Tale MC — Algorithmes & Scripts PYTHON- Correction — Nov 2020

Ex 1 :

La suite (u_n) est définie par $u_0=500$ et $\forall\, n\geqslant 1 \quad : \quad u_{n+1}=0{,}9\times u_n+1$

Le tableau d'avancement du « pseudo-code » est le suivant :

n	0	1	2	3	4	5
u	500	451	406,9	367,21	332,49	300,24
test	VRAI	VRAI	VRAI	VRAI	VRAI	FAUX

Le programme PYTHON corrigé est donné ci-contre

```
File  Edit  Format  Run  O
# CODE PYTHON
def suite(N):
    assert N>=0
    u=500
    n=0
    while n<N:
        n=n+1
        u=0.9*u+1
    return n,u
```

<u>Conjecture</u> : $\lim\limits_{n\to+\infty}(u_n)=10$

<u>Preuve</u> : On pose la suite auxiliaire $v_n=u_n-10$

donc $v_{n+1}=u_{n+1}-10=0{,}9\,u_n+1-10=0{,}9\,u_n-9$

$=0{,}9(u_n-10)=0{,}9\,v_n$

donc (v_n) est une suite géométrique de 1er terme $v_0=490$ et de raison $q=0{,}9$

donc $v_n=490\times 0{,}9^n$

donc $u_n=10+490\times 0{,}9^n$

or $0<0{,}9<1$ donc $\lim\limits_{n\to+\infty}(0{,}9^n)=0$

donc $\lim\limits_{n\to+\infty}(u_n)=10+490\times 0=10$

Ex 2 :

La suite (u_n) est définie par $u_0=1000$ et $\forall\, n\geqslant 1 \quad : \quad u_{n+1}=1{,}02\times u_n+2400$

Le rôle de ces 2 programmes est de déterminer au bout de combien d'années une certaine somme (nommée « seuil ») sera atteinte avec ce placement bancaire

En utilisant le script PYTHON ci-contre on obtient le résultat suivant :

```
Python 3.6.1 Shell
File  Edit  Shell  Debug  Options  Window  Help
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:5
 on win32
Type "copyright", "credits" or "license()" for
>>>
 RESTART: f:/Users/Utilisateur/Desktop/Lycée 2(
ites/ex2_algo_fiche1.py
>>> recherche(18000)
(7, 18990.96578556288)
>>> |
```

```
File  Edit  Format  Run  Options
# CODE PYTHON
def recherche(seuil):
    assert seuil>=1000
    u=1000
    n=0
    while u<=seuil:
        n=n+1
        u=1.02*u+2400
    return n,u
```

soit au bout de 7 années

Ex 3 : On considère la suite (v_n)

définie par $v_{n+1}=\dfrac{v_n-2}{3\,v_n-4}$

On obtient le script PYTHON ci-contre

```
File  Edit  Format  Run  Options  Window
# CODE PYTHON
def suite2(N,v0):
    assert N>=0
    v=v0
    for i in range(1,N+1):
        v=(v-2)/(3*v-4)
    return v
```

on obtient les résultats suivants :

```
ites/ex3_algo_fiche1.py
>>> suite2(100,2)
0.6666666666666666
>>> suite2(100,3)
0.6666666666666664
>>> suite2(100,1)
1.0
>>> suite2(100,4)
0.6666666666666664
>>> suite2(100,-1)
0.6666666666666667
>>> |
```

Ainsi si $v_0=1$ alors $\lim\limits_{n\to+\infty}(v_n)=1$

si $v_0=2$ alors $\lim\limits_{n\to+\infty}(v_n)=\dfrac{2}{3}$

si $v_0=3$ alors $\lim\limits_{n\to+\infty}(v_n)=\dfrac{2}{3}$

si $v_0=-1$ alors $\lim\limits_{n\to+\infty}(v_n)=\dfrac{2}{3}$

<u>Conjecture</u> : la suite (v_n) converge vers $\dfrac{2}{3}$ si $v_0\neq 1$

<u>Preuve</u> : On pose la suite (w_n) définie par $w_n=\dfrac{3\,v_n-3}{3\,v_n-2}$

Ainsi $$w_{n+1}=\frac{3\,v_{n+1}-3}{3\,v_{n+1}-2}=\frac{3\left(\dfrac{v_n-2}{3\,v_n-4}\right)-3}{3\left(\dfrac{v_n-2}{3\,v_n-4}\right)-2}=\frac{3(v_n-2)-3(3\,v_n-4)}{3(v_n-2)-2(3\,v_n-4)}$$

$$=\frac{-6\,v_n+6}{-3\,v_n+2}=\frac{6\,v_n-6}{3\,v_n-2}=\frac{2(3\,v_n-3)}{v_n-2}=2\,w_n$$

donc (w_n) est une suite géométrique de 1er terme $w_0=\dfrac{3\,v_0-3}{v_0-2}\neq 1$ et de raison $q=2$ donc on déduit que $w_n=w_0\times 2^n \quad , \quad \forall\, n\in\mathbb{N}$

par ailleurs $w_n=\dfrac{3\,v_n-3}{3\,v_n-2}$ donc $w_n(3\,v_n-2)=3\,v_n-3$

donc $3\,w_n\times v_n-2\,w_n=3\,v_n-3$ donc $3\,w_n\times v_n-3\,v_n=2\,w_n-3$

donc $v_n(3\,w_n-3)=2\,w_n-3$ donc $v_n=\dfrac{2\,w_n-3}{3\,w_n-3}=\dfrac{2\,w_n-2-1}{3\,w_n-3}$

soit encore $v_n=\dfrac{2(w_n-1)-1}{3(w_n-1)}=\dfrac{2(w_n-1)}{3(w_n-1)}-\dfrac{1}{3(w_n-1)}=\dfrac{2}{3}-\dfrac{1}{3(w_n-1)}$

On remarquera que cette écriture reste valide uniquement si $w_n\neq 1$ ce qui est vraie par construction même de la suite (w_n)

On obtient alors $v_n=\dfrac{2}{3}-\dfrac{1}{3(w_0\times 2^n-1)}$

or $2>1$ donc $\lim\limits_{n\to+\infty}(2^n)=+\infty$ donc $\lim\limits_{n\to+\infty}(3(w_0\times 2^n-1))=+\infty$

donc $\lim\limits_{n\to+\infty}\left(\dfrac{-1}{3(w_0\times 2^n-1)}\right)=0$ et (enfin !) $\lim\limits_{n\to+\infty}(v_n)=\dfrac{2}{3}$

Rque : Bien sûr, j'ai déposé cette preuve pour que chacun puisse comprendre l'origine de cette limite ; cependant, son calcul proposé ici est du programme de « spé maths »

Ex 4 : Soient les suites (u_n) et (v_n) définies par
$u_{n+1}=0{,}8\,u_n+45$; $u_0=150$ et $v_n=u_n-225$

on obtient les valeurs suivantes :
$u_1=165$; $u_2=177$; $u_3=186{,}6$; $u_4=194{,}28$;

$u_5=200{,}424$; $v_1=-60$; $v_2=-48$

$v_3=-38{,}4$; $v_4=-30{,}72$; $v_5=-24{,}58$

```
File  Edit  Format  Run  Options
# CODE PYTHON
def f(N):
    assert N>=0
    n,u,v=0,150,-75
    while n<N:
        n=n+1
        u=0.8*u+45
        v=u-225
    return (n,u,v)
```

Le script PYTHON est donné ci-contre
on obtient les résultats suivants :

```
>>>
 RESTART: f:/Users/Utilisateur/Desktop/Lycée 202
ites/ex4_algo_fiche1.py
>>> f(20)
(20, 224.13530887154494, -0.864691128455064)
>>> f(100)
(100, 224.9999999847223, -1.527769200038165e-08)
>>> f(200)
(200, 225.0, 0.0)
>>>
```

Conjectures : $\lim\limits_{n\to+\infty}(u_n)=225$ et $\lim\limits_{n\to+\infty}(v_n)=0$

Preuve :

$v_n=u_n-225$ donc on déduit que
$v_{n+1}=u_{n+1}-225=0{,}8\,u_n+45-225=0{,}8\,u_n-180=0{,}8(u_n-225)=0{,}8\,v_n$

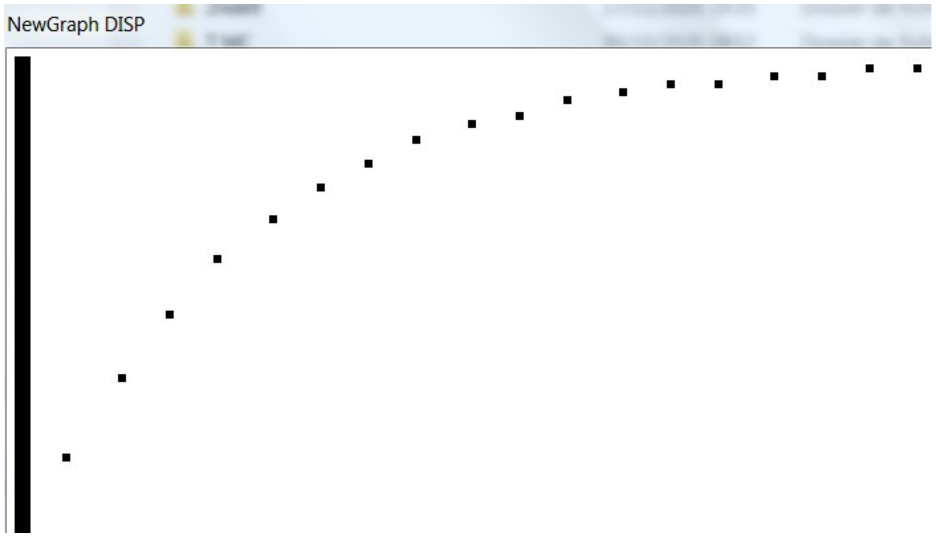
donc la suite (v_n) est géométrique de 1er terme $v_0=-75$ et de raison $q=0{,}8$

donc $v_n=-75\times 0{,}8^n$, $\forall\, n\in\mathbb{N}$
ainsi on obtient : $u_n=225-75\times 0{,}8^n$, $\forall\, n\in\mathbb{N}$

or $0<00{,}8<1$ donc $\lim\limits_{n\to+\infty}(0{,}8^n)=0$

donc $\lim\limits_{n\to+\infty}(v_n)=-75\times 0=0$ et $\lim\limits_{n\to+\infty}(u_n)=225-75\times 0=225$

On vérifie ces résultats avec le nuage de points de la suite (v_n)



Rques : on observe également que :

- (u_n) *est croissante*
- (u_n) *est minorée par 150 et majorée par 225*
- (u_n) *converge vers 225*