De la géométrie, des séries et du Python

1 On se lance

1. Je vous laisse faire...
Il semblerait sur vos dessin que C_1 appartienne à la droite (B_2C_2),ainsi que $C_2 \in (B_3C_3)$. Vous avez creusé ? A posteriori, j'ai trouvé une jolie preuve. Ceux qui l'auront cherchée pourront venir demander !
2. Dans le triangle (OA_nB_n) rectangle en A_n, Pythagore nous donne : $a_{n+1}^2 = OA_{n+1}^2 = OB_n^2 = OA_n^2 + A_nB_n^2$ et comme l'aire $OA_n \times A_nB_n$ est constante égale à $\mathcal{A} = ab$ on a $A_nB_n = \dfrac{\mathcal{A}}{a_n}\cdot$
$$\boxed{\text{Pour tout } n \in \mathbb{N}^*,\ a_{n+1} = \sqrt{a_n^2 + \frac{\mathcal{A}^2}{a_n^2}}\cdot}$$
Bien entendu, et comme vous aviez plissé les yeux quand j'avais annoncé qu'une formule de récurrence ne se montrait presque jamais par récurrence, je ne vais lire aucune récurrence. BIEN ENTENDU...
3. La tangente de l'angle recherché vaut $\dfrac{A_nB_n}{OA_n} = \dfrac{\mathcal{A}}{a_n^2}$, et comme on a choisi pour θ_n la mesure dans $[0, \pi/2[$, elle vaut l'arctangente de sa tangente !
$$\boxed{\theta_n = \operatorname{Arctan}\frac{\mathcal{A}}{a_n^2}\cdot}$$
4. Les rectangles semblent devenir de plus en plus fin, et les angles θ_n semblent donc décroissants. On peut imaginer qu'ils tendent vers 0 (ce qui n'est pas évident, sauf si on pense que toute suite décroissante de réels positifs tend vers 0...). Avant d'avoir traité la suite du problème, la question de l'angle total $\theta_1 + \cdots + \theta_n$ n'est pas claire : il augmente bien entendu, mais encore ?
Si on croit que toute suite croissante tend vers $+\infty$ (ce qui n'est pas votre cas n'est-ce pas ?) ou encore que toute série à termes positifs est divergente (ce qui est la même chose...) alors on est convaincu que cet angle total tend vers $+\infty$. Mais si on a déjà rencontré dans sa vie des suites croissantes convergentes, ou encore des séries à termes positifs convergentes... alors rien n'est clair ! La suite du problème va confirmer que l'angle total tend vers $+\infty$. Espérons que ça ne confortera pas ceux ayant fait l'erreur de raisonnement signalée plus haut (« ben voila, je savais que mon raisonnement était juste... »).

2 Divergence, équivalent des sommes partielles

1. Puisque $a_{n+1} = a_n \underbrace{\left(1 + \dfrac{\mathcal{A}^2}{a_n^4}\right)^{1/2}}_{>1} > a_n$ la suite $(a_n)_{n\in\mathbb{N}^*}$ est strictement croissante, donc convergente ou bien divergente vers $+\infty$. Dans le premier cas, la limite ℓ vérifierait d'abord (passage d'inégalité large à la limite) $\ell \geqslant a_1 > 0$ et par ailleurs (passage de la relation de récurrence à la limite, les deux membres convergeant) $\ell^2 = \ell^2 + \dfrac{\mathcal{A}^2}{\ell^2}$ donc $\dfrac{\mathcal{A}^2}{\ell^2} = 0$, ce qui n'est pas possible car $\mathcal{A} > 0$. Ainsi (a_n) est croissante et non convergente, donc :
$$\boxed{a_n \underset{n\to+\infty}{\longrightarrow} +\infty}$$

Ensuite, $a_{n+1} = a_n \underbrace{\left(1+\frac{\mathcal{A}^2}{a_n^4}\right)^{1/2}}_{\underset{n\to+\infty}{\longrightarrow} 1}$ donc

$$\boxed{a_{n+1} \sim a_n}$$

2. Tout d'abord, $\ln(a_{n+1}) - \ln(a_n) = \dfrac{1}{2}\ln\left(1+\dfrac{\mathcal{A}^2}{a_n^4}\right)$, mais on a vu que $a_n \underset{n\to+\infty}{\longrightarrow} +\infty$, donc $\dfrac{\mathcal{A}^2}{a_n^4} \underset{n\to+\infty}{\longrightarrow} 0$, ce qui fournit l'équivalent demandé :

$$\boxed{\ln(a_{n+1}) - \ln(a_n) \sim \frac{\mathcal{A}^2}{2a_n^4}\cdot}$$

Ça y est vous êtes en spé, donc vous savez que $\ln(1+u) \sim u$ *sans passer par un développement limité !*

Dans l'équivalent précédent, les termes sont **positifs**, donc les séries associées sont de même nature ; or $\sum(\ln(a_{n+1}) - \ln(a_n))$ diverge (cette **série** est de même nature que la **suite** $(\ln(a_n))_{n\in\mathbb{N}}$ qui tend vers $+\infty$), donc $\sum \frac{\mathcal{A}^2}{2a_n^4}$ également :

$$\boxed{\sum \frac{1}{a_n^4} \text{ diverge.}}$$

Puisque $\theta_n \sim \dfrac{\mathcal{A}}{a_n^2} \geqslant \dfrac{\mathcal{A}}{a_n^4}$ (à partir d'un certain rang $a_n \geqslant 1$), alors par comparaison de séries à termes positifs :

$$\boxed{\sum \theta_n \text{ diverge.}}$$

Puisque les θ_n sont positifs, la divergence de la série signifie que la suite des sommes partielles tend vers $+\infty$. Or $\theta_1 + \cdots + \theta_n$ est l'angle entre $\overrightarrow{OA_1}$ et $\overrightarrow{OA_{n+1}}$: les rectangles « tournent autour de l'origine sans angle limite ».

3. Tout d'abord, $a_{n+1} = a_n\left(1+\dfrac{\mathcal{A}^2}{a_n^4}\right)^{1/2} = a_n + \dfrac{\mathcal{A}^2}{2a_n^3} + o(1/a_n^3)$, donc $a_{n+1} - a_n \sim \dfrac{\mathcal{A}^2}{2a_n^3}$ et enfin :

$$\boxed{(a_{n+1} - a_n)a_n^3 \sim \frac{\mathcal{A}^2}{2}\cdot}$$

Puisque $\sum \frac{\mathcal{A}^2}{2}$ est à termes positifs et est divergente, il en est de même pour $\sum(a_{n+1} - a_n)a_n^3$, avec de plus (d'après le résultat admis dans l'énoncé) :

$$\boxed{\sum_{k=1}^{n}(a_{k+1} - a_k)a_k^3 \sim \sum_{k=1}^{n}\frac{S^2}{2} = \frac{S^2}{2}n.}$$

4. Par croissance de $t \mapsto t^3$ on a pour tout $k \geqslant 1$:

$$(a_{k+1} - a_k)a_k^3 \leqslant \int_{a_k}^{a_{k+1}} t^3 dt.$$

Vous avez bien entendu fait un dessin où on voit un rectangle situé sous une courbe... sans quoi ceci ne sera pas corrigé.

En sommant, on obtient :

$$T_n = \sum_{k=1}^{n}(a_{k+1} - a_k)a_k^3 \leqslant \int_{a_1}^{a_{n+1}} t^3 dt = \frac{a_{n+1}^4 - a_1^4}{4}\cdot \qquad (I_1)$$

Puisque $a_{n+1} \sim a_n \underset{n\to+\infty}{\longrightarrow} +\infty$, le membre de droite de (I_1) est équivalent à $\dfrac{a_n^4}{4}\cdot$ Le même procédé permet de minorer $(a_{k+1} - a_k)a_{k+1}^3$, puis :

$$U_n = \sum_{k=1}^{n}(a_{k+1} - a_k)a_{k+1}^3 \geqslant \int_{a_1}^{a_{n+1}} t^3 dt = \frac{a_{n+1}^4 - a_1^4}{4}\cdot \qquad (I_2)$$

Mais T_n et U_n sont les sommes partielles de deux séries divergentes de termes généraux positifs et équivalents, donc sont eux-même équivalents. Ainsi en reprenant (I_1) et (I_2) :

$$\frac{T_n}{U_n}\frac{a_{n+1}^4 - a_1^4}{4} \leqslant \frac{T_n}{U_n} U_n = T_n \leqslant \frac{a_{n+1}^4 - a_1^4}{4}.$$

Divisant les trois termes par $\dfrac{a_n^4}{4}$, on constate que les termes extérieurs tendent vers 1, et on peut alors appliquer le théorème des gendarmes, qui nous donne :

$$\boxed{T_n = \sum_{k=1}^{n} (a_{k+1} - a_k) a_k^3 \sim \frac{a_n^4}{4}.}$$

Il reste à combiner ce résultat avec celui de la question précédente pour obtenir :

$$\boxed{a_n \sim \sqrt{\mathcal{A}}\sqrt[4]{2n}.}$$

5. Puisque $\sum \theta_n$ est divergente à termes positifs et $\theta_n \sim \dfrac{1}{\sqrt{2n}}$, les sommes partielles sont équivalentes, et il reste à appliquer le résultat du poly de cours (mais qui est hors-programme) concernant la deuxième :

$$\boxed{\sum_{k=1}^{n} \theta_k \sim \frac{1}{\sqrt{2}} \sum_{k=1}^{n} \frac{1}{\sqrt{k}} \sim \sqrt{2n}.}$$

3 Un peu de Python

1. Certains écrivent des boucles `while`. D'où vient cette étrange perversion ? On sait ici qu'on va faire une boucle avec exactement $n-1$ étapes élémentaires (la complexité sera donc de l'ordre de n).

```
from math import sqrt, atan, cos, sin


def a(n, a1, b1):
    aka, S = a1, a1*b1
    for k in range(2, n+1):
        aka = sqrt(aka**2+S**2/aka**2)
    return aka
"""
>>> [a(n, 1, 2) for n in range(1, 10)]
[1, 2.23606797749979, 2.4083189157584592, 2.5474801613386107,
 2.6657120513960044, 2.769282160710955, 2.861906578579664, 2.945993785711146,
 3.0232049848985354]
"""
```

2. Ensuite, peu de suspens pour le calcul de θ_n. Regardez tout de même la façon dont je teste ladite fonction.

```
def theta(n, a1, b1):
    return atan(a1*b1/a(n, a1, b1)**2)
"""
>>> [theta(2**k, 1, 2) for k in range(10)]
[1.1071487177940904, 0.3805063771123648, 0.2989469267775927,
0.22649051208098023, 0.1670989719219272, 0.12113922974457861,
0.08688050412824432, 0.06191948584267785, 0.043972884027206116,
0.031166246368457454]

>>> [theta(2**k, 2, 1) for k in range(10)]
```

```
[0.4636476090008061, 0.3805063771123648, 0.2989469267775927,
0.22649051208098023, 0.1670989719219272, 0.12113922974457861,
0.08688050412824432, 0.06191948584267785, 0.043972884027206116,
0.031166246368457454]
"""


def SP1(n, a1, b1):  # quadratique
    return sum(theta(k, a1, b1) for k in range(1, n+1))
"""
>>> SP1(100, 2, 1)
12.204624421923855
>>> SP1(1000, 2, 1)
42.63630355942422
>>> SP1(10000, 2, 1)
139.28047047131153
"""
```

Chaque calcul de θ_n demande de l'ordre de n opérations élémentaires, donc le calcul de S_n en demandera de l'ordre de $1 + 2 + \cdots + n$ c'est-à-dire de l'ordre de n^2.

3. La fonction `SP1` est de complexité quadratique : en déclarant raisonnable les calculs demandant moins d'un milliard d'opérations élémentaires, le calcul est d'une durée acceptable pour $n = 10^3$, mais plus pour $n = 10^6$.
   Pour passer à une complexité linéaire dans le calcul de S_n, il suffit de ne pas faire appel à la fonction calculant a_n, mais de calculer les arc-tangentes à la volée, pour obtenir une complexité linéaire, donc tout à fait acceptable pour $n = 10^6$:

```
def SP2(n, a1, b1):  # linéaire
    aka, S, SomPart = a1, a1*b1, atan(b1/a1)
    for k in range(2, n+1):
        aka = sqrt(aka**2+S**2/aka**2)
        SomPart += atan(S/aka**2)
    return SomPart, SomPart/sqrt(2*n)
"""
>>> SP2(100, 2, 1)
(12.204624421923855, 0.8629972690577306)
>>> SP2(1000, 2, 1)
(42.63630355942422, 0.953376730681888)
>>> SP2(10000, 2, 1)
(139.28047047131153, 0.9848616515711708)
>>> SP2(10**6, 1, 1)
(1412.82956274673, 0.9990213644790377)
>>> SP2(10**8, 1, 1)
(14140.748747049623, 0.9999019330093964)
"""
```