

TP1 : Introduction à Scilab

Extrait du programme officiel

L'objectif est de poursuivre la formation initiée au lycée des étudiants concernant l'algorithmique et l'utilisation de l'informatique en mathématiques au travers de thèmes empruntés au programme pour comprendre, illustrer et éclairer les notions introduites. Dès qu'un calcul numérique est envisagé, dès qu'un problème incite à tester expérimentalement un résultat, dès qu'une situation aléatoire peut être modélisée avec des outils informatiques, le recours à des algorithmes et des logiciels devra devenir naturel.

Un mot sur Scilab

Scilab est un logiciel libre gratuit, créé par l'INRIA avec le soutien de grandes entreprises européennes. Il s'agit d'un logiciel de calcul numérique (par opposition au calcul formel) et est utile et utilisé dans de très nombreux domaines des mathématiques. Nous verrons quelques exemples d'utilisation au cours des ... séances tout au long de l'année.

L'environnement Scilab

Le logiciel affiche quatre fenêtres à l'ouverture :

- Console Scilab
- Fenêtre SciNote

Première utilisation

Lorsque l'on tape une instruction dans la ligne de commande, elle est exécutée dès la validation (Entrée). La réponse est enregistrée dans la variable **ans**, qui peut être ensuite réutilisée. Une instruction validée ne peut plus être modifiée. On peut toutefois rappeler une instruction dans la ligne de commande à l'aide des flèches ↑ et ↓.

```
-->1+1
ans =
    2.
```

Q.1 Faire des essais! On pourra utiliser les opérations élémentaires $+$, $-$, $*$, $/$, $^$.

```
-->ans^2
ans =
    4.
```

Des constantes prédéfinies sont connues de Scilab : `%pi` et `%e` sont des approximations de π et e .

Q.2 Exécuter les instructions et interpréter ($1.e-16 = 10^{-16}$).

```
1.e-16
1+1.e-16
ans-1
```

On retiendra que Scilab est un logiciel de calcul **approché**. Des **erreurs d'arrondis** peuvent apparaître et devenir **significatives**!

Scilab connaît les fonctions usuelles suivantes du programme d'ECE :

- **log** pour le logarithme népérien,
- **exp** pour l'exponentielle,
- **floor** pour la partie entière,
- **abs** pour la valeur absolue,
- **sqrt** pour la racine carrée.

Par ailleurs, Scilab dispose d'un générateur de nombres aléatoires. L'instruction **rand()** renvoie un nombre au hasard entre 0 et 1, suivant la loi uniforme sur $[0,1]$. On tapera l'instruction **help rand** pour plus d'informations sur les options de **rand**.

Plus généralement, savoir consulter l'aide de Scilab est indispensable. On écrira **help nom de la fonction** pour savoir comment utiliser cette fonction.

Variables

```
-->a=2
a =
    2.
```

Intuitivement, une variable est un *nom* auquel on associe une *valeur*. L'action d'assigner une valeur à une variable est appelée **affectation**. En Scilab, on utilisera la syntaxe **a=2** pour affecter à **a** la valeur 2.

```
-->a=3
a =
    3.
```

On modifie la valeur d'une variables en faisant une nouvelle affectation. L'ancienne valeur est alors perdue. On écrira **help clear** pour savoir comment supprimer des variables.

Q.4 On dispose de deux variables **a** et **b** dont on veut échanger les valeurs. Écrire un algorithme sur votre feuille permettant de réaliser l'échange. Le transcrire en Scilab et tester.

Q.5 Écrire les résultats de chacune des instructions suivantes puis vérifier avec Scilab :

x=3	x=6	x=1
y=5	y=2	y=4
x=2*x+1	x=y	x=x+y
y-x	y=x	y=x-y
	x	x=x-y
	y	x
		y

Scilab permet d'utiliser plusieurs *types* de variables. On utilisera principalement les variables numériques, les variables matricielles, ainsi que les chaînes de caractères.

Q.6 Exécuter les instructions suivantes et interpréter.

```
L=[1,2,3,4]
C=[1;2;3;4]
M=[1,2;3,4]
```

Fonctions et programmation

On peut définir des fonctions numériques dans Scilab. La méthode est en fait la même que pour écrire un programme. Examinons un exemple.

Q.7 On définit la fonction affine $f : x \mapsto 3x + 1$. Exécuter les instructions suivantes.

```
function y=f(x)
y=3*x+1
endfunction
```

On appelle ensuite la fonction simplement par $\mathbf{f}(\dots)$. Vérifier le résultat de $\mathbf{f}(\mathbf{0})$.

Q.8 Définir la fonction \mathbf{g} telle que $g : x \mapsto \ln(x - 2)$. Que devrait valoir $g(0)$? Essayer d'exécuter $\mathbf{g}(\mathbf{0})$. On retiendra qu'il arrive que Scilab effectue des calculs que nous ne comprenons pas, là où l'on voudrait voir apparaître une erreur.

Pour écrire une fonction, pas nécessairement mathématique, on utilise la même construction. On écrira

```
function variable(s)_de_sortie = nom_de_la_fonction (variable(s)_d_entree)
... //instructions
endfunction
```

La série d'instruction découle de la traduction en Scilab d'un algorithme préconçu. Dans cet algorithme, on donne des valeurs aux variables de sortie. Les dernières valeurs assignées aux variables `_de_` `_sortie` sont celles qui sont renvoyées.

Pour écrire les programmes en Scilab, on peut utiliser la console ou bien créer un document texte contenant les instructions et en l'enregistrant au format `.sce`. On utilisera plutôt l'application SciNote, incluse dans Scilab, qui facilite la mise en forme.

Q.9 Ouvrir une fenêtre SciNote et écrire un programme **addition** qui prend en arguments deux nombres réels et renvoie leur somme. Enregistrer le fichier `.sce` dans le dossier courant. Pour exécuter un fichier `.sce`, on écrira dans la console **exec nom_du_fichier.sce**.

Q.10 Écrire un programme **plus-moins** qui prend en arguments deux nombres réels et renvoie leur somme et leur différence.

Q.11 Écrire un programme **addition2** qui ne prend pas d'argument mais demande à l'utilisateur deux nombres réels et renvoie leur somme. On consultera l'aide de la commande **input**.