

**Université Mohamed Boudiaf, M'sila**  
**Faculté de technologie**  
**Socle commun**

**Module : TP-informatique 3**  
**2<sup>ième</sup> année ST**

## **TP3- Initiation à MATLAB**

### **1. Introduction**

*Matlab* est un environnement puissant au calcul scientifique. Il intègre le calcul matriciel et l'analyse numérique dans ses fonctions de base. De plus son architecture graphique orientée objets permet la génération de graphiques d'excellente qualité.

*Matlab* tire son nom de l'anglais et constitue une contraction de **MAT**rix **LAB**oratory. C'est donc un logiciel qui a été développé pour traiter spécifiquement les problèmes nécessitant un formalisme matriciel important.

Le logiciel **MATLAB** était intuitivement codé en Fortran. Au fil des années, *Matlab* a bien sûr s'est enrichi d'un grand nombre de fonction, les domaines couverts ont augmenté et le langage C a remplacé le Fortran. De plus pour optimiser les performances du logiciel, de nombreuses routines internes ont directement été écrites en assembleur.

On ce conçoit plus aujourd'hui d'enseignements d'automatique de traitement du signal, ..., sans y associer des travaux pratiques sous *Matlab*.

### **2. Structure**

Il comprend le langage *Matlab* et intègre de nombreuses fonctions mathématiques codées en langage *Matlab* sous forme *.m* (les *m-files*) par exemple, la fonction *abs.m* permet d'obtention de la valeur absolue d'un nombre.

Les bibliothèques de fonction de *Matlab* spécialisées permettent de personnaliser l'environnement de travail.

### **3. Commande Help**

La commande *help* est l'une des commandes les plus utiles pour commencer l'apprentissage de *Matlab*, elle donne accès à l'aide en ligne. Pour l'utiliser, on a le choix entre plusieurs méthodes.

**Exp.**

```
>> help plot
PLOT Linear plot.
    PLOT(X,Y) plots vector Y versus vector X. If X or Y is a matrix
>> help graph2d
Two dimensional graphs
```

**Remarque :** La connaissance du mot-clef anglais correspond à ce que l'on cherche.

### **4. Editeur ligne ou fichier**

**4.1. Editeur ligne :** On peut travailler dans l'environnement *Matlab* en ligne (ligne à ligne) ou bien exécutant un fichier écrit à l'avance.

Le mode ligne à ligne est celui que l'on a rencontré jusqu'à présent, on tape à la suite du prompt ' »' dans la fenêtre de commande la commande que l'on souhaite exécuter suivie d'un retour chariot (entrée).

*Matlab* est un langage interprété et le résultat est immédiatement disponible dans l'espace de travail.

#### 4.2. Création d'un fichier

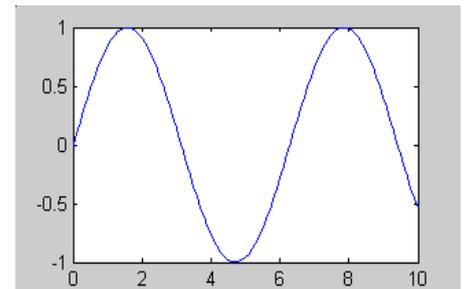
Comme dans tout langage de programmation, on peut écrire un programme puis en demandant l'exécution. Exp : on réalise un programme en langage *Matlab* qui consiste à écrire toutes les commandes au moyen d'un éditeur de texte. Une fois le programme tapé le fichier est sauvé avec un certain nom et une extension *.m* pour que *Matlab* reconnaisse un *M-file* lorsqu'on demande l'exécution de ce programme.

#### 5. Affichage graphique

Les résultats présentés sous forme graphique, sous *Matlab* n'est pas un casse-tête les possibilités graphique sont énormes et de nombreuses fonctions toutes prêtes les rendent accessibles.

Voyons quelques commandes, traçons une courbe très simple temporel  $y(t) = \sin(t)$  de 10 seconds.

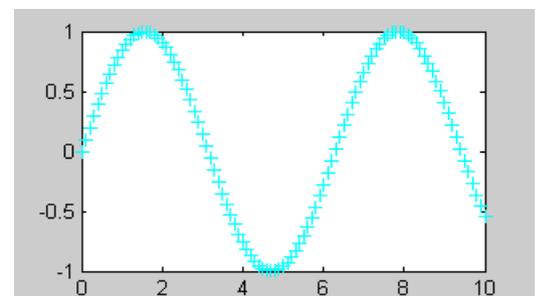
Exp. `>>%tracer un graphe`  
`t=[0:0.1:10]; % t=[début:pas:fin]`  
`f=inline('sin(t)');% on donne la fonction`  
`plot(t,f,'b');% tracer la courbe f en fonction de t`  
`>>     □ Matlab rend la main`  
**'b' c'est la couleur bleu**



En utilisant la commande `plot` pour tracer la courbe  $f(t)$  un extrait du help de cette commande nous indique comment l'utiliser `>> help plot`.

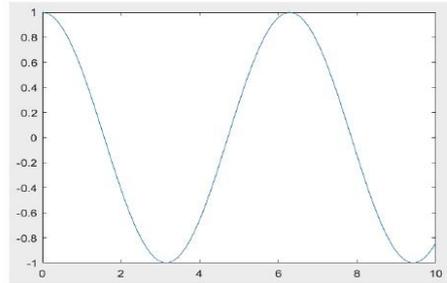
'b' indique la couleur de cette courbe (r : red, g : green, w : white, k :black, y : yellow, m : magenta ....).

Exp. `>> plot(t,f,'c+'); % tracer la courbe en cyan`  
**avec toute la ligne de (+)**



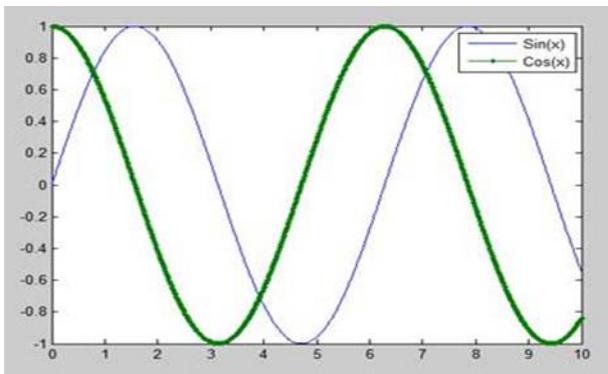
Pour la courbe  $g(t)=\cos(t)$  sur un horizon temporel de 10 seconds

```
Exp>>%tracer un graphe
t=[0:0.1:10]; % t=[début:pas:fin]
g=inline('cos(t)');% on donne la fonction
plot(t,g,'g');% tracer la courbe g en
fonction de t
>>  Matlab rend la main
```



Les deux courbes dans le même axe

```
Plot(t, f(t),'b', t, g(t),'g-');
```



## 6. Polynôme

Soit le polynôme de la forme  $a_{m-1}x^{m-1} + a_mx^m + \dots + a_1x^1 + a_0$  a est représenté par Matlab comme  $[a_{m-1} \ a_m \ \dots \ a_1 \ a_0]$ , par exemple  $x^2 + 4x - 1$  est représenté par  $[1 \ 4 \ -1]$ .

Pour chercher les solutions de ce polynôme en appliquant la commande roots qui permet de calculer les solutions de tel polynôme

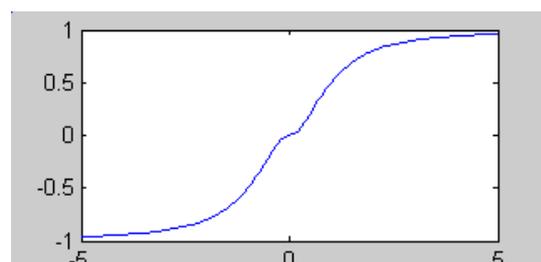
```
>> v=[1 -3 -4]; % le polynôme est : y(x)=x^2-3*x-4
>> roots(v) ans =
     4
    -1
>>
>> polyval(v,4) % pour calculer y(4)
```

## 7. Graphismes

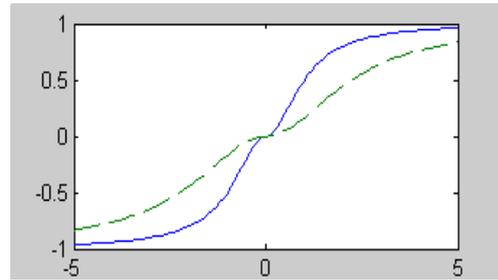
Considérons le problème de graphisme d'une fonction, par exemple la fonction suivante

$f(x) = x|x|/(1+x^2)$  dans l'intervalle  $[-5 \ 5]$ .

```
>> x=(-5:1:5)';
>> y=x.*abs(x)./(1+x.^2);
>> plot(x,y,'-')
>>
```



```
>> y1=x.*abs(x)./(5+x.^2);  
>> plot(x,y,'-',x,y1,'--')
```



**Exercice** – Tracer la courbe de la fonction suivante :

$$f(x) = \frac{1}{(x-0.3)^2 + 0.01} + \frac{1}{(x-0.9)^2 + 0.04} - 6$$