I. Résumé du chapitre I : Initiation à la programmation (Matlab)

I.1. Objectifs et Compétences Visées

A travers ce résumé, l'étudiant(e) peut:

• Langage Matlab :

- Avoir une connaissance de la syntaxe du langage Matlab.
- Avoir une connaissance des fonctions classiques vectorielles et matricielles de Matlab.
- Se servir de l'aide en ligne du Matlab.
- Algorithme :
 - Concevoir les différentes étapes d'un algorithme.
 - Organiser des petits programmes (scripts, fonctions, boucles).
 - Ecrire des programmes lisibles par des gens extérieurs.
 - Valider et tester ses programmes de façon autonome.

• Méthodes Numériques :

- Saisir le raisonnement des méthodes classiques suivantes : intégration numérique, résolution des équations non linéaires, résolution des équations différentielles ordinaires et résolution numérique des systèmes d'équations linéaires.
- Connaitre les avantages et les limites de ces méthodes.
- Maitriser quelques exemples d'application et d'utilisation de ces méthodes.

I.2. Lancement de Matlab

On lance Matlab, dans la fenêtre qui apparait on voit (>>) qui indique que Matlab attend une commande.



- 1 : Command Window : fenêtre principale pour l'exécution des instructions
- 2: Command History: historique des commandes
- 3 :Workspace :affichage des variables
- 4 : Current Folder : liste des fichiers précédemment ouverts

Pour faire des calculs simples ou des affichages de graphes, on travaille dans la fenêtre de commande (« command window »)

Dans cet exemple « a » est un vecteur ligne ; en frappant « b=a+2 » on crée une seconde vectrice ligne « b » en ajoutant 2 à chacun des éléments de « a ».

On cherche à tracer la courbe b=f(a) en utilisant la fonction « plot », le résultat s'affiche dans une nouvelle fenêtre : en abscisse on obtient le numéro d'indice et en ordonnée la valeur de la composante correspondante de « b ».





Si on écrit un point-virgule à la fin de la ligne de commande, comme : >> b = a + 2; Alors la commande est exécutée mais le résultat ne sera pas affiché Si on veut entrer une séquence longue d'intervalles <u>réguliers</u> sans entrer toutes les valeurs on utilise les deux points « : »



Il est possible d'appliquer des fonctions sur la séquence « a » :



Cela affiche le sinus de la séquence :



En abscisse il y a un simple numéro : la position de l'élément a dans le calcul (on notera que Matlab commence les numéros d'indice à 1 et non à zéro, le cas de Fortran). Pour se déplacer dans la liste des instructions, on peut utiliser es flèches du clavier :

Si on frappe « Entrée », c'est la ligne sélectionnée qui est exécutée.

I.3. Scripts et Fonctions

Pour enregistrer un programme dans un fichier dont on l'exécutera lorsqu'il sera prêt, on click sur « File » puis on choisit « New Script » ou « New Function ».

Ceci fait apparaître une nouvelle fenêtre "Editor" dans laquelle on éditera le texte du programme.

New	- P.	Script %N	elourme/Documents/MATLAB	····· ···· ····
Open	80	Function		
Close Command Window	×w	Class	Command Window	× 7 +1
Import Data		Figure		TH 150 1
Save Workspace As	æs.	Variable		Name +
Set Path		Model		
Preferences	ж,	CUI	-	
Page Setup	PMP	Deployment Project		
Print	₩P	Code Generation Project		
Print Selection	CMP	Simuline Project		
1 /Users/ODE/Exercice2	.m			
2 /Users/ODE/Exercice1	.m			
4 ANT/onde rue droite	m 4 m			
EXIT MATLAB	₩Q			- 1 mm
				× + +!
				v:
				hs
				v. v:
				v
				h
				м
				M
				hs
				M
				- B ()
· · · · · · · · · · · · · · · · · · ·				

Script	Fonction		
 Suite d'instructions Pas de paramètre d'entrée Ne revoie aucune valeur 	 Peut prendre des arguments Retourne une ou plusieurs valeurs N'accède pas aux variables de l'environnement 		
 Crée ou modifie des variables d'environnement Possible d'appel d'autre scripts ou fonctions 	 Les variables locales inaccessibles depuis l'extérieur Contrainte syntaxique : seule la fonction qui porte le nom du M.File qui est accessible 		
Command Window			



Une fois qu'on aura fini programme, il faut le sauvegarder : on choisit par exemple « save as » et le nom du fichier qui doit nécessairement avoir l'extension « .m » :

Matlab reconnaitra par la suite cette extension lorsqu'on lui demandera l'exécution dans la fenêtre de commande, par exemple pour exécuter le programme enregistré dans le fichier « PremierExemple.m » il suffit juste de l'écrire dans la fenêtre « Command Window » comme suit :

📣 MATLAB R20	115a				- 8 ×				
HOME	PLOTS APPS	EDITOR	PUBLISH VIEW	🛃 🗃 🖌 🕸 💷 🗇 🔄 🔁 🕐 Search Do	cumentation 🔎 🔼				
🕹 🗂 🖥	Find Files 😔 😒	Insert fx							
New Open Sa	Ve	Indent 5	Select File for Save As	×					
F	FILE NAVIGATE	EDIT	🖌 🔶 🗉 🛧 📙 « MATLAB Pro 🔹 R2015a 🗧	✓ Ŏ	and the second se				
← → 🖸 🕅 Current Folder	C: Program Files N	IATLAB + MATLAB	Organiser 👻 Nouveau dossier	₩== ▼ (2)	 Q ▼ ▼ 				
🗋 Name 🔶		Unti	OneDrive Nom	Modifié le Type					
 appdata bin bugreport 		^ 1 ~	a Ce PC appdata bin bin bigreport	18/01/2020 12: Dossier de fichi 18/01/2020 12: Dossier de fichi 18/01/2020 12: Dossier de fichi	-				
Details 🗸 🗸		~	Dictiments dient dient etc Musique examples Objets 3D extern	18/01/2020 12: Dossier de ferei 18/01/2020 12: Dossier de f 18/01/2020 12: Dossier de f 18/01/2020 12: Dessier de f	u programme				
			Téléchargemer help Jidéos Disoue local (ć Y <	18/01/2020 12 Herro1/2020 12 Dossier de fichi					
			Type : MATLAB Code files (*.m)	~					
Workspace		•							
Name 🔶	Value	Comman	🔥 🔺 Masquer les dossiers	Enregistrer Annuler	•				
a [1 2 3 4 6 4 3 4 5] b [3 4 5 6 8 6 5 6 7]		New to	New to MATLAB? See resources for <u>Getting Started</u> .						
		>> p: >> g: ft; >>	3 4 5 6 8 6 5 6 Lot(b) cidion	7	J				
4444 J				script	In 1 Col 13				

>> PremierExemple

I.4. Comment trouver et exécuter des programmes déjà sauvegardés ?

Lorsque le fichier script est sauvegardé dans un dossier créé préalablement, on va le lancer par une simple recherche dans la rubrique "Open", après çà, on peut facilement le modifier et l'exécuter

📣 MATLAB R	2015a											ð X
HOME	PLOTS	APPS	EDITOR	PUBLISH	VEW						🖻 🕐 Search Documentatio	n 👂 🛛
New New Script +	Open Compare	Import S Data Work	ave space VARIABLE	Variable Var	Analyze Code Run and Time Clear Commands CODE	Simulink L Library		? Help	Community	rt		
🕈 🌩 🖬 🞘	C: 🕨 Prog	ram Files 🔸	MATLAB • MA	TLAB Productio	n Server 🕨 R2015	a 🕨						م -
Current Fold	er		• 🛛 🗹	Editor - Untitleo	d*							⊙×
Name	<u>~</u>			Untitled* 🛛	+							
🗷 📃 appdat	3		^ 1	[xi,yi]	meshgrid(-pi/2	:pi/30:pi	/2);					
⊞ bin ⊞ bugrop	ort		-		\sim —			1				
Detelle	011		×		Séle	oction	du					
Detana					oen							
				dossier								
								1				
	Select a file to view	v details										
Workspace			۲									
Name 🔺	Value		Co	mmand Windov	v							۲
a	[1 2 3 4 6	4345]	N	New to MATLAB? See resources for <u>Getting Started</u> .						×		
tt b	[34568	6567]		b =								^
				3 4	5 6	8 4	6 5 6	7				
				_								
				>> pict(b) >> grid on								
			fx,	>>								*
1111-										script	Ln 2	Col 1

I.5. Graphiques sous Matlab

Nous avons vu un exemple de tracé simple ; en voici un autre où on donne l'abscisse et l'ordonnée pour deux graphes >> x = [0:0.01:2*pi];





Pour rajouter un titre et une légende :

```
title('sinus et cosinus'); xlabel('x'); ylabel('f(x)')
legend('cos(x)','sin(x)',0) % le « 0 » place la légende à côté des
Dr. MODERRES MOURA
```



Pour afficher plusieurs figures sur le même écran, on utilise la fonction" subplot"

🚸 MATLAB								
<u>Eile E</u> dit <u>T</u> ext <u>C</u> ell T <u>o</u> ols De <u>b</u> ug <u>D</u> esktop <u>W</u> indow <u>H</u> elp								
🗅 😅 👗 ங 🛍 🕫 🖙 🎁 🎁 🖆 🛃 💡 🖾 "Program Files" 🔽 🤅								
Shortcuts 🖉 How to Add 💽 What's New								
🗅 🚅 🖩 👗 🖻 🗠 🖙 🎒 🗰 📴 🔻 🛛 💌								
427 + □ 1,□ ↓□ - 1.0 + ÷ 1.1 × % ² % ² √								
1 function sixgraphiques								
2 - t=[0:0.1:2*pi];								
3 - for casse=[1:6]								
- subplot(2,3,casse)								
plot(t,sin(casse*t));								
6 - end 💌								
📣 <u>S</u> tart sixgraphiques Ln 4 Col 23 OVR 🛒								



Exemples de commandes expliquées dans le manuel Matlab

grid
ylabel
bar

Dr. MODERRES Mourad

figure	step
disp	input

I.6. Variables et scalaires

- Pas de différents types de pour les nombres (Matlab fait les conversions)
- Pas de déclaration de variables

>> a=2; >> b=2.5; >> c=a*b;

- Liste des variables (commande who)

```
>> a=2;
>> b=2.5;
>> c=a*b;
>> who
Your variables are:
a b c
```

Suppression des variables (commande clear)

- » clear a : supprime la variable a
- \gg clear all : supprime toutes les variables.

: supprime l'écran de Command Window.

I.7. Tableaux et matrices

 \gg clc

I.7.1. Création des tableaux et des matrices :

Les vecteurs sont des tableaux à une ligne ou à une colonne, ses éléments sont séparés par des espaces ou par des virgules.

≫ *V*1 = [1, 2, -2, 6, 8] % vecteur ligne contenant : 1, 2, -2, 6, 8

 \gg *V*1 = [1 2 - 2 6 8] % vecteur ligne contenant : 1, 2, -2, 6, 8

≫ *V*2 = [1; 2; -2; 6; 8] % vecteur colonne contenant :
$$V_2 = \begin{pmatrix} 1 \\ 2 \\ -2 \\ 6 \\ 8 \end{pmatrix}$$

Quand les éléments sont séparés par des points-virgules, cela va créer des colonnes d'une une matrice :

Dr. MODERRES Mourad

 $M_1 = [1 \ 2 \ -2; \ 6 \ 8 \ 4]$ % une matrice (2 lignes, 3 colonnes): $M_1 = \begin{pmatrix} 1 & 2 & -2 \\ 6 & 8 & 4 \end{pmatrix}$

I.7.2. Taille d'un vecteur, dimension d'une matrice : (length, size)

 \gg length(V1) % affiche : 5

 \gg length(V2) % affiche : 5

 \gg size(V2) % affiche (5 1)

 \gg length(M1) % affiche (2 3)

I.7.3. Créer des progressions arithmétiques





Les fonctions : **linspace** et **logspace** permettent de créer des séquences régulièrement espacées.



I.7.5. Affichage

» disp(a) : pour obtenir un résultat déjà utilisé

» disp('c"est le résultat recheché ') : pour afficher un message

I.7.6. Entrée des variables

» input(' La valeur b est = ') : Le résultat de cette commande est l'affichage de message « Lavaleur de b= », le Matlab va attendre la saisie de la valeur 'b'

I.8. Les boucles

Les principales instructions de contrôle proposées par Matlab sont : for, if, while et switch. Elles fonctionnent quasiment comme leurs équivalents dans les autres langages de programmation.

Syntaxe de la boucle for :

for compteur=expression
Instructions
end

Syntaxe de la condition if :

```
if expression
Instruction 1
else
Instruction 2
end
```

Syntaxe de la boucle while :

while compteur=expression Instructions end

Syntaxe de la condition switch :

switch expression du choix case expression du cas instructions

... **case** expression du cas instructions

•••

otherwise

instructions

end