Suites numériques

PREMIÈRE · SPÉCIALITÉ MATHÉMATIQUES · CORRIGÉS

Les rappels de méthode sont en italique. Les corrigés sont rédigés comme une copie : à l'épreuve anticipée, la qualité de la rédaction compte pour deux points sur vingt.

Exercice 1

1. *On remplace n par sa valeur.* $u_0 = 0$, $u_1 = 2 - 1 = 1$ et $u_5 = 2 \times 25 - 5 = 45$.
2. $v_0 = \dfrac{-1}{1} = -1$, $v_1 = \dfrac{0}{2} = 0$, $v_3 = \dfrac{2}{4} = \dfrac{1}{2}$ et $v_9 = \dfrac{8}{10} = \dfrac{4}{5}$.
3. $w_0 = 5 \times 1 - 2 = 3$, $w_2 = 5 \times 9 - 2 = 43$ et $w_4 = 5 \times 81 - 2 = 403$.
4. $t_0 = 1 + 0 = 1$, $t_3 = -8 + 3 = -5$ et $t_4 = 16 + 4 = 20$.
5. a. La suite commence au rang 1, donc le cinquième terme est $a_5 = 3 \times 5 - 2 = 13$.
   b. On résout $3n - 2 = 40$, ce qui donne $3n = 42$, soit $n = 14$. Le terme égal à 40 est celui de rang 14.

Exercice 2

Partie A

1. Le prix est formé de 120 € de frais fixes et de $0{,}35$ € par affiche, donc $u_n = 120 + 0{,}35\,n$ pour tout entier naturel n.
2. $u_{100} = 120 + 35 = 155$ et $u_{1\,000} = 120 + 350 = 470$. Cent affiches coûtent 155 €, et mille affiches coûtent 470 €.
3. On résout $120 + 0{,}35\,n = 225$, soit $0{,}35\,n = 105$, donc $n = \dfrac{105}{0{,}35} = 300$. On obtient 300 affiches pour 225 €.

Partie B

On remplace n partout, avec des parenthèses.

1. $a_{n+1} = (n+1)^2 - 2(n+1) = n^2 + 2n + 1 - 2n - 2 = n^2 - 1$.
2. $a_n + 1 = n^2 - 2n + 1$. Ce n'est pas a_{n+1}.
3. $a_{2n} = (2n)^2 - 2 \times 2n = 4n^2 - 4n$.
4. $a_{n-1} = (n-1)^2 - 2(n-1) = n^2 - 2n + 1 - 2n + 2 = n^2 - 4n + 3$.

Exercice 3

Partie A

1. $a_1 = 3 \times 5 - 4 = 11$, $a_2 = 3 \times 11 - 4 = 29$ et $a_3 = 3 \times 29 - 4 = 83$.
2. $3^1 + 2 = 5$, $3^2 + 2 = 11$, $3^3 + 2 = 29$ et $3^4 + 2 = 83$. La formule donne bien les quatre termes.
3. $3\left(3^{n+1} + 2\right) - 4 = 3^{n+2} + 6 - 4 = 3^{n+2} + 2$. On vérifie ainsi que, si la formule est vraie au rang n, alors elle est vraie au rang $n+1$: elle respecte la relation.
4. $a_6 = 3^7 + 2 = 2\,187 + 2 = 2\,189$.
5. Avec $a_0 = 2$, on trouve $a_1 = 3 \times 2 - 4 = 2$, et ainsi de suite : la suite est constante, égale à 2.

Partie B

1. $b_1 = \dfrac{1}{1+1} = \dfrac{1}{2}$, puis $b_2 = \dfrac{\frac{1}{2}}{\frac{1}{2}+1} = \dfrac{\frac{1}{2}}{\frac{3}{2}} = \dfrac{1}{3}$, et de même $b_3 = \dfrac{\frac{1}{3}}{\frac{4}{3}} = \dfrac{1}{4}$.
2. On conjecture que $b_n = \dfrac{1}{n+1}$. Dans la relation :

$$\frac{\frac{1}{n+1}}{\frac{1}{n+1}+1} = \frac{\frac{1}{n+1}}{\frac{n+2}{n+1}} = \frac{1}{n+2}.$$

C'est bien la formule au rang $n+1$.

Exercice 4

Une solution de BarroTech

1. a. *On écrit ce qui se passe en une étape.* Utiliser la moitié du stock, c'est multiplier le stock par $0{,}5$. On ajoute ensuite les 30 kg livrés. Donc $p_{n+1} = 0{,}5\,p_n + 30$, avec $p_0 = 200$.
   b. $p_1 = 100 + 30 = 130$, $p_2 = 65 + 30 = 95$ et $p_3 = 47{,}5 + 30 = 77{,}5$.
2. a. Perdre $12\,\%$ des membres, c'est multiplier par $0{,}88$, donc $u_{n+1} = 0{,}88\,u_n + 50$.
   b. $u_{n+1} = 2u_n - 3$.
   c. $u_{n+1} = u_n^2$.
3. Chaque année, le solde du compte augmente de $5\,\%$, puis on retire 20 €.

Exercice 5

1. En comptant sur les figures, $b_1 = 8$, $b_2 = 10$ et $b_3 = 12$.
2. Une dalle bleue de plus allonge la frise d'une colonne : on ajoute une dalle blanche en haut et une en bas. Donc $b_{n+1} = b_n + 2$.
3. Les rangées du haut et du bas comptent chacune $n+2$ dalles blanches. Il reste une dalle blanche à chaque bout de la rangée du milieu. Donc $b_n = 2(n+2) + 2 = 2n + 6$.
4. $b_{50} = 2 \times 50 + 6 = 106$.
5. On cherche le plus grand entier n tel que $2n + 6 \leq 200$, soit $n \leq 97$. La plus grande frise compte 97 dalles bleues, et elle utilise 200 dalles blanches.

Exercice 6

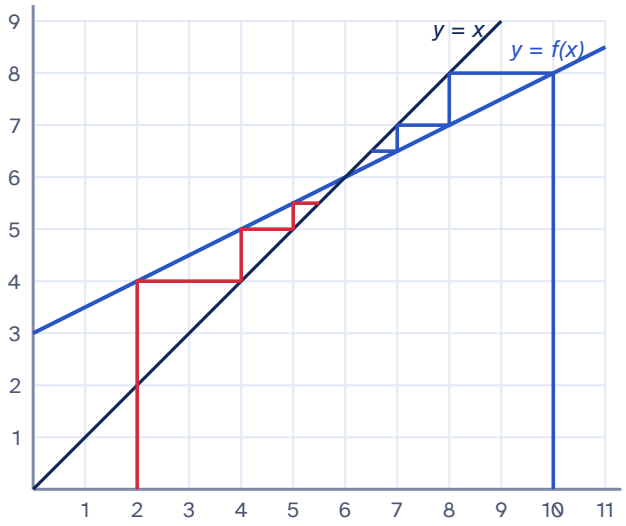
1. Un triangle n'a pas de diagonale, donc $d_3 = 0$. On compte $d_4 = 2$, $d_5 = 5$ et $d_6 = 9$.
2. Un sommet est relié à tous les autres, sauf à lui-même et à ses deux voisins : il est l'extrémité de $n - 3$ diagonales. Cela fait $n(n-3)$ extrémités, mais chaque diagonale est comptée deux fois, une par extrémité. Donc $d_n = \dfrac{n(n-3)}{2}$.
3. $d_{10} = \dfrac{10 \times 7}{2} = 35$ et $d_{20} = \dfrac{20 \times 17}{2} = 170$.
4. $d_{n+1} - d_n = \dfrac{(n+1)(n-2) - n(n-3)}{2} = \dfrac{n^2 - n - 2 - n^2 + 3n}{2} = \dfrac{2n-2}{2} = n - 1$.
5. Pour $n = 6$, k prend les valeurs 3, 4 et 5. La variable d prend les valeurs 2, 5, puis 9. L'algorithme renvoie 9, c'est-à-dire d_6 : il calcule d_n, en ajoutant $k - 1$ à chaque tour comme le dit la question 4.

Exercice 7

1. On lit $u_0 = -1$, $u_3 = 8$ et $u_5 = 4$.
2. La droite d'équation $y = 7$ passe par les points d'abscisses 2 et 4 : $u_n = 7$ pour $n = 2$ et $n = 4$.
3. Les termes augmentent jusqu'au rang 3, où ils atteignent 8, puis ils diminuent.
4. $u_{10} = 8 - 7^2 = 8 - 49 = -41$.
5. La suite n'est définie que pour les entiers naturels : il n'y a aucun terme entre deux rangs consécutifs.

Une solution de Barrotech

Exercice 8



1. $u_1 = f(2) = 4$, $u_2 = f(4) = 5$ et $u_3 = f(5) = 5{,}5$. Sur la figure, c'est l'escalier rouge.
2. $v_1 = f(10) = 8$, $v_2 = f(8) = 7$ et $v_3 = f(7) = 6{,}5$. Sur la figure, c'est l'escalier bleu.
3. La suite (u_n) semble croissante, et la suite (v_n) semble décroissante. Leurs termes semblent se rapprocher de 6.
4. $0{,}5x + 3 = x$ équivaut à $3 = 0{,}5x$, soit $x = 6$. Le nombre 6 est l'abscisse du point d'intersection des deux droites : c'est vers lui que semblent aller les deux escaliers.

Une solution de BarroTech

Exercice 9

1. $a_{n+1} - a_n = -2 < 0$: la suite est strictement décroissante.
2. $b_{n+1} - b_n = (n+1)^2 - 5(n+1) - n^2 + 5n = 2n - 4$. Cette différence est négative pour $n < 2$, nulle pour $n = 2$ et positive pour $n > 2$. La suite est croissante à partir du rang 2. Ses premiers termes sont 0, -4, -6, -6, -4.
3. $c_{n+1} - c_n = \dfrac{n+1}{n+2} - \dfrac{n}{n+1} = \dfrac{(n+1)^2 - n(n+2)}{(n+1)(n+2)} = \dfrac{1}{(n+1)(n+2)} > 0$. La suite est strictement croissante.
4. Tous les termes sont strictement positifs, et
$$\frac{d_{n+1}}{d_n} = \frac{3^{n+1}}{n+2} \times \frac{n+1}{3^n} = \frac{3(n+1)}{n+2}.$$
Pour tout entier naturel n, $3n + 3 > n + 2$, donc ce quotient est strictement supérieur à 1. La suite est strictement croissante.
5. $e_{n+1} - e_n = -n^2 - 1 < 0$ pour tout entier naturel n. La suite est strictement décroissante.

Exercice 10

1. **Faux.** La fonction $f(x) = \frac{1}{2}x + 2$ est croissante, mais avec $u_0 = 7$, on a $u_1 = 5{,}5 < u_0$. La suite n'est pas croissante.
2. **Faux.** La suite $(-1)^n$ n'est pas croissante, et elle n'est pas décroissante non plus.
3. **Faux.** Avec $f(x) = (x - 0{,}4)^2$, la fonction f est décroissante puis croissante sur $[0\,;+\infty[$, donc elle n'est pas monotone. Pourtant, $u_{n+1} - u_n = (n + 0{,}6)^2 - (n - 0{,}4)^2 = 2n + 0{,}2 > 0$. La suite est strictement croissante.
4. **Vrai.** $v_0 = 0$ et $v_1 = 1 - 7 = -6$, donc $v_1 < v_0$. Ce contre-exemple suffit.
5. **Faux.** La négation est : « il existe un entier naturel n tel que $u_n \leq 0$ ».

Exercice 11

1. `[1, 3, 5, 7, 9]`
2. `[1, 4, 9, 16]`. *`range(1, 5)` donne les entiers de 1 à 4.*
3. `[1, 5, 9, 13]`. *On garde les nombres dont le reste de la division par 4 vaut 1.*
4. `[10, 20]`. *On garde les multiples de 2 et de 5, c'est-à-dire les multiples de 10.*
5. `[2, 5, 8, 11]`. *À chaque tour, on ajoute le terme d'indice k, augmenté de 3.*
6. `len(L)` vaut 3 et `L[2]` vaut 5. La liste n'a pas d'indice 3 : `L[3]` provoque une erreur (`IndexError`).

Une solution de BarroTech

Exercice 12

1. `[3 * n - 1 for n in range(8)]` donne `[-1, 2, 5, 8, 11, 14, 17, 20]`.
2. `[n for n in range(50) if n % 7 == 0]` donne `[0, 7, 14, 21, 28, 35, 42, 49]`, soit 8 éléments.
3. Le programme suivant donne `[1, 5, 17, 53, 161, 485]`.

```
L = [1]
for k in range(5):
    L.append(3 * L[k] + 2)
```

4. `[n for n in range(1, 31) if n % 4 == 0 or n % 6 == 0]` donne `[4, 6, 8, 12, 16, 18, 20, 24, 28, 30]`.
5. Avec `and`, on obtient `[12, 24]`. Ce sont les multiples communs de 4 et de 6, c'est-à-dire les multiples de 12.

Exercice 13

1. Après chaque ligne, `T` contient : `[52, 48, 50, 47]`, puis `[52, 48, 49, 47]`, puis `[52, 48, 49, 47, 46]`, puis `[48, 49, 47, 46]`, et enfin `[48, 49, 46]`.
2. `T[1]` vaut 49 et `len(T)` vaut 3.
3. Le programme affiche trois lignes : `1 48`, `2 49` et `3 46`. Il numérote les temps à partir de 1, alors que les indices commencent à 0.

Exercice 14

1. a. Les éléments strictement supérieurs à 5 sont 12, 7 et 25 : la fonction renvoie 3.
   b. Elle compte les éléments de `L` strictement supérieurs à `s`.
2. On adapte la fonction `maximum` de la leçon.

```
def minimum(L):
    m = L[0]
    for x in L:
        if x < m:
            m = x
    return m
```

3. On cherche un élément suivi d'un plus grand : un seul contre-exemple suffit.

Une solution de BarroTech

```
def decroissante(L):
    for i in range(len(L) - 1):
        if L[i + 1] > L[i]:
            return False
    return True
```

4. Le dernier élément n'a pas de suivant. Avec `range(len(L))`, la boucle demanderait `L[len(L)]`, qui n'existe pas, et le programme s'arrêterait sur une erreur.
5. On garde en mémoire l'indice, et non la valeur. Une boucle `for x in L` donne les valeurs, sans leur place.

```
def indice_max(L):
    r = 0
    for i in range(len(L)):
        if L[i] > L[r]:
            r = i
    return r
```

Exercice 15

1. 5, 16, 8, 4, 2, 1 : il faut 5 étapes.
2. 12, 6, 3, 10, 5, 16, 8, 4, 2, 1 : il faut 9 étapes, et le plus grand terme est 16.
3. a. `etapes(5)` renvoie 5 et `etapes(7)` renvoie 16.
   b. La boucle s'arrête seulement si la suite arrive à 1. Or personne n'a démontré que c'est le cas pour tout entier $a > 0$: c'est une conjecture. Une vérification sur de nombreux exemples ne prouve pas que la boucle s'arrête toujours.

Exercice 16

1. Pour une marche, la seule façon est 1. Pour deux marches, ce sont $1+1$ et 2. Pour trois marches, ce sont $1+1+1$, $1+2$ et $2+1$. Donc $e_1 = 1$, $e_2 = 2$ et $e_3 = 3$.
2. Si le premier pas est d'une marche, il reste $n+1$ marches, soit e_{n+1} façons de finir. Si le premier pas est de deux marches, il reste n marches, soit e_n façons. Ces deux cas sont distincts et couvrent toutes les possibilités. Donc $e_{n+2} = e_{n+1} + e_n$.
3. $e_4 = 3+2 = 5$ et $e_5 = 5+3 = 8$. Ensuite, on trouve 13, 21, 34, 55, puis $e_{10} = 89$.
4. Seuls les deux premiers termes changent.

```
def escalier(n):
    L = [1, 2]
    for k in range(n - 2):
        L.append(L[k] + L[k + 1])
    return L
```

Une solution de BarroTech

Exercice 17

1. $a_0 = 4$, $a_1 = 3{,}5$ et $a_{99} = 3{,}01$. La suite semble tendre vers 3.
2. $b_0 = -7$, $b_{10} = 13$ et $b_{100} = 193$. La suite semble tendre vers $+\infty$.
3. $c_0 = 10$, $c_{10} = -90$ et $c_{100} = -9\,990$. La suite semble tendre vers $-\infty$.
4. d_n vaut 3 si n est pair, et -1 si n est impair. La suite oscille entre deux valeurs : elle n'a pas de limite.
5. $e_0 = 6$, $e_1 = 4{,}5$, $e_{99} = \dfrac{499}{100}$ et $e_{100} = \dfrac{506}{101}$. La suite semble tendre vers 5.
6. $f_1 = \dfrac{1}{2}$, $f_2 = 2$, $f_3 = \dfrac{1}{2}$. La suite oscille entre deux valeurs : elle n'a pas de limite.

La suite (e_n) n'est pas monotone : $e_1 < e_0$, mais $e_2 = \dfrac{16}{3} > e_1$. Elle semble pourtant avoir une limite : une suite peut avoir une limite sans être monotone.

Exercice 18

Partie A

1. Perdre $20\,\%$, c'est multiplier par $0{,}8$. On ajoute ensuite 2 grammes. Donc $u_{n+1} = 0{,}8\,u_n + 2$.
2. $u_1 = 2$, $u_2 = 0{,}8 \times 2 + 2 = 3{,}6$ et $u_3 = 0{,}8 \times 3{,}6 + 2 = 4{,}88$.
3. Les valeurs $8{,}93$; $9{,}88$; $9{,}99$ font conjecturer que la suite tend vers 10. Quand il y a 10 grammes de chlore, $20\,\%$ de cette masse représente 2 grammes : il en disparaît alors autant qu'on en ajoute.

Partie B

On peut conjecturer que le quotient de deux termes consécutifs de la suite de Fibonacci tend vers un nombre proche de $1{,}618$. Ce n'est pas une démonstration : on a seulement observé un nombre fini de termes.

Exercice 19

1. $T_1 = 0{,}9 \times 80 + 2 = 74$ et $T_2 = 0{,}9 \times 74 + 2 = 68{,}6$.
2. *La condition de la boucle est le contraire de ce qu'on cherche.*

```
n ← 0
T ← 80
Tant que T ⩾ 40
    T ← 0,9 × T + 2
    n ← n + 1
Fin Tant que
```

Une solution de BarroTech

3. La fonction suivante traduit cet algorithme.

```
def seuil(s):
    n = 0
    T = 80
    while T >= s:
        T = 0.9 * T + 2
        n = n + 1
    return n
```

4. `seuil(40)` renvoie 11 : d'après le tableau, $T_{10} \approx 40{,}92 \geq 40$ et $T_{11} \approx 38{,}83 < 40$. La soupe passe sous 40 degrés à la onzième minute.
5. La température reste au-dessus de 20 degrés, donc elle ne passe jamais sous 15 degrés. La condition reste toujours vraie, et le programme ne s'arrête pas.

Exercice 20

1. On cherche $f_n > s$: la boucle tourne tant que `f <= s`.
2. Le suivi de la boucle pour $s = 1\,000$ est le suivant.

n	f	**Test** $f \leq 1\,000$
0	500	vrai
1	700	vrai
2	940	vrai
3	$1\,228$	faux

La boucle s'arrête, et la fonction renvoie 3.

3. On contrôle avec f_9 et f_{10} : on doit avoir $f_9 \leq 5\,000 < f_{10}$. En effet, $f_9 \approx 4\,659{,}78$ et $f_{10} \approx 5\,691{,}74$.
4. Avec `while f > s:`, le test est faux dès le départ, car $500 > 1\,000$ est faux. La boucle ne tourne pas, et la fonction renvoie 0. L'élève a écrit ce qu'il cherche, au lieu d'écrire le contraire.

Une solution de BarroTech