200	Erreur	0,02	10 ²
3	>> 0.5^2	0.2500	50.0000	25	0.55
4	>> 2*(2+7)	11	18	16	Erreur
5	>> 2*2+7	11	18	16	Erreur
6	>> (2*2)+7	11	18	16	Erreur
7	>> 0.1+3/(1-6)	0.6000	0,6000	- 0.5000	0.5000
8	>> (0.1+3)/(1-6)	-0.6000	0,6000	- 0.6200	0.5000
9	>> 3x4	12	12.0000	12,0000	Error
10	>> 2\1	0.5000	0,5000	Erreur	0.5
11	>> format bank >> 0.5	0.5000000000000000	0.5000	0.50	0.5\$
12	>> format rat >> 0.5	0.5	0.5000	1/2	Error
13	>> format short >> 0.5	0.5	0.5000	1/2	Error
14	>> T='année'	Error	character	365 j	année
15	>> pi	3.1416	180°	Erreur	Π
	Après les instructions suivantes a=4 ; b=0.5 ; c=2 ;				
16	>> a+b\c	Error	4.2500	1,7500	5.5000
17	>> A+b/c	Error	4.2500	1,7500	5.5000
18	>> axb	Error	2.0000	2	2.00
19	length([a a b c])	4 4 0.5 2	4	3	Error
20	>> a^c	2	16	2.0000	Error

1.10. SOLUTIONS EXERCICES CHAPITRE 1

Exercice n°1 : Ordre et résultats des opérations suivantes

Ordre des opérations (ou priorité des opérateurs)

Les opérations sont évaluées en donnant la priorité aux opérateurs selon l'ordre suivant :

- $()$ parenthèse
- $^$ puissance
- $*$, $/$ la première dans l'instruction, multiplication et division, sera exécutée
- $+$, $-$ la première dans l'instruction, addition ou soustraction, sera exécutée.

a) L'ordre des priorités des opérations de l'exercice n°1

$2+3/6$	1 ^{ère} opération $3/6=0.5$, puis $2+0.5=2.5000$
$(2+3)/5$	1 ^{ère} opération $2+3=5$, puis $5/5=1$
$6/3-5$	1 ^{ère} opération $6/3=2$, puis $2-5=-3$
$2/(3-5)$	1 ^{ère} opération $3-5=-2$, puis $2/-2=-1$
$(2+5)-7$	1 ^{ère} opération $2+5=7$ puis $7-7=0$
$0.1+3*(1+5)$	1 ^{ère} opération $1+5=6$, puis $6*3=18$, puis $0.1+18=18.1000$
$0.1/10$	une seule opération $0.1/10=0.0100$
$0.5*4.8/4$	1 ^{ère} opération $4.8/4=1.2$, puis $0.5*1.2=0.600$
$2*3^{0.76}$	1 ^{ère} opération $3^{0.76}=2.3047$, $2*2.3047=4.6094$
$18/3^2$	1 ^{ère} opération $3^2=9$, puis $18/9=2$
$3^2/6$	1 ^{ère} opération $3^2=9$, puis $9/6=1.5000$
$2*4^2/0.5$	1 ^{ère} opération $4^2=16$, puis $2*16=32$, puis $32/0.5=64$
$2*3/0.2$	1 ^{ère} opération $2*3=6$, puis $6/0.2=30$
$2/3*0.2$	1 ^{ère} opération $2/3=0.6666$, puis $0.6666*0.2=0.1333$
$(2*3)^2$	1 ^{ère} opération $2*3=6$, puis $6^2=36$
$2*(3^2)$	1 ^{ère} opération $3^2=9$, puis $2*9=18$

b) Faire attention à la virgule, l'instruction $3^{0,76}$ donne deux valeurs $3^0=1$ et 76

De même la réponse à l'instruction $1/0,56$ donne : inf (pour $1/0$) et 56.

Après l'instruction :

>> $S=3^{0,76}; 1/0,56$

S=

1

inf

56

Remarque : le ; (point-virgule) empêche l'affichage de 76

Exercice n°2

```
>> pi           ↪      3.1416 (donne la valeur de  $\pi$ )
>> Pi           ↪      variable indéfinie
>> cos(pi/4)-sqrt(4)/2 ↪ -2      car (cos(pi/2) = 0 et sqrt(4)=2 )
>> tand(45)     ↪      1          calcule la tangente en degré
>> tan(pi/4)     ↪      1          calcule la tangente en radian
>> tan(45)       ↪      1.6198    calcule la tangente en radian
>> cosd(60)      ↪      0.5000    calcule le cosinus en degré
>> sin(pi/6)     ↪      0.5000    calcule le sinus en radian
```

Exercice n°3

Instruction	Résultat	Observation																				
1°)	variable indéfinie																					
2°)	yz= 3.1416																					
3°)	yadi = 2																					
4°)	a = 3																					
5°)	Vos variables sont : yz yadi a	Matlab répond en Anglais : Your variables are																				
6°)		Aucune réponse car il n'y pas de variable y																				
7°)	Vos variables sont : yz yadi	Affiche toutes les variables qui commencent par y																				
8°)	<table><tr><td>Name</td><td>Size</td><td>Bytes</td><td>Class</td><td>Attributes</td></tr><tr><td>a</td><td>1x1</td><td>8</td><td>double</td><td></td></tr><tr><td>yadi</td><td>1x1</td><td>8</td><td>double</td><td></td></tr><tr><td>yz</td><td>1x1</td><td>8</td><td>double</td><td></td></tr></table>	Name	Size	Bytes	Class	Attributes	a	1x1	8	double		yadi	1x1	8	double		yz	1x1	8	double		Donne le détail des variables stockées en mémoire
Name	Size	Bytes	Class	Attributes																		
a	1x1	8	double																			
yadi	1x1	8	double																			
yz	1x1	8	double																			
9°)		n'exécute rien car il n'y pas de variable y																				
10°)		Efface la variable a																				
11°)		Efface les variables commençant par y																				
format long		affiche 15 chiffres décimales																				
format short		affiche 4 chiffres décimales																				
format bank		affiche 4 chiffres décimales																				
format rat		affiche la valeurs en format rationnelle, P/Q (P et Q deux entiers)																				

Exercice n°4: Les résultats

```
❑ u=5*a-b=59.5000 ; ❑ w=a/c=120
❑ X= a+b/c=15 ; ❑ Y=(a+0.8)/(c*(8.1-c)^2)=2
❑ b*C erreur C inconnu ; ❑ (a+(b/c)^2 erreur parenthèse
❑ X/Y=7.5000 ; ❑ X\Y=0.1333
❑ a=5.5=5.5000 ; ❑ b=6
❑ C=2^.7=1.6245 ; ❑ b*C =9.7470
❑ (a+(b/c)^2 erreur parenthèse ; ❑ s=6X erreur pas d'opérateur
```

1.11. REPONSES QUIZ N°1

N°	réponse		N°	Réponse		N°	Réponse		N°	Réponse
1	A		6	A		11	C		16	D
2	A		7	C		12	C		17	A
3	A		8	C		13	B		18	D
4	B		9	D		14	D		19	B
5	A		10	A		15	A		20	A