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

Informatique - T.P n°2

Suites numériques

On commencera par créer, dans son dossier personnel, un dossier intitulé "TP2" dans lequel on pourra enregistrer tous les programmes (au format *.sce*) ou fonctions (au format *.sci*) relatifs à cette nouvelle feuille d'exercices.

1. Calcul des termes d'une suite

Exercice 1.

(1) Deviner l'affichage du programme suivant avant de l'exécuter:

```
u=0
for i=1:6
    u=2*u+3
    disp(u)
end
```

(2) Quelle est la suite définie dans ce programme?
(3) Modifier le programme précédent pour qu'il affiche uniquement u_{100}.
(4) Ecrire un programme sans boucle `for` qui affiche le terme u_n.

Exercice 2.

(1) Dans chacun des cas suivants, écrire un programme qui demande à l'utilisateur un entier n et affiche le terme u_n :
  (a) $u_0 = 0.1$ et, pour tout $n \geq 0$, $u_{n+1} = e^{-2u_n}$.
  (b) $u_1 = 1$ et, pour tout $n \geq 1$, $u_{n+1} = 2nu_n + 3$.
(2) Modifier les programmes précédents pour en faire des fonctions.

Exercice 3.

Soit (u_n) la suite définie par

$$\begin{cases} u_1 & = & 0 \\ u_2 & = & -9 \\ u_{n+2} & = & 6u_{n+1} - 9u_n \end{cases}$$

On pose, pour tout $n \geq 1$, S_n la somme des n premiers termes de la suite.

(1) Ecrire un programme qui demande n à l'utilisateur et affiche la valeur de u_n.
(2) Compléter le programme précédent afin qu'il renvoie également la valeur de S_n.

Exercice 4. Soit (u_n) la suite définie par

$$\begin{cases} u_0 & \in & \mathbb{R} \\ u_{n+1} & = & u_n - u_n^2 \end{cases}$$

(1) Que peut-on dire du sens de variation de (u_n)?

(2) Ecrire un programme qui demande à l'utilisateur un réel u_0 et un entier n et affiche le terme u_n.
(3) On suppose que $u_0 = \frac{3}{4}$.
   (a) Calculer u_n pour des valeurs de plus en plus grandes de n. Que semble-t-il se produire?
   (b) Ecrire un programme qui détermine la plus petite valeur de n telle que $u_n < 10^{-3}$.
(4) On suppose que $u_0 = 2$.
   (a) Calculer u_n pour des valeurs de plus en plus grandes de n. Que semble-t-il se produire?
   (b) Ecrire un programme qui détermine la plus petite valeur de n telle que $u_n < -10^8$.

Exercice 5. Soit (u_n) la suite définie par

$$\begin{cases} u_0 & = & 4 \\ u_{n+1} & = & u_n^2 + 1 \end{cases}$$

(1) Ecrire un programme qui demande à l'utilisateur un réel x et renvoie la valeur du premier terme de la suite qui soit supérieur ou égal à x.
(2) Modifier le programme pour qu'il renvoie également l'indice de ce terme.
(3) Réécrire ce programme pour en faire une fonction (attention, il y aura deux valeurs de sortie!).

Exercice 6.

(1) Ecrire une fonction qui, prenant pour argument l'entier n, permet de calculer $S_n = \sum_{k=1}^{n} \frac{1}{k}$.
(2) Utiliser cette fonction pour conjecturer les valeurs de

$$\lim_{n\to+\infty} S_n, \qquad \lim_{n\to+\infty} \frac{S_n}{n}, \qquad \lim_{n\to+\infty} \frac{S_n}{\ln(n)}.$$

(3) Ecrire une fonction qui à deux entiers n et p associe

$$\sum_{k=1}^{n} k^p.$$

(4) Ecrire un programme qui compare les valeurs obtenues avec la fonction précédentes et les formules classiques du cours.

Exercice 7.

On veut déterminer la limite α de la suite (u_n) définie par

$$\begin{cases} u_0 & = & 0 \\ u_{n+1} & = & \sqrt{a + u_n}, \end{cases}$$

où $a > 0$ est un paramètre fixé.

(1) On fait l'hypothèse suivante: si $|u_{n+1} - u_n| < 0.0005$, alors u_n est est une approximation de la limite de la suite à 0.001 près. Ecrire un programme qui utilise cette hypothèse pour donner une valeur approchée de la limite de (u_n) à partir du paramètre a entré par l'utilisateur.
(2) On peut montrer, mathématiquement, que cette limite vaut

$$\alpha = \frac{1 + \sqrt{1 + 4a}}{2}.$$

Compléter le programme précédent pour vérifier que l'approximation précédente est bien vérifiée.

Remarque 1. *L'hypothèse précédente n'est pas toujours vraie, mais elle l'est ici. On peut l'utiliser par exemple si on arrive à montrer que*

$$|u_{n+1} - \alpha| \leq \frac{1}{2}|u_n - \alpha|.$$

En effet, dans ce cas l'inégalité triangulaire nous donne

$$|u_n - \alpha| \leq |u_{n+1} - \alpha| + |u_{n+1} - u_n| \leq \frac{1}{2}|u_n - \alpha| + |u_{n+1} - u_n|$$

ou encore, en factorisant par $|u_n - \alpha|$

$$|u_n - \alpha| \leq 2|u_{n+1} - u_n|.$$

Ainsi, si $|u_{n+1} - u_n| \leq 0.0005$*, on a bien* $|u_n - \alpha| \leq 0.001$.

2. Suite en tant que vecteur ligne

Les exercices précédent avaient tous pour principe le calcul d'un terme de la suite. Ainsi, on calculait les termes individuellement. On peut également définir la suite comme "liste" de valeurs, et le terme de rang n sera donc le $n-$ième terme de cette liste (ou $(n+1)-$ième si la suite commence pour $n = 0$).

On représente une liste sous `SciLab` sous forme de **vecteur-ligne**. On définit un vecteur-ligne en indiquant la suite des nombres qui le composent, séparés par des virgules, le tout entre crochets [ ].

Si le vecteur-ligne `u` est défini par `u=[0,3,7,12,15]`, on accède au $k-$ième terme via `u(k)`. Dans le cas précédent, `u(3)` renvoie donc `7`.

L'instruction `length(u)` renvoie la longueur de `u` (c'est à dire le nombre de coefficients qui le composent, ici `length(u)` va donc renvoyer `5`).

Si un vecteur-ligne possède n coefficient, on peut le compléter en affectant directement la valeur du $p-$ième coefficient (avec $p > n$). Si $p > n + 1$, les coefficients intermédiaires seront automatiquement affectés de la valeur 0.
Par exemple, si on écrit `u(9)=20`, `SciLab` affichera ensuite
`u=0.   , 3.   , 7.   , 12.   , 15.   , 0.   , 0.   , 0.   , 20.`

On peut donc facilement remplir, à l'aide d'une boucle `for` par exemple, toutes les composantes d'une suite (définie sous forme de vecteur-ligne).

Cela dit, cette commande est chère en temps de calcul. Il vaut mieux définir dès le départ la taille du vecteur-ligne, puis modifier ensuite chaque terme. La commande `zeros(1,n)` permet par exemple de créer un vecteur ligne de n termes, pré-rempli avec des zéros.

De plus, la fonction `find()` renvoie la position (c'est à dire le rang) des éléments du vecteur-ligne qui vérifient une condition donnée. Par exemple, `find(u>10)` renverra `4.   , 5.   , 9.` qui sont les valeurs de k telles que $u_k > 10$.

Enfin, la fonction `sum()` prenant en argument un vecteur-ligne renvoie la somme de toutes les composantes de ce vecteur-ligne, ce qui est bien pratique.

Exercice 8. On considère la suite (u_n) définie par

$$\begin{cases} u_1 & = & 5 \\ u_{n+1} & = & u_n - 2, \end{cases}$$

(1) Ecrire une suite d'instruction qui permet d'obtenir les 20 premiers termes de la suite sous forme de vecteur-ligne.
(2) En déduire la valeur de la somme des 20 premiers termes de (u_n).
(3) Reprendre les mêmes questions avec $u_1 = 16$.

Exercice 9. Ecrire, sur une seule ligne de commande, les intructions qui permettent de calculer la somme des 15 premiers termes de la suite géométrique de premier terme $v_1 = 3$ et de raison $\frac{4}{3}$.

Exercice 10. Reprendre l'Exercice 5 avec la notion de vecteur.

Si a, b, c sont trois réels avec $b > 0$ et $a \leq c$, la commande `SciLab a : b : c` permet de définir (en une seule instruction) le vecteur-ligne composé des termes d'une suite arithmétique de premier terme a, de raison b et dont le dernier terme sera le plus grand terme de la suite inférieur à c.

Par exemple, `1:1:20` renvoie le vecteur-ligne composé des 20 premiers entiers consécutifs.

Exercice 11. (Sommes)

(1) Ecrire une fonction, qui ne comporte qu'une seule instruction, prenant pour argument n et renvoyant

$$\sum_{k=0}^{n} (3n+4).$$

(2) Ecrire une fonction prenant pour argument n et renvoyant

$$\sum_{1 \leq i \leq j \leq n} ij.$$

Exercice 12.

(1) Definir un vecteur-ligne `u` de longueur 40 composé des 40 premiers termes de la suite de Fibonacci, notée ici (u_n), à partir de $u_1 = 1$.
(2) Définir, à partir de `u`, un vecteur-ligne `v` représentant les 40 premiers termes de la suite (v_k) définie pour $k \geq 0$ par $v_k = u_{k+1}$.
(3) Que fait la commande `a=v./u` ?
(4) Calculer les 20 premiers termes de la suite (ϕ_n) définie pour $n \geq 1$ par

$$\phi_n = f\left(\frac{u_{n+1}}{u_n}\right), \quad \text{où} \quad f(x) = x^2 - x - 1.$$

(5) On admet que (ϕ_n) converge vers une limite ϕ. Au vu des dernières questions, que peut-on conjecturer quant à la valeur exacte de ϕ?