Planche HEC : étude d'une suite implicite

Exercice avec préparation 1

Pour tous $n \in \mathbb{N}^*$ et $x \in \mathbb{R}$, on pose $P_n(x) = \sum_{k=0}^{2n} x^k$.

1. Cours : Énoncer les trois théorèmes de comparaison de séries à termes positifs.

2. Écrire un script **Python** permettant de représenter la fonction P_{100} sur $[-1, 1]$.

3. Pour $x \in \mathbb{R} \setminus \{1\}$ et $n \in \mathbb{N}^*$, montrer que $P_n'(x) = \dfrac{Q_n(x)}{(x-1)^2}$ où Q_n est une fonction polynomiale à déterminer.

4. Déterminer les variations de P_n pour $n \in \mathbb{N}^*$. Montrer en particulier que P_n admet un minimum sur $\mathbb{R}$ atteint en un unique point u_n de $]-1, 0[$.

5. Compléter le code ci-dessous afin que l'instruction `u(n)` renvoie une approximation de u_n à 10^{-3} près.

```
def Q(n,x):
    return ...

def u(n):
    a, b = ..., ...
    while b-a > ...:
        c = (a+b)/2
        if Q(n,a)*Q(n,c) < 0:
            ...
        else:
            ...
    return ...
```

6. Déterminer un équivalent simple de $\ln(2n + 1 - 2nu_n)$.

7. En déduire la limite de u_n lorsque n tend vers $+\infty$.

8. On pose, pour tout $n \in \mathbb{N}^*$, $h_n = u_n + 1$. Déterminer un équivalent de h_n lorsque n tend vers $+\infty$.

Réponses de l'exercice avec préparation 1 :

1. Soient $\sum u_n$ et $\sum v_n$ deux séries à termes positifs.

- Si, pour tout $n \in \mathbb{N}$, $0 \leqslant u_n \leqslant v_n$, et si la série $\sum v_n$ converge, alors la série $\sum u_n$ converge.
- Si $u_n \underset{n\to+\infty}{=} o(v_n)$ et si la série $\sum v_n$ converge, alors la série $\sum u_n$ converge.
- Si $u_n \underset{n\to+\infty}{\sim} v_n$, alors les séries $\sum u_n$ et $\sum v_n$ ont même nature.

2. On donne ci-dessous un script **Python** permettant de représenter la fonction P_{100} sur $[-1, 1]$.

```
def P(n,x):
    S = 0
    for k in range(2*n+1):
        S += x**k
    return S

n = 100
abscisse = np.linspace(-1,1,1000)
ordonnee = [P(n,x) for x in abscisse]
plt.plot(abscisse, ordonnee)
plt.grid()
plt.show()
```

On peut améliorer le code en faisant moins de calculs de produits pour les puissances de x.

```
def P(n,x):
    S = 0
    puissancex = 1
    for k in range(2*n+1):
        S += x**k
        puissancex *= x
    return S

n = 100
abscisse = np.linspace(-1,1,1000)
ordonnee = [P(n,x) for x in abscisse]
plt.plot(abscisse, ordonnee)
plt.grid()
plt.show()
```

3. Soit $n \in \mathbb{N}^*$. Soit $x \in \mathbb{R} \setminus \{1\}$.
On remarque que $P_n(x)$ est une somme géométrique de raison $x \neq 1$, on peut donc écrire :

$$P_n(x) = \frac{1 - x^{2n+1}}{1 - x}$$

La fonction P_n est dérivable sur $\mathbb{R} \setminus \{1\}$ comme fonction polynomiale :

$$P_n'(x) = \frac{-(2n+1)x^{2n}(1-x) + (1 - x^{2n+1})}{(1-x)^2} = \frac{Q_n(x)}{(x-1)^2}$$

où

$$\begin{aligned} Q_n(x) &= -(2n+1)x^{2n}(1-x) + (1 - x^{2n+1}) \\ &= 1 - x^{2n+1} - (2n+1)x^{2n} + (2n+1)x^{2n+1} \\ &= 1 - (2n+1)x^{2n} + 2nx^{2n+1} \end{aligned}$$

La fonction Q_n est bien polynomiale.

4. Soit $n \in \mathbb{N}^*$.

- La fonction Q_n est dérivable sur $\mathbb{R}$. Soit $x \in \mathbb{R}$.

$$Q_n'(x) = -2n(2n+1)x^{2n-1} + 2n(2n+1)x^{2n} = 2n(2n+1)x^{2n-1}(x-1)$$

Dressons le tableau de variations de Q_n :

x	$-\infty$		0		1		$+\infty$
Signe de $Q_n'(x)$		$+$	0	$-$	0	$+$	
Variations de Q_n	$-\infty$	$\nearrow$	1	$\searrow$	0	$\nearrow$	$+\infty$

Par théorème de la bijection, l'équation $Q_n(x) = 0$ admet une unique solution dans $]-\infty, 0[$, que l'on note u_n.

× Il est clair que $u_n < 0$.

× $Q_n(-1) = 1 - (2n+1) - 2n = -4n < 0$ donc $u_n > -1$.

$$\boxed{u_n \in \left]-1, 0\right[}$$

- Pour $x \geqslant 0$, la formule

$$P_n'(x) = \sum_{k=0}^{2n} kx^{k-1} \geqslant 0$$

prouve que P_n est croissante sur $[0, +\infty[$.

- Pour $x < 0$, $P_n'(x)$ a le même signe que $Q_n(x)$. On en déduit le tableau de variations de P_n :

x	$-\infty$		u_n		$+\infty$
Signe de $P_n'(x)$		$-$	0	$+$	
Variations de P_n	$+\infty$	$\searrow$	$P_n(u_n)$	$\nearrow$	$+\infty$

$$\boxed{\text{La fonction } P_n \text{ admet un minimum atteint en } u_n \text{ et seulement en ce point.}}$$

5. On implémente l'algorithme de dichotomie.

```
def Q(n,x):
    return 1 - (2*n+1) * x**(2*n) + (2*n) * x**(2*n+1)

def u(n):
    a, b = -1, 0
    while b-a > 10**(-3):
        c = (a+b)/2
        if Q(n,a)*Q(n,c) < 0:
            b = c
        else:
            a = c
    return c
```

6. Pour tout $n \in \mathbb{N}^*$, $-1 < u_n < 0$ donc $\ln(2n+1) < \ln(2n+1-2nu_n) < \ln(4n+1)$.
Puisque $\dfrac{\ln(2n+1)}{\ln(n)} \underset{n\to+\infty}{\longrightarrow} 1$ et $\dfrac{\ln(4n+1)}{\ln(n)} \underset{n\to+\infty}{\longrightarrow} 1$, il vient par théorème d'encadrement :

$$\boxed{\ln(2n+1-2nu_n) \underset{n\to+\infty}{\sim} \ln(n).}$$

7. Soit $n \in \mathbb{N}^*$. Par définition, $Q_n(u_n) = 0$. Or,

$$\begin{aligned}
Q_n(u_n) = 0 &\iff 1-(2n+1)u_n^{2n} + 2nu_n^{2n+1} = 0\\
&\iff 1 - u_n^{2n}(2n+1-2nu_n) = 0\\
&\iff 2n+1-2nu_n = \frac{1}{u_n^{2n}}\\
&\iff \ln(2n+1-2nu_n) = -n\ln(u_n^2)\\
&\iff \ln(2n+1-2nu_n) = -n\ln((-u_n)^2)\\
&\iff \ln(2n+1-2nu_n) = -2n\ln(-u_n) && \textit{(car } u_n < 0\textit{)}
\end{aligned}$$

On en déduit que $\ln(-u_n) \underset{n\to+\infty}{\sim} -\dfrac{\ln(n)}{2n}$ donc $\ln(-u_n) \underset{n\to+\infty}{\longrightarrow} 0$ par croissances comparées.

$$\boxed{\text{D'où : } u_n \underset{n\to+\infty}{\longrightarrow} -1.}$$

8. On remarque que, pour tout $n \in \mathbb{N}^*$, $-u_n = 1 - h_n$. On en déduit, puisque $h_n \underset{n\to+\infty}{\longrightarrow} 0$, que :

$$-h_n \underset{n\to+\infty}{\sim} \ln(1-h_n) \underset{n\to+\infty}{\sim} -\frac{\ln(n)}{2n}$$

$$\boxed{\text{D'où : } h_n \underset{n\to+\infty}{\sim} \frac{\ln(n)}{2n}.}$$