Université Batna-2
Faculté Technologie
Département d'Electronique
Licence 2 : Génie Biomédical-Télécommunication-Electronique- Automatique

TP Informatique 03

TP N°02 : Initiations à Matlab

1. Introduction

Ce TP a plusieurs objectifs :

1- D'initier l'étudiant à l'éditeur de commande de Matlab afin qu'il puisse créer un programme et l'enregistrer sur PC.
2- La maitrise des différentes commandes de représentations graphique ;
3- Utilisation des instructions de control (if, while et for) ;
4- Créer des fonctions propres au programmeur avec l'instruction Function

2. Représentation graphique des résultats

2.1 Présentation des commandes

Représentations de points dans le plan. Il existe plusieurs possibilités pour représenter un ensemble de points (x(i) ; y(i)). Les plus utilisées sont énumérées ci-dessous :

plot (x,y,'s')	tracé d'une courbe ou d'un nuage de points
bar(x,y,'s')	tracé sous forme d'un histogramme
stem(x,y,'s')	diagramme en bâtons
stairs(x,y)	tracé en escalier des valeurs discrètes
fplot	représente des fonctions
hist	trace des histogrammes

's' : est un paramètre facultatif constitué d'une chaine de caractères qui spécifique le type de tracé (couleur, différents tracés en pointillés, symboles pour le tracé de points). Par défaut, le tracé est continu.

2.2 Gestion de la fenêtre graphique.

hold on	les prochains tracés se superposeront aux tracés déjà effectués
hold off	le contenu de la fenêtre graphique active sera effacé lors du prochain tracé
clf	efface le contenu de la fenêtre graphique active
figure(n)	affiche ou rend active la fenêtre graphique numéro n
close	ferme la fenêtre graphique active
close all	ferme toutes les fenêtres graphiques
subplot (n,m,p)	partage la fenêtre graphique active en m x n espaces graphiques et sélectionne le p-ième.

2.3 Axes et légendes.

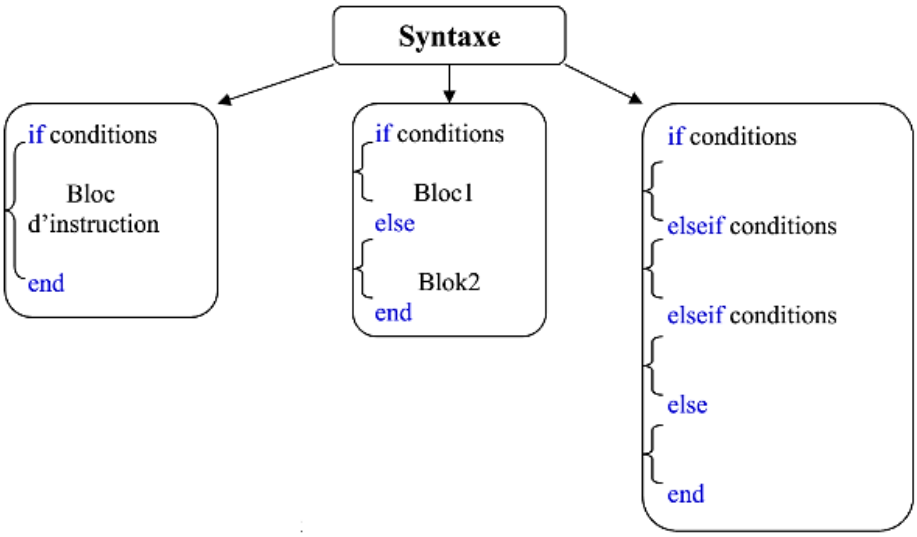
axis ([xmin xmax ymin ymax])	pour définir les échelles des axes
grid	quadrillage du graphique
title ('titre')	titre pour le graphique
xlabel ('titre')	légende pour l'axe des abscisses
ylabel ('titre')	légende pour l'axe des ordonnées
legend ('titre1','titre2',...)	légende pour chaque courbe du graphique
text (x, y, 'texte')	texte explicatif _a la position (x, y)
gtext ('texte')	texte positionné à l'aide de la souris

Programme 1 :

```
clc
close all
clear all
x= -pi : 0.1: 3*pi;
y=x.*sin(x);

fig1=figure;
fig1=plot (x,y)
clf

fig1=plot (x,y)
axis([-pi, 3*pi, -6, 9])
xlabel ('x')
ylabel ('y')
%title ('graphe de la fonction x sin(x) sur l''intervalle ['num2str(x(1)) ' , 'num2str(x (end)) ']')
title(['graphe de la fonction x sin(x) sur l''intervalle    ',  [num2str(x(1))  ,'    ' ,     num2str(x(end))] ])

fig2=figure;
fig2=plot(x, y, x, 2*y)

fig3=figure;
fig3=plot(x,[y; 2*y])

fig4=figure;
fig4=plot(x, y, 'r--',x,2*y, 'g+')

fig5=figure;
fig5=fplot (@(x) [x.*sin(x),2.*x.*sin(x)],[-pi,3*pi])

fig6=figure;
fig6=fplot (@(x)x.*sin(x),[-pi,3*pi],'b+')
hold on
fig6=fplot(@(x) 2*x.*sin(x),[-pi,3*pi],'yo')
hold off

t=0:0.1:2*pi;
fig7=figure;
fig7=plot(sin(t),sin(2*t))

fig8=figure;  fig8=plot(sin(t),sin(2*t),'c-')
```

3 Instructions de contrôles

3.1 Le teste conditionnel 'if'



Programme 2

```
   a = 4;
   b = 5;
   c = 2;
   delta =
   if delta > 0
disp('2 solutions réelles');
elseif delta == 0
disp('solution double');
else
disp('2 solutions complexes');
end
```

3.2 La boucle FOR . . . END.

Elle répète l'exécution d'un bloc d'instructions tant que l'indice de la boucle n'arrive pas à la valeur finale.

Syntaxe:

```
for k = valeur de départ : pas : valeur finale
        Bloc d'instructions
end
```

Programme 3:

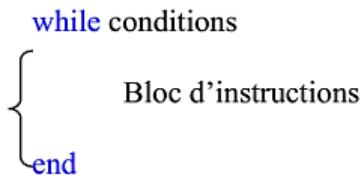
Calcul de somme de N premier élément de l'ensemble naturel

```
N = 10;
S = 0;
for k=1:N
    S = S+k;
end
```

3.3 La boucle WHILE . . . END

Elle répète un bloc tant que la condition d'arrêt n'est pas vérifiée.

Syntaxe :



Programme 4:

Calcul de produit de N premier élément de l'ensemble naturel

```
P = 1;
N = 4;
k = 1;
while k<=N
      P = P * k;
      k = k + 1;
end
P
```

4 Les fonctions

Il existe de nombreuses fonctions prédéfinies en Matlab, mais il arrivera forcément un moment où vous voudrez utiliser une fonction qui n'est pas définie. Heureusement, il est possible de définir ses propres fonctions et de s'en servir exactement comme les fonctions préexistantes.

Syntaxe

function [paramètres de sorties] = nom_ fonction (paramètres d'entrées)

```
function [ output_args ] = Untitled6( input_args )
%UNTITLED6 Summary of this function goes here
%   Detailed explanation goes here


end
```

Exemple :
La fonction qui calcule la moyenne des trois éléments :
function [m] = moy(a, b, c)
m = (a+b+c)/3 ;
end