UFR Mathématiques — Université Rennes 1
Licence 2ème année — Année 2006/2007

Étude de fonctions, équations différentielles
TP1 – Manipulation de MATLAB

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.

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 du TP. S'il se produit une erreur lors de l'utilisation de la commande `#1` consultez l'aide, tapez :

```
>> help #1
```

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 + 9.7i    1.25*exp(j*0.246).
```

Que vaut `j` ? Pourquoi ?

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^5;    (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=3*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(1,7,3)   >> v=1:.25:4
>> y=[4;5;6]          >>z                        >> u=0:0.1:1           >> w=logspace(1,3,7)
```

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 rentrer 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        >> inv(a), a^2
```

ainsi que celles-ci :

```
>> 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)
>> clf
>> plot(x,t,x,z)
>> clf
>> plot(x,t,'r',x,z,'b--') .
```

Quitez 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 fichier `C02` qui contient un fichier `TP1` et lancez-y MATLAB.

commandes Unix : `>cd >mkdir -p C02/TP1 >cd C02/TP1 >matlab &`

Ouvrez-y EMACS : tapez `emacs &` à la suite des commandes précédentes (vous pouvez utiliser l'éditeur de texte de votre choix).

Écrire un script

Exercice 1 – *Exemple de boucle*
Créez un fichier `exo1.m` contenant :

```
% ceci est le script du premier 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 :

```
>> what   (donne la liste des fichiers .m)
>> exo1
>> A
```

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

Écrire une fonction

Exercice 2 – *Une fonction définie par morceaux*
Créez un fichier `F.m` contenant :

```
function y=F(x)
% Mettez vos commentaires ici
if x < -1
   y= x^2+2-1/(x+1);
elseif x>-1
   y=exp(1/(x+1));
else
   y=0;
end
```

Pour utiliser cette fonction sous MATLAB taper

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

Modifiez cette fonction pour que `F([-5,-1,2;3,4])` renvoie `[F(-5),F(-1),F(2);F(3),F(4)]`.

Exercice 3 – *Splines cubiques*
Étant donnés n nombres réels distincts $a_1 < \ldots < a_n$ et n autres nombres $b_1, \ldots, b_n$, on cherche à construire une fonction $f : [a_1, a_n] \to \mathbb{R}$ de classe $\mathcal{C}^2$ qui prenne les valeurs b_i en a_i et dont les dérivées à droite en a_1 et à gauche en a_n sont nulles. Lorque cette fonction est définie sur $[a_i, a_{i+1}]$ par un polynôme de degré trois P_i pour tout i, f s'appelle une spline cubique.

1°) Lorsque $n = 3$, écrivez le système linéaire à résoudre pour trouver les trois polynômes définissant la spline.

2°) Écrivez une fonction `spline.m` dépendant des arguments $[a_1, a_2, a_3; b_1, b_2, b_3]$ qui donne les coefficients des polynômes définissant la spline cubique qui prend les valeurs b_i au point a_i.
Modifiez cette fonction en une fonction des a_i, b_i et d'un vecteur x de telle sorte que la sortie de la fonction soit le vecteur de valeurs de la spline en x.

3°) Modifiez cette fonction en une fonction de $[a_1, \ldots, a_n; b_1, \ldots, b_n]$ et d'un vecteur x dont la sortie donne les valeurs en x de la spline prenant les valeurs a_i en b_i.

Tracez la sur des exemples

Fonctions mathématiques

Déterminez les fonctions mathématiques suivantes :

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