Université Badji Moukhtar- ANNABA

Faculté Sciences de la terre

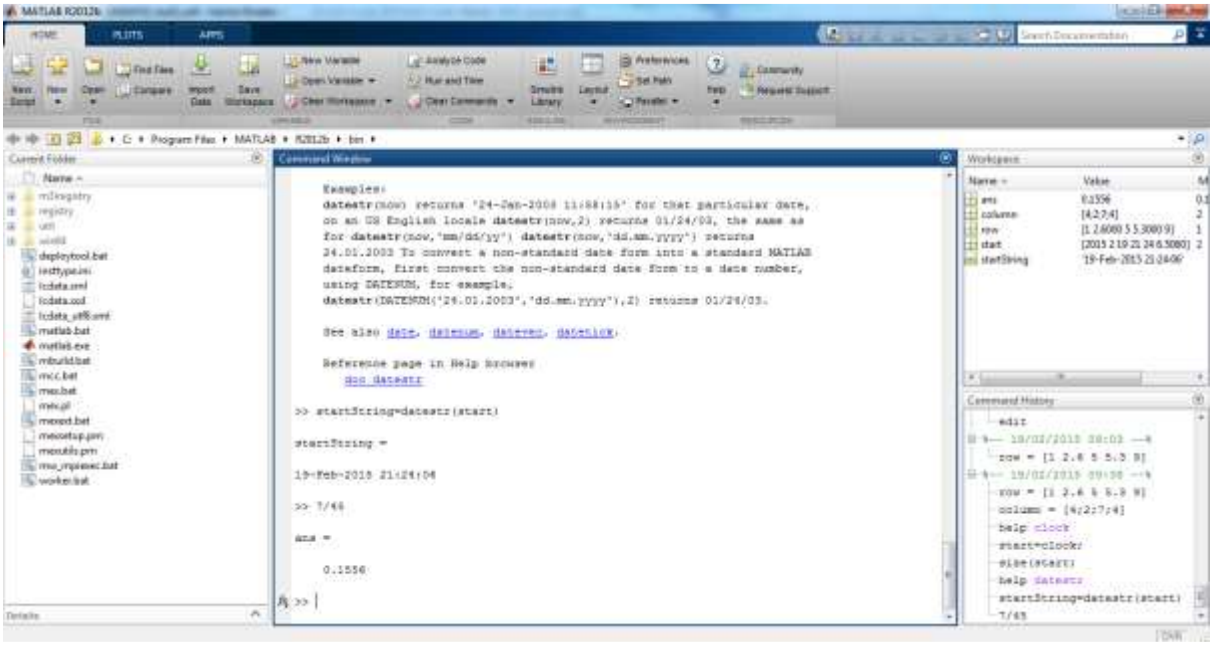
Département des Mines

Support TP

Introduction au MatLab

1. Introduction

- *MatLab (matrix laboratory)* est un système interactif de calcul numérique, qui dispose d'un *grand nombre de fonctions*, d'un *langage de programmation* et d'*outils de visualisation graphique.*
- *Matlab* est un langage *interprété*, c'est à dire qu'il exécute directement (sans compilation) les commandes, que vous entrez dans la fenêtre de commandes.
- La structure de base sur laquelle travaille *MatLab* est la *matrice.*
- ***Interface MatLab***



Au démarrage *MatLab* affiche plusieurs fenêtres sur l'écran, les fenêtres les plus importantes sont :

- ✓ *Command Window:* permit d'entrer les commandes *MatLab (Mode interactif).* L'utilisateur peut introduire des commandes des équations, après *les chevrons '>>',* sur les coté gauche de la fenêtre. Avec un simple *'Entrée'*, l'exécution s'affiche.
- ✓ *Command History:* Historique de commandes, contient tous les commandes, qu'ont été exécuté.
- ✓ *Current Folder:* Permet la visualisation de répertoire du travail courant.
- ✓ *Workspace:* Permet la visualisation des variables qu'ont été initialisée jusqu'à présent.

- ✓ *Help browser:* Ce navigateur nous permettre de parcourir l'aide MatLab *(Inteface graphique de l'aide).*

- *Matlab* fonctionne suivant deux modes, le *mode interactif* et le *mode programmation*.
- Matlab fait la distinction *entre minuscules et majuscules*.
- Commencer toute nouvelle page d'instructions *Matlab* par *clear all (clear var)*. Cette instruction détruit toutes les variables personnelles créées auparavant.
- *exit* ou *quit* : permet de quitter l'interface *MatLab.*
- Tout ce qui suit un *%* est considéré comme un commentaire (Dans le script).
- Deux méthodes sont possibles pour l'aide sur *MatLab*, soit en *mode texte*, soit via *l'interface graphique*. En mode texte, la commande *help* vous donne un aperçu des commandes disponibles.
- Pour avoir de l'aide directement sur une commande, entrez *help commande* :

>> *help sin* (Exmple)

- Chercher une commande sans avoir une idée sur le nom de cette commande, entrez *lookfor* : >> *lookfor cosinus*

2. Généralités sur MatLab

2.1. Créations et Manipulations dans l'interface de Commandes

Dans le mode interactif, on peut écrire directement :

>> 10 → ans = 10

a=4 *; (Le point-virgule) ;* (a= 4 ; b= 5)

b=pi (π)

clear a clear all

>> s='Hello' → s = Hello

>> s(2) → ans = e

Vecteurs

X=[1:20] *vecteur de lignes*

Matrices

A=[1 2 3;4 5 6; 7 8 9]

>> A(2,1) *(2eme ligne 1ere colonne)*

ans =

4

2.2. **Les matrices**

2.2.1. **Taille d'une matrice et déterminant** : S = size(A), d= det (A)

Exemple : calculer la taille de la matrice suivante :

$$A = \begin{pmatrix} 1 & 2 & 3 & 0 \\ 4 & 5 & 6 & 6 \\ 0 & 8 & 7 & 9 \end{pmatrix}$$

S= [3 4] ; d= det (A)

2.2.2. **Le nombre de lignes et colonne dans une matrice** :

Exemple : A =la même matrice ;

[m,n] = size(A)

m = 4

n = 3

A(L,C) : (ligne, colonne)

2.2.3. **Les entrées et sorties**

Utilisation de help pour : input, disp, sprintf

2.2.3.1. **Lecture de données :**

Exemple 1 :

A= input ('tapez un nombre \n') résultat

B= input('entrez une phrase', 's') Chaine de caractère

2.2.3.2. **Affichage de données :**

***Exemple2* :** disp('afficher le déterminant de la matrice A')

***Exemple3* :** disp(det(A))

Exemple4 : disp (['le déterminant de A est :',num2str(det(A))])

Exemple 5 :

```
detrA = 'Le déterminant de A est :';

D = det (A);

X = [detrA, num2str(D)];

disp(X)
```

Exemple 6 : Sprintf (Format, Arguments)

```
formatSpec = "On est le: %s,%d";

A1 = 'Mardi';

A2 = 28 01 2020;

str = sprintf(formatSpec,A1,A2)
```

Exemple 5 :

```
C = π; % changer le variable et le nombre de chiffre après la virgule

str= sprintf('%0.5f',C)
```

Les principaux types de formats d'écriture sont :

%d entier %5d : entier de longueur 5, ex : 34562

%f réel %5.2f : réel de longueur 5 avec 2 chiffres après la virgule

%e exponentiel %10.8e : nombre de la forme -21.01e+05

%g réel double précision %e et %f

%s chaine de caractères

3. Graphisme et Courbes

Pour tracer le graphique d'une fonction, vous devez suivre les étapes suivantes :

-Définissez x, en spécifiant la plage de valeurs pour la variable x, pour laquelle la fonction doit être tracée,

-Définissez la fonction, y = f (x)

-Appelez la commande *plot* : plot (x, y)

Exemple 1 : tracer f(x) = x. De 0 à 100 ; Pas 5

```
>> x = [0 :5 :100] ;
Y= x ;
Plot(x,y)
```

Exemple 2 : tracer f(x) = x^2. De -100 à 100 ; Pas 20

Exemple 3 : tracer d(x) = cos πx.

```
>> X = [0:0.1:6]; Y = cos(X*pi);
figure(1);
plot(X,Y)
```

4. Programmation

4.1 Fonctions

help function

Utilisé de cette manière, les scripts permettent simplement de mémoriser une suite de commande. Pour pouvoir faire des programmes modulaires, vous devez utiliser des fonctions. La syntaxe d'une fonction *Matlab* est la suivante :

function [args1,args2,...] = nomfonction(arge1,arge2,...)

Instructions

end

args1,args2,... sont les arguments de sortie de la fonction et peuvent être de n'importe quel type

arge1,arge2,... sont les arguments d'entrée de la fonction et peuvent être de n'importe quel type

instructions est un bloc d'instructions quelconque devant affecter les arguments de sortie *args1,args2,...*

Lorsqu'il n'y a qu'un seul argument de sortie, on peut utiliser la syntaxe plus simple :

function args = nomfonction(arge1,arge2,...)

Exemple : Ecrire une fonction pour calculer (t_1, t_2)= f(x,y,z)

t_1= x^2+3y-9z ; t_2 = y^6+2z

Solution :

function [t1,t2]= fonc (x,y,z)
t1= x^2+3*y-9*z ; t2=y^2+2*z ;
end
>> [a,b]= fonc (3,1,2)
a = -6 / b=5

4.2. Structures de contrôle

La programmation fait souvent usage de *tests conditionnels ou de boucles*. Toutes ces structures sont disponibles en *Matlab* :

Utiliser le help pour : if, switch, while, for, break

♦ Syntaxe du test (**if**)

```
if expression booléenne
   instructions
end
```

```
if expression booléenne
   instructions
else
   instructions
end
```

```
if expression booléenne
   instructions
elseif expression booléenne
      instructions
else
   instructions
end
```

♦ Syntaxe du branchement (**switch**)

```
switch expression          (expression est un scalaire ou une chaîne de caractères)
   case value1
      instructions         (instructions effectuées si expression=value1)
   case value2
      instructions
   ...
   otherwise
      instructions
end
```

♦ Syntaxe de boucle (**while** et **for**)

```
while expression
   instructions
end
```

```
for indice=debut:pas:fin   (si le pas n'est pas précisé, par défaut il vaut 1)
   instructions
end
```

Pour sortir d'un test ou d'une boucle, on utilise la commande **break** (voir l'aide en ligne).

Break : qui permet de sortir d'une boucle vent la fin.

Ces structures font appel aux tests mentionner dans les tableaux précédents : *tableaux (Opérateurs relationnels et Operations Logique, Instructions de contrôle) pages (09, 10)* successivement.

Exemple1 :

```
for i=1:10;
 y1(i)=i^2;
 end;
  y1
```

Exemple2 :

```
v(0)=1;v(1)=1;
i=1; while v(i)<=100
i=i+1;
v(i)=v(i-1)+v(i-2);
i
```

Exemple3 :

```
rep = input('Votre reponse (Allemagne, Italie, USA) :');
switch rep
case {'Allemagne','A'},
disp('liste de voitures allemandes: Audi et porsche');
case {'Italie','I'}
disp('On a : Alfa Romeo');
case {'USA','U'},
disp(' on parle de JEEP');
end
```

clc - clears the command window

5. Applications

Savoir si un nombre impaire ou non :

```
function Exercice_N1()
a=input('Please choose a number. a =')
 if (mod(a,2)==0)
    sprintf('%d is an odd number.',a)
  else
    sprintf('%d is an even number',a)
  end
end
```

Changer la valeur de deux variables

```
function Exercice_N2()
clc
 a = input('Please choose 1st number. a =');
 b = input('Please choose 2nd number. b =');
 temp=a;
 a=b;
 b=temp;
 sprintf('a = %d',a)
 sprintf('b = %d',b)
end
```

6. Résumé des commandes

Gestions des commandes et des fonctions	
help	aide
what	Listing du nom des M_files présents
type	impression d'un M_file
lookfor	recherche d'une entrée dans le help
which	localise les fonctions et fichiers
demo	lance la démonstration
path	Défini les chemins d'accès aux fichiers et fonctions
cedit	paramètres d'édition d'une ligne de commande
whatsnew	affiche les fichiers README de la toolbox
info	information sur MATLAB et The MathWorks
why	renvoie une réponse aléatoire non 'neutre'

Gestion des variables et de l'espace de travail	
who	affiche les variables courantes
whos	affiche les variables courantes, format long
save	Sauve l'espace de travail sur disque
load	restaure l'espace de travail à partir du disque
clear	efface les variables et fonctions de la mémoire
pack	réorganise la mémoire
size	renvoie la taille d'une matrice
length	renvoie la longueur d'un vecteur
disp	affiche une matrice de texte

Fenêtre de commande MatLab	
clc	efface la fenêtre de commande
home	curseur en haut de l'écran
format	définit le format d'affichage
echo	affiche les instructions exécutées par un script
more	contrôle de l'affichage paginé

Commandes système	
cd	change le directory courant
pwd	affiche le directory courant
dir, ls	liste les fichiers
delete	suppression de fichiers
getenv	renvoie la variable d'environnement
!	appelle et exécute une commande système
diary	sauvegarde le texte d'une session MATLAB

Démarrer et quitter MatLab	
quit, exit	quitter MATLAB
startup	M_file de lancement de MatLab
matlabrc	M_file principal de lancement

Opérateurs sur les Matrices		Opérateurs sur les Tableaux	
+	addition	+	addition
-	soustraction	-	soustraction
*	multiplication	.*	multiplication
^	puissance	.^	puissance
/	division à droite	./	division à droite
\	division à gauche	.\	division à gauche
'	transpose conjugué		
.'	transpose		
kron	produit de Kronecker		

Intégration numérique	
quad	intégrale de Simpson
quad8	intégrale de Newton-Cotes
trapz	méthode des trapèzes

Opérateurs Relationnels		Opérateurs Logiques	
<	inférieur à	&	et
>	supérieur à	|	ou
<=	inférieur ou égal à	~	non
>=	Supérieur ou égal à	xor	ou exclusif
==	égal à		
~=	différent de		

Matrices prédéfinies	
zeros	matrice de 0
ones	matrice de 1
eye	matrice identité
diag	matrice diagonale
toeplitz	matrice de Toeplitz
magic	carré magique
compan	matrice compagnon
linspace	vecteurs linéairement espacés
logspace	vecteurs logarithmiquement espacés
meshgrid	grille pour les graphiques 3D
rand	nombres aléatoires à répartition uniforme
randn	nombres aléatoires à répartition normale
hilb	Hilbert
invhilb	inverse de Hilbert (exact)
vander	Vandermonde
pascal	Pascal
hadamard	Hadamard
hankel	Hankel
rosser	matrice test pour le calcul des valeurs propres
wilkinson	matrice test pour le calcul des valeurs propres
gallery	deux matrices test spéciales

Variables prédéfinies	
ans	réponse à une expression sans assignation
eps	précision de la virgule flottante
realmax	plus grand nombre flottant
realmin	plus petit nombre flottant positif
pi	π
i, j	[√(-1)]
inf	∞
NaN	Not a Number
flops	nombre d'opérations flottantes par seconde
nargin	nombre d'arguments d'entrée d'une fonction
nargout	nombre d'arguments de sortie d'une fonction
computer	type du calculateur

Durées et Date	
date	date courante
clock	horloge
etime	durée d'exécution
tic, toc	affiche le début et la fin d'exécution
cputime	temps CPU écoulé

Caractères spéciaux	
=	assignation
[ ]	définition de matrices ou vecteurs;
( )	gère la priorité des opérations arithmétique
.	point décimal
..	directory parent
...	indique une ligne suite
,	séparateur d'arguments ou d'instructions
;	fin de lignes (matrices) ou suppression de l'affichage
%	commentaires
:	manipulation de sous matrices ou génération de vecteurs

Manipulation de Matrices	
diag	création ou extraction de la diagonale
rot90	rotation de 90°
fliplr	retournement gauche-droit
flipud	retournement haut-bas
reshape	redimensionnent
tril	partie triangulaire inférieure
triu	partie triangulaire supérieure
.'	transposition
:	conversion matrice → vecteur

Sauvegarde et copie graphique	
print	imprime ou sauve dans un fichier
printopt	configuration de l'imprimante
orient	orientation paysage ou portait

Fonctions trigonométriques	
sin, asin, sinh, asinh	
cos, acos, cosh, acosh	
tan, atan, tanh, atanh	
cot, acot, coth, acoth	
sec, asec, sech, asech	1./cos(z), acos(1./z), 1./cosh(z), acosh(1./z)
csc, acsc, csch, acsch	1./sin(z), asin(1./z), 1./sinh(z), asinh(1./z)

Fonctions prédéfinies	
bessel	fonction de Bessel
beta	fonction beta
gamma	fonction gamma
rat	approximation par un rationnel
rats	format de sortie pour rat
erf	fonction erreur erf
erfinv	inverse de erf
ellipke	intégrale elliptique complète
ellipj	fonction elliptique de Jacobi
expint	fonction intégrale exponentielle pour n=1
log2	logarithme base 2 ou décomposition mantisse, exposant
pow2	calcule 2 puissance y

Analyse de données par colonne	
max	valeur max
min	valeur min
mean	valeur moyenne
median	valeur médiane
std	écart type
sort	tri en ordre croissant
sum	somme des éléments
prod	produit des éléments
cumsum	vecteur des sommes partielles cumulées
cumprod	vecteur des produits partiels cumulés
hist	histogramme

Animations	
moviein	initialise l'espace mémoire pour l'animation
getframe	enregistre une image pour l'animation
movie	joue l'animation

Décompositions et factorisations de Matrices	
inv	inversion
lu	décomposition LU
rref	réduction de lignes
chol	factorisation de Cholesky
qr	décomposition QR
nnls	moindres carrés non-négatifs
lscov	moindres carrés avec covariance connue
null	noyau
orth	orthogonalisation
eig	valeurs et vecteurs propres
hess	forme de Hessenberg
schur	décomposition de Schur
cdf2rdf	forme complexe diagonale vers forme réelle diagonale par blocs
rsf2csf	forme réelle diagonale par blocs vers forme complexe diagonale
balance	mise à l'échelle pour le calcul des valeurs propres
qz	valeurs propres généralisées
polyeig	polynôme aux valeurs propres
svd	décomposition en valeurs singulières
pinv	pseudo-inverse

Traitement de signal	
corrcoef	coefficients de corrélation
cov	matrice de covariance
filter	filtrage monodimensionnel
filter2	filtrage bidimensionnel
cplxpair	tri en paires complexes
unwrap	suppression des sauts de phase
nextpow2	puissance de 2 immédiatement supérieure
fft	FFT monodimensionnel (fréquences de 0 à 1)
fft2	FFT bidimensionnel
ifft	FFT inverse
ifft2	FFT inverse
fftshift	FFT (fréquences de -1/2 à 1/2)

Conditionnement	
cond	suivant norme L_2 ($[(\lambda_{max})/(\lambda_{min})]$)
rcond	≈ 1 bon, ≈ 0. mauvais
condest	suivant norme L_1
norm	normes L_1, L_2, L_p et L_∞
normest	estimateur de la norme L_2
rank	rang

Université Badji Moukhtar- ANNABA

Faculté Sciences de la terre

Département des Mines

Fonctions logiques	
exist	teste l'existence d'une variable ou d'une fonction
any	vrai si un élément est vrai
all	vrais si tous les éléments sont vrais
find	cherche l'indice des éléments non nuls
isnan	vrai si l'élément n'est pas un nombre
isinf	vrai pour tout élément infini
finite	vrai pour tout élément fini
isieee	vrai si la représentation est au format IEEE
isempty	vrai pour une matrice vide
issparse	vrai pour une matrice creuse
isstr	vrai pour une chaîne de caractères
strcmp	comparaison de deux chaînes

Instruction de contrôle	
if	test conditionnel
else	complète if
elseif	complète if
end	terminaison de if, for et while
for	instruction de répétition avec compteur
while	instruction de répétition avec test
break	interrompt une boucle for ou while
return	retour
error	affiche un message et interrompt l'exécution

Instructions spécifiques	
input	indicateur d'attente d'entrée
keyboard	considère le clavier comme un fichier script
menu	génère un menu de choix pour l'utilisateur
pause	attente
function	définition de fonction
eval	exécute un chaîne de caractère
feval	exécute une fonction définie dans une chaîne
global	définit les variables comme globales
nargchk	valide le nombre d'arguments d'entrée

Traitement du son	
saxis	modification de l'échelle d'amplitude
sound	convertit un vecteur en son
auread	lit un fichier audio au format SUN
auwrite	écrit un fichier audio au format SUN
lin2mu	conversion loi linéaire vers loi μ
mu2lin	conversion loi μ vers loi linéaire

Textes et chaînes	
string	à propos des chaînes dans MATLAB
abs	convertit une chaîne en valeur numérique
blanks	une chaîne d'espaces
eval	évalue une chaîne contenant du code MATLAB
num2str	convertit un nombre en chaîne
int2str	convertit un nombre entier en chaîne
str2num	convertit une chaîne en nombre
isstr	vrai si l'élément est une chaîne
strcmp	comparaison de chaînes
upper	conversion en majuscule
lower	conversion en minuscule
hex2num	convertit une chaîne hexadécimale en flottant
hex2dec	convertit une chaîne hexadécimale en entier
dec2hex	convertit un entier en une chaîne hexadécimale

Mise au point (debug)	
dbstop	met un point d'arrêt
dbclear	supprime un point d'arrêt
dbcont	reprend l'exécution
dbdown	change le contexte local
dbstack	affiche qui appelle qui
dbstatus	liste des points d'arrêt
dbstep	exécute une ou plusieurs lignes
dbtype	affiche un M_files avec lignes numérotées
dbup	inverse de dbdown
dbquit	sortie du mode debug

Fonctions mathématiques élémentaires	
abs	valeur absolu ou module
angle	argument d'un complexe
sqrt	racine carrée
real	partie réelle
imag	partie imaginaire
conj	complexe conjugué
gcd	PGCD
lcm	PPCM
round	arrondi à l'entier le plus proche
fix	troncature
floor / ceil	arrondi vers $-\infty$ / vers $+\infty$
sign	signe de
rem	reste de la division
exp	exponentiel
log / log10	log népérien / log décimal

Pour plus d'information visitez : https://www.tutorialspoint.com/matlab/index.htm

Contactez SAADI Hayet : saadihayet_inf@yahoo.fr