

LM206 : Initiation à Scilab, TP 1

1 Brève présentation du logiciel

L'objectif de ce module est d'acquérir les bases d'un logiciel de calcul scientifique, en l'occurrence **Scilab**. De tels logiciels (**Scilab**, **Matlab**, **Octave**, ...) sont utilisés en particulier par les ingénieurs pour résoudre numériquement de nombreux problèmes mathématiques comme la résolution d'équations différentielles ou aux dérivées partielles, la résolution de systèmes linéaires, la recherche de valeurs propres de matrices, la résolution d'équations non linéaires, etc... Contrairement à **Matlab**, **Scilab** est un logiciel gratuit téléchargeable sur le web à l'adresse suivante

<http://www.scilab.org>

Scilab est un langage de programmation, comme Fortran, C, ... et a l'avantage sur ces langages de ne pas nécessiter de compilation. De plus, il possède un grand nombre de fonctions préprogrammées ainsi que des outils de représentation graphique. À noter enfin que la connaissance de **Scilab** (ou **Matlab**, **Octave**, ...) est requise pour les candidats à l'Agrégation de Mathématiques dans le cadre de l'épreuve de modélisation.

Note L'ensemble du cours se base sur les versions de Scilab 4 et plus, sous Linux ou Windows.

1.1 Première utilisation

Le logiciel, une fois lancé, se présente sous la forme d'une fenêtre de commande possédant une barre de menu. L'exercice 1 permet de découvrir la première utilisation de **Scilab**, en tant que calculatrice scientifique.

Exercice 1 Lancer **Scilab** et taper successivement les instructions suivantes dans la fenêtre de commande

```
a=5+6
sqrt(12)
ans -1
(1+%i)^3
a=[1,2,3], b=[1 2 3]
cos(a)
c=[-1;2;1;-3]
A=[1,-1,3,4;-1,2,7,8;0,2,-4,-2]
```

```
d=exp(A)
e=A*c;
```

1.2 Premier exemple de script

Afin d'exécuter une suite d'instructions plus longues ou plus complexes, il est préférable d'écrire celles-ci dans un fichier plutôt que de les taper dans la fenêtre de commande comme précédemment. Un fichier formé d'instructions est appelé **script**. Il est ensuite possible d'exécuter ce script en utilisant la commande **exec** accessible avec la barre de menu de la fenêtre principale¹. L'exercice 2 propose un premier exemple d'écriture et d'exécution d'un script.

Exercice 2 Créer un nouveau dossier où seront enregistrés vos programmes. Ouvrir un éditeur de texte² et créer un fichier nommé `essai1.sci` (ou un autre nom !) contenant le texte suivant

```
A=rand(4,4);b=ones(4,1);x=inv(A)*b;s=norm(A*x-b);
x,s
```

Exécuter le script.

1.3 Premier exemple de fonction

Il est également possible de construire avec **Scilab** de nouvelles fonctions mathématiques, en complément de celles existantes (comme la fonction **inv** pour calculer l'inverse d'une matrice de l'exercice 2). L'exercice 3 propose un exemple d'écriture et de chargement d'une fonction.

Exercice 3 Ouvrir l'éditeur et créer un fichier nommé `fct1.sci` contenant le texte suivant

```
function c=fct1(n)
if (n<0) | (int(n)<>n) then
    c=0;
else
    c=1;
    for i=1:n
        c=c*i;
    end
end
endfunction
```

¹À partir du menu principal : File → File operations → Exec

²ou l'éditeur intégré de **Scilab**

Charger la fonction³ `fct1`. Puis tester celle-ci pour différentes valeurs de son argument en tapant les instructions ci-dessous dans la ligne de commande

```
fct1(2), fct1(5), fct1(-1), fct1(3.5), ...
```

Quelle fonction bien connue reconnaît-on ? Commenter les résultats.

1.4 Aide en ligne et démos

La première prise en main du logiciel passe aussi par la découverte de l'aide en ligne. Cette aide, en anglais ou en français suivant la version du logiciel installée sur votre ordinateur, est accessible à partir de l'option *Help* du menu. Elle se présente sous la forme d'une liste de fichiers d'aide pour chaque fonction de `Scilab`. Pour effectuer une recherche par mot clé dans l'index (par exemple, trouver la fonction qui retourne le cosinus hyperbolique), on peut taper `cosine` dans la fenêtre de commande (et découvrir que la fonction s'appelle `cosh`). On peut aussi taper `hyperbolic`. Faire un essai. Lorsqu'une instruction est connue (par exemple `rand`), il est possible d'accéder directement à l'aide de celle-ci en tapant `help rand` sur la ligne de commande.

Exercice 4 Trouver avec l'aide en ligne, la fonction `Scilab` qui permet de construire une matrice formée de zéros de taille donnée, la fonction qui permet de construire une matrice identité de taille donnée, ainsi que celle qui permet de construire une matrice diagonale.

Exercice 5 Taper dans un fichier la fonction suivante qui calcule les moyennes arithmétique et géométrique de deux réels.

```
function [a,g]=arge(x,y)
a=(x+y)/2;
if x*y>0 then
    g=sqrt(x*y)
else
    disp('la moyenne geometrique n''est pas definie');
end
endfunction
```

Lancer les instructions suivantes.

```
[u,v]=arge(5,6)
arge(3,2)
u=arge(1,3)
u=arge(5,-3.6)
[u,w]=arge(5,-3.6)
```

³À partir du menu principal : File → File operations → Getf

Commenter les résultats.

Les programmes de démonstration (option *Demos* du menu) permettent pour leur part de découvrir la plupart des caractéristiques de **Scilab** ainsi que ses nombreux champs d'application.

Exercice 6 Lancer les démos suivantes

– Trajet d'un vélo.

Demos → Simulation → Bike Simulation → bike : unstable trajectory

– Mouvement d'un N-pendule

Demos → Graphics → Animation → N like pendulum movement

pour découvrir des exemples de modélisation complexe avec **Scilab**.

Exercice 7 On rappelle l'encadrement suivant de nombre d'Euler e : pour tout entier n non nul

$$c_n < e < c_n + \frac{1}{n \times n!} \quad \text{avec } c_n = \sum_{k=0}^n \frac{1}{k!}$$

1. Écrire un programme permettant de calculer c_n étant donné n .
2. Pour différentes valeur de n , calculer $e - c_n$.