ECE 1 - Année 2015-2016
Lycée français de Vienne
Mathématiques - F. Gaunard
`http://frederic.gaunard.com`

Informatique - T.P n°1

À la découverte de SciLab

On commencera par créer, dans son dossier personnel, un dossier intitulé "TP1" dans lequel on pourra enregistrer tous les programmes (au format *.sce*) ou fonctions (au format *.sci*) relatifs à cette première feuille d'exercices.

1. Calculs non programmés

Quand les commandes à taper sont simples et peu nombreuses, on peut directement utiliser la feuille de travail SciLab (appelée *console*).
L'invite `-->` de la console indique que SciLab est prêt pour exécuter de nouvelles commandes.

Exercice 1. (SciLab comme une calculatrice)

(1) Calculer le résultat des opérations suivantes

$$\frac{1}{2\times 3};\qquad \frac{1}{2}\times 3;\qquad 7^4;\qquad \frac{1+\sqrt{5}}{2};\qquad \frac{1}{1+\frac{1}{1+2}};\qquad \left|\frac{2}{3}-\frac{\sqrt{2}}{2}\right|.$$

(2) Déterminer la valeur du plus grand entier naturel inférieur ou égal à $\ln(385)$.
(3) Quelle est la valeur de π? De e? De $\pi\times e$?
(4) (i) Calculer $\ln(3)\cdot e^2$.
(ii) Puis, à l'aide de la variable `ans`, calculer

$$\frac{\ln(3)\cdot e^2}{1+\ln(\frac{7}{8})}.$$

(iii) Peut-on encore utiliser `ans` pour calculer $\ln(3)\cdot e^2\cdot\sqrt{0,1}$? Taper la commande nécessaire au calcul de l'expression précédente (on pourra utiliser les touches $\uparrow$ ou $\downarrow$).

Exercice 2. (Affectation)

(1) (i) Créer une variable `a` puis lui affecter la valeur 10. Taper ensuite dans l'invite de commande `a=a*10`. Que vaut maintenant `a`?
(ii) Créer et affecter la valeur $5e$ à une variable `b`. Calculer $a+b$.
(iii) Taper la commande `clear` puis à nouveau `a`. Que se passe-t-il?

(2) (i) Écrire une suite de deux instructions qui affecte à la variable `a` le nombre 10, calcule le nombre $(1+\frac{1}{a})^a$ et fait afficher cette valeur.
(ii) En rappelant les instructions précédentes dans la ligne de commande, faire afficher le nombre $(1+\frac{1}{100})^{100}$, puis $(1+\frac{1}{1000})^{1000}$.
(iii) Faire afficher la valeur du nombre e. Que peut-on conjecturer?

Exercice 3. (Une limite connue)

(1) Rappeler la valeur de
$$\lim_{\substack{x\to 0\\ \neq}} \frac{e^x-1}{x}.$$
(2) Illustrer cette propriété à l'aide d'une suite d'instructions dans l'invite de commande.
(3) Faire le calcul de $\frac{e^x-1}{x}$ avec $x = 10^{-15}$, $x = 5.0 \times 10^{-16}$ ou encore $x = 10^{-16}$. Que se passe-t-il?

Remarque 1. *(Erreurs d'arrondi)*
Les nombres réels sont représentés par des flottants (sorte d'équivalent informatique de l'écriture scientifique). Cela dit, il n'est pas possible de représenter tous les réels exactement sur un ordinateur (par exemple, $0,2$ *s'écrit - en base* 2 *- 0,00110011...). La machine arrondit donc tout résultat vers le plus proche nombre flottant. La précision de cet arrondi est définie par l'*$\epsilon-$*machine; c'est la distance entre* 1 *et le plus proche nombre flottant qui lui est supérieur. Il vaut*

```
%eps = 2.220E-16
```

En particulier, aucun nombre compris entre 1 *et* $1+\frac{\epsilon}{2}$ *n'est représentable. Il y a donc un arrondi dans la représentation machine et des erreurs d'arrondi qui peuvent s'accumuler au fur et à mesure du calcul.*

Exercice 4. (Erreurs d'arrondi et erreur numérique)

(1) Définir et affecter la valeur $1+\epsilon$ à une variable `a`.
(2) On pose $m = (a+1)/2$ et $n = (a-1)/2$. Qu'observe-t-on?
(3) Calculer $2(m-n)$. Commenter.

2. Premiers programmes

Un programme est une suite d'instructions regroupées dans un fichier. On écrira les programmes avec l'éditeur de textes `SciNotes`, intégré à `SciLab`.

La commande `input` permet de faire lire la valeur d'une variable au cours de l'exécution d'un programme (on peut par exemple écrire la commande `n=input('entrez un nombre entier')` pour que le programme demande à l'utilisateur de rentrer un entier dont la valeur sera affecté à la variable n.

On utilisera la commande `disp` (pour *display*) pour afficher la valeur d'une variable (ou une chaîne de caractères).

Il ne faut pas hésiter à utiliser le symbole `//` afin de commenter les lignes de code pour s'y retrouver!

Enfin, on signale que les eux opérareurs logiques à connaître sont `&` (signifiant "et") et `|` (signifiant "ou").

La syntaxe des alternatives `if...` et des boucles `for` et `while` sera précisée au cours du TP.
Si la syntaxe des fonctions sera également précisée, on insiste sur le fait que la suite d'instructions qui définit une fonction doit toujours être délimitée par les mots `function` et `endfunction` et qu'avant d'être utilisée (dans un calcul, un programme ou une autre fonction), le fichier la contenant doit être chargé dans l'environnement de travail (avec la commande `exec`).

Exercice 5. (Petits programmes entre amis)

(1) Écrire un programme qui renvoie le pourcentage d'évolution entre deux valeurs rentrées par l'utilisateur.
(2) Écrire un programme qui demande un nombre réel x et, après discussion, renvoie la racine carrée de celui-ci quand cela est possible.
(3) Écrire un programme qui, en fonction des trois paramètres a, b, c rentrés par l'utilisateur, renvoie les solutions réelles de l'équation $ax^2 + bx + c = 0$.
(4) Écrire un programme qui demande un nombre réel x et renvoie 1 si $x \in [0; 1]$ et 0 sinon.
(5) Écrire un programme qui demande un nombre réel x et renvoie 1 si $x \in]-\infty; -2] \cup] \ln(2); +\infty[$ et 0 sinon.
(6) Écrire un programme, faisant intervenir une boucle `for`, qui demande un entier $n \geq 0$, calcule et affiche $n! = 1 \times 2 \times 3 \times \cdots \times n$. (On rappelle - notamment pour l'initialisation - que, par convention, $0! = 1$.)
(7) Écrire un programme, faisant intervenir une boucle `while`, qui demande un nombre positif t, calcule et affiche en combien d'années une valeur subissant une hausse annuelle de $t\%$ aura doublé.

Exercice 6. (Un comportement asymptotique)

Écrire un programme qui demande un entier $n > 0$ et qui, en utilisant une boucle `for`, calcule et affiche

$$1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} - \ln(n).$$

Faire tourner ce programme pour de plus en plus grandes valeurs de n. Que remarque-t-on?

Exercice 7. (Programme vs. Fonction)

(1) Ecrire les programmes des (4) et (5) de l'Exercice 5 sous forme de fonctions.
(2) (i) Utiliser les instructions du programme du (6) de l'Exercice 5 pour écrire une fonction intitulée `factorielle` prenant en argument un entier $n \geq 0$ et renvoyant la valeur $n!$.
(ii) Utiliser cette fonction pour écrire un programme qui demande à l'utilisateur un nombre réel positif x et renvoie le plus petit entier naturel n tel que $n! \geq x$. (On utilisera une boucle `while`.)

La fonction `rand` *renvoie un nombre au hasard selon une loi de probabilité définie à l'avance. Par exemple, en tapant* `rand('uniform')`*, on indique à la machine, que chaque appel* `rand()` *simule une loi uniforme* $\mathcal{U}([0; 1])$.

Exercice 8. (Boules)

(1) Écrire un programme qui simule le tirage au hasard d'une boule dans une urne qui en contient trois blanches et une noire et qui renvoie la couleur de la boule choisie.
(2) Modifier le programme pour effectuer un second tirage, sans remise, et afficher le couple de couleurs obtenues.

Exercice 9. (Un dé à 6 faces)

(1) Écrire une instruction qui permet de simuler le jet d'un dé (non truqué) à six faces.
(2) Écrire ensuite un programme qui, à partir d'un nombre n de lancers entré par l'utilisateur renvoie la fréquence d'obtention d'un 6.
(3) Faire tourner le programme précédent pour des valeurs de plus en plus grandes de n. Que semble-t-on observer?

Exercice 10. (Un jeu et un compteur)

Écrire un programme qui, après avoir tiré (secrètement) au hasard un nombre entier entre 1 et 10, demande à l'utilisateur de le deviner et compte le nombre de coups nécessaires.