



TP d'informatique n°16



Utilisation d'un environnement de calcul scientifique : SCILAB

L'objectif du TP est de découvrir le logiciel Scilab.

I Brève présentation du logiciel

L'objectif de ce TP est de découvrir les bases d'un logiciel de calcul scientifique, en l'occurrence Scilab. De tels logiciels (Scilab, Matlab, etc....) sont utilisés en particulier par les ingénieurs pour résoudre numériquement de nombreux problèmes mathématiques comme la résolution d'équations différentielles, la résolution de systèmes linéaires, 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 (Institut de Recherche en Informatique et en Automatique), il est téléchargeable sur le web à l'adresse <http://www.scilab.org>. Il utilise un langage propre, proche dans sa structure des langages informatiques courants (Fortran, C, Pascal...). Un des principaux avantages par rapport à ces langages est qu'il possède un grand nombre de fonctions préprogrammées ainsi que des outils de représentation graphique.

II Première utilisation de Scilab

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

1. Cliquer sur l'icône Scilab et taper successivement les lignes suivantes, appelées aussi instructions, dans la fenêtre de commande à la flèche clignotante ---> (en appuyant sur la touche Enter pour changer de ligne) :

```
2 + 2 , (1 - %i) ^2 , 1/3 , a = 7
a, 1 + a
x = [8, 2, 7, 4]
2 * x, x + 0.2 , x + %i
y = [0 : 0.2 : 1]
y(2)
A = [1, 2 ; 3, 4] , B = [4, 5; 6, 7]
A(1, 2), A* 4, A + B
A * B , B * A
sin([%pi, %pi/2]) , log([3 : 0.2 : 4])
```

Comment ferions-nous pour obtenir la table des valeurs de cos pour toutes les valeurs $0 + 0.1 k$ où k entier entre 0 et 10 ? De même avec exp en $0 + (0.1 + 0.2 * i) * k$ (où k entier entre 0 et 10 et i est le complexe tel que $i^2 = -1$) ?

III Premier exemple de script Scilab

Avant 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.

La question 2 propose un premier exemple d'écriture et d'exécution d'un script.

- Créer un nouveau dossier sur le disque dur où seront enregistrés vos programmes le temps de la séance. Ouvrir l'éditeur *SciNotes* en cliquant sur l'icône en haut à gauche «démarrer SciNotes» (ou aller dans *Applications* > *SciNotes*) et créer un fichier nommé *essai.sce* contenant le texte suivant :

```
A = rand(1, 4), B = - A
C = A + B
```

Deux manières pour exécuter le script :

- aller dans la fenêtre Scilab (Console Scilab) et en utilisant l'option *Fichier/Exécuter* dans la barre de menu
- faire *Ctrl + E* dans la fenêtre SciNotes. Attention, ne seront exécutées que les instructions comprises entre le début du script et le curseur.

Réessayer avec :

```
A = [1, 2, 3, 4] ; B = - 2 * A
C = A + B
```

Remarques :

- le « ; » permet d'exécuter une instruction sans afficher le résultat.
- « , » permet d'écrire deux instructions à la suite l'une de l'autre sur la même ligne.

IV Premier exemple de fonction Scilab

Il est également possible de construire avec Scilab de nouvelles fonctions mathématiques, en complément de celles existantes (comme `sin()` par exemple pour calculer le sinus d'un angle). Ces fonctions peuvent prendre divers types d'arguments (scalaire, matrice, etc....) et peuvent réaliser des opérations mathématiques plus ou moins complexes. Les questions suivantes proposent des exemples d'écriture et de chargement d'une fonction.

- Ouvrir l'éditeur *SciNotes* et créer un fichier nommé *ft1.sce* contenant le texte suivant :

```
function y = F(x) // ces deux barres permettent de placer un commentaire
    y = cos(x) - x ;
endfunction
```

En faisant *Ctrl + E* définissez la fonction puis tester celle-ci sur différentes valeurs en entrée. Essayer par exemple `F(%pi)`

4. Une autre fonction à tester. Attention à rectifier les indentations !

```
function c = fct2(n)
    if (n < 0) | (int(n) <> n) then c = 0 ;
    else c = 1;
        for i = 1:n
            c = c * i ;
        end
    end
endfunction
```

Que calcule cette fonction ?

V Aide et démos Scilab

a) Aide

La première prise en main du logiciel passe aussi par la découverte de l'aide. Cette aide est accessible en cliquant sur l'icône « ? ». Elle se présente sous la forme d'une liste de fichiers d'aide pour chaque instruction disponible sous Scilab.

On peut ainsi :

- effectuer une recherche par mot clé. Exemple : trouver la fonction qui retourne le cosinus hyperbolique. Taper *cosinus hyperbolique* dans la barre *trouver* (après avoir cliqué sur la loupe) et vérifier que cette fonction s'écrit **cosh**.
- lorsqu'une instruction est connue (par exemple **rand**), il est possible d'accéder directement à l'aide de celle-ci en tapant **rand** dans la barre.
- rechercher une fonction dans la table des matières (icône sous forme de livre).

5. Trouver avec l'aide, comment se servir de la fonction Scilab **integrate** qui permet d'intégrer (numériquement) une fonction. Calculer alors l'intégrale de la fonction sinus entre 0 et π

b) Programmes de démonstration

Les programmes de démonstration permettent pour leur part de découvrir la plupart des fonctionnalités et caractéristiques de Scilab ainsi que ses nombreux champs d'application. Ils sont accessibles en cliquant sur l'icône en forme de roue dentée (à côté de l'icône d'aide).

6. Exécuter les démos *Simulation > Simulation de vélo > unstable trajectory* pour découvrir des exemples de modélisation complexe avec Scilab. Noter que l'on a accès au script du programme utilisé...

Exécuter deux ou trois autres démos en traitement du signal, le pendule, en chimie...

VI Tracé de graphiques

a) La commande plot2d

Pour pouvoir obtenir une représentation graphique en Scilab, il faut se donner :

- un vecteur d'abscisses, par exemple à l'aide d'incrémentes : $t = 0 : 0.1 : 2 * \pi$
- un vecteur d'ordonnées. On peut le construire à l'aide d'une fonction prise en un vecteur : si f est une fonction définie et v est le vecteur $[v_1, v_2, \dots, v_n]$, $f(v)$ est le vecteur $[f(v_1), f(v_2), \dots, f(v_n)]$

7. Exécuter :

```
t = 0 : 0.1 : 2 * %pi ;
plot2d(t, sin(t)) // courbe simple
xtitle('sin(x)', 'x', 'sin(x)') ; // pour mettre un titre et nommer les axes
plot2d(sin(t), cos(t)) //courbe paramétrée
clf() // pour réinitialiser la figure
plot2d ([t,t,t], [sin(t), sin(2*t),cos(2*t)])
clf()
t = [0 : 0.1 : 2 * %pi]'
plot2d(t, [sin(t), sin(2*t),cos(2*t)])
```

On pourra trouver comment améliorer la présentation du graphe en utilisant l'aide de la fonction plot2d

8. Résonance en élongation :

```
x = 0.1 : 0.1 : 5 ;
plot2d(x, ((1 - (x/2)^2)^2 + (x/4)^2)^(-0.5))
```

Observer l'influence du facteur de qualité et de la pulsation de résonance (Dans l'exercice 8 de la feuille 5 sur les équations différentielles, on avait montré que, sous certaines conditions, les solutions de l'équation : $m y'' + c y' + k y = A e^{i \omega t}$ avait des solutions d'amplitude $H = \frac{1}{\sqrt{(k - m \omega^2)^2 + \omega^2 c^2}}$)

b) La commande plot3d

Scilab offre la possibilité de représenter des graphiques 3D. La commande de base est plot3d(x,y,z) où x, y et z sont trois vecteurs (de même dimension).

9. Exécuter :

```
t = [0 : 0.3 : 2 * %pi]' ;
z = sin(t)*cos(t)';
plot3d(t,t,z)
xtitle('Vague sur la mer', 'Axe X', 'Axe Y', 'Axe Z' ) ;
```