Plaquette 4 – Seuils, algorithmes et tableur

Suites numériques et modèles — Terminale STMG

Corrections détaillées

Boîte à outils

Algorithme de seuil (boucle « tant que ») : on répète le passage au terme suivant tant que le seuil n'est pas franchi, en comptant les étapes.

```
u = 2000
n = 0
while u < 4000:
    u = u * 1.06
    n = n + 1
```

Accumulateur (boucle « pour ») : une variable initialisée à 0 reçoit un terme à chaque tour ; à la fin, elle contient $\sum u_k$.

Tableur. Formule de récurrence recopiée vers le bas (B3 : *=B2*1,06*) ou formule directe à partir du rang (*=2000*1,06^A3*). Une référence absolue (symbole dollar devant la ligne et la colonne) fige une cellule lors de la recopie.

Contrôle par une formule. Un seuil se vérifie par le calcul des termes de part et d'autre ; un cumul par la formule de la somme :

$$\sum_{k=0}^{n-1} u_0 q^k = u_0 \times \frac{q^n - 1}{q - 1}$$

Correction de l'exercice 1 – Compléter un algorithme de seuil

On répète « une année de plus » tant que le capital est inférieur à 4000 €, et chaque année multiplie le capital par 1,06 :

```
def seuil():
    u = 2000
    n = 0
    while u < 4000:
        u = u * 1.06
        n = n + 1
    return n
```

La fonction renvoie 12. Contrôle : $2000 \times 1{,}06^{11} \approx 3796{,}60$ et

$$2000 \times 1{,}06^{12} \approx 4024{,}39$$

Réponse. *while u < 4000* et *u = u * 1.06* ; la fonction renvoie 12.

Correction de l'exercice 2 – Un seuil vers le bas

Le stock diminue : on répète tant qu'il est encore supérieur ou égal à 1000, en le multipliant par 0,85 chaque mois.

```
def seuil():
    s = 5000
    n = 0
    while s >= 1000:
        s = s * 0.85
```

mathlvl.fr — Tous droits réservés. Toute reproduction, diffusion, revente ou exploitation commerciale, même partielle, est strictement interdite.
Tout contrevenant s'expose à des poursuites en vertu du Code de la propriété intellectuelle (art. L.335-2 et suivants).

```
        n = n + 1
    return n
```

La fonction renvoie 10 : $s_9 \approx 1158$ et $s_{10} \approx 984$. Avec le logarithme : $n > \dfrac{\log(0{,}2)}{\log(0{,}85)} \approx 9{,}9$.

Réponse. *while s >= 1000* avec *s = s * 0.85* ; la fonction renvoie 10.

Correction de l'exercice 3 – Trace d'exécution

On multiplie chaque valeur par 1,15 :

n	0	1	2	3	4	5
u_n	100	115	132,25	152,09	174,90	201,14

On garde toutes les décimales dans les calculs et l'on n'arrondit qu'à l'affichage : sinon les erreurs d'arrondi s'accumulent d'une ligne à l'autre. Contrôle direct :

$$u_5 = 100 \times 1{,}15^5 \approx 201{,}14$$

Le nombre d'abonnés a doublé en cinq périodes.

Réponse. 100 ; 115 ; 132,25 ; 152,09 ; 174,90 ; 201,14.

Correction de l'exercice 4 – Formules de tableur

a) B2 contient le premier terme : *=1200*.

b) Chaque mois ajoute 150 au précédent : en B3, *=B2+150*, recopié vers le bas (B4 devient *=B3+150*, etc.).

c) La suite est arithmétique, $p_n = 1200 + 150n$: on écrit en B2 la formule *=150*A2+1200*, recopiée vers le bas.

Les deux méthodes donnent la même colonne. La formule directe a l'avantage de calculer n'importe quel rang sans les précédents.

Réponse. a) *=1200* — b) *=B2+150* — c) *=150*A2+1200*.

Correction de l'exercice 5 – Tableur et modèle géométrique

a) Chaque année multiplie le capital par 1,06 : *=B2*1,06*.

b) Avec $u_n = 2000 \times 1{,}06^n$: *=2000*1,06^A3*.

c) On écrit *=B2*(1+D1)* en faisant de D1 une **référence absolue** (symbole dollar devant la colonne et la ligne), pour que la recopie vers le bas ne la décale pas : la formule reste $u_{n+1} = u_n \times (1 + t)$ avec t lu en D1.

Il suffit alors de changer D1 pour recalculer toute la colonne avec un autre taux.

Réponse. a) *=B2*1,06* — b) *=2000*1,06^A3* — c) *=B2*(1+D1)* avec D1 en référence absolue.

Correction de l'exercice 6 – Que renvoie cet algorithme ?

On suit l'exécution pas à pas :

n	0	1	2	3	4	5
u	500	560	620	680	740	800

Après le cinquième passage, $u = 800$ et la condition $u < 800$ est fausse : la boucle s'arrête et la fonction renvoie 5.

mathlvl.fr — Tous droits réservés. Toute reproduction, diffusion, revente ou exploitation commerciale, même partielle, est strictement interdite.
Tout contrevenant s'expose à des poursuites en vertu du Code de la propriété intellectuelle (art. L.335-2 et suivants).

On peut le confirmer par le calcul : $500 + 60n \geq 800 \Leftrightarrow n \geq 5$.

Réponse. 5.

Correction de l'exercice 7 – Modifier un algorithme

Seule la dernière ligne change. Pour le capital, on écrit *return u* : la fonction renvoie environ 4024,39, le premier capital qui atteint 4000 €.

Pour les deux, on écrit *return n, u* : la fonction renvoie le couple

$$(12\,;\,4024{,}39)$$

C'est utile pour vérifier qu'on a bien franchi le seuil, et de combien.

Réponse. *return u* donne ≈ 4024,39 ; *return n, u* donne le couple.

Correction de l'exercice 8 – Un cumul par boucle

On utilise un accumulateur *total* et une boucle « pour » sur les rangs 0 à $n-1$:

```
def cumul(n):
    total = 0
    for k in range(n):
        total = total + 1200 + 150 * k
    return total
```

cumul(20) renvoie 52 500. Contrôle par la somme arithmétique : le dernier mois produit $1200 + 150 \times 19 = 4050$ unités, et

$$\sum_{k=0}^{19} (1200 + 150k) = 20 \times \frac{1200 + 4050}{2} = 52\,500$$

Réponse. Le cumul vaut 52 500 unités sur 20 mois.

Correction de l'exercice 9 – Un seuil sur un cumul

On combine accumulateur et boucle « tant que » :

```
def seuil_cumul():
    total = 0
    n = 0
    while total <= 50000:
        total = total + 1200 + 150 * n
        n = n + 1
    return n
```

La fonction renvoie 20 : après 19 mois, le cumul vaut $52\,500 - 4050 = 48\,450$ unités, sous le seuil ; après 20 mois, il vaut 52 500.

$$48\,450 \leq 50\,000 < 52\,500$$

Réponse. 20 mois, pour un cumul de 52 500 unités.

Correction de l'exercice 10 – Corriger un algorithme

L'ordre des deux lignes de la boucle n'a pas d'importance ici : n et u avancent ensemble. L'erreur est dans la dernière ligne, qui renvoie le capital : la fonction renvoie environ 4024,39.

Il faut écrire *return n* : la fonction renvoie alors 12, le rang cherché.

$$u_{12} = 2000 \times 1{,}06^{12} \approx 4024{,}39$$

mathlvl.fr — Tous droits réservés. Toute reproduction, diffusion, revente ou exploitation commerciale, même partielle, est strictement interdite.
Tout contrevenant s'expose à des poursuites en vertu du Code de la propriété intellectuelle (art. L.335-2 et suivants).

Réponse. Elle renvoie ≈ 4024,39 ; avec *return n*, elle renvoie 12.

Correction de l’exercice 11 – Un cumul au tableur

a) B2 : *=400* ; B3 : *=B2*1,05*. C2 : *=B2* ; C3 : *=C2+B3*, recopiées vers le bas : chaque cellule de C ajoute la vente du mois au cumul précédent.

b) C13 contient le cumul des 12 mois, somme de 12 termes d’une suite géométrique de raison 1,05 :

$$\sum_{k=0}^{11} 400 \times 1{,}05^{k} = 400 \times \frac{1{,}05^{12}-1}{0{,}05} \approx 6367$$

La boutique vend environ 6367 articles sur l’année.

Réponse. a) *=400*, *=B2*1,05*, *=B2*, *=C2+B3* — b) ≈ 6367 articles.

Correction de l’exercice 12 – Une épargne annuelle

a) Chaque année, le capital rapporte 3 %, puis reçoit un versement :

```
def epargne():
    u = 0
    n = 0
    while u <= 100000:
        u = 1.03 * u + 15000
        n = n + 1
    return n
```

La fonction renvoie 7.

b) La formule vient d’une somme géométrique (les versements de 15 000, 15 000 × 1,03, …). Pour $n = 6$, $u_6 \approx 97\,026{,}15$ € et pour $n = 7$:

$$u_7 = 15\,000 \times \frac{1{,}03^{7}-1}{0{,}03} \approx 114\,936{,}93$$

Réponse. 7 ans, pour ≈ 114 937 € accumulés.

Correction de l’exercice 13 – Un parc à renouveler

Le parc est multiplié par 0,88 chaque année :

```
def parc():
    p = 12000
    n = 0
    while p >= 6000:
        p = p * 0.88
        n = n + 1
    return n
```

La fonction renvoie 6. Contrôle :

$$12\,000 \times 0{,}88^{5} \approx 6332 \qquad 12\,000 \times 0{,}88^{6} \approx 5573$$

Au bout de 6 ans, il reste moins de la moitié du parc.

Réponse. 6 ans, le parc tombant à ≈ 5573 machines.

Correction de l’exercice 14 – Repérer une erreur d’algorithme

Dans la boucle, seul n change : u reste égal à 5000, la condition $u > 1000$ reste vraie pour toujours, et le programme ne s’arrête jamais. C’est une **boucle infinie**.

mathlvl.fr — Tous droits réservés. Toute reproduction, diffusion, revente ou exploitation commerciale, même partielle, est strictement interdite.
Tout contrevenant s’expose à des poursuites en vertu du Code de la propriété intellectuelle (art. L.335-2 et suivants).

Il manque la ligne qui fait évoluer la suite :

```
def seuil():
    u = 5000
    n = 0
    while u > 1000:
        u = u * 0.85
        n = n + 1
    return n
```

Corrigée, la fonction renvoie 10, car $5000 \times 0{,}85^{10} \approx 984$.

Réponse. Boucle infinie : il manque *u = u * 0.85* ; corrigée, elle renvoie 10.

Correction de l’exercice 15 – Synthèse : un plan d’épargne progressif

a) Les versements forment une suite géométrique : $v_k = 150 \times 1{,}02^k$.

b) Accumulateur et boucle « tant que » :

```
def epargne():
    total = 0
    n = 0
    while total <= 5000:
        total = total + 150 * 1.02 ** n
        n = n + 1
    return n
```

c) Après n mois, le total vaut $150 \times \dfrac{1{,}02^n - 1}{0{,}02}$. Il dépasse 5000 si $1{,}02^n > 1 + \dfrac{5000 \times 0{,}02}{150} \approx 1{,}667$, soit

$$n > \frac{\log(1{,}667)}{\log(1{,}02)} \approx 25{,}8$$

La fonction renvoie 26 : environ 4805 € après 25 mois, 5051 € après 26.

Réponse. a) $v_k = 150 \times 1{,}02^k$ — b) et c) 26 mois.

mathlvl.fr — Tous droits réservés. Toute reproduction, diffusion, revente ou exploitation commerciale, même partielle, est strictement interdite.
Tout contrevenant s’expose à des poursuites en vertu du Code de la propriété intellectuelle (art. L.335-2 et suivants).