

Initiation à Scilab

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**, ...) 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** a le bon goût d'être un logiciel gratuit (freeware). Développé par l'INRIA, il est téléchargeable sur le web à l'adresse suivante

<http://www.scilab.org>

Scilab utilise un langage propre, proche dans sa structure des langages informatiques courants (Fortran, C, Pascal, ...). Les principaux avantages par rapport à ces langages est qu'il ne nécessite pas d'étape de compilation mais aussi qu'il possède un grand nombre de fonctions préprogrammées ainsi que des outils de représentation graphique.

À noter enfin que la connaissance de **Scilab** est requise pour tous les candidats à l'Agrégation de Mathématiques dans le cadre de la nouvelle épreuve orale de modélisation.

Première utilisation. Le logiciel, une fois lancé, se présente sous la forme d'une fenêtre de commande possédant une barre de menu. L'exercice 1.1 permet de découvrir la première utilisation possible de **Scilab**, en tant que calculatrice scientifique.

Exercice 1.1 Lancer **Scilab** et taper successivement les lignes suivantes, appelées aussi instructions, dans la fenêtre de commande à la suite de la flèche clignotante -> (en appuyant sur la touche Enter pour changer de ligne) :

```
2+2
(1-%i)^2
1/3, format('v',16); 1/3
%eps, 1+%eps
A=[1,2;3,4]
A*[2,3;1,1]
sin([%pi,%pi/2])
```

Premier exemple de script Scilab. 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. On parle dans ce cas de l'écriture d'un script. Il est ensuite possible d'exécuter ce script en utilisant la commande **exec** accessible avec la barre de menu de la fenêtre principale. L'exercice 1.2 propose un premier exemple d'écriture et d'exécution d'un script.

Exercice 1.2 Créer un nouveau dossier sur le disque dur où seront enregistrés vos programmes le temps de la séance. Ouvrir un éditeur de texte et créer un fichier nommé `essai.sci` contenant le texte suivant :

```
A=rand(4,4);B=inv(A);
C=A*B
```

Exécuter le script précédent dans la fenêtre **Scilab** et en utilisant l'option¹ *File/Exec* dans la barre de menu.

A noter que **Scilab** dispose depuis la version 2.7 d'un éditeur de texte intégré et interfacé accessible par le menu de la fenêtre principale.

Premier exemple de fonction Scilab. Il est également possible de construire avec **Scilab** de nouvelles fonctions mathématiques, en complément de celles existantes (comme $\text{inv}(A)$ par exemple pour calculer l'inverse d'une matrice). Ces fonctions peuvent prendre divers types d'arguments (scalaire, matrice, ...) et peuvent réaliser des opérations mathématiques plus ou moins complexes. L'exercice 1.3 propose un premier exemple d'écriture et de chargement d'une fonction.

Exercice 1.3 Ouvrir l'éditeur et créer un fichier nommé `fct1.sci` contenant le texte suivant :

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

En allant dans l'option *File/Getf* du menu de la fenêtre de commande **Scilab**, charger la fonction `fct1.sci` puis tester celle-ci pour différentes valeurs en entrée. Quelle fonction bien connue reconnaît-on ?

Aide en ligne et démos Scilab. La première prise en main du logiciel passe aussi par la découverte de l'aide en ligne dont le recours sera ensuite constant. Cette aide, entièrement en anglais, 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 instruction disponible sous **Scilab**. Pour effectuer une recherche par mot clé dans l'index (par exemple, trouver la fonction qui retourne le cosinus hyperbolique), on peut taper `apropos cosine` dans la fenêtre de commande (et découvrir que la fonction s'appelle `cosh`). 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 1.4 Trouver avec l'aide en ligne, la fonction **Scilab** qui permet de construire une matrice identité de taille donnée ainsi que celle qui permet de construire une matrice diagonale.

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

Exercice 1.5 Lancer les démos `car_parking`, `bike_simulation` pour découvrir des exemples de modélisation complexe avec **Scilab**.

¹dépendant de la version utilisée de **Scilab**!

Un premier exemple d'application. L'exercice suivant donne un premier exemple d'application de Scilab pour l'approximation d'un nombre célèbre.

Exercice 1.6 Ecrire à l'aide de Scilab, une fonction appelée `euler` permettant d'approcher le nombre d'Euler e à une précision donnée ε supérieure à la précision machine `%eps`.

Rappel : pour tout n entier : $1 + 1 + 1/2 + 1/(3!) + \dots + 1/(n!) < e < 1 + 1 + 1/2 + 1/(3!) + \dots + 1/(n!) + 1/(n \times n!)$.