

Chapitre V: Traitement des données sous Matlab

1. Les graphiques simples

La fonction *plot* permet de tracer des courbes en Matlab. Les arguments de cette fonction sont les vecteurs de variables indépendantes et dépendantes, comme dans l'exemple suivant:

```
>> x = [0:0.01:2*pi]
>> plot(x, cos(x), x, sin(x))
```

On peut aussi simplifier comme suit:

```
>> plot(x', [cos(x)' sin(x)'])
```

Ce graphique manque de clarté. Il faut toujours nommer les axes (et mettre les unités si possible), proposer un légende, etc.

```
% graphique.m
```

```
clear all
close all      % fermer les anciennes figures
```

```
x = [0:0.01:2*pi];
y1 = cos(x);
y2 = sin(x);
```

```
figure(1)
plot(x, y1, 'o', x, y2, '+')      % cos(x) en points, sin(x) en +
```

```
title('sinus et cosinus')
xlabel('x')
ylabel('f(x)')
```

```
legend('cos(x)', 'sin(x)', 0)     % le 0 place la légende à coté des courbes
```

Fonctions de graphiques

Fonction	Description
semilogx	Axe des x logarithmique
semilogy	Axe des y logarithmique
loglog	Axes des x et des y logarithmique
polar	Graphique en coordonnées polaires
mesh	Graphique 3D

2. Graphique d'une fonction

La commande *fplot* permet de tracer le graphique d'une fonction sur un intervalle donné.

La syntaxe est:

Fplot('nomf', [xmin, xmax])

Où:

- nomf est le nom de la fonction.
- [xmin, xmax] est l'intervalle pour lequel est tracé le graphe de la fonction.

Exemple 1

Le graphe de la fonction incorporée sinus entre -2π et 2π

Fplot('sin', [-2*pi 2*pi])

Exemple 2

Le graphe de la fonction:

$$h(x) = x \sin(x), \text{ entre } -2\pi \text{ et } 2\pi.$$

On peut définir la fonction utilisateur *h* dans un fichier *h.m*

function y=h(x)

y=x.*sin(x);

On obtient alors le graphe de la fonction h par l'instruction:

```
fplot('h', [-2*pi 2*pi])
```

Exemple 3

Une autre façon pour tracer le graphe de $x \sin(x)$

```
fplot('x*sin(x)', [-2*pi 2*pi])
```

Dans le 3eme exemple on a le choix entre l'écriture de $x.\sin(x)$ ou $x*\sin(x)$.

3. Sauvegarder une figure

La commande **print** permet de sauvegarder la figure d'une fenêtre graphique dans un fichier sous divers formats d'images.

La syntaxe est la suivante:

```
print -f<num> -d<format> <nomfich>
```

Où:

- <num> désigne le numéro de la fenêtre graphique. Si ce paramètre n'est pas spécifié, c'est la fenêtre active qui est prise en compte.
- <nomfich> est le nom du fichier dans lequel est sauvegardée la figure. Si aucune extension de nom n'est donnée, un nom par défaut est ajoutée à l'extension du fichier en fonction du format choisi (.ps pour du PostScript, .jpeg pour du jpeg, par exemple).
- <format> est le format de sauvegarde de la figure. Ces formats sont nombreux; les principaux sont:

ps	: PostScript noir et blanc
psc	: PostScript couleur
eps	: PostScript Encapsulé noir et blanc
epsc	: PostScript Encapsulé couleur
jpeg	: Format d'image JPEG
tiff	: Format d'image TIFF

4. Importation et exportation des données

Par fois, il est nécessaire de conserver vos solutions numériques d'une session de travail à l'autre. La méthode la plus facile est d'utiliser les fonctions *save* et *load* de Matlab. Les fichiers créés portent l'extension ".mat" par défaut (ceux-ci ne sont accessible qu'avec Matlab).

Exemple

```
>> A = ones(3)      % matrice
>> b = zeros (3);  % vecteur
>> save ma_session % le fichier ma_session.mat est crée dans le répertoire courant

>> clear all
>> whos            % les variables ne sont plus dans la mémoire
>> load ma_session % on va charger tous les éléments de la session
>> whos
```

Name	Size	Bytes	Class
A	3x3	72	double array
B	3x1	24	double array

```
>>
```

Il est aussi possible de sauvegarder les variables dans un fichier ASCII et lui donner un format désiré. Elles pourront ensuite être lues par d'autres logiciels (par exemple: EXCEL) et avec Matlab (en utilisant aussi la fonction *load*).

Dans le cas suivant, on ne sauvegarde que la matrice A de l'exemple précédant.

```
>> save ma_session.dat -ASCII A
>> load ma_session.dat
```

La fonction *xlsread* permet de charger en Matlab des fichiers EXCEL.

```
>> xlsread classeur.xls
```

Il est possible d'avoir plus d'informations à ce sujet, on consultant:

```
>> help fileformats
```