

Fiche d'initiation à MATLAB

Carvalho Estéban

MATLAB est un logiciel utilisé pour le calcul numérique et la visualisation graphique. L'origine de son nom vient de **MAT**rix **LAB**oratory, indiquant que l'on va pouvoir manipuler et travailler avec des matrices. On cherchera souvent à éviter les boucles for, de tests, etc. pour plutôt se servir de la puissance des calculs matriciels. MATLAB inclut de nombreuses fonctions permettant la manipulation, le traitement, le calcul, l'affichage, etc. de données. A cela s'ajoute des Toolbox permettant de travailler dans de nombreux domaines tels que : l'automatique, l'optimisation, le traitement du signal, des images, etc.

Ce document est une fiche non exhaustive des fonctionnalités de base de MATLAB afin de bien débiter et de connaître quelques astuces rapidement. Elle n'est en aucun cas complète, ici l'utilisation "basique" des fonctions est présentée. Très généralement, les fonctions peuvent prendre de nombreux paramètres additionnels ou facultatifs permettant des utilisations plus larges, de manières différentes mais aussi de personnalisation, pour les tracés par exemple. Pour une utilisation plus complète, l'utilisation de la documentation est nécessaire.

1 - Documentation

<code>doc nom_de_la_fonction</code>	Ouvre une fenêtre contenant la documentation de la fonction cherchée
<code>help nom_de_la_fonction</code>	Décrit brièvement l'utilisation de la fonction dans la console
<code>lookfor mot_cle</code>	Recherche les fonctions liées au mot clé

2 - Constantes

<code>pi</code>	3.1415...
<code>eps</code>	epsilon machine = 2.2204e-16
<code>Inf</code>	Infini
<code>NaN</code>	Not a Number
<code>i</code> ou <code>j</code>	Nombre imaginaire

3 - Console

<code>ans</code>	dernière réponse
<code>clc</code>	nettoie la console
<code>home</code>	réinitialise la position du curseur et la vue sans supprimer l'historique

4 - Variables du workspace

<code>who</code>	liste les variables du workspace
<code>whos</code>	... et leurs propriétés
<code>clear</code>	supprimer toutes les variables du workspace
<code>clear name</code>	supprime la variable name (cumuler les variables est possible)

5 - Opérations et Opérateurs

<code>+</code> <code>-</code>	addition, soustraction
<code>*</code> <code>/</code> <code>^</code>	multiplication, division, puissance
<code>.*</code> <code>./</code> <code>.^</code>	opérations terme à terme
<code>mod(a,b)</code>	reste de la division euclidienne de a par b
<code>A \ b</code>	solution de $Ax=b$
<code>b / A</code>	solution de $xA=b$
<code><</code> , <code><=</code>	supérieur, supérieur ou égal
<code>></code> , <code>>=</code>	inférieur, inférieur ou égal
<code>==</code> , <code>~=</code>	égalité, différent
<code>&</code> <code> </code> <code>~</code>	et, ou, non

6 - Vecteurs et Matrices

<code>x = [1; 2; 3]</code>	créer un vecteur colonne
<code>x = [1, 2, 3]</code>	créer un vecteur ligne
<code>x = [1 2 3]</code>	(espace ou virgules)
<code>x'</code>	transposée
<code>x = a:h:b</code>	créer un vecteur de a à b avec pas de h
<code>x = [1 ;2;3 4]</code>	créer une matrice (; passage à la ligne suivante)
<code>y = [x ;x]</code>	concaténation de matrices
<code>x = []</code>	création d'un vecteur / matrice vide
<code>x(i,j)</code>	composante ligne i, colonne j
<code>x(:,j)</code>	sélectionne toutes les lignes de la colonne j
<code>x(i,:)</code>	sélectionne toutes les colonnes de la ligne i
<code>x(:,j) = []</code>	suppression de la colonne j
<code>x(i,:) = []</code>	suppression de la ligne i

7 - Matrices particulières

<code>eye(n)</code>	matrice identité d'ordre n
<code>zeros(m,n)</code>	matrice nulle de taille (m,n)
<code>ones(m,n)</code>	matrice de remplie de 1 de taille (m,n)
<code>diag(x)</code>	matrice diagonale formée à partir du vecteur x
<code>diag(x,k)</code>	matrice dont la k ^{ième} diagonale est formée des éléments de x (k = 0, représente la diagonale principale)
<code>blkdiag(x1,...,xn)</code>	matrice diagonale formée à partir des matrices x_1, \dots, x_n
<code>tril(x)</code>	renvoie la partie triangulaire inférieur de la matrice x
<code>triu(x)</code>	renvoie la partie triangulaire supérieur de la matrice x
<code>magic(n)</code>	carré magique d'ordre n
<code>rand(n)</code>	matrice (m,n) de nombres aléatoires uniformément distribués dans l'intervalle (0,1)
<code>randi(imax,n)</code>	matrice carré d'ordre n d'entiers pseudo-aléatoires de distribution uniforme discrète sur l'intervalle [1,imax].
<code>randn(n)</code>	matrice (m,n) de nombres aléatoires normalement distribués

8 - Fonctions vectorielles

length(x)	renvoie la plus grande dimension de x
size(x)	dimension de x : vecteur[nb ligne,nb colonne]
rank(x)	rang de x
det(x)	déterminant de x
trace(x)	trace de x
inv(x)	inverse de x
poly(x)	polynôme caractéristique de x
eig(x)	valeur propre de x
[V,D] = eig(x)	vecteur et valeur propre de x
norm(x)	norme 2 de x
reshape(x,m,n)	créer une matrice (m,n) à partir de x
Pour les fonctions suivantes si x n'est pas un vecteur, l'opération s'effectue selon la première dimension	
sum(x)	somme les éléments de x
mean(x)	renvoie la moyenne de x
var(x)	renvoie la variance de x
std(x)	renvoie l'écart type de x
median(x)	renvoie la médiane de x
sort(x)	trie les éléments de x
min(x)	renvoie le minimum (les, dans le cas multidimensionnel)
max(x)	renvoie le maximum (les, dans le cas multidimensionnel)
find(x)	renvoie les indices des éléments non nuls de x

9 - Fonctions math

sign(x)	signe de x (+1 ou -1)
sqrt(x)	racine de x
sin(x) ou sind(x)	sinus de x radian ou degré
cos(x) ou cosd(x)	cosinus de x radian ou degré
tan(x) ou tand(x)	cosinus de x radian ou degré
asin(x) ou asind(x)	arc sinus de x radian ou degré
acos(x) ou acosd(x)	arc cosinus de x radian ou degré
atan(x) ou atand(x)	arc tangente de x radian ou degré
exp(x)	exponentielle
log(x)	logarithme népérien
log10(x)	logarithme décimal
abs(x)	valeur absolue ou module
angle(x)	phase
real(x)	partie réelle
imag(x)	partie imaginaire

10 - Fonctions d'arrondi

round(x,n)	arrondit chaque élément de x à l'entier le plus proche avec n décimales
ceil(x,n)	arrondit chaque élément de x à l'entier supérieur ou égal le plus proche
floor(x,n)	arrondit chaque élément de x à l'entier inférieur ou égal le plus proche

11 - Affichage graphiques

11.1 - Fonctions de tracés

<code>plot(x,y)</code>	tracé de y en fonction de x
<code>plot3(x,y,z)</code>	tracé des coordonnées en 3D
<code>stem(x,y)</code>	tracé en bâtons
<code>stairs(x,y)</code>	tracé en escalier
<code>bar(x,y)</code>	barre de taille y en position x
<code>histogram(x)</code>	histogramme
<code>line(x,y)</code>	tracé de lignes
<code>surf(X,Y,Z)</code>	tracé d'une surface 3D (peinte)
<code>mesh(x,y,z)</code>	tracé d'une surface 3D

11.2 - Fonctions légendes

<code>title('texte')</code>	définit le titre de la figure
<code>xlabel('texte')</code>	légende sur x
<code>ylabel('texte')</code>	légende sur y
<code>zlabel('texte')</code>	légende sur z
<code>xlim([xmin xmax])</code>	définit les limites de l'axe x
<code>ylim([ymin ymax])</code>	définit les limites de l'axe y
<code>zlim([zmin zmax])</code>	définit les limites de l'axe z
<code>axis([xmin xmax ymin ymax zmin zmax])</code>	définit les limites des axes spécifiés
<code>legend('texte', ..., 'texte n')</code>	ajoute une légende
<code>grid</code>	active le quadrillage
<code>grid on</code> ou <code>off</code>	active ou désactive le quadrillage

11.3 - Gestion des affichages

<code>figure</code> ou <code>figure(nb)</code>	créer un figure ou la figure n°nb
<code>subplot(m,n,p)</code> ou <code>subplot(mnp)</code>	divise la figure actuelle en une grille m par n et créer des axes à la position p
<code>hold on</code> ou <code>off</code>	maintient ou non du prochain tracé sur la même figure
<code>shg</code>	affiche la figure courante au premier plan
<code>gca</code>	retourne les axes courants
<code>gcf</code>	retourne la figure courante
<code>close all</code>	ferme toutes les figures ouvertes
<code>close</code>	ferme la figure courante
<code>clf</code>	efface le contenu de la figure courante

12 - Boucles

12.1 - Boucle if

<code>if condition 1</code> <code>instruction 1</code> <code>elif condition 2</code> <code>instruction 2</code> <code>else condition 3</code> <code>instruction 3</code> <code>end</code>

12.2 - Boucle for

<code>for indice = valeur</code> <code>instruction</code> <code>end</code>
--

12.3 - Switch

```
switch expression
  case cas_expression 1
    instruction 1
  case cas_expression 2
    instruction 2
    ...
  otherwise
    instruction
end
```

12.4 - Boucle while

```
while condition
  instruction
end
```

13 - Définir des fonctions

13.1 - Fonctions

```
function [y1,...,yN] = nom(x1,...,xM)
% [y1,...,yN] = nom(x1,...,xM)
% Description de la fonction en commentaires
% Permet de générer la documentation pour les fonctions help / doc
% Descriptions des variables d'entrées, de sorties, etc.
% Auteurs, date
instructions
end
```

13.2 - Autour des fonctions

<pre>fun = @(x) x*exp(-x) fun(x) fun = @cos fun = @(x,y) cos(x)+sin(y)</pre>	'function handle' pour créer ou appeler des fonctions existantes ou anonymes appel de la fonction créée (autres exemples)
<pre>nargin nargin(<i>fonction</i>)</pre>	nombre d'entrées (utilisation dans la fonction uniquement) nombre d'entrées de la fonction précisée
<pre>nargout nargout(<i>fonction</i>)</pre>	nombre de sorties (utilisation dans la fonction uniquement) nombre de sorties de la fonction précisée

14 - Évaluation du temps

<pre>tic time = toc clock profile start profile viewer</pre>	démarre le chronomètre stop le chrono et récupère la valeur renvoie la date et l'heure actuels (fonction avancée) démarre l'évaluation de temps : permet d'obtenir de façon détaillée le temps d'utilisation des fonctions appelées, nombre d'appel, etc. affiche les résultats du profiler dans une fenetre
---	--