ED1 – Équations différentielles
TP1 – Manipulation de MATLAB (2 séances)

MATLAB est un logiciel de calcul matriciel. Les données qu'il manipule doivent être rentrées sous forme matricielle. Il faudra donc commencer par discretiser les calculs qu'on veut lui faire executer. Pour vous familiariser avec ce logiciel, entrez les nombres et commandes suivantes.

Comment lancer MATLAB ?

Ouvrez une fenêtre `Xterm` et tapez-y `matlab &`. La fenêtre MATLAB s'ouvre. Les commandes MATLAB doivent être entrées après le prompt `>>`.

Pour quitter MATLAB, tapez

```
>> quit
```

MATLAB contient plusieurs commandes et fonctions prédéfinies, une liste est donnée à la fin de cette feuille de TP. S'il se produit une erreur lors de l'utilisation d'une commande, par exemple la commande `plot`, consultez l'aide: tapez

```
>> help plot
```

Utilisation en mode "ligne par ligne"

Nombres

Les nombres réels peuvent être écrit sous différentes formats :

`5    1.0549    0.5245E-12    12.78e6    0.000054    -235.087.`

Les nombres complexes peuvent être écrit sous forme cartésienne ou polaire :

`0.5 +i*2.7    -1.2+j*0.789    2.5 + 9.7*i    1.25*exp(j*0.246).`

Que vaut `j` ? (Explication : cette notation est pour les physiciens.)

Pour choisir le format d'affichage des nombres, on utilise la commande `format` :

```
format short      0.1234
format long       0.12345678901234
format short e    1.234E+002
format long e     0.123456789012345E+002
```

Tester ces différents formats sur $\sqrt{3}$:

```
>> format short, sqrt(3)
>> format long, sqrt(3)
```

etc ...

```
>> format short   (retour au format normal)
```

Opérations

Les opérations arithmétiques sont `+` addition, `-` soustraction, `*` multiplication, `/` division à droite, `\` division à gauche, `^` puissance.

L'opération d'affectation d'une valeur à une variable se fait grace à `=` :

```
>> x=2
>> y=x^4;    (notez l'effet du point-virgule)
>> y/x
```

La dernière réponse est appelée `ans` si on ne l'a pas affectée. On peut ainsi l'utiliser :

```
>> ans
>> z=factorial(ans), z=4*z
```

La liste actuelle des variables de votre espace de travail est donnée par :

```
>> who
>> whos
```

Matrices

On définit une matrice d'une des manières suivantes :

`>> x=[1,2,3]`	`>>z=[1,2,5;3,4,6;5,7,3];`	`>> t=linspace(1,11,6)`	`>> v=1:.25:4`
`>> y=[4;5;6]`	`>>size(z)`	`>> u=0:0.1:1`	`>> w=logspace(-1,2,4)`

Voici quelques matrices particulières :

`>> id=eye(5)`	`>> zz=zeros(5,3)`	`>> diag(x)`	`>> un=ones(6,2)`
`>> eye(2,3)`	`>> zeros(size(y))`	`>> rand(1,3)`	`>> [z;z;z]`

Les coefficients d'une matrice peuvent être extraits en utilisant les commandes suivantes :

`>> x(2)`	`>> z(:,2)`	`>> x(2:3)`	`>> u(t)`
`>> z(1,2)`	`>> z(3,:)`	`>> z(1:2,1:3)`	`>> z(t(1:2),x(2))`

Un peu d'algèbre linéaire

Après avoir rentré les données suivantes :

```
>> a=[1 2 3; 4 5 6; 7i 8 10], b=[1 1 1]';
```

effectuez et identifiez les opérations suivantes :

`>> 2*a, a/4`	`>> a', a.', a+z`	`>> a*b, b'*a`	`>> b*a`

ainsi que celles-ci :

`>> inv(a), a^2`	`>> a+1, a+2`	`>> a.^2,a.*a, b.*b`	`>> 1./a, 1./a.^2` .

Fonctions élémentaires et graphiques

MATLAB traite une variable comme un vecteur :

`>> x=0:0.5:10;` (x est un maillage de pas $1/2$ de l'intervalle $[0, 10]$)

et on peut définir des fonctions de cette variable :

```
>> y=0.25*x; z=y.^2;
```

`>> t=5*exp(-0.4*x).*sin(20.5*y);` (à quoi sert le '.' avant le '*' ?).

On trace les graphes de ces fonctions en utilisant la commande `plot` :

```
>> plot(x,y)
>> plot(x,z)
>> hold on
>> plot(x,y,'x')
>> clf
>> plot(x,t,x,z)
>> clf
>> plot(x,t,'r',x,z,'b--')
>> legend('t','z') .
```

Exercice 1 – *Tracez des courbes*

Créez un maillage de 15 points uniformement répartis entre $t_0 = 0$ et $t_1 = 4\pi$. Tracez sur un même graphique les graphes de $x(t) = e^{-t}\cos t$ et $y(t) = e^{-t}\sin t$ (n'oubliez pas de mettre une légende). Augmentez le nombre de points du maillage afin que le résultat vous paraisse satisfaisant. Tracez ensuite la courbe paramétree par (x, y) avec

`>> plot(x,y)`.

Quittez MATLAB.

Utilisation en mode "programme"

Un script ou une fonction MATLAB est un fichier `.m` où l'on a écrit toutes les instructions que MATLAB doit exécuter. Il suffira ensuite de demander à MATLAB de lire le fichier. Avant de commencer, il faut s'organiser : créez un dossier `ED1` qui contient un dossier `TP1` et lancez-y MATLAB.

commandes Unix : `>cd >mkdir -p ED1/TP1 >cd ED1/TP1 >matlab &`

Ouvrez-y ensuite un éditeur de texte de votre choix, EMACS ou GEDIT sont disponibles.

Écrire un script

Exercice 2 – *Exemple de boucle*
Créez un fichier `exo2.m` contenant :

```
% ceci est le script du deuxieme exercice du TP1
% ajouter tous commentaires dont vous avez besoin
for j=1:9
   for k=1:10
      A(k,j)=k*j;
   end
end
```

Pour exécuter le script sous MATLAB :

```
>> exo1
>> A
```

Pourquoi est-il important d'avoir le point virgule dans le script ?

Écrire une fonction

Exercice 3 – *Une fonction $F(x)$ définie par morceaux*
Créez un fichier `F.m` contenant :

```
function y=F(x)
% Mettez vos commentaires ici
if x < -1
   y= x^2;
elseif x>1
   y=x^3;
else
   y=exp(1-x^4);
end
```

Vous venez de créer une function MATLAB (aussi appelée "M-file") pour la fonction F. Testez cette fonction sous MATLAB :

```
>> F(3)
>> F([-5,-1,2])
>> F([2,3,4])
```

Vous verrez que les deux dernières lignes ne marchent pas (pourquoi ?). Créez ensuite, dans un fichier `Fmatrice.m`, une autre fonction *Fmatrice* (qui fera appel à la fonction F) qui accepte une matrice de dimension arbitraire et applique la fonction F à toutes ses coordonnées. Par exemple,

```
>> Fmatrice([-5,-1,2;1,2,3])
```

doit renvoyer la matrice `[F(-5),F(-1),F(2);F(1),F(2),F(3)]`.
Écrivez enfin un script `exo3.m` qui crée un maillage x d'abscisses de -3 à 3 par pas de $0,1$. Représentez la fonction F aux points x_i , avec la commande `plot`.

Exercice 4 – *La fonction cosinus et son développement de Taylor jusqu'à l'ordre n*
Considérons une fonction $G(x) = \cos(x)$. On rappelle que la dérivée $k$ième de cette fonction est donnée par $G^{(k)}(x) = \cos(x + k \cdot \frac{\pi}{2})$. Notre but est de dessiner le polynôme P qui est le développement de Taylor de G. Créer un fichier `P.m` dont la première ligne est

```
function y=P(x,n,a)
```

où `x` et `y` sont des vecteurs ligne, l'entier `n` est le dégré du polynôme, et le réel `a` est le nombre autour duquel le dévéloppement est fait.

Créer ensuite un fichier `exo4.m` qui fait un plot de G et de P pour $n = 4$ et $a = 1$, sur l'intervalle $[-2, 5]$ avec un maillage de pas 0,1

Ne pas faire confiance aveugle à l'ordinateur

Exercice 5 – *Fibonacci*
Donnez une matrice A telle que la suite f_n définie par $\begin{bmatrix} f_n \\ f_{n+1} \end{bmatrix} = A \begin{bmatrix} f_{n-1} \\ f_n \end{bmatrix}$ et $\begin{bmatrix} f_0 \\ f_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ soit la suite de Fibonacci. Vérifiez à la main que $A \begin{bmatrix} 1 \\ \frac{1+\sqrt{5}}{2} \end{bmatrix} = \frac{1+\sqrt{5}}{2} \begin{bmatrix} 1 \\ \frac{1+\sqrt{5}}{2} \end{bmatrix}$ et $A \begin{bmatrix} 1 \\ \frac{1-\sqrt{5}}{2} \end{bmatrix} = \frac{1-\sqrt{5}}{2} \begin{bmatrix} 1 \\ \frac{1-\sqrt{5}}{2} \end{bmatrix}$. En déduire que $A^n \begin{bmatrix} 1 \\ \frac{1-\sqrt{5}}{2} \end{bmatrix} \underset{n\to\infty}{\longrightarrow} \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.

Écrivez un script qui calcule la suite $\begin{bmatrix} a_n \\ b_n \end{bmatrix} = A^n \begin{bmatrix} 1 \\ \frac{1-\sqrt{5}}{2} \end{bmatrix}$, pour $n = 0, 1, 2, \ldots, 20$, et dessinez la suite des points du plan ainsi obtenus. Refaire la même expérience pour $n = 0, 1, 2, \ldots, 100$.

Que remarquez-vous ? Donnez une explication.

Fonctions mathématiques

Déterminez les fonctions mathématiques suivantes :

`sign` =	`acosh` =
`exp` =	`asinh` =
`log` =	`cosh` =
`log10` =	`sinh` =
`sqrt` =	`tanh` =
`round` =	`sin` =
`rem` =	`tan` =
`atan` =	`real` =
`acos` =	`imag` =
`abs` =	`angle` =

Référence complémentaire
http://www.ann.jussieu.fr/~postel/matlab/