



SÉRIE D'EXERCICES 1

Le but de cette séance est de se familiariser avec le logiciel MATLAB qui sera utilisé pendant le semestre pour implémenter les différentes méthodes numériques présentées au cours.

MATLAB est un logiciel de calcul matriciel à syntaxe simple. Avec ses fonctions spécialisées, il peut être considéré comme un langage de programmation adapté pour les problèmes scientifiques. MATLAB est un interpréteur, c'est-à-dire que les instructions sont interprétées et exécutées ligne par ligne.

Ce document n'a aucunement la prétention d'être une introduction complète à MATLAB, mais doit permettre une prise en main de ce logiciel dans l'optique des futures applications numériques du cours. Pour plus d'information voici quelques sites qui fourniront un tutorial plus complet :

Sites en anglais :

<http://www.mathworks.com/products/matlab/>

<http://www.math.mtu.edu/~msgocken/intro/intro.html>

Sites en français :

<http://www.mathworks.fr/products/matlab/>

<http://www.gel.ulaval.ca/~lehuy/intromatlab/index.html>

D'autre part, les manuels sur MATLAB sont disponibles dans la rubrique Exercices de votre site web préféré :

http://transp-or.epfl.ch/courses/optimization2012/syllabus_matlab.pdf

<http://transp-or.epfl.ch/courses/optimization2012/MATH600-MATLABTutorial.pdf>

0. Démarrer :

Dans la fenêtre de commande de votre système, taper « matlab ». MATLAB répondra par un symbole « ».

Dans cette fenêtre, on tape les instructions une ligne à la fois et chaque ligne est exécutée immédiatement après la touche « Return ». Une ligne peut contenir plusieurs instructions séparées par des « , ». Des boucles `if`, `for`, `if ... else` peuvent être écrites sur plusieurs lignes.

Lorsque les fonctions graphiques sont appelées, une fenêtre Graphique s'ouvre.

1. Fonction HELP :

Pour obtenir de l'aide sur un sujet, une instruction ou une fonction, on tape `help` suivi par le sujet, l'instruction ou la fonction désirée.

Exemple :

» *help exp*

EXP Exponential.

EXP(X) is the exponential of the elements of X, e to the X.

*For complex $Z=X+i*Y$, $EXP(Z) = EXP(X)*(COS(Y)+i*SIN(Y))$.*

See also LOG, LOG10, EXPM, EXPINT.

Overloaded methods

help sym/exp.m

2. Matrices :

On définit une matrice A en donnant ses éléments:

» $A = [1 \ 2 \ 3; \ 4 \ 5 \ 6]$

A =

1 2 3

4 5 6

Matrice identité:

» $B = eye(3)$

B =

1 0 0

0 1 0

0 0 1

Les éléments d'un vecteur ou d'une matrice peuvent être adressés en utilisant les indices sous la forme suivante:

$t(10)$ élément no. 10 du vecteur t

$A(2,9)$ élément se trouvant à ligne 2, colonne 9 de la matrice A

$B(:,7)$ la colonne 7 de la matrice B

$C(3,:)$ la ligne 3 de la matrice C

Les opérations matricielles exécutées par MATLAB sont illustrées dans le tableau suivant:

$B = A'$ La matrice B est égale à la matrice A transposée

$E = inv(A)$ La matrice E est égale à la matrice A inversée

$C = A + B$ Addition

$D = A - B$ Soustraction

$Z = X*Y$ Multiplication

$X = A \setminus B$ Équivalent à $inv(A)*B$

$X = B/A$ Équivalent à $B*inv(A)$

Les opérations « élément par élément » sur les vecteurs et les matrices sont effectuées en ajoutant un `.` devant les opérateurs :

Exemple :

```
» A = [ 3 4 5];
» B = [ 3 2 1];
» C = A.*B
C =
    9 8 5
```

3. Graphiques :

On utilise l'instruction `plot` pour tracer un graphique 2D :

Exemple :

```
» plot(x,y) % Trace le vecteur y en fonction du vecteur x
```

On utilise l'instruction « `mesh` » pour tracer un graphique 3D :

Exemple:

```
» [X,Y] = meshgrid(-2:.2:2, -2:.2:2); % construire une "grille"
» Z = X .* exp(-X.^2 - Y.^2); % évalue la fonction pour chaque paire de points
» surf(X,Y,Z) % dessine la courbe
```

4. Programmation avec MATLAB :

Les fichiers M sont des fichiers « texte » contenant des suites d'instructions MATLAB dont le nom a comme extension `.m`, par exemple « `test1.m` ». Si on tape `test1` dans la fenêtre Commande, les instructions contenues dans le fichier correspondant `test1.m` seront exécutées une par une.

On peut créer des fichiers M à l'aide de l'éditeur de MATLAB que l'on lance avec la commande « `edit` » mais aussi à partir de n'importe quel éditeur de texte.

Exemple:

```
% Ceci est un exemple de fichier M
% Les lignes « commentaires » commencent par "%"
for i=1:10
    for j=1:4
        x=0.005*i;
        y=30+j;
        z(i,j)=10*exp(-y*x)*cos(120*pi*x);
    end
end
end
```

Remarque:

Le point virgule à la fin de chaque ligne empêche MATLAB d'afficher le resultat de la ligne dans la fenêtre de commande.

Une fonction MATLAB est un fichier M particulier dont la première ligne commence par « **function** ». Une fonction peut être utilisée dans les expressions mathématiques ou dans les instructions MATLAB. Afin de pouvoir utiliser une fonction, il faut qu'elle se trouve dans le répertoire de travail et que le nom du fichier « .m » qui lui est associé corresponde exactement avec le nom de la fonction.

Exemple:

```
function y = NomdelaFonction(x)
% NomdelaFonction demande un argument x, i.e un vecteur ou une matrice, et retourne
un vecteur de dimension 2, y.
```

```
y(1) = sin(x(1));
y(2) = cos(x(2));
```

A faire :

(a) Utiliser Matlab afin de trouver x tel que $Ax \cong b$, avec:

$$A = \begin{pmatrix} 2 & 3 & 4 \\ 5 & 1 & 1 \\ 5 & 3 & 2 \\ 4 & 1 & 2 \end{pmatrix} \quad \text{et} \quad b = \begin{pmatrix} 21 \\ 19 \\ 24 \\ 19 \end{pmatrix}$$

(b) Implémenter une fonction qui retourne la valeur de la fonction:

$$f(x, y) = x^2 + 3y^2$$

ainsi que son gradient et son hessien.

(c) Utiliser la fonction *mesh* afin de visualiser la fonction $f(x, y)$.

(d) Utiliser la fonction *fminsearch* de Matlab pour trouver le minimum de la fonction $f(x, y)$.

(e) Mêmes questions pour la fonction de Rosenbrock donnée par :

$$f(x, y) = 100(y - x^2)^2 + (1 - x)^2.$$