1. Prise en Main de Matlab :

Matlab est l'abréviation de MATrix LABoratory

1.1. Une session matlab

Ouvrir une fenêtre de commande dans laquelle il y a un symbole (>>) qui indique que matlab attend des instructions+commandes Matlab (exp : quit)

Chaque ligne d'instructions doit se terminer par un retour chariot. La commande pour quitter matlab est quit.

>> en début de ligne constituent le prompt de Matlab

1.2. L'espace de travail (workspace)

Comme tout langage de programmation, le Matlab gère les données sous forme de variables sauf que la représentation des variables qui est différente dans Matlab.

Le workspace sert d'un espace de stockage des variables utilisées lors d'une session de travail et peuvent être utilisées dans les calculs subséquents.

Pour obtenir la liste des variables actives de l'espace de travail
On dispose des commandes who et whos. La commande who affiche le nom des variables actives. La commande whos donne plus d'informations : le nom, la taille du tableau (nombre de lignes et de colonnes) associé, l'espace mémoire utilisé (en Bytes) et la classe des données (principalement double array ou char).

Effacer les variables de l'espace de travail
''clear nom_var'' ou ''clear'' pour effacer toutes les var stockées

Sauvegarder et charger un espace de travail dans un fichier pour une utilisation ultérieure :
save nom-fic load nom-fich

Syntaxe d'une ligne d'instructions

>> A = [8 1 6; 3 5 7; 4 2 9]; (n'affiche pas de résultat ans) >> A A = 8 1 6 3 5 7 4 9 2 >> A*A;	>> ans ans = 91 67 67 67 91 67 67 67 91 >>
>> B = [1 3; 4 2]; B*B ans =	

Activité (A):

- Dans la fenêtre de commandes, tapez ce qui suit :

>> x=2*pi/3; y=sin(x); z=cos(x);
>> A = [1 3; 4 2]; B = A*A;

>> t = ’bonjour’;

- Afficher la liste des variables
- Afficher le type et la taille de chaque variable
- Afficher la valeur de la variable z
- Tapez : >> 2*3;
- Affichez la liste des variables ; que remarquez-vous ?
- Effacer les variables x, y et t, vérifier ensuite que les variables ont été effectivement effacées.
- Sauvegarder l’espace de travail sous le nom ‘’myspace’’, puis quitter Matlab
- Ouvrez à nouveau Matlab
- Charger l’espace de travail ‘’myspace’’
- Afficher toutes les variables ainsi que leurs valeurs se trouvant dans l’espace de travail ‘’myspace’’
- Sauvegardez à nouveau l’espace de travail, mais uniquement les variables A et z.
- Quitter Matlab, puis charger l’espace de travail.

1.3. Types de données et variables

Attention, matlab différencie majuscules et minuscules

L’utilisation de variables avec matlab ne nécessite pas de déclaration de type ou de dimension. Le type et la dimension d’une variable sont déterminés de manière automatique à partir de l’expression mathématique ou de la valeur affectée à la variable.

Pour matlab toute variable est considérée comme étant une matrice d’éléments d’un type donné. matlab différencie trois formes particulières de matrices. scalaire est une matrice de dimension 1*x*1, un vecteur colonne de dimension *n* est une matrice nx1, un vecteur ligne de dimension *n*, une matrice 1*xn*.

Les scalaires se déclarent directement, par exemple :
>> x = 0;
>> a = x;
Les vecteurs ligne se déclarent de la manière suivante :
>> V_ligne = [0 1 2]
V_ligne =
0 1 2
Pour les vectrices colonnes, on sépare les éléments par des points-virgules :
>> V_colonne = [0;1;2]
V_colonne =
0
1
2

>> V1 = [1 2];
>> V2 = [3 4];
>> V = V1 + V2 % addition de vecteurs
V =
4 6
>> V = V2 - V1 % soustraction de vecteurs
V =
2 2
>> V = 2*V1 % multiplication par un scalaire
V =

2 4

La Matrice :

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

Tapez : >> C = [1 2 3; 4 5] ; qu'est ce que vous remarquez ?

Matrices spéciales

eye(n) : la matrice identité
ones(m,n) : la matrice à m lignes et n colonnes dont tous les éléments valent 1
zeros(m,n) : la matrice à m lignes et n colonnes dont tous les éléments valent 0
rand(m,n) : une matrice à m lignes et n colonnes dont les éléments sont générés de manière aléatoire entre 0 et 1

Le type chaîne de caractères

Une chaîne de caractères est un tableau de caractères, il est possible de manipuler chaque lettre de la chaîne en faisant référence à sa position dans la chaîne.

Manipulations d'une chaîne de caractères :
>> ch = 'abc' ; % affectation
>> strcat(ch1,ch2) % concaténation de deux chaines de caractères.
>> ch(i), % le ième élément de la chaine 'ch'
>> ch(i : j), % les éléments entre i et j

Activité (B):

- Tapez :

>> ch1 = 'bon'
>> ch2 = 'jour'
>> whos

- Constituez le mot 'bonjour' à partir de ch1 et ch2
- Tapez : >> ch3 = 'soi';
- Constituez le mot bonsoir à partir des deux variable 'ch' et 'ch3'

Le type logique

>> tst = (x==y);
>> if tst, disp('x est egal a y '), else disp('x est different de y '), end
x est different de y
>> whos

>> bool = (a>b)
bool = 0

Opérateur	Description
$\sim$ a	NOT - retourne 1 si a égal 0, 1 si a égal 0
a == b	retourne 1 si a égal b, 0 autrement
a < b	retourne 1 si a est plus petit que b, 0 autrement
a > b	retourne 1 si a est plus grand que b, 0 autrement
a <= b	retourne 1 si a est plus petit ou égal à b, 0 autrement
a >= b	retourne 1 si a est plus grand ou égal à b, 0 autrement
a $\sim$=b	retourne 1 si a est différent de b, 0 autrement

1.4. Instructions de contrôle

Boucle FOR : parcours d'un intervalle

for indice = borne_inf : borne_sup % incrémentation à 1 de l'indice
séquence d'instructions
end

ou

for indice = borne_inf : valeur d'incrémentation (vi) : borne_sup % incrémentation à vi de l'indice
séquence d'instructions
end

exp :
>> for r=1.1:-0.1:0.75
disp(['r = ', num2str(r)]);
end
r = 1.1
r = 1
r = 0.9
r = 0.8
>>

Boucle WHILE : tant que . . . faire

while expression logique
séquence d'instructions
end

L'instruction conditionnée IF

Syntaxe :
if expression logique
séquence d'instructions 1
else
séquence d'instructions 2
end

1.5. Entrées-sorties

N = input('Nombre de boucles désirées >'); % entrée interactive
disp(N) % affiche la valeur de N

fileID=fopen('file.txt','r') % ouvrir un fichier en lecture ('r'),
fileID=fopen('file.txt','w+') % ouvrir ou créer un fichier en lecture et écriture (écraser le contenu)
C=fscanf(fileID,'%c') % lire le contenu du fichier ouvert et le convertir en chaine de caractère (%c)
fprintf(fileID,txt) % ecrire le contenu txt dans le fichier spécifié
fclose(fileID) % fermer le fichier

Scripts et fonctions

Il est possible d'enregistrer une séquence d'instructions dans un fichier (appelé un «M-file »)
On distingue 2 types de M-file, les fichiers de scripts et les fichiers de fonctions.

Fichiers FUNCTION :

function a = ma_function(x,y)
a = x + y;
b = x * y;
produit la sortie suivante :
>> a = ma_fonction(4,2)
a = 6

function [a,b] = ma_function(x,y)
a = x + y;
b = x * y;
pour obtenir :
>> [a,b] = ma_fonction(4,2)
a = 6
b = 8

Activité (C):

Ecrire une fonction qui calcule la factorielle d'un nombre en utilisant la boucle for ensuite en utilisant la boucle while :