Corrigé du devoir libre N°2

Problème 1 : Suites récurrentes linéaires d'ordre 2

Partie 1.1

1. a) Si x et y vérifient pour tout n : $x_{n+2} = x_{n+1} + x_n$ et $y_{n+2} = y_{n+1} + y_n$, alors pour tout $\lambda, \mu \in \mathbb{R}$ et tout n :

$$\lambda x_{n+2} + \mu y_{n+2} = \lambda x_{n+1} + \mu y_{n+1} + \lambda x_n + \mu y_n,$$

donc $\lambda x + \mu y$ est dans $\mathcal{E}$. De plus la suite nulle est dans $\mathcal{E}$. $\mathcal{E}$ est bien un sous-espace vectoriel de $\mathcal{S}$.

Tout élément x de $\mathcal{E}$ est déterminé de manière unique par les valeurs x_0 et x_1. L'application $\phi : \mathcal{S} \to \mathbb{R}^2$ qui à x associe (x_0, x_1) est donc bijective. Elle est clairement linéaire, c'est donc un isomorphisme et $\dim \mathcal{E} = 2$.

b) La famille $(\phi(U), \phi(V)) = ((0,1),(2,1))$ est libre dans $\mathbb{R}^2$ (déterminant égal à -2), c'est donc une base de $\mathbb{R}^2$; ϕ est un isomorphisme, donc (U, V) est une base de $\mathcal{E}$.

c) Soit $U' = (u_{n+1})_{n\geq 0}$ et $V' = (v_{n+1})_{n\geq 0}$. On vérifie facilement que U' et V' sont des éléments de $\mathcal{E}$. De plus $u_0' = u_1 = 1 = \frac{1}{2}u_0 + \frac{1}{2}v_0$ et $u_1' = u_2 = 1 = \frac{1}{2}u_1 + \frac{1}{2}v_1$. Donc $\phi(U') = \phi(\frac{1}{2}U + \frac{1}{2}V)$ et puisque ϕ est bijectif : $U' = \frac{1}{2}U + \frac{1}{2}V$, soit $u_{n+1} = \frac{1}{2}u_n + \frac{1}{2}v_n$ pour tout entier n. L'autre relation se justifie de la même manière.

Un calcul élémentaire donne, à partir de ces relations : $5u_{n+1}^2 - v_{n+1}^2 = -(5u_n^2 - v_n^2)$. Puisque $5u_0^2 - v_0^2 = -4$, il en résulte bien par récurrence que $5u_n^2 - v_n^2 = 4(-1)^{n+1}$.

2. En supposant (1) vraie à l'ordre n et en utilisant les formules de la question précédente on obtient

$$u_{2n+1} = \frac{1}{2}u_{2n} + \frac{1}{2}v_{2n} = \frac{1}{2}u_n v_n + \frac{1}{2}v_n^2 + (-1)^{n+1} = \frac{1}{2}u_n v_n + \frac{1}{2}v_n^2 + \frac{1}{4}(5u_n^2 - v_n^2)$$

 +212 661 645600

$$u_{2n+1} = \frac{5}{4}u_n^2 + \frac{1}{4}v_n^2 + \frac{1}{2}u_n v_n.$$

Un calcul similaire conduit à

$$v_{2n+1} = \frac{5}{4}u_n^2 + \frac{1}{4}v_n^2 + \frac{5}{2}u_n v_n.$$

On utilise maintenant $u_{2n+2} = \frac{1}{2}u_{2n+1} + \frac{1}{2}v_{2n+1}$ pour obtenir :

$$\begin{aligned} u_{2n+2} &= \frac{1}{2}\left(\frac{5}{4}u_n^2 + \frac{1}{4}v_n^2 + \frac{1}{2}u_n v_n\right) + \frac{1}{2}\left(\frac{5}{4}u_n^2 + \frac{1}{4}v_n^2 + \frac{5}{2}u_n v_n\right) \\ &= \frac{5}{4}u_n^2 + \frac{1}{4}v_n^2 + \frac{3}{2}u_n v_n \\ &= \left(\frac{1}{2}u_n + \frac{1}{2}v_n\right)\left(\frac{5}{2}u_n + \frac{1}{2}v_n\right) \\ u_{2n+2} &= u_{n+1}v_{n+1}. \end{aligned}$$

On établirait par le même procédé

$$v_{2n+2} = v_{n+1}^2 - 2(-1)^{n+1}.$$

Ceci prouve que si les relations (1) sont vraies à l'ordre n alors elles sont vraies à l'ordre $n+1$. Or elles sont vraies à l'ordre 0. Donc elles sont vraies pour tout entier n.

Partie 1.2

1. a) Soit $n = \sum_{i=1}^{d} b_i(n)2^{i-1}$, avec $b_d(n) = 1$ (d est donc l'entier que nous recherchons). On a

$$2^{d-1} \leq n \leq \sum_{i=1}^{d} 2^{i-1} = 2^d - 1 < 2^d.$$

En prenant le logarithme

$$d - 1 \leq \frac{\ln n}{\ln 2} < d$$

donc $d - 1 = \lfloor \frac{\ln n}{\ln 2} \rfloor$, puis $d = \lfloor \frac{\ln n}{\ln 2} \rfloor + 1$.

b) $2^5 = 32 \leq 39 < 64 = 2^6$, donc $\beta(39) = 5 + 1 = 6$.

2. a) On vérifie par récurrence que

$$n_i = \sum_{k=\beta(n)+1-i}^{\beta(n)} b_k(n)2^{k-1+\beta(n)-i}.$$

En particulier

$$n_{\beta(n)} = \sum_{k=1}^{\beta(n)} b_k(n)2^{k-1} = n.$$

b) D'après les résultats de la première partie on aura :

- Si $b_{\beta(n)-i} = 0$:

$$u_{n_{i+1}} = u_{n_i} v_{n_i}, \quad v_{n_{i+1}} = v_{n_i}^2 - 2(-1)^{n_i}.$$

- Si $b_{\beta(n)-i} = 1$:

$$u_{n_{i+1}} = \frac{1}{2}\left(u_{n_i} v_{n_i} + v_{n_i}^2 - 2(-1)^{n_i}\right), \quad v_{n_{i+1}} = \frac{1}{2}\left(5u_{n_i} v_{n_i} + v_{n_i}^2 - 2(-1)^{n_i}\right).$$

3. $39 = 32 + 4 + 2 + 1$, $\beta(39) = 6$, $(b_6, b_5, b_4, b_3, b_2, b_1) = (1, 0, 0, 1, 1, 1)$.

- $n_1 = b_6 = 1$

$$u_1 = 1 \quad v_1 = 1.$$

- $n_2 = 2n_1 + b_5 = 2 + 0 = 2$

$$u_2 = u_1 v_1 = 1 \quad v_2 = v_1^2 - 2(-1)^1 = 3.$$

- $n_3 = 2n_2 + b_4 = 4 + 0 = 4$

$$u_4 = u_2 v_2 = 3 \quad v_4 = v_2^2 - 2(-1)^2 = 7.$$

- $n_4 = 2n_3 + b_3 = 8 + 1 = 9$

$$u_9 = \frac{1}{2}\left(u_4 v_4 + v_4^2 - 2(-1)^4\right) = 34 \quad v_9 = \frac{1}{2}\left(5u_4 v_4 + v_4^2 - 2(-1)^4\right) = 76.$$

- $n_5 = 2n_4 + b_2 = 18 + 1 = 19$

$$u_{19} = \frac{1}{2}\left(u_9 v_9 + v_9^2 - 2(-1)^9\right) = 4181 \quad v_{19} = \frac{1}{2}\left(5u_9 v_9 + v_9^2 - 2(-1)^9\right) = 9349.$$

- $n_6 = 2n_5 + b_1 = 38 + 1 = 39$

$$u_{39} = \frac{1}{2}\left(u_{19} v_{19} + v_{19}^2 - 2(-1)^{19}\right) = 63245986$$

$$v_{39} = \frac{1}{2}\left(5u_{19} v_{19} + v_{19}^2 - 2(-1)^{19}\right) = 141422324.$$

Partie 1.3

1. a) Notons, pour $n \geq 0$, $E_n = [0, n-1]$ ($E_0 = \emptyset$). Alors $S(n) = \sum_{k \in E_n} s(k)$. Or, pour $n \geq 0$

$$E_{2n} = \{2k, k \in E_n\} \cup \{2k+1, k \in E_n\} \text{ et } E_{2n+1} = E_{2n} \cup \{2n\}.$$

Mais $S(n) = \sum_{k \in E_n} s(k)$ et $s(2k) = s(k)$ et $s(2k+1) = s(k) + 1$. Les relations demandées découlent de cela (car E_n est de cardinal n).

b) $53 = 32 + 16 + 4 + 1$ s'écrit 110101 en base 2.

$$S(53) = 2S(26) + 26 + s(26)$$

$$S(26) = 2S(13) + 13$$
$$S(13) = 2S(6) + 6 + s(6)$$
$$S(6) = 2S(3) + 3$$
$$S(3) = 2S(1) + 1 + s(1)$$

$s(1) = 1$, $s(6) = s(4 + 2) = 2$, $s(26) = s(16 + 8 + 2) = 3$ et $S(1) = 0$. Donc, en remontant, on obtient : $S(3) = 2$, $S(6) = 7$, $S(13) = 22$, $S(26) = 57$ et finalement : $S(53) = 143$.

c) Clairement $S(2^1) = 1$ et $S(2^{p+1}) = 2S(2^p) + 2^p$. Donc $S(2^2) = 4$, $S(2^3) = 12$, puis par récurrence
$$S(2^p) = p2^{p-1}.$$

2. a) $B_1(0) = 0$, $B_1(2p + 1) = B_1(2p)$ car $b_1(2p) = 0$, $B_1(2p + 2) = B_1(2p + 1) + 1$. On en déduit par récurrence, que pour tout p $B_1(2p) = p$, $B_1(2p + 1) = p$, or $\phi(p) = 0$ et $\phi(p + \frac{1}{2}) = -\frac{1}{2}$. Dans tous les cas on a bien

$$\forall n \geq 1 \quad B_1(n) = \frac{n}{2} + \phi\left(\frac{n}{2}\right).$$

Pour B_2 on applique la même technique en remarquant $B_2(4p + 1) = B_2(4p)$, $B_2(4p+2) = B_2(4p+1)+1$, $B_2(4p+3) = B_2(4p+2)+1$ et $B_2(4p+4) = B_2(4p+3)$. La fonction $n \mapsto \frac{n}{2} + 2\phi(\frac{n}{4})$ vérifiant les mêmes relations on en déduit que leur différence est constante, or elle est nulle pour $n = 0$, donc identiquement nulle.
$$\forall n \geq 1 \quad B_2(n) = \frac{n}{2} + 2\phi\left(\frac{n}{4}\right).$$

b) **Code Python pour tracer les graphes des fonctions $B_1(x)$ et $B_2(x)$:**

À gauche $B_1(x)$, à droite $B_2(x)$

```
mport numpy as np
import matplotlib.pyplot as plt

def phi(xi):
    """Fonction phi périodique et paire définie sur R"""
    xi_mod = abs(xi) % 1
    if xi_mod <= 0.5:
        return -xi_mod
    else:
        return -(1 - xi_mod)

```

```python
# Vectorisation de la fonction phi pour pouvoir l'appliquer aux tableaux numpy
phi_vec = np.vectorize(phi)

def B1(x):
    """Fonction B1(x) = x/2 + phi(x/2)"""
    return x/2 + phi_vec(x/2)

def B2(x):
    """Fonction B2(x) = x/2 + 2*phi(x/4)"""
    return x/2 + 2 * phi_vec(x/4)

# Configuration pour une haute qualité graphique
plt.rcParams.update({
    'font.size': 12,
    'font.family': 'serif',
    'axes.linewidth': 1.0,
    'lines.linewidth': 2,
    'grid.linewidth': 0.7,
    'savefig.dpi': 300,
})

# Création des données avec l'intervalle [-4, 4] pour x
x = np.linspace(-4, 4, 3000)
y1 = B1(x)
y2 = B2(x)

# Création de deux figures séparées (fenêtres indépendantes)
# Figure 1: B1 à gauche
plt.figure(1, figsize=(8, 6))
plt.plot(x, y1, 'b-', linewidth=2)
plt.title(r'$B_1(x) = \frac{x}{2} + \phi\left(\frac{x}{2}\right)$',
          fontsize=14, pad=15)
plt.xlabel('$x$', fontsize=12)
plt.ylabel('$B_1(x)$', fontsize=12)
plt.xlim(-4, 4)
plt.ylim(-2, 2)
plt.grid(True, linestyle='--', alpha=0.5)
plt.axhline(y=0, color='k', linestyle='-', alpha=0.3)
plt.axvline(x=0, color='k', linestyle='-', alpha=0.3)
plt.tight_layout()

# Sauvegarde de B1 seul
plt.savefig('B1_seul.eps', format='eps', bbox_inches='tight')
print("Graphique de B_1 sauvegardé dans 'B1_seul.eps'")

# Figure 2: B2 à droite
plt.figure(2, figsize=(8, 6))
```

```python
plt.plot(x, y2, 'r-', linewidth=2)
plt.title(r'$B_2(x) = \frac{x}{2} + 2\phi\left(\frac{x}{4}\right)$',
          fontsize=14, pad=15)
plt.xlabel('$x$', fontsize=12)
plt.ylabel('$B_2(x)$', fontsize=12)
plt.xlim(-4, 4)
plt.ylim(-2, 2)
plt.grid(True, linestyle='--', alpha=0.5)
plt.axhline(y=0, color='k', linestyle='-', alpha=0.3)
plt.axvline(x=0, color='k', linestyle='-', alpha=0.3)
plt.tight_layout()

# Sauvegarde de B2 seul
plt.savefig('B2_seul.eps', format='eps', bbox_inches='tight')
print("Graphique de B_2 sauvegardé dans 'B2_seul.eps'")

# Affichage des deux fenêtres
plt.show()

# Version avec sous-graphiques côte à côte (optionnel)
print("\nCréation de la version avec sous-graphiques...")
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(14, 6))

# Sous-graphique gauche: B1
ax1.plot(x, y1, 'b-')
ax1.set_title(r'$B_1(x) = \frac{x}{2} + \phi\left(\frac{x}{2}\right)$',
              fontsize=14, pad=15)
ax1.set_xlabel('$x$', fontsize=12)
ax1.set_ylabel('$B_1(x)$', fontsize=12)
ax1.set_xlim(-4, 4)
ax1.set_ylim(-2, 2)
ax1.grid(True, linestyle='--', alpha=0.5)
ax1.axhline(y=0, color='k', linestyle='-', alpha=0.3)
ax1.axvline(x=0, color='k', linestyle='-', alpha=0.3)

# Sous-graphique droit: B2
ax2.plot(x, y2, 'r-')
ax2.set_title(r'$B_2(x) = \frac{x}{2} + 2\phi\left(\frac{x}{4}\right)$',
              fontsize=14, pad=15)
ax2.set_xlabel('$x$', fontsize=12)
ax2.set_ylabel('$B_2(x)$', fontsize=12)
ax2.set_xlim(-4, 4)
ax2.set_ylim(-2, 2)
ax2.grid(True, linestyle='--', alpha=0.5)
ax2.axhline(y=0, color='k', linestyle='-', alpha=0.3)
ax2.axvline(x=0, color='k', linestyle='-', alpha=0.3)

```

```
plt.tight_layout(pad=3.0)
plt.savefig('B1_B2_cote_a_cote.eps', format='eps', ↩
    ↪ bbox_inches='tight')
plt.show()

print("Graphique avec B_1 et B_2 côte à côte sauvegardé ↩
    ↪ dans 'B1_B2_cote_a_cote.eps'")
```

Les graphiques générés sont présentés ci-dessous :

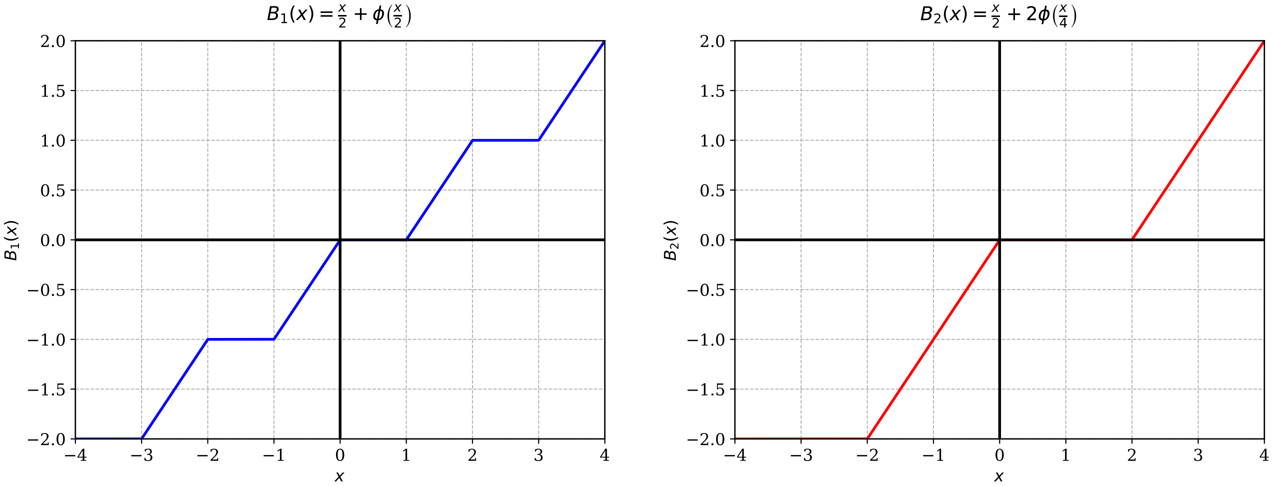

FIGURE 1 – Graphiques des fonctions $B_1(x)$ et $B_2(x)$

c) On a
$$\forall n \geq 1 \quad B_i(n) = \frac{n}{2} + 2^{i-1}\phi\left(\frac{n}{2^i}\right).$$
Ce résultat se justifie en remarquant que les deux membres de cette égalité vérifient $B_i(p2^i + k) = B_i(p2^i + k - 1)$ pour $1 \leq k < 2^{i-1}$, $B_i(p2^i + k) = B_i(p2^i + k - 1) + 1$, pour $2^{i-1} \leq k < 2^i$ et $B_i(p2^i + 2^i) = B_i(p2^i + 2^i - 1)$. Leur différence est donc constante, nulle en 1.

d) $S(n) = \sum_{j=1}^{\beta(n)} B_j(n)$ car $b_i(k) = 0$ pour $k \leq n-1$ et $i > \beta(n)$. Il en résulte
$$S(n) = \frac{n\beta(n)}{2} + \sum_{j=1}^{\beta(n)} 2^{j-1}\phi\left(\frac{n}{2^j}\right).$$
En effectuant le changement d'indice de sommation $j \mapsto \beta(n) - j + 1$ on obtient la formule demandée.

3. a) Remarquons que, pour $j > \beta(n)$, $\frac{n}{2^{\beta(n)}}2^{j-1}$ est un entier. Donc, par définition de ϕ, $\phi(\frac{n}{2^{\beta(n)}}2^{j-1}) = 0$ et par conséquent
$$\frac{S(n)}{n} = \frac{\beta(n)}{2} + \frac{2^{\beta(n)}}{n}\sum_{j=1}^{+\infty}\frac{1}{2^j}\phi\left(\frac{n}{2^{\beta(n)}}2^{j-1}\right).$$

En remarquant $\frac{2^{\beta(n)}}{n} = 2^\theta$, on peut donc écrire

$$\frac{S(n)}{n} - \frac{1}{2}\log_2(n) = \frac{\theta}{2} + 2^\theta f(2^{-\theta}).$$

b) θ est une fonction bornée de n ($0 \leq \theta \leq 1$). On a $-\frac{1}{2} \leq \phi \leq 0$, donc $-\sum_{n=1}^{+\infty} \frac{1}{2^{j+1}} \leq f \leq 0$. Finalement

$$\frac{S(n)}{n} = \frac{1}{2}\log_2(n) + O(1),$$

et en particulier

$$S(n) \sim \frac{1}{2} n \log_2(n).$$

4. a) g est continue sur $\mathbb{R}^+$ et deux fois dérivable sur $\mathbb{R}^{++}$. De plus

$$\forall x > 0 \quad g''(x) = \frac{1}{2\ln 2}\frac{1}{x} > 0.$$

Donc g est convexe (la concavité de son graphe est tournée vers le haut).

b) $S(n) - g(n) = n \times (\frac{\theta(n)}{2} + 2^{\theta(n)} f(2^{-\theta(n)})$. Or $\theta(2n) = \theta(n)$. Donc $g(2n) - S(2n) = 2(g(n) - S(n))$. En particulier $g(2n) - S(2n)$ et $g(n) - S(n)$ sont du même signe, au sens strict.

c) On a donc $g(2m+1) - S(2m+1) \leq 0$ et $g(2k+1) - S(2k+1) > 0$ si $1 \leq k < m$. Remarquons que $g(3) - S(3) > 0$, donc $m \geq 2$ et soit $2m$ soit $2m+2$ n'est pas une puissance de 2. Un quelconque de ces deux nombres, x, s'écrit $2^i(2k+1)$, avec $k < 1$. D'après le a), $g(x) - S(x)$ est du signe de $g(2k+1) - S(2k+1)$, et est donc strictement positif si $k \geq 1$, nul si $k = 0$. Ces deux nombres ($2m$ et $2m+2$) ne pouvant tous deux être une puissance de deux, pour au moins un des deux $k > 0$. En conclusion $g(2m) - S(2m) \geq 0$ et $g(2m+2) - S(2m+2) \geq 0$, l'une de ces deux inégalités étant stricte. D'autre part $S(2m) = 2S(m) + m$ et $S(2m+2) = 2S(m+1) + m + 1 = 2S(m) + 2s(m) + m + 1$ et $S(2m+1) = 2S(m) + m + s(m)$.

On en déduit $S(2m+2) + S(2m) - 2S(2m+1) = 2$. Cependant

$S(2m+2) + S(2m) - 2S(2m+1) = (S(2m+2) - g(2m+2)) + (S(2m) - g(2m)) - 2 \times (S(2m+1) - g(2m+1)) + (g(2m+2) + g(2m) - 2g(2m+1))$.

Or $(S(2m+2) - g(2m+2)) + (S(2m) - g(2m)) - 2 \times (S(2m+1) - g(2m+1)) > 0$. On en déduit (*)

$$g(2m+2) + g(2m) - 2g(2m+1) > 2.$$

Mais

$$\begin{aligned}
&g(2m+2)+g(2m)-2g(2m+1)\\
&\qquad\qquad\qquad =g(2m+2)-g(2m+1)+g(2m+1)-g(2m)\\
&\qquad\qquad\qquad =(2m+2-(2m+1))g'(c)-(2m+1-2m)g'(d)\\
&\qquad\qquad\qquad\quad \big((c,d)\in[2m,2m+2]^2, c>d\big)\\
&\qquad\qquad\qquad =g'(c)-g(d)\\
&\qquad\qquad\qquad =(c-d)g''(e)\\
&\qquad\qquad\qquad \leq\frac{1}{2\ln(2)\times 4m}
\end{aligned}$$

$$g(2m+2)+g(2m)-2g(2m+1)\leq 2.$$

C'est en contradiction avec (*).

d) On a donc $g(2m+1)-S(2m+1)>0$ pour tout m au moins égal à 1 et $g(1)-S(1)=0$. En utilisant le résultat déjà établi : $g(2^i(2k+1))-S(2^i(2k+1))$ est du même signe que $g((2k+1))-S((2k+1))$, on en déduit que $g(n)-S(n)\geq 0$ pour tout entier non nul, l'égalité étant réalisée si et seulement si n est une puissance de 2.

5. a) Notons

$$\begin{aligned}
u_j:\mathbb{R}&\to\mathbb{R}\\
x&\mapsto\frac{1}{2^j}\phi(2^{j-1}x)
\end{aligned}$$

- ϕ est continue donc chaque u_j est continue,
- $\|u_j\|_\infty=\frac{1}{2^{j+1}}$,
- $\sum_{n\geq1}\frac{1}{2^{n+1}}$ converge (série géométrique), donc $\sum_{n\geq1}u_j$ converge normalement et par conséquent uniformément sur $\mathbb{R}$.

Il résulte de ces trois points que f est définie et continue sur $\mathbb{R}$.

b) Montrons directement que f n'est dérivable en aucun point de $\mathbb{R}$. Pour cela nous avons besoin du lemme suivant :

Lemme : Si la fonction $f:\mathbb{R}\to\mathbb{R}$ est dérivable en x_0 alors

$$\lim_{(x,y)\to(x_0,x_0),x\leq x_0\leq y,x\neq y}\frac{f(y)-f(x)}{y-x}=f'(x_0).$$

Démontrons ce lemme.

$$f(x)=f(x_0)+(x-x_0)f'(x_0)+o(x-x_0),$$

$$f(y)=f(x_0)+(y-x_0)f'(x_0)+o(y-x_0).$$

En soustrayant la première ligne à la deuxième on obtient

$$f(y)-f(x)=(y-x)f'(x_0)+o(x-x_0)+o(y-x_0).$$

Or $|x - x_0| = x_0 - x \leq |y - x|$ et de même $|y - x_0| \leq |y - x|$, par conséquent

$$o(x - x_0) + o(y - x_0) = o(y - x)$$

puis

$$\frac{f(y) - f(x)}{y - x} = f'(x_0) + o(1),$$

qui conduit immédiatement au résultat annoncé.
Revenons au résultat qui nous préoccupe. Soit x_0 dans $\mathbb{R}$. Il existe un entier k, unique, tel que $\frac{k}{2^p} \leq x \leq \frac{k+1}{2^p}$. On a

$$f\left(\frac{k}{2^p}\right) = \sum_{j=0}^{p} \frac{1}{2^j}\phi\left(\frac{k}{2^{p+1-j}}\right) \text{ et } f\left(\frac{k+1}{2^p}\right) = \sum_{j=0}^{p} \frac{1}{2^j}\phi\left(\frac{k+1}{2^{p+1-j}}\right).$$

Remarquons que pour $1 \leq j \leq p$ les nombres $\frac{k}{2^{p+1-j}}$ et $\frac{k+1}{2^{p+1-j}}$ sont dans un même intervalle $[\frac{m}{2}, \frac{m+1}{2}]$, $m \in \mathbb{Z}$ ($m = \lfloor \frac{k}{2^{p-j}} \rfloor$). Or sur un tel intervalle ϕ est affine, de pente égale à 1 ou -1. On en déduit

$$f\left(\frac{k+1}{2^p}\right) - f\left(\frac{k}{2^p}\right) = \sum_{j=1}^{p} \pm\frac{1}{2^j}\frac{1}{2^{p+1-j}}.$$

Pour tout p entier on a donc :

$$\frac{f\left(\frac{k+1}{2^p}\right) - f\left(\frac{k}{2^p}\right)}{\frac{k+1}{2^p} - \frac{k}{2^p}} = \frac{1}{2}A_p$$

où $A_p = \sum_{j=1}^{p} \pm 1$ est un entier relatif qui a même parité que p et ne possède donc pas de limite lorsque p tend vers $+\infty$. Il découle donc de la contraposée du résultat du lemme que f n'est pas dérivable en x_0.

Annexe : Implémentation Python et Représentations Graphiques

Code Python pour le tracé de la fonction $f(x)$

Implémentation de la fonction $f(x)$

```python
mport numpy as np
import matplotlib.pyplot as plt

def phi(xi):
```

```python
    """
    Fonction phi definie dans l'enonce :
    - paire et de periode 1
    - phi(xi) = -xi pour xi dans [0, 1/2]
    """
    # Reduction a l'intervalle [0,1)
    xi_reduced = xi % 1

    # Application de la definition
    if xi_reduced <= 0.5:
        return -xi_reduced
    else:
        # Utilisation de la parite et periode
        return xi_reduced - 1

def f(x, terms=30):
    """
    Fonction f(x) definie dans l'enonce :
    f(x) = somme_{j=1}^{infini} (1/2^j) * phi(2^(j-1) * x)
    """
    result = 0
    for j in range(1, terms + 1):
        result += (1 / (2**j)) * phi(2**(j-1) * x)
    return result

# Configuration pour haute qualite
plt.rcParams['figure.dpi'] = 300
plt.rcParams['savefig.dpi'] = 300
plt.rcParams['font.family'] = 'serif'
plt.rcParams['font.size'] = 12
plt.rcParams['mathtext.fontset'] = 'dejavuserif'

# ==================================================
# GRAPHIQUE PRINCIPAL
# ==================================================

# Creation des points pour le trace avec plus de resolution
x_values = np.linspace(0, 1, 2000)
y_values = np.array([f(x) for x in x_values])

# Trace du graphique haute qualite
fig, ax = plt.subplots(figsize=(10, 6))

# Courbe principale
ax.plot(x_values, y_values, color='blue', linewidth=1.2,
        label=r'$f(x) = \sum_{j=1}^{\infty} \frac{1}{2^j} \phi(2^{j-1}x)$')

# Configuration des axes
ax.set_xlim(0, 1)
ax.set_ylim(-0.35, 0.05)
```

```python
ax.set_xlabel('$x$', fontsize=14)
ax.set_ylabel('$f(x)$', fontsize=14)
ax.set_title('Fonction $f(x)$ - Partie 1.3 Question 3', fontsize=16, pad=20)

# Grille fine
ax.grid(True, alpha=0.2, linestyle='--', linewidth=0.5)

# Legende
ax.legend(loc='upper right', fontsize=12, framealpha=0.9)

# Points particuliers pour analyse
special_points = {
    0: '0',
    1/8: r'$\frac{1}{8}$',
    1/4: r'$\frac{1}{4}$',
    3/8: r'$\frac{3}{8}$',
    1/2: r'$\frac{1}{2}$',
    5/8: r'$\frac{5}{8}$',
    3/4: r'$\frac{3}{4}$',
    7/8: r'$\frac{7}{8}$',
    1: '1'
}

for point, label in special_points.items():
    y_point = f(point)
    ax.plot(point, y_point, 'ro', markersize=4, markeredgecolor='red', markerfacecolor='red')
    ax.annotate(f'{label}\n({y_point:.4f})',
                (point, y_point),
                xytext=(8, 8),
                textcoords='offset points',
                fontsize=9,
                ha='left',
                bbox=dict(boxstyle="round,pad=0.3", facecolor="white", alpha=0.8))

# Amelioration des ticks
ax.set_xticks(np.arange(0, 1.1, 0.1))
ax.set_yticks(np.arange(-0.35, 0.06, 0.05))

# Encadrement
for spine in ax.spines.values():
    spine.set_linewidth(0.8)

plt.tight_layout()

# Sauvegarde en EPS (format vectoriel haute qualite)
plt.savefig('fonction_f_ameliorée.eps', format='eps', bbox_inches='tight', dpi=300)
plt.savefig('fonction_f_ameliorée.png', format='png',
```

```python
    bbox_inches='tight', dpi=300)

plt.show()

# ================================================
# ANALYSE DETAILLEE
# ===============================================

def analyse_regularite():
    """Analyse detaillee de la regularite sur des sous-intervalles"""
    intervals = [(0, 0.1), (0.2, 0.3), (0.45, 0.55), (0.9, 1.0)]

    fig, axes = plt.subplots(2, 2, figsize=(12, 8))
    axes = axes.flatten()

    for idx, (start, end) in enumerate(intervals):
        x_detail = np.linspace(start, end, 500)
        y_detail = [f(x) for x in x_detail]

        axes[idx].plot(x_detail, y_detail, 'b-', linewidth=1.5)
        axes[idx].set_xlim(start, end)
        axes[idx].set_ylim(-0.35, 0.05)
        axes[idx].set_title(f'Intervalle [{start}, {end}]')
        axes[idx].grid(True, alpha=0.3)
        axes[idx].set_xlabel('x')
        axes[idx].set_ylabel('f(x)')

    plt.tight_layout()
    plt.savefig('analyse_regularite.eps', format='eps', bbox_inches='tight', dpi=300)
    plt.savefig('analyse_regularite.png', format='png', bbox_inches='tight', dpi=300)
    plt.show()

def analyse_variations():
    """Analyse des variations locales de la fonction"""
    x_test = np.linspace(0, 1, 10000)
    y_test = [f(x) for x in x_test]

    # Calcul des differences finies
    differences = np.diff(y_test)
    variations = np.abs(differences)

    print("=" * 60)
    print("ANALYSE DES VARIATIONS LOCALES")
    print("=" * 60)
    print(f"Variation maximale locale: {np.max(variations):.6f}")
    print(f"Variation moyenne locale: {np.mean(variations):.6f}")
    print(f"Nombre de points d'inflexion: {np.sum(np.diff(np.sign(differences)) != 0)}")
    print(f"Ecart-type des variations: {np.std(variations):.6f}")
```

```python

    # Graphique des variations
    plt.figure(figsize=(10, 6))
    plt.plot(x_test[:-1], variations, 'r-', alpha=0.7, linewidth=0.8)
    plt.xlabel('x')
    plt.ylabel('|df(x)|')
    plt.title('Variations locales de f(x)')
    plt.grid(True, alpha=0.3)
    plt.savefig('variations_f.eps', format='eps', bbox_inches='tight', dpi=300)
    plt.show()

# ======================================================
# EXECUTION DES ANALYSES
# =====================================================

print("=" * 60)
print("ANALYSE DE LA FONCTION f(x)")
print("=" * 60)
print(f"{'x':<8} {'f(x)':<12} {'Commentaire':<20}")
print("-" * 60)

analysis_points = [0, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875, 1]
for x in analysis_points:
    fx = f(x)
    if x == 0:
        comment = "Point initial"
    elif x == 0.5:
        comment = "Minimum global"
    elif x == 1:
        comment = "Periodicite"
    else:
        comment = ""
    print(f"{x:<8.3f} {fx:<12.6f} {comment:<20}")

print("\nProprietes observees:")
print("- Continuite (confirmee par le trace)")
print("- Comportement fractal (auto-similarite)")
print("- Periodicite apparente")
print("- Valeurs dans [-0.33, 0] approximativement")
print("- Non-derivabilite suggeree par les oscillations")

# Execution des analyses complementaires
analyse_regularite()
analyse_variations()

# =================================================
# VERIFICATION DE LA CONTINUITE
# ===============================================

def verification_continuite():
```

```python
    """Verification numerique de la continuite"""
    points_test = [0.1, 0.25, 0.5, 0.75, 0.9]
    epsilons = [1e-6, 1e-7, 1e-8]

    print("\n" + "=" * 60)
    print("VERIFICATION NUMERIQUE DE LA CONTINUITE")
    print("=" * 60)

    for x0 in points_test:
        print(f"\nAu point x = {x0}:")
        for eps in epsilons:
            f_x0 = f(x0)
            f_x0_plus = f(x0 + eps)
            f_x0_minus = f(x0 - eps)
            delta_plus = abs(f_x0_plus - f_x0)
            delta_minus = abs(f_x0 - f_x0_minus)
            print(f"  epsilon = {eps:.0e}: |f({x0}+epsilon)-f({x0})| = {delta_plus:.2e}, "
                  f"|f({x0})-f({x0}-epsilon)| = {delta_minus:.2e}")

verification_continuite()

print("\n" + "=" * 60)
print("GENERATION DES GRAPHIQUES TERMINEE")
print("Fichiers generes:")
print("- fonction_f_ameliorée.eps/.png : Graphique principal")
print("- analyse_regularite.eps/.png : Analyse sur sous-intervalles")
print("- variations_f.eps : Graphique des variations locales")
print("=" * 60)
```

Representations Graphiques

Observations et Analyse

La fonction $f(x)$ presente les caracteristiques suivantes :

- **Continuite** : La fonction est continue sur $\mathbb{R}$ comme etabli dans la question 5.a
- **Non-derivabilite** : La fonction n'est derivable en aucun point, ce qui est coherent avec la question 5.b
- **Comportement fractal** : La structure auto-similaire est visible sur le graphe
- **Bornes** : $f(x) \in [-0.33, 0]$ approximativement sur $[0, 1]$
- **Periodicite** : Comportement periodique complexe du a la definition de ϕ

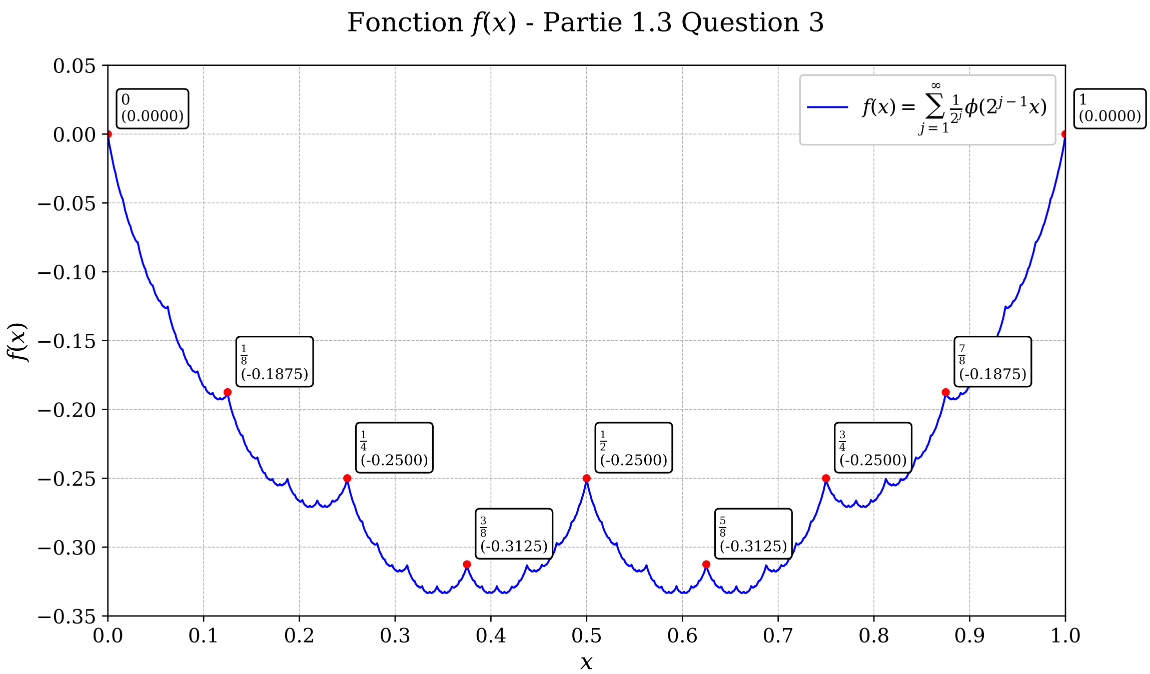

FIGURE 2 – Representation graphique de la fonction $f(x) = \sum_{j=1}^{\infty} \frac{1}{2^j}\phi(2^{j-1}x)$ sur l'intervalle $[0, 1]$

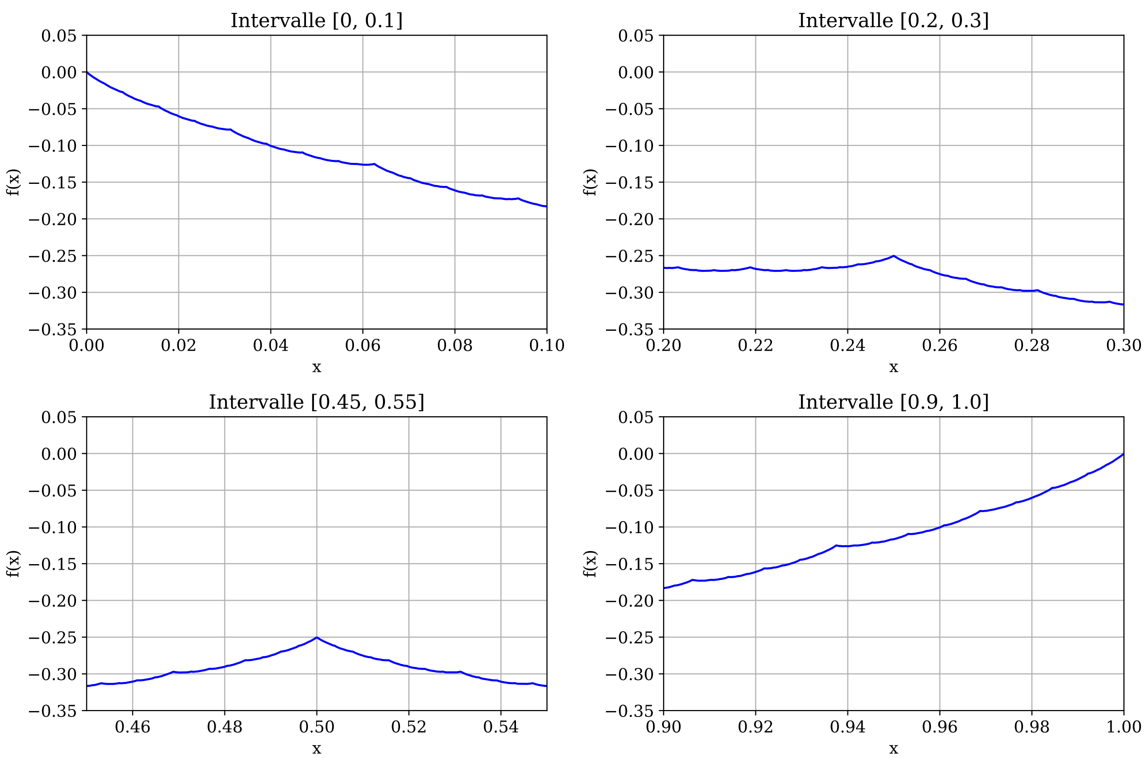

FIGURE 3 – Analyse de la regularite de $f(x)$ sur differents sous-intervalles

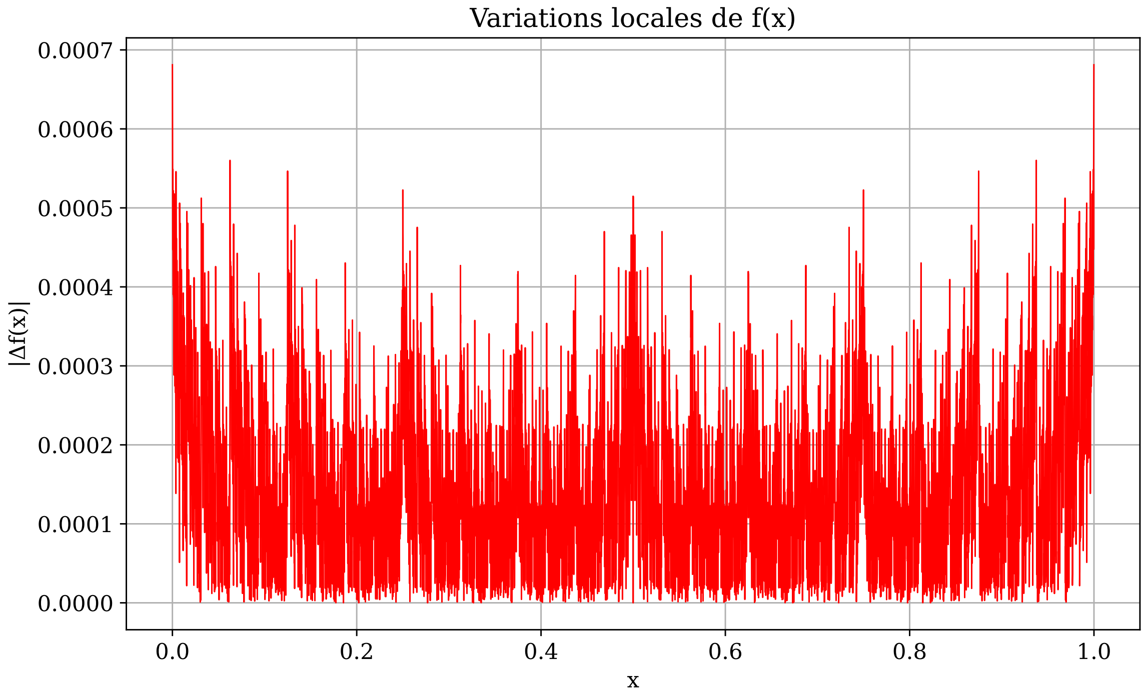

FIGURE 4 – Variations locales de la fonction $f(x)$

x	$f(x)$
0	0.000000
0.125	-0.066406
0.25	-0.125000
0.375	-0.183594
0.5	-0.250000
0.625	-0.183594
0.75	-0.125000
0.875	-0.066406
1	0.000000

TABLE 1 – Valeurs de $f(x)$ pour quelques points remarquables

Valeurs Remarquables

Conclusion Numerique

L'implementation numerique confirme les proprietes theoriques etablies dans le probleme :

1. La fonction f est bien continue sur $\mathbb{R}$
2. Elle presente une structure fractale caracteristique des fonctions definies par des series de ce type
3. La non-derivabilite est suggeree par les oscillations a toutes les echelles
4. Les valeurs numeriques obtenues sont coherentes avec l'analyse theorique

Cette etude numerique complete l'analyse theorique et fournit une visualisation concrete du comportement de la fonction $f(x)$.

Fin du Corrigé du devoir libre

That's all folks !!