Chapitre 7 : La boucle TANT QUE.

Voici une deuxième technique pour effectuer des boucles, c'est-à-dire itérer un processus. Dans une boucle « for », le nombre d'itérations est connu à l'avance. Dans une boucle « tant que », on répète une suite d'instructions un nombre de fois qui nous est inconnu au moment d'effectuer la première itération.

I Exemple

Écrire le script suivant et le tester pour les valeurs suivantes de a : 3, 10, 100, -1.

```
a=input('entrez un seuil ')
u=1
while u<a then u=2*u
end
disp(u,'valeur de u :  ')
```

Faire un tableau montrant l'évolution des variables u<a (booléen) et a. Décrire l'affichage en sortie, c'est-à-dire le contenu de la variable u.

Rajouter quelques lignes comme ci-dessous, puis faire fonctionner à nouveau le programme pour les mêmes valeurs de a que précédemment. A quoi sert la variable k ?

```
a=input('entrez un seuil ')
u=1
k=0
while u<a then u=2*u,k=k+1
end
disp(u,'valeur de u :  ')
disp(k,'valeur de k  : ')
```

II Syntaxe de la boucle WHILE avec Scilab

L'instruction `while condition then instruction(s) ; end` permet de répéter la ou les instructions un certain nombre de fois (tant que la condition est vraie).

Remarques :

- Le « then » est facultatif, mais l'instruction apparaît ainsi dans le programme officiel ECS.
- La boucle peut ne pas tourner du tout si le test est faux dès le premier passage.
- Cette boucle sert surtout à afficher le plus petit entier tel que la condition est fausse, d'où l'importance de la notion de compteur.

TP 7 :

Ex 1 : Soit la suite définie par $U_0=1$ et pour tout entier n supérieur ou égal à 0 : $U_{n+1}=U_n^2+1$.
On admet que cette suite diverge vers $+\infty$.
Ecrire un script qui demande à l'utilisateur un seuil A puis qui calcule et affiche le rang du premier terme de cette suite qui est strictement supérieur à A.

Ex 2 : Soit la suite définie par $U_0=1$ et pour tout entier n supérieur ou égal à 0 :
$U_{n+1}=\frac{1}{2}\left(U_n+\frac{2}{U_n}\right)$. On admet que cette suite converge vers $\sqrt{2}$.
Écrire un script qui demande à l'utilisateur un seuil ε puis qui calcule et affiche le rang du premier terme de cette suite qui vérifie $|U_n-\sqrt{2}|\leq\varepsilon$. Faire fonctionner le programme pour ε=0,01 puis ε=0,000 001.
Noter les résultats pour comparer plus tard avec ceux du dernier exercice.

Ex 3 : Soit $S_n=\sum_{k=1}^{n}\frac{1}{k^2}$ pour tout entier n supérieur ou égal à 1.

On admet que cette suite converge vers $\frac{\pi^2}{6}$. Calculer le plus petit entier n tel que $\left|S_n-\frac{\pi^2}{6}\right|\leq 0{,}01$.

Ex 4 : La suite dite « de Syracuse » est la suite (u_n) définie par : $u_0=a$, a étant un entier naturel non nul entré par l'utilisateur, puis pour tout $n\in\mathbb{N}$: $u_{n+1}=\frac{u_n}{2}$ si u_n est pair et $u_{n+1}=3u_n+1$ si u_n est impair.

Une conjecture célèbre (non encore prouvée) dit que l'on finit toujours par obtenir 1.
Écrire un programme pour qu'il affiche le plus petit entier n tel que $u_n=1$.

Rappel : n pair ssi n=2xpartie entière de n/2

Ex 5 : On a montré (feuille sur les suites) que les suites suivantes étaient adjacentes : $S_n=\sum_{k=0}^{n}\frac{1}{k!}$ et $T_n=S_n+\frac{1}{n.n!}$ définies à partir de n=1. Si on note L leur limite commune, on a (voir toujours l'exercice) : $\left|L-\frac{S_n+T_n}{2}\right|\leq\frac{T_n-S_n}{2}$. Ecrire un script qui demande à l'utilisateur un réel ε puis qui calcule et affiche une valeur approchée de L à ε près (c'est-à-dire α tel que $|L-\alpha|\leq\varepsilon$) .

Ex 6 : **Résolution d'équation par dichotomie**.

On veut résoudre l'équation f(x)=0. On sait que dans l'intervalle]a,b[, il existe une unique solution x_0 grâce au théorème de la limite monotone : f est strictement monotone et continue sur [a,b] et 0 est dans l'intervalle image f(]a,b[), autrement dit f(a)f(b)<0.

Alors x_0 est dans $\left]a,\frac{a+b}{2}\right]$ ou bien dans $\left]\frac{a+b}{2,}b\right[$

selon que $f(a)f(\frac{a+b}{2})>0$ ou bien $f(a)f(\frac{a+b}{2})\leq 0$. On définit donc par récurrence les suites (a_n) et (b_n) par : $a_0=a$, $b_0=b$ et pour tout entier naturel n :

$$a_{n+1}=\begin{pmatrix} a_n \ \ si \ \ f(a_n)f\left(\frac{a_n+b_n}{2}\right)\leq 0 \\ \frac{a_n+b_n}{2} \ \ si \ \ f(a_n)f\left(\frac{a_n+b_n}{2}\right)>0 \end{pmatrix} \quad \text{et} \quad b_{n+1}=\begin{pmatrix} \frac{a_n+b_n}{2} \ \ si \ \ f(a_n)f\left(\frac{a_n+b_n}{2}\right)\leq 0 \\ b_n \ \ si \ \ f(a_n)f\left(\frac{a_n+b_n}{2}\right)>0 \end{pmatrix}$$

1) On prend f(x)=x-1, a=0 et b=3. Calculer les quatre premiers termes des suites (a_n) et (b_n) définies ci-dessus.

Retour au cas général :

2) Exprimer $a_{n+1}-b_{n+1}$ et fonction de a_n-b_n. En déduire que la suite (a_n-b_n) converge vers 0.

Comme la suite (a_n) est croissante et (b_n) décroissante par construction, on a donc une couple de suites adjacentes.

3) Supposons f strictement croissante sur [a,b]. On a alors pour tout entier n : $f(a_n)\leq 0$ et $f(b_n)\geq 0.$ En déduire que la limite commune de (a_n) et (b_n) est x_0 .

Le principe de résolution d'une équation par dichotomie est de calculer les termes des suites (a_n) et (b_n) jusqu'à s'être suffisamment approché de leur limite.

On a $0\leq x_0-a_n\leq b_n-a_n$ donc a_n est une approximation de x_0 par défaut à la précision $\varepsilon=b_n-a_n$.

Mise en œuvre sur des exemples :

4) On considère l'équation $\ln(x)+x=0$. Montrer qu'il existe une unique solution dans $\left]\frac{1}{e};1\right[$.

5) Écrire un script permettant de trouver et afficher une valeur approchée de la solution à 0,001 près.

6) Modifier le script pour qu'il affiche aussi le nombre d'itérations effectuées.

7) Faire fonctionner le principe de dichotomie pour approcher le réel $\sqrt{2}$ avec l'équation $x^2-2=0$ sur $]1;2[$ à ε=0,01 puis ε=0,000 001 près et noter le nombre d'itérations. Comparer les résultats avec ceux obtenus dans l'exercice 2.