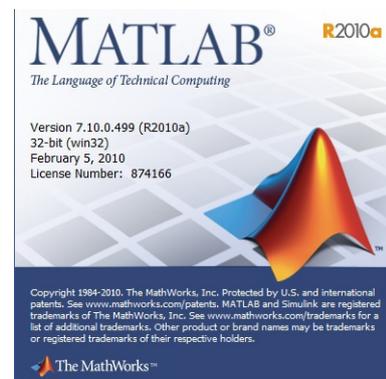


# Chapitre 01 : Introduction a l'environnement MATLAB

*Université de M'sila*



Dr. Chouder. R

M.C.B. université de M'sila

# Table des matières



<b>Introduction</b>	3
<b>I - L'environnement de travail</b>	4
<b>II - Premières commandes</b>	6
<b>III - Les types de variables</b>	10
<b>IV - Les opérateurs Matlab</b>	13
<b>V - Les fonctions prédéfinies</b>	15
<b>VI - L'aide de Matlab</b>	20
<b>VII - Exercice</b>	22

# Introduction



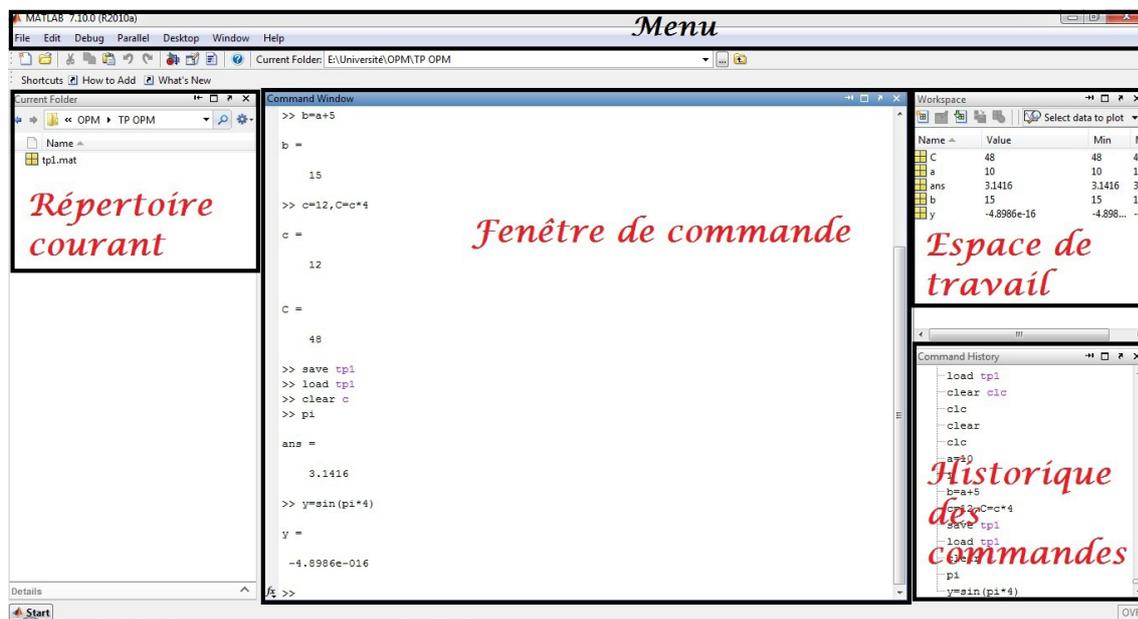
MATLAB (*MATrix LABoratory*) est un environnement de programmation interactif pour le calcul scientifique, la programmation et la visualisation des données. Il est très utilisé dans les domaines d'ingénierie et de recherche scientifique, ainsi qu'aux établissements d'enseignement supérieur. Sa popularité est due principalement à sa forte et simple interaction avec l'utilisateur mais aussi aux points suivants :

- Sa richesse fonctionnelle : avec MATLAB, il est possible de réaliser des manipulations mathématiques complexes en écrivant peu d'instructions. Il peut évaluer des expressions, dessiner des graphiques et exécuter des programmes classiques. Et surtout, il permet l'utilisation directe de plusieurs milliers de fonctions prédéfinie.
- La possibilité d'utiliser les boîtes à outils (*toolboxes*) : ce qui encourage son utilisation dans plusieurs disciplines (simulation, traitement de signal, imagerie, intelligence artificielle,...etc.).
- La simplicité de son langage de programmation : un programme écrit en MATLAB est plus facile à écrire et à lire comparé au même programme écrit en C ou en PASCAL.
- Sa manière de tout gérer comme étant des matrices, ce qui libère l'utilisateur de s'occuper de typage de données et ainsi de lui éviter les problèmes de transtypage.



# L'environnement de travail

Après lancement du MATLAB. On va voir apparaître une ou plusieurs nouvelles fenêtres sur l'écran . Selon la version on peut trouver les fenêtres suivantes

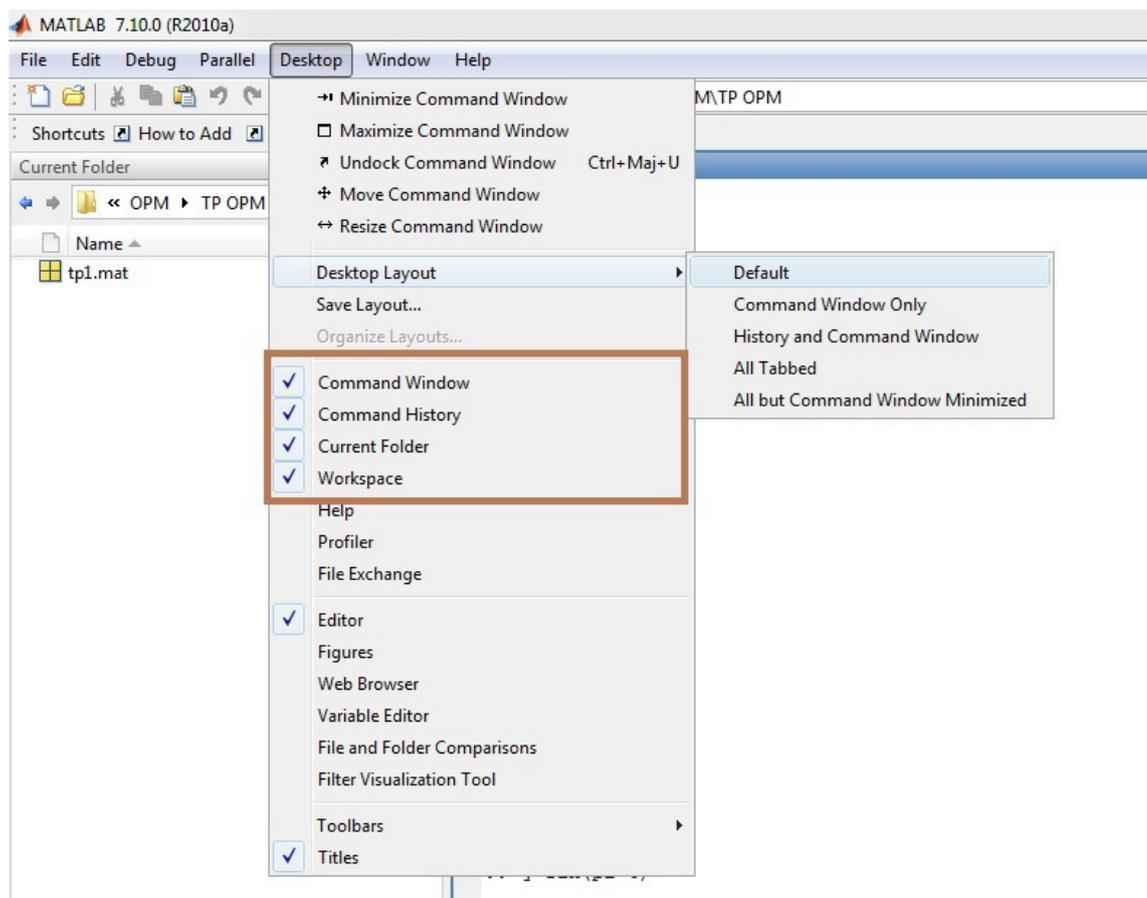


- **Fenêtre de commandes (Command Window)** : L'une des plus importantes fenêtres de MATLAB, elle permet de traiter les instructions données. C'est après l'invite ou prompt (Les caractères >>), qu'on tape les commandes qui seront exécutées par le logiciel après avoir appuyer sur la touche entrée . Le résultat s'affiche dans la fenêtre, mais il peut être représenté graphiquement dans une nouvelle fenêtre spécifique (avec possibilité de zoom, d'impression, etc).
- **Espace de travail (Workspace)** : Permet de visualiser les variables (nom, dimension et type). Matlab est basé sur les matrices, toutes les variables sont constituées de plusieurs dimensions : un scalaire est une matrice  $1 \times 1$  et un vecteur est une matrice  $1 \times n$  ou  $n \times 1$ , etc.
- **Le répertoire courant ( Current Directory)** : C'est le répertoire où sont enregistrés les fichiers -M. Il est fortement conseillé de se créer un répertoire autre que celui fournit par Matlab afin de mieux gérer les fichiers.
  - Pour utiliser un fichier-M, il doit être enregistré dans le répertoire courant.

- Si un fichier -M appelle une fonction autre qu'une fonction de Matlab (i.e. une fonction créée par l'utilisateur), le fichier -M appelant la fonction ainsi que le fichier-M définissant la fonction doivent être dans le même répertoire courant.
- *Historique de commandes (Command History)* : Permet d'inscrire les commandes au fur et à mesure qu'elles sont appelées dans Command Window. Elle garde en mémoire ces commandes, ainsi que la date et l'heure d'ouverture de chacune des sessions de Matlab.
- *Barres de menus et barre d'outils* : Permet d'accéder aux commandes du Matlab. Voici quelques exemples:  
*Edit → Clear Command Window* : Efface les instructions et/ou les résultats visibles dans la Command Window.  
*Edit → Clear Command History* : Efface les commandes précédentes mises en mémoire.  
*Edit → Clear Command Workspace* : Efface de la mémoire les variables stockées.  
 La barre de boutons d'outils : Permet un accès rapide à certaines commandes.

### Remarque : Changer l'affichage

Il est possible d'afficher ou de masquer chacune des 4 fenêtres vues précédemment en utilisant le menu *Desktop* puis en cochant/décochant les fenêtres à afficher/masquer.



Il est aussi possible de rétablir l'affichage par défaut à tout moment : Menu *Desktop* → *Desktop Layout* → *Default*

# Premières commandes



Le moyen le plus simple d'utiliser MATLAB est d'écrire directement dans la fenêtre de commande (Command Window) juste après le curseur (prompt) >>

Pour calculer une expression mathématique il suffit de l'écrire comme ceci :

```
>> 7+8 Puis on clique sur la touche Entrer pour voir le résultat
```

```
ans =
```

```
15
```

Si nous voulons qu'une expression soit calculée mais sans afficher le résultat, on ajoute un point-virgule ';' à la fin de l'expression comme suit :

```
>> 7+8 ;
```

```
>>
```

## *Affectation*

Pour créer une variable on utilise la structure simple : '*variable = définition*' sans se préoccuper du type de la variable.

Par exemple :

```
>> a = 10 ;
```

```
>> u = cos(a) ;
```

```
>> v = sin(a) ;
```

```
>> u^2+v^2
```

```
ans =
```

```
1
```

```
>> ans+10
```

```
ans =
```

```
11
```

```
>>
```

### Remarque : Le nom d'une variable

- Le nom d'une variable ne doit contenir que des caractères *alphanumériques* ou le symbole '\_' (underscore), et doit commencer par un alphabet. Nous devons aussi faire attention aux majuscules car le MATLAB est sensible à la casse (*A* et *a* sont deux identifiants différents).
- Lorsque le calcul d'une opération n'est pas affecté à une variable, Matlab crée de manière automatique une variable *ans* (answer) qui contient le résultat de l'opération.

### Liste des variables utilisées

Pour voir la liste des variables utilisées, soit on regarde à la fenêtre 'Workspace' soit on utilise les commandes 'whos' ou 'who'.

*whos* donne une description détaillée (le nom de la variable, son type et sa taille), par contre *who* donne juste les noms des variables.

Par exemple, dans ce cours on a utilisé 3 variables *a*, *u* et *v* :

```
>> who
```

```
Your variables are:
```

```
a ans u v
```

```
>> whos
```

```
Name Size Bytes Class Attributes
```

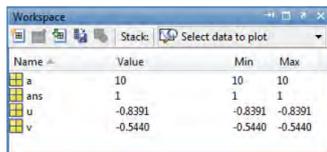
```
a 1x1 8 double
```

```
ans 1x1 8 double
```

```
u 1x1 8 double
```

```
v 1x1 8 double
```

L'utilisation de ces deux commandes peut être omise car des informations sur les variables sont visibles directement dans la fenêtre workspace.



Name	Value	Min	Max
a	10	10	10
ans	1	1	1
u	-0.8391	-0.8391	-0.8391
v	-0.5440	-0.5440	-0.5440

La commande *clear* permet de supprimer une variable, plusieurs variables, ou toutes les variables du workspace.

```
>> clear a , supprimer la variable a
```

```
>> clear a u , supprimer les variables a et u
```

```
>> clear , supprimer toutes les variables
```

## *Sauvegarder, quitter Matlab et restaurer une session de travail*

- Pour *sauvegarder* l'ensemble de variables : déjà manipulées en Matlab, dans une session de travail on utilise la commande « save »

Par exemple

```
>> save tp1
```

Dans le répertoire courant (ou Current Directory) un nouveau fichier portant le nom `tp1.mat` a été créé.

- *Quitter* Matlab : Pour quitter Matlab on peut utiliser soit le menu File → exit Mtlab ou d'utiliser directement la commande suivante

```
>> quit
```

- *Restaurer* un espace de travail : (exemple : `tp1.mat`) , dans ce cas on tape la commande « load »

```
>> load tp1
```

## *Plusieurs expressions dans la même ligne*

Il est possible d'écrire plusieurs expressions dans la même ligne en les faisant séparées par des virgules ou des points virgules. Par exemple :

```
>> x=5+6, y=2*5-1, 12-4
```

```
x =
```

```
11
```

```
y =
```

```
9
```

```
ans =
```

```
8
```

```
>> 5+6; 2*5-1, 12-4;
```

```
ans =
```

```
9
```

## *Lecture et affichage simple*

*La fonction input* : permet la saisie d'une valeur depuis le clavier:

- Pour les valeurs numériques, `n = input('message')` affiche message et affecte à la variable `n` la valeur numérique entrée au clavier.
- Pour les chaînes de caractères, `str = input('message','s')` affiche message et affecte à la variable `str` la valeur entrée au clavier considérée alors comme une chaîne de caractères.

```
>> A=input('Entrée la valeur de A : ')
Entrée la valeur de A : 3
A =
3
>> S=input('Entrée une phrase : ','s')
Entrée une phrase : bonjour tout le monde
S =
bonjour tout le monde
```

La fonction *disp* : permet d'afficher un tableau de valeurs numériques ou de caractères. L'autre façon d'afficher un tableau est de taper son nom.

```
>> a=[1 2 3]; b=['O' 'P' 'M'];
>> disp(a)
1 2 3
>> disp(b)
OPM
```



### Remarque

Matlab utilise toujours les nombres réels ( double précision ) pour faire les calculs , ce qui permet d'obtenir une précision de calcul allant jusqu'aux 16 chiffres significatifs. Mais il faut noter les points suivants :

- Le résultat d'une opération de calcul est par défaut affichée avec quatre chiffres après la virgule.
- Pour afficher davantage de chiffres utiliser la commande *format long* (14 chiffres après la virgule).
- Pour retourner à l'affichage par défaut, utiliser la commande *format short*.
- Pour afficher uniquement deux chiffres après la virgule, utiliser la commande *format bank*.
- Pour afficher les nombres sous forme d'une ration, utiliser la commande *format rat*.

# Les types de variables



En Matlab, l'utilisation de variables ne nécessite pas de déclaration de type. Les types de variables sont déterminés d'une façon dynamique (c.à.d. durant le temps d'exécution). Les quatre principaux types de variables utilisés par Matlab sont :

- *Le type réel*
- *Le type complexe*
- *Le type logique (Booléen) : Peut avoir deux valeurs ( true=1 ou false=0)*
- *Le type chaîne de caractères : Délimité par le symbole '*   
*Toute variable dans Matlab quelque soit son type est considérée comme une matrice.*

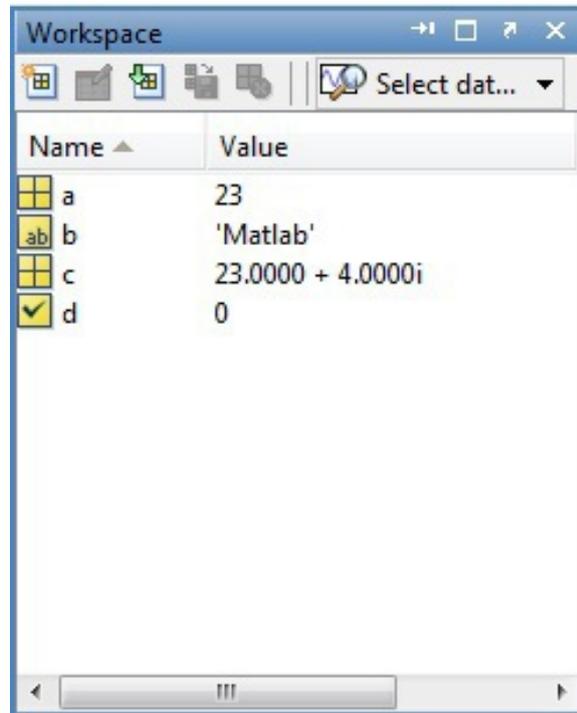
## Exemple : Types de variables

---

```
>> a=23; b='Matlab'; c=a+4i; d=false;
```

4 variables créées :

- *a* de type *réel* et vaut *23*
- *b* de type *chaîne de caractères* et vaut *'Matlab'*
- *c* de type *complexe* et vaut *23+4i*
- *d* de type *booléen* (logique) et vaut *0=false*



### *Variables spéciales (prédéfinies)*

Matlab possède des variables prédéfinies, ces variables existent même si elles ne sont pas présentes dans le workspace.

*La variable pi :*

```
>> pi
ans =
3.1416
```

*Les variables i et j :* les unités imaginaires des nombres complexes ( $i^2 = -1$ )

```
>> i
ans =
0 + 1.0000i
```

```
>> j
ans =
0 + 1.0000i
```

*La variable Inf (L'infini) :* Elle est obtenu par exemple en effectuant l'opération  $1/0$ .

*La variable NaN (Not -a -Number) :* La valeur d'un NaN est obtenue des opérations n'ayant pas valeur numérique en retour. Par exemple, multiplication de  $0*Inf$ , l'addition  $Inf-Inf$ .

 **Remarque**

---

Dans Matlab, les variables prédéfinies sont réservées, mais, on peut également les utiliser comme d'autres variables (entières ou réelles) si on les précise. Par exemple, la commande suivante crée une variable *pi* qui vaut 25 en écrasant la valeur 3.1416

```
>> pi=25
```

```
pi =
```

```
25
```

# Les opérateurs Matlab

## IV

### Les opérateurs arithmétiques

Les opérations de base dans une expression sont résumées dans le tableau suivant :

Symbole	Opération
+	addition
-	soustraction
*	multiplication
/	division
\	division gauche (ou la division inverse)
^	puissance

#### Remarque : Priorités des opérateurs arithmétiques

Priorité 1 : ^ (La puissance)

Priorité 2 : \* et / (La multiplication et la division, même priorité)

Priorité 3 : + et - (L'addition et la soustraction, même priorité)

Il est possible de modifier cet ordre de priorité en intégrant les parenthèses ().

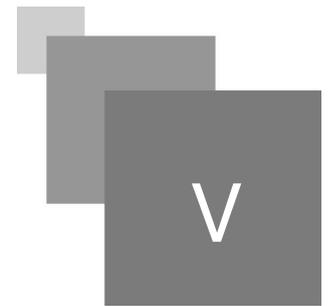
### Les opérateurs Logiques

Symbole	Opération
&	ET
	OU
~	NON

*Les opérateurs relationnels*

<i>Symbole</i>	<i>Opération</i>
<	<i>inférieur strictement</i>
>	<i>supérieur strictement</i>
<=	<i>inférieur ou égal</i>
>=	<i>supérieur ou égal</i>
==	<i>égal</i>
~=	<i>est différent</i>

# Les fonctions prédéfinies



Matlab propose plusieurs fonctions prédéfinies pour le calcul arithmétique. Nous allons voir uniquement les fonctions les plus utilisées.



Calculs		Trigonométriques	
L a fonction	Sa signification	L a fonction	Sa signification
exp	fonction exponentielle	sin	Sinus d'un angle (en radian)
log	logarithme naturel (à base $e$ ) $ln$	sind	Sinus d'un angle (en degré)
log10	logarithme décimal (à base 10)	cos	Cosinus d'un angle (en radian)
sqrt	la racine carrée	cosd	Cosinus d'un angle (en degré)
abs	la valeur absolue	tan	Tangent d'un angle (en radian)
conj	le conjugué d'un nombre complexe	cot	Cotangente d'un angle (en radian)
imag	la partie imaginaire d'un nombre complexe	asin	Arc sinus (résultat en radian)
real	la partie réelle d'un nombre complexe	acos	Arc cosinus (résultat en radian)
complex	cette fonction calcule un nombre complexe à partir de ces parties réelle et imaginaire	atan	Arc tangent (résultat en radian)
angle	l'argument d'un nombre complexe (en radian)	acosd	Arc cosinus (résultat en degré)

Une fonction est appelée par son nom suivi de ces variables (paramètres) entre parenthèses ().

### Exemple : Utilisation des fonctions prédéfinies (Calculs)

```
>> sqrt(4)
```

```
ans = 2
```

```
>> abs(-3)
```

```
ans = 3
```

```
>> exp(1)
```

```
ans = 2.7183
```

```
>> log(exp(4))  
ans = 4  
  
>> log10(10^3)  
ans = 3  
  
>> z=1+i ;  
  
>> abs(z)  
ans = 1.4142 ( $\sqrt{2}$ )  
  
>> real(z)  
ans = 1  
  
>> imag(z)  
ans = 1  
  
>> conj(z)  
ans = 1.0000 - 1.0000i  
  
>> imag(conj(z))  
ans = -1  
  
>> angle(z)  
ans = 0.7854 ( $\pi/4$ )  
  
>> complex(2,-3)  
ans = 2.0000 - 3.0000i
```

 ***Exemple : Utilisation des fonctions prédéfinies (Trigonométriques)***

---

```
>> sin(pi/2)  
ans = 1  
  
>> cos(-pi)  
ans = -1  
  
>> sind(-90)  
ans = -1  
  
>> tand(45)  
ans = 1.0000  
  
>> cot(pi/4)  
ans = 1.0000  
  
>> asin(0)
```

```
ans = 0
>> acosd(-1)
ans = 180
>> atand(1)
ans = 45
```

Arrondi et reste		Chaîne de caractères	
La fonction	Sa signification	La fonction	Sa signification
round	arrondir à l'entier le plus proche	strcat	concaténer des chaînes de caractères horizontalement
floor	arrondir à l'entier le plus proche vers $-\infty$	strvcat	concaténer des chaînes de caractères verticalement
ceil	arrondir à l'entier le plus proche vers $+\infty$	upper	converti tous les caractères d'une chaîne en majuscule
fix	arrondir vers zéro	lower	converti tous les caractères d'une chaîne en minuscule
mod	reste de division	length	calcule le nombre d'éléments d'un vecteur ou la longueur d'une chaîne

 Exemple : Utilisation des fonctions prédéfinies (Arrondi et reste)

```
>> round(5.3)
ans = 5
>> round(-15.6)
ans = -16
>> floor(6.87)
ans = 6
>> floor(-7.08)
ans = -8
>> ceil(6.17)
ans = 7
```

```
>> ceil(-7.08)

ans = -7

>> fix(3.4)

ans = 3

>> fix(-3.4)

ans = -3

>> mod(13,7)

ans = 6

>> mod(14,3)

ans = 2
```

 ***Exemple : Utilisation des fonctions prédéfinies (Chaîne de caractères)***

---

```
>> strcat('mat','lab')

ans =matlab

>> strcat('mat','lab','2020')

ans =matlab2020

>> strvcat('matlab','2020')

ans =

matlab

2020

>> upper('MaTlaB')

ans = MATLAB

>> lower('MaTlaB')

ans = matlab

>> length('matlab')

ans = 6
```

# L'aide de Matlab

VI

Matlab met à la disposition des utilisateurs, une vaste bibliothèque de fonctions (toolbox). On peut accéder à la documentation concernant ces fonctions de plusieurs façons:

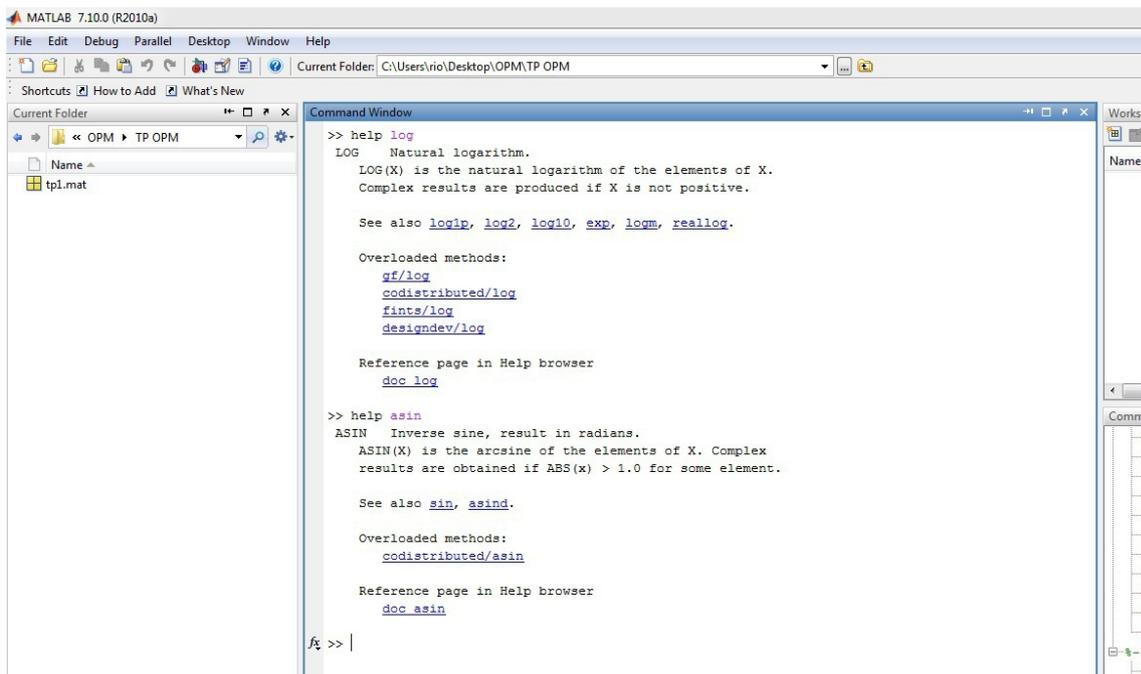
- *La commande help* : permet d'afficher dans l'interpréteur toute la documentation d'une fonction.
- *Utilisation de l'index* : dans le *menu help* , il est possible d'appeler l'index, puis taper le mot recherché. Et Matlab affiche les fonctions qui s'y rapportent.

## Exemple : La commande help

Voici deux exemples :

```
>> help log
```

```
>> help asin
```



```

MATLAB 7.10.0 (R2010a)
File Edit Debug Parallel Desktop Window Help
Current Folder: C:\Users\vrio\Desktop\OPM\TP OPM
Shortcuts How to Add What's New
Current Folder: << OPM >> TP OPM
Name
tpl.mat

>> help log
LOG Natural logarithm.
LOG(X) is the natural logarithm of the elements of X.
Complex results are produced if X is not positive.

See also logip, log2, log10, exp, logm, reallog.

Overloaded methods:
gf/log
codistributed/log
fints/log
designdev/log

Reference page in Help browser
doc_log

>> help asin
ASIN Inverse sine, result in radians.
ASIN(X) is the arcsine of the elements of X. Complex
results are obtained if ABS(x) > 1.0 for some element.

See also sin, asind.

Overloaded methods:
codistributed/asin

Reference page in Help browser
doc_asin

fx >> |
  
```

## Exemple : Utilisation de l'index

Dans le *menu help* , il est possible d'appeler l'index, puis taper le mot recherché.

Help

File Edit View Go Favorites Desktop Window Help

asin

Contents Search Results

Type	Relevance	Product
fx	asin - Inverse sine; result in radians	
	Y = asin(X) returns the inverse sine ... The asin function operates element-wise on ... of X in the domain [-1,1]. asin(X) is in the range ... of x	MATLAB
	<a href="#">Trigonometric Function</a> - Specified trigonometric function on input	
	asin ... asin(u) ... asin ... asin ... asin, acos, or atanh	Simulink
	<a href="#">Fcn</a> - Apply specified expression to input	
	functions --- abs, acos, asin, atan, atan2, ceil, cos, cosh, exp	Simulink
fx	<a href="#">registerFcnFunctionEntry</a> - Create TFL function entry based on specified pa...	
	asin	Real-Time Workshop Embedded Coder
fx	<a href="#">para2fan</a> - Convert parallel-beam projections to fan-beam	
	180/P/ASIN(PSPACE/D) ... D*ASIN(PSPACE/D)	Image Processing Toolbox
	<a href="#">equations</a> - Define component equations	
	asin	Simscape
fx	<a href="#">setTFCFunctionEntryParameters</a> - Set specified parameters for function ...	
	asin	Real-Time Workshop Embedded Coder
fx	<a href="#">registerCPPFunctionEntry</a> - Create TFL C++ function entry based on spe...	
	asin	Real-Time Workshop Embedded Coder
	<a href="#">Simulink Software Now Has Internal MATLAB Functions for Math On</a>	
	<a href="#">Search Online Support for asin</a>	

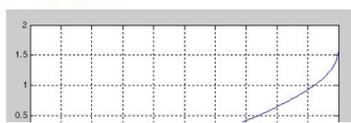
**asin**  
Inverse sine; result in radians

**Syntax**  
Y = asin(X)

**Description**  
Y = asin(X) returns the inverse sine (arcsine) for each element of X. The asin function operates element-wise on arrays. The function's domains and ranges include complex values. All angles are in radians. For real elements of X in the domain [-1,1], asin(X) is in the range  $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ .  
For real elements of x outside the range [-1,1], asin(x) is complex.

**Definitions**  
The arcsine is defined as:  
$$\sin^{-1}(z) = i \log \left[ iz + (1 - z^2)^{1/2} \right]$$

**Examples**  
Graph the inverse sine function over the domain  $-1 \leq x \leq 1$ .  
x = -1:0.01:1;  
plot(x, asin(x), grid on)



# Exercice

VII

Créer une variable  $x$  et donnez-la la valeur 4, puis écrivez les expressions suivantes :

$$3x^4 + 2x^3 - x^2 + 4x - 1$$

$$\frac{e^x - 1}{\sqrt{x} + \ln(x + 1)}$$

$$\sin(2x) + \cos(x - 1) + \tan(x^2)$$

$$\left| \cot\left(\frac{\pi}{x}\right) + \arcsin\left(\frac{1}{x}\right) + \cos^{-1}(1) \right|$$