Correction des questions faisant intervenir un algorithme

Exercice 1

On veut calculer plusieurs termes de la suite sans savoir à l'avance le nombre de termes que l'on veut calculer : il faut utiliser une boucle while. On appelle ce genre de question un problème de seuil.

```
def seuil():
    u = 80
    n = 0
    while u >= 40 :
        u = 0.8*u + 2
        n = n + 1
    return n
```

On peut calculer *(même sans avoir trouvé le programme)* que $u_3 = 45{,}84$ et $u_4 = 38{,}672$, donc l'entier n recherché est 4 : c'est ce que renverra la fonction.

Exercice 2

a.

$$v_1 = \frac{2 \times \frac{1}{2}}{1 + \frac{1}{2}} = \frac{1}{\frac{3}{2}} = \frac{2}{3}$$

b. *Ici, on sait à l'avance combien de répétitions sont nécessaires dans la boucle : il s'agit donc d'une boucle for.*

```
def suite(n):
    v = 1/2
    for k in range(n):
        v = (2*v)/(1 + v)
    return v
```

Exercice 3

a. D'après l'algorithme, $T_{n+1} = 0{,}82T_n + 3{,}6$.

b. *On peut reproduire le programme, mais on peut aussi rentrer la suite dans la calculatrice et afficher le tableau de valeurs.* $T_4 \approx 463$.

c.

```
def temps():
    T = 1000
    n = 0
    while T >= 70
        T = 0.82*T + 3.6
        n = n + 1
    return n
```

Au Bac, il ne serait sans doute pas pénalisé de mettre « T > 70 », même si le contraire de « inférieur » est « supérieur ou égal ».

d. *À nouveau, on peut utiliser le programme ci-dessous, ou bien poursuivre le tableau de valeurs de la suite à la calculatrice.* On trouve $n = 15$.

Exercice 4

1. *Le but est de vérifier que vous avez bien compris l'expression de la suite. Attention, le premier terme est* $u_1 = 0$.

$$u_2 = (1+1)u_1 - 1 = 2 \times 0 - 1 = -1$$
$$u_3 = (2+1)u_2 - 1 = 3 \times (-1) - 1 = -4$$
$$u_4 = (3+1)u_3 - 1 = 4 \times (-4) - 1 = -17$$

2. *En Python, range(1,n) contient tous les entiers de 1 inclus à n exclu. Cela correspond à ce qu'on veut faire, car le premier terme est d'indice 1 et non 0. Remarquez que dans le programme, le k de la boucle correspond au n de la formule.*

```
def suite(x,n):
    u = x
    for k in range(1,n):
        u = (k + 1)*u - 1
    return u
```

3. La suite semble diverger vers $-\infty$ si $u_1 = 0{,}7$ et diverger vers $+\infty$ si $u_1 = 0{,}8$.

Exercice 5

1a. $f(20) \approx 0{,}0307$.

1b. Le taux maximal est $f(1{,}75) \approx 0{,}70$, soit 70%.

2a. Sur $[1{,}75; 20]$, f est continue, strictement décroissante, et $0{,}035 \in [0{,}0307; 0{,}70]$.

Donc d'après le corollaire du TVI, l'équation $f(t) = 0{,}035$ admet une unique solution T.

2b. L'algorithme recherche cette solution T à 0,1 près. Il renvoie 15,7 : le temps en minutes au bout duquel le taux de CO2 retrouve une valeur inférieure à 3,5%.

Exercice 6

1. Initialisation : on a $u_0 = 5$ et $u_1 = \sqrt{5+1} = \sqrt{6}$. Ainsi, on a bien $1 \leq u_1 \leq u_0$.

Hérédité : soit n entier naturel, et supposons que $1 \leq u_{n+1} \leq u_n$.

Alors $2 \leq u_{n+1} + 1 \leq u_n + 1$ et ensuite $\sqrt{2} \leq \sqrt{u_{n+1} + 1} \leq \sqrt{u_n + 1}$ car la fonction racine carrée est croissante.

Or $\sqrt{2} > 1$, ainsi on retrouve bien $1 \leq u_{n+2} \leq u_{n+1}$.

2. (u_n) est décroissante et minorée par 1, donc elle converge.

3. La fonction $f(x) = \sqrt{x+1}$ étant continue, la limite de (u_n) est solution de l'équation $f(x) = x$.

Or $\sqrt{x+1} = x \Leftrightarrow x + 1 = x^2 \Leftrightarrow x^2 - x - 1 = 0$.

Ce polynôme du second degré admet deux solutions, mais l'une d'entre elles est négative alors que la suite est minorée par 1. L'autre limite est le nombre ℓ donné dans l'énoncé *(c'est le nombre d'or !!! Surtout ne faites pas d'exposé au grand oral sur le nombre d'or, les examinateurs en ont déjà vu des dizaines, souvent copiés-collés).*

4. C'est plus complexe ici : l'algorithme va calculer tous les termes de la suite (u_n) jusqu'à ce que la distance entre u_n et la limite ℓ soit inférieure à 10^{-n}.

a. Ainsi, `seuil(2)` renvoie 5, car u_5 est à une distance de ℓ inférieure à 10^{-2} ($u_5 \approx 1{,}624$ alors que $\ell \approx 1{,}618$)

b. De même, `seuil(4)` renvoie 9 car u_9 est le premier terme de (u_n) qui est à une distance de ℓ inférieure à 10^{-4}.

Exercice 7

a. $a_2 = a_1 + a_0 = 0 + 1 = 1$

$a_3 = a_2 + a_1 = 1 + 1 = 2$

$a_4 = a_3 + a_2 = 2 + 1 = 3$

$a_5 = a_4 + a_3 = 3 + 2 = 5$

$a_6 = a_5 + a_4 = 5 + 3 = 8$

b. *Il faut pour cela comprendre qu'à chaque étape dans la boucle, b doit être remplacé par a+b, et a doit être remplacé par b. Mais si on écrivait b = a+b dès le départ, on perdrait la valeur initiale de b et on ne pourrait plus faire a = b. C'est l'intérêt de la variable c.*

```
def fibonacci(n):
    a = 0
    b = 1
    for i in range(n):
        c = a + b
        a = b
        b = c
    return a
```

Exercice 8 **1.** $u_0 = 1$; $u_1 = 1 + \frac{3}{4} = 1{,}75$ et $u_2 = 1 + \frac{3}{4} + \left(\frac{3}{4}\right)^2 = 2{,}3125$.

2. A priori tout est bon : on répète $n + 1$ fois (pour aller de 0 à n) l'ajout de $\left(\frac{3}{4}\right)^n$ ... sauf qu'ici c'est toujours le même n, alors que nous voulons ajouter $\left(\frac{3}{4}\right)^0$, puis $\left(\frac{3}{4}\right)^1$, puis $\left(\frac{3}{4}\right)^2$, etc.

Le programme est donc faux : la bonne ligne dans la boucle for est `S = S + (3/4)**k`

3. (u_n) est en fait la somme des termes d'une suite géométrique, de premier terme 1 et de raison $\frac{3}{4}$.

D'après une propriété du cours dont vous vous rappelez sans nul doute, la somme des termes de u_0 à u_n d'une suite géométrique est donnée par la formule :

$$u_0 \frac{1 - q^{n+1}}{1 - q}$$

où q est la raison de la suite. Ainsi, pour tout n entier naturel :

$$u_n = 1 \times \frac{1 - \left(\frac{3}{4}\right)^{n+1}}{1 - \frac{3}{4}} = \frac{1 - \left(\frac{3}{4}\right)^{n+1}}{\frac{1}{4}} = 4 \times \left(1 - \left(\frac{3}{4}\right)^{n+1}\right)$$

Or $\lim\limits_{n\to+\infty} \left(\frac{3}{4}\right)^{n+1} = 0$ car c'est la limite d'une suite géométrique avec $-1 < q < 1$. Ainsi $\lim\limits_{n\to+\infty} u_n = 4$.

Exercice 9

1. f est continue, strictement croissante, $f(a) = -1$, $f(b) = 1$ et $0 \in [-1; 1]$.

D'après le corollaire du TVI, l'équation $f(x) = 0$ admet une unique solution α sur l'intervalle $[a; b]$.

2. Il s'agit d'un algorithme de recherche du nombre α par dichotomie. Le principe est le suivant :

• on part de l'intervalle $[a; b]$

• on pose $m = \frac{a+b}{2}$ le centre de l'intervalle $[a; b]$. On calcule $f(m)$.

• si $f(m)$ est négatif, c'est que la solution recherchée est « à droite de m » : dans l'intervalle $[m; b]$.

On répète donc de nouveau l'algorithme en remplaçant a par m.

• sinon, si $f(m)$ est positif, c'est que la solution recherchée est « à gauche de m » : dans l'intervalle $[a; m]$.

On répète donc de nouveau l'algorithme en remplaçant b par m.

• On continue de répéter ainsi jusqu'à ce que la distance entre les deux bornes soit inférieure ou égale à 0,001.

Le nombre m est alors une valeur approchée à 0,001 près du nombre α.

Le seul algorithme qui fait ça correctement est l'algorithme d.