Takagi01

October 25, 2020

1 La fonction de Takagi

Marc Lorenzi

17 octobre 2020

[1]:
```
import matplotlib.pyplot as plt
import math
from fractions import Fraction
```

[2]:
```
plt.rcParams['figure.figsize'] = (10, 6)
```

1.1 1. Introduction

Teiji Takagi a introduit en 1903 la fonction qui porte son nom comme un exemple de fonction continue partout et dérivable nulle part. Cette fonction est définie pour tout réel x sous forme de série par

$$\tau(x) = \sum_{k=0}^{\infty} \frac{1}{2^k} \ll 2^k x \gg$$

où $\ll x \gg = d(x, \mathbb{Z})$ est la *distance de x à $\mathbb{Z}$*.

Il y a beaucoup à dire sur cette fonction, et la place disponible dans ce notebook a obligé à faire des choix. Nous nous concentrerons sur les points suivants :

- Une présentation de fonctions affines par morceaux τ_n qui approchent (nous verrons en quel sens) la fonction de Takagi.
- La continuité et la non dérivabilité de τ.
- Le calcul exact de $\tau(x)$ pour tout $x \in \mathbb{Q}$.

1.1.1 1.1 Les fonctions α et $\ll \gg$

Soient α et $\ll \gg$ les deux fonctions $\mathbb{R} \to \mathbb{R}$ définies pour tout réel x par

$$\alpha(x) = \left\lfloor x + \frac{1}{2} \right\rfloor$$

et

$$\ll x \gg = |x - \alpha(x)|$$

[3]:
```
def alpha(x): return math.floor(x + 1/2)
```

Nous appellerons `delta` la fonction Python qui calcule $\ll x \gg$.

[4]:
```
def delta(x): return abs(x - alpha(x))
```

Proposition. Soit $x \in [0, 1]$.

- Si $0 \leq x < \frac{1}{2}$, $\ll x \gg = x$.
- Si $\frac{1}{2} \leq x \leq 1$, $\ll x \gg = 1 - x$.

Démonstration.

- Si $0 \leq x < \frac{1}{2}$, on a $\frac{1}{2} \leq x + \frac{1}{2} < 1$, et donc $\alpha(x) = 0$, d'où $\ll x \gg = |x - 0| = x$.
- Si $\frac{1}{2} \leq x < 1$, on a $1 \leq x + \frac{1}{2} \leq \frac{3}{2}$, et donc $\alpha(x) = 1$, d'où $\ll x \gg = |x - 1| = 1 - x$.

Proposition. La fonction $\ll \gg$ est de période 1.

Démonstration. Pour tout réel x,

$$\alpha(x+1) = \left\lfloor x + 1 + \frac{1}{2} \right\rfloor = \left\lfloor x + \frac{1}{2} \right\rfloor + 1 = \alpha(x) + 1$$

De là,

$$\ll x + 1 \gg = |x + 1 - \alpha(x+1)| = |x + 1 - (\alpha(x) + 1)| = |x - \alpha(x)| = \ll x \gg$$

En fait, 1 est la plus petite période de $\ll \gg$. En effet, soit $T \in]0, 1[$.

- Si $0 < t \leq \frac{1}{2}$, $\ll T \gg = T \neq 0 = \ll 0 \gg$ donc T n'est pas période de $\ll \gg$.
- Si $\frac{1}{2} < T < 1$, $\ll T \gg = 1 - T \neq 0 = \ll 0 \gg$ donc T n'est pas période de $\ll \gg$.

1.1.2 1.2 Tracer une courbe

Définissons tout d'abord une fonction `subdi` qui prend en paramètres deux réels a et b et un entier n, et renvoie la liste

$$\left[a + k\frac{b-a}{n}, 0 \leq k \leq n \right]$$

[5]:
```
def subdi(a, b, n):
    d = (b - a) / n
    return [a + k * d for k in range(n + 1)]
```

[6]:
```
print(subdi(-1, 1, 10))
```

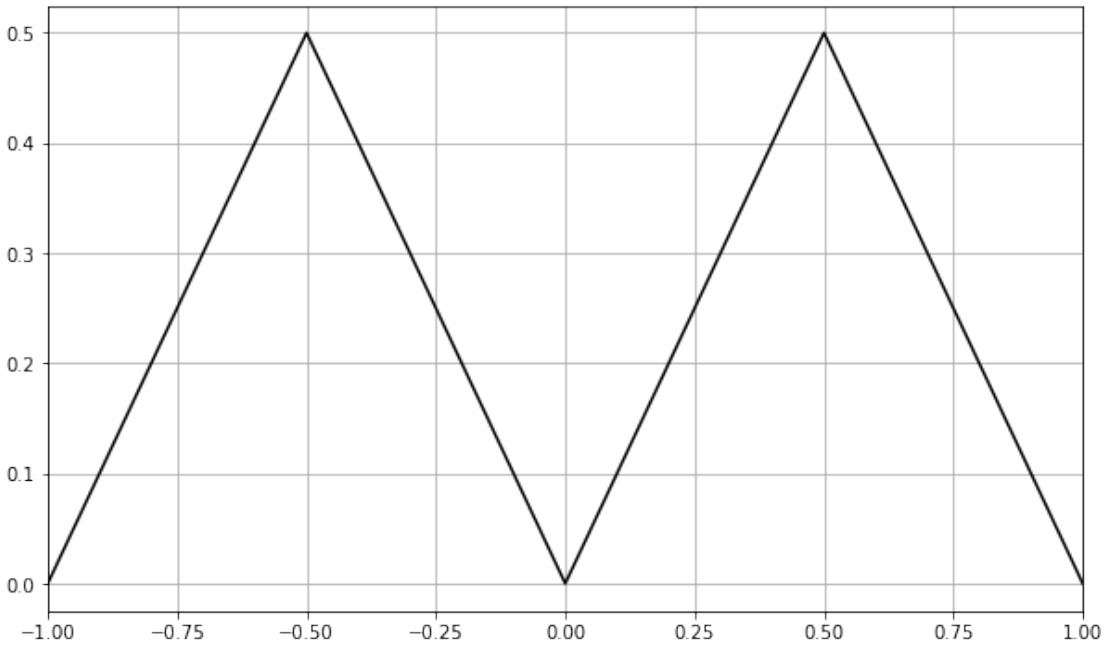
```
[-1.0, -0.8, -0.6, -0.3999999999999999, -0.19999999999999996, 0.0,
0.20000000000000018, 0.40000000000000013, 0.6000000000000001, 0.8, 1.0]
```

Définissons également une fonction `plot_fun` qui prend en paramètres une fonction f et deux réels a et b, et qui trace la courbe représentative de f sur le segment $[a, b]$.

[7]:
```
def plot_fun(f, a, b, n, dots=False):
    xs = subdi(a, b, n)
    ys = [f(x) for x in xs]
    plt.xlim(a, b)
    plt.plot(xs, ys, 'k')
    if dots: plt.plot(xs, ys, 'or')
    plt.grid()
```

Voici par exemple le graphe de la fonction $x \mapsto \ll x \gg$ sur $[-1, 1]$.

[8]:
```
plot_fun(delta, -1, 1, 4)
```



1.1.3 1.3 La fonction τ_n

Pour tout entier n, soit $\tau_n : \mathbb{R} \to \mathbb{R}$ définie pour tout réel x par

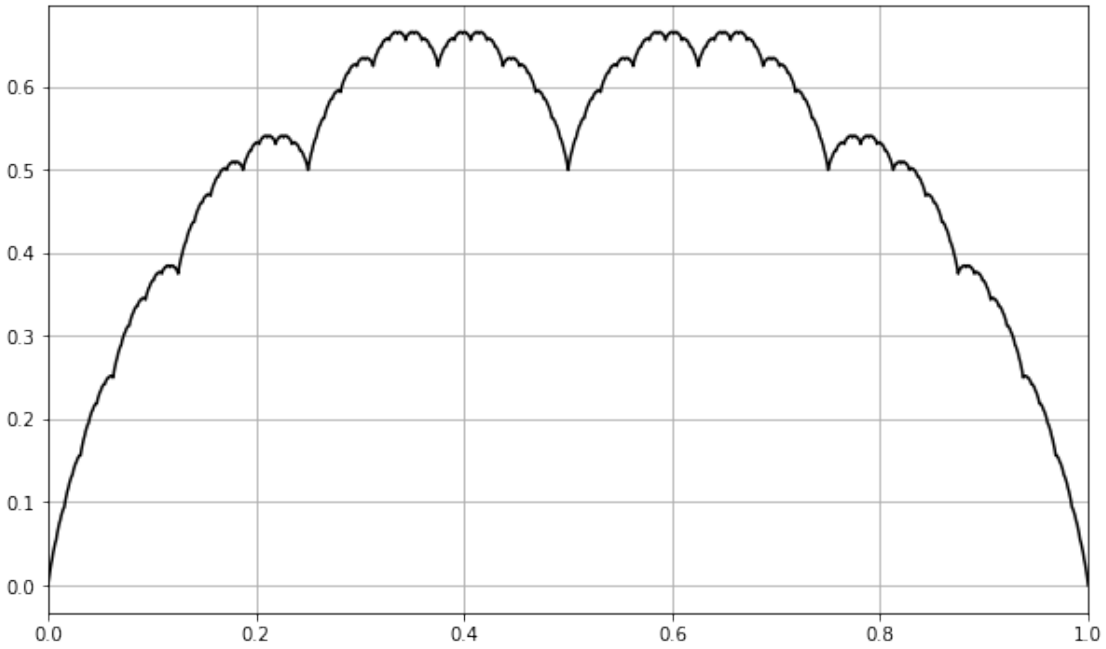
$$\tau_n(x) = \sum_{k=0}^{n} \frac{1}{2^k} \ll 2^k x \gg$$

La fonction τ_n est de période 1. Elle est définie très simplement en Python par

[9]:
```
def tau(n, x):
    return sum([delta(2 ** k * x) / 2 ** k for k in range(n + 1)])
```

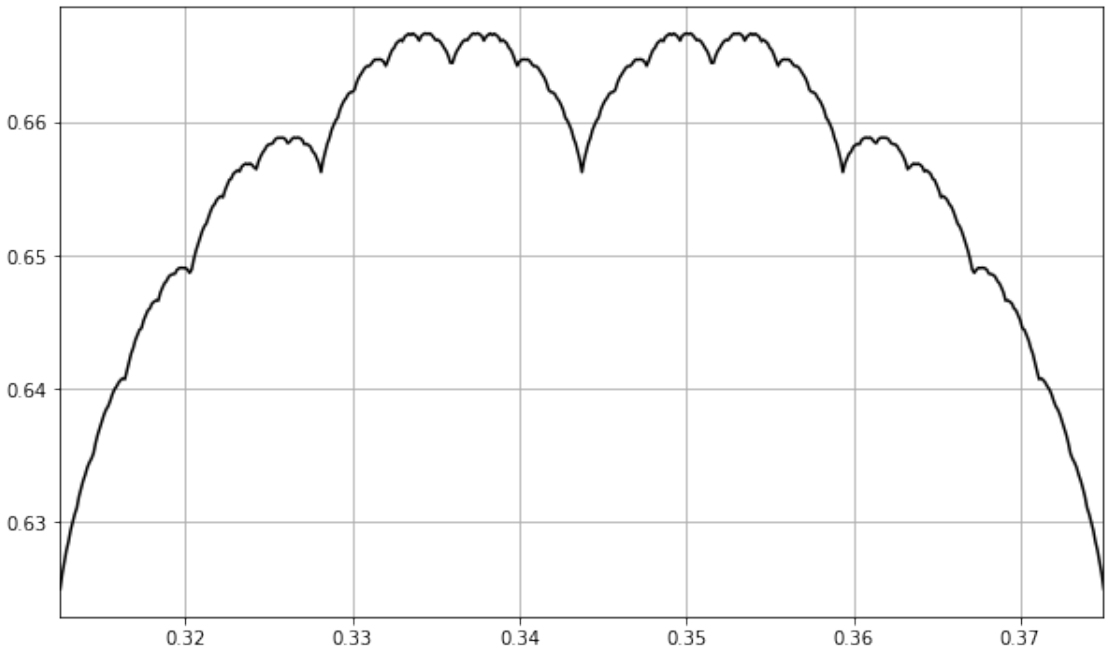
Voici le graphe de τ_{40} sur $[0, 1]$.

[10]:
```
plot_fun(lambda x: tau(40, x), 0, 1, 2 ** 10, dots=False)
```



Voici un zoom de τ_{40}.

[11]:
```
plot_fun(lambda x: tau(40, x), 5/16, 6/16, 500, dots=False)
```



Point n'est besoin d'être très observateur pour remarquer que les deux courbes précédentes se ressemblent. Nous parlerons de ceci dans un autre notebook : la courbe de la fonction de Takagi possède des propriétés que l'on attribue à ce que l'on appelle souvent les fractales.

1.1.4 1.4 Nombres dyadiques

Notons $\mathbb{D} = \left\{ \frac{p}{2^n}, p \in \mathbb{Z}, n \in \mathbb{N} \right\}$. L'ensemble $\mathbb{D}$ est l'ensemble des *nombres dyadiques*.

Proposition. Soit $x \in \mathbb{R}$. Pour tout $n \in \mathbb{N}$ il existe $u, v \in \mathbb{D}$ tels que

$$u \leq x < v \quad \text{et} \quad v - u = \frac{1}{2^n}$$

Démonstration. Soit $x \in \mathbb{R}$. Soit $n \in \mathbb{N}$. Posons $u = \frac{1}{2^n} \lfloor 2^n x \rfloor$ et $v = u + \frac{1}{2^n}$. On a alors $u, v \in \mathbb{D}$. De plus,

$$\lfloor 2^n x \rfloor \leq 2^n x < \lfloor 2^n x \rfloor + 1$$

et donc, en divisant par 2^n, $u \leq x < v$.

Corollaire. $\mathbb{D}$ est dense dans $\mathbb{R}$.

1.2 2. Convergence, continuité

Remarque. Les preuves ci-dessous peuvent être simplifiées en utilisant des résultats sur les séries de fonctions. J'ai dans la mesure du possible fait les démonstrations à un niveau « élémentaire ».

1.2.1 2.1 La fonction de Takagi

Proposition. Soit $x \in \mathbb{R}$. La suite $(\tau_n(x))_{n\geq 0}$ converge lorsque n tend vers l'infini.

Démonstration. Pour tout $n \geq 0$, on a

$$\tau_{n+1}(x) - \tau_n(x) = \frac{1}{2^{n+1}} \ll 2^{n+1}x \gg \geq 0$$

Ainsi, la suite $(\tau_n(x))_{n\geq 0}$ est croissante. De plus, pour tout $n \geq 0$,

$$\begin{array}{rcl} \tau_n(x) & = & \sum_{k=0}^{n} \frac{1}{2^k} \ll 2^k x \gg \\ & \leq & \frac{1}{2} \sum_{k=0}^{n} \frac{1}{2^k} \\ & = & \frac{1}{2} \frac{1-\frac{1}{2^{n+1}}}{1-\frac{1}{2}} \\ & \leq & 1 \end{array}$$

La suite est donc majorée par 1. Ainsi, cette suite converge. Notons dorénavant pour tout réel x

$$\tau(x) = \lim_{n\to\infty} \tau_n(x) = \sum_{k=0}^{\infty} \frac{1}{2^k} \ll 2^k x \gg$$

Définition. La fonction $\tau : \mathbb{R} \to \mathbb{R}$ est appelée la **fonction de Takagi**.

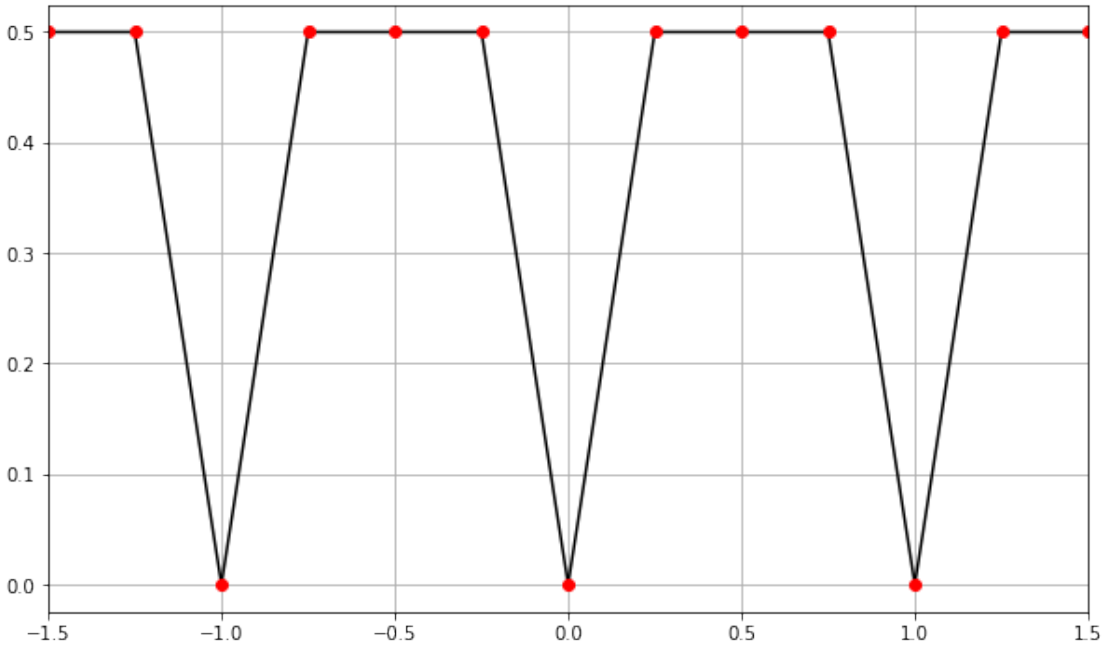
La preuve de la proposition précédente montre que pour tout réel x, $\tau(x) \leq 1$. Comme pour tout entier n, $\tau_n \geq 0$, on a par passage à la limite $\tau \geq 0$. Enfin, toujours par passage à la limite, τ est de période 1.

Remarque. La majoration de τ par 1 n'est pas optimale. On peut faire mieux en remarquant que pour tout $x \in \mathbb{R}$,

$$\tau_1(x) = \ll x \gg + \frac{1}{2} \ll 2x \gg \leq \frac{1}{2}$$

Le faire en exercice, ou alors regarder le graphe de τ_1 ci-dessous.

[12]:
```
plot_fun(lambda x: tau(1, x), -1.5, 1.5, 12, dots=True)
```



De là, en regroupant les termes deux par deux,

$$\begin{array}{rcl} \tau(x) & = & \sum_{k=0}^{\infty} \frac{1}{2^k} \ll 2^k x \gg \\ & = & \sum_{j=0}^{\infty} \left(\frac{1}{2^{2j}} \ll 2^{2j} x \gg + \frac{1}{2^{2j+1}} \ll 2^{2j+1} x \gg \right) \\ & = & \sum_{j=0}^{\infty} \frac{1}{2^{2j}} \tau_1(2^{2j} x) \\ & \leq & \frac{1}{2} \sum_{j=0}^{\infty} \frac{1}{2^{2j}} \\ & = & \frac{2}{3} \end{array}$$

Le majorant $\frac{2}{3}$ est optimal, c'est le maximum de la fonction τ. En effet,

Proposition. $\tau\left(\frac{1}{3}\right) = \frac{2}{3}$.

Démonstration. Pour tout $k \in \mathbb{N}$, $2^k \equiv (-1)^k \bmod 3$ et donc $\ll \frac{2^k}{3} \gg = \ll \frac{(-1)^k}{3} \gg = \frac{1}{3}$. De là,

$$\tau\left(\frac{1}{3}\right) = \frac{1}{3} \sum_{k=0}^{\infty} \frac{1}{2^k} = \frac{2}{3}$$

Nous reviendrons plus loin sur le calcul de $\tau(x)$ lorsque $x \in \mathbb{Q}$.

1.2.2 2.2 Le cas des nombres dyadiques

Soit $a = \frac{p}{2^N} \in \mathbb{D}$. Pour tout $k \geq N$, $2^k a \in \mathbb{Z}$ et ainsi $\ll 2^k a \gg = 0$. On en déduit que pour tout $n \geq N$, $\tau_n(a) = \tau_N(a) = \tau(a)$. On peut donc calculer $\tau(a)$ de façon exacte au moyen d'une somme finie.

La fonction `tau_dyadique` ci-dessous prend en paramètre un élément $a \in \mathbb{D}$ (appartenant à la classe `Fraction`) et renvoie la valeur exacte de $\tau(a)$.

[13]:
```
def log2(m):
    l = 0
    while m != 1:
        l = l + 1
        m = m // 2
    return l
```

[14]:
```
log2(1024)
```

[14]: 10

[15]:
```
def tau_dyadique(a):
    N = log2(a.denominator)
    s = 0
    p2k = 1
    for k in range(N):
        s = s + delta(p2k * a) / p2k
        p2k = 2 * p2k
    return s
```

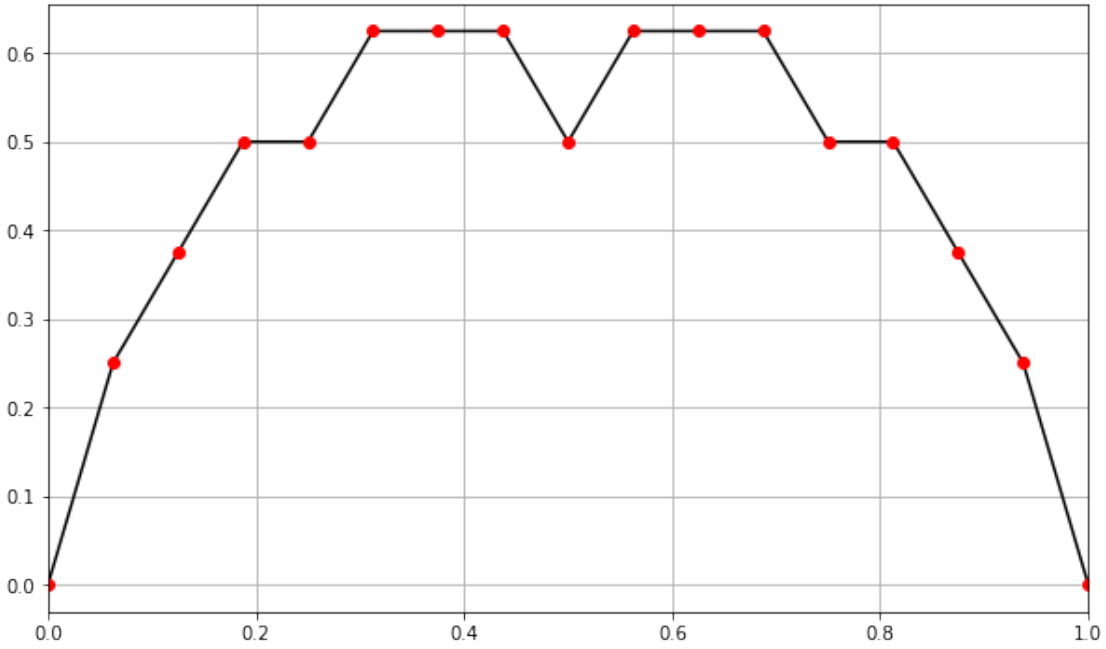
Calculons $\tau(\frac{k}{16})$ pour $0 \leq k \leq 16$.

[16]:
```
for k in range(17):
    x = Fraction(k, 16)
    print('%10s%20s' % (x, tau_dyadique(x)))
```

```
         0                   0
      1/16                 1/4
       1/8                 3/8
      3/16                 1/2
       1/4                 1/2
      5/16                 5/8
       3/8                 5/8
      7/16                 5/8
       1/2                 1/2
      9/16                 5/8
       5/8                 5/8
     11/16                 5/8
       3/4                 1/2
     13/16                 1/2
       7/8                 3/8
     15/16                 1/4
        1                   0
```

On retrouve bien ces valeurs sur le graphe de τ_4.

[17]:
```
plot_fun(lambda x: tau(4, x), 0, 1, 2 ** 4, dots=True)
```



1.2.3 2.3 Continuité

Attaquons-nous à la continuité de la fonction de Takagi.

Lemme. Pour tous $x \in \mathbb{R}$ et $n \in \mathbb{N}$,

$$0 \leq \tau(x) - \tau_n(x) \leq \frac{1}{2^{n+1}}$$

Démonstration. Soient $m, n \in \mathbb{N}$ vérifiant $n+1 \leq m$. On a pour tout réel x

$$0 \leq \tau_m(x) - \tau_n(x) = \sum_{k=n+1}^{m} \frac{1}{2^k} \ll 2^k x \gg \leq \frac{1}{2} \sum_{k=n+1}^{m} \frac{1}{2^k}$$

Or,

$$\sum_{k=n+1}^{m} \frac{1}{2^k} = \frac{1}{2^{n+1}} \sum_{k=n+1}^{m} \frac{1}{2^{k-n-1}} = \frac{1}{2^{n+1}} \sum_{k=0}^{m-n-1} \frac{1}{2^k} \leq \frac{1}{2^n}$$

Ainsi,

$$0 \leq \tau_m(x) - \tau_n(x) \leq \frac{1}{2^{n+1}}$$

Faisons maintenant tendre m vers $+\infty$. Il vient

$$0 \leq \tau(x) - \tau_n(x) \leq \frac{1}{2^{n+1}}$$

Remarque. Nous venons de majorer $\tau(x) - \tau_n(x)$ par une quantité qui ne dépend pas de x. Cette majoration s'appelle une majoration **uniforme** et sera la clé de la preuve de la proposition suivante.

Proposition. τ est continue sur $\mathbb{R}$.

Démonstration. Soit $x \in \mathbb{R}$. Montrons que τ est continue en x.

On peut facilement montrer que la fonction $\ll \gg$, affine par morceaux, est continue sur $\mathbb{R}$. De là, par les opérations sur les fonctions continues, les fonctions τ_n sont aussi continues sur $\mathbb{R}$.

Soit $\varepsilon > 0$. Il existe $N \in \mathbb{N}$ tel que pour tout $n \geq N$, $\frac{1}{2^{n+1}} \leq \frac{1}{3}\varepsilon$. On a alors pour tout $n \geq N$, pour tout $t \in \mathbb{R}$,

$$0 \leq \tau(t) - \tau_n(t) \leq \frac{1}{3}\varepsilon$$

La fonction τ_N étant continue en x, il existe $\alpha > 0$ tel que pour tout réel t,

$$|t - x| \leq \alpha \implies |\tau_N(t) - \tau_N(x)| \leq \frac{1}{3}\varepsilon$$

Pour tout $t \in \mathbb{R}$ vérifiant $|t - x| \leq \alpha$, on a alors

$$\begin{array}{rcl} |\tau(t) - \tau(x)| & = & |\tau(t) - \tau_N(t) + \tau_N(t) - \tau_N(x) + \tau_N(x) - \tau(x)| \\ & \leq & |\tau(t) - \tau_N(t)| + |\tau_N(t) - \tau_N(x)| + |\tau_N(x) - \tau(x)| \\ & \leq & \frac{1}{3}\varepsilon + \frac{1}{3}\varepsilon + \frac{1}{3}\varepsilon \\ & = & \varepsilon \end{array}$$

1.3 3. Non-dérivabilité

Avant de passer au gros morceau, regardons un peu ce que nous pouvons dire sur les fonctions $x \mapsto \frac{1}{2^k} \ll 2^k x \gg$.

1.3.1 3.1 Fonctions affines par morceaux

Pour tout $k \in \mathbb{N}$ et tout réel x, notons

$$\psi_k(x) = \frac{1}{2^k} \ll 2^k x \gg$$

Proposition. Soient $p \in \mathbb{Z}$, $n \in \mathbb{N}^*$ et $0 \leq k \leq n-1$. La fonction ψ_k est affine sur $[\frac{p}{2^n}, \frac{p+1}{2^n}]$. Sa pente sur cet intervalle est égale à ± 1.

Démonstration. Effectuons la division euclidienne de p par 2^{n-k} : $p = q2^{n-k} + r$ où $0 \leq r < 2^{n-k}$.

Posons $u = \frac{p}{2^n}$ et $v = \frac{p+1}{2^n}$. On a $2^k u = q + \frac{r}{2^{n-k}}$ et $2^k v = q + \frac{r+1}{2^{n-k}}$.

- Cas 1 : $0 \leq r < 2^{n-k-1}$. Alors $r + 1 \leq 2^{n-k-1}$. Pour tout $x \in [u, v]$, on a

$$q \leq q + \frac{r}{2^{n-k}} \leq 2^k x \leq q + \frac{r+1}{2^{n-k}} \leq q + \frac{1}{2}$$

et donc, $2^k x - q \in [0, \frac{1}{2}]$. Ainsi, $\ll 2^k x - q \gg = 2^k x - q$. Mais $\ll \gg$ est de période 1, donc $\ll 2^k x \gg = \ll 2^k x - q \gg$. Finalement, en divisant par 2^n,

$$\psi_k(x) = x - \frac{q}{2^n}$$

- Cas 1 : $2^{n-k-1} \leq r < 2^{n-k}$. Alors $2^{n-k-1} \leq r + 1 \leq 2^{n-k}$. Pour tout $x \in [u, v]$, on a

$$q + \frac{1}{2} \leq q + \frac{r}{2^{n-k}} \leq 2^k x \leq q + \frac{r+1}{2^{n-k}} \leq q + 1$$

et donc, $2^k x - q \in [\frac{1}{2}, 1]$. Ainsi, $\ll 2^k x - q \gg = 1 - (2^k x - q) = 1 + q - 2^k x$. Mais $\ll \gg$ est de période 1, donc $\ll 2^k x \gg = \ll 2^k x - q \gg$. Finalement, en divisant par 2^n,

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

1.3.2 3.2 La non-dérivabilité de τ

Nous y voilà. Tout d'abord un petit lemme très général concernant les fonctions dérivables.

Lemme. Soit f une fonction dérivable en un réel x. Soient (u_n) et (v_n) deux suites de réels telles que

- Pour tout entier n, $u_n \leq x < v_n$
- u_n tend vers x lorsque $n \to \infty$.
- v_n tend vers x lorsque $n \to \infty$.

Alors $\frac{f(v_n) - f(u_n)}{v_n - u_n} \to f'(x)$ lorsque n tend vers l'infini.

Démonstration. On a pour tout réel t voisin de x

$$f(t) = f(x) + (t - x)f'(x) + (t - x)\varepsilon(t)$$

où $\varepsilon(t) \to 0$ lorsque $t \to x$. En appliquant cette égalité à u_n et v_n, il vient

$$f(v_n) = f(x) + (v_n - x)f'(x) + (v_n - x)\varepsilon(v_n)$$
$$f(u_n) = f(x) + (u_n - x)f'(x) + (u_n - x)\varepsilon(u_n)$$

Soustrayons, divisons par $v_n - u_n$. On obtient

$$\frac{f(v_n)-f(u_n)}{v_n-u_n} = f'(x) + A(n,x)$$

où

$$|A(n,x)| = \left|\frac{(v_n-x)\varepsilon(v_n)-(u_n-x)\varepsilon(u_n)}{v_n-u_n}\right| \leq \frac{v_n-x}{v_n-u_n}|\varepsilon(v_n)| + \frac{x-u_n}{v_n-u_n}|\varepsilon(u_n)| \leq |\varepsilon(v_n)| + |\varepsilon(u_n)|$$

Ainsi, $A(n,x)$ tend vers 0 lorsque n tend vers l'infini, puisque v_n et u_n tendent vers x.

Proposition. τ n'est dérivable nulle part.

Démonstration. Soit $x \in \mathbb{R}$. Pour tout $n \in \mathbb{N}$, donnons nous $u_n = \frac{p}{2^n} \in \mathbb{D}$ tel que

$$u_n \leq x < v_n = u_n + \frac{1}{2^n}$$

Pour tout entier $k \geq n$, $2^k u_n$ et $2^k v_n$ diffèrent d'un entier. Par périodicité de $\ll \gg$, on a donc $\ll 2^k u_n \gg = \ll 2^k v_n \gg$. Ainsi,

$$\tau(v_n) - \tau(u_n) = \sum_{k=0}^{n-1} (\psi_k(v_n) - \psi_k(u_n))$$

et donc

$$\frac{\tau(v_n)-\tau(u_n)}{v_n-u_n} = \sum_{k=0}^{n-1} \frac{\psi_k(v_n)-\psi_k(u_n)}{v_n-u_n}$$

La fonction ψ_k est affine sur le segment $[u_n, v_n]$ ainsi,

$$\frac{\psi_k(v_n)-\psi_k(u_n)}{v_n-u_n} = \psi_k^+(x)$$

où $\psi_k^+(x)$ est la dérivée à droite de ψ_k en x (qui, en fait, ne dépend pas de x puisqu'on dérive une fonction affine). De là,

$$\frac{\tau(v_n)-\tau(u_n)}{v_n-u_n} = \sum_{k=0}^{n-1} \psi_k^+(x)$$

Mais $\psi_k^+(x) = \pm 1$ ne tend pas vers 0 lorsque k tend vers l'infini. La somme $\sum_{k=0}^{n-1} \psi_k^+(x)$ ne converge donc pas lorsque n tend vers l'infini. On en déduit par le lemme précédent que la fonction τ n'est pas dérivable en x.

1.3.3 3.3 Non-dérivable, mais ...

Bien que non dérivable, la fonction de Takagi possède tout de même des propriétés de régularité intéressantes.

Définition. Soit $f : \mathbb{R} \to \mathbb{R}$. Soit $s \in]0, 1]$. On dit que f est s-holdérienne lorsqu'il existe $M \in \mathbb{R}_+$ tel que pour tous $x, y \in \mathbb{R}$,

$$|f(y) - f(x)| \leq M|y - x|^s$$

Le cas $s = 1$ redonne la notion de fonction lipschitzienne. La fonction de Takagi ne peut pas être lipschitzienne, car un résultat très général dit que toute fonction lipschitzienne est dérivable presque partout. En revanche, on a le résultat suivant.

Proposition. Pour tout $s \in]0, 1[$, la fonction τ est s-holdérienne.

Démonstration. Soient $x, t \in \mathbb{R}$. Supposons tout d'abord $0 < |t| \leq 1$. Soit $n \in \mathbb{N}$ tel que

$$\frac{1}{2^{n+1}} < |t| \leq \frac{1}{2^n}$$

Rappelons-nous que la fonction ψ_k est affine par morceaux, avec des pentes égales à ± 1. Il en résulte que ψ_k est 1-lipschitzienne : pour tous réels x, t,

$$|\psi_k(x + t) - \psi_k(x)| \leq |t|$$

De là,

$$\begin{array}{rcl} |\tau_n(x + t) - \tau_n(x)| & = & |\sum_{k=0}^{n}(\psi_k(x + t) - \psi_k(x))| \\ & \leq & (n + 1)|t| \\ & = & (n + 1)|t|^{1-s}|t|^s \\ & \leq & (n + 1)\frac{1}{2^{n(1-s)}}|t|^s \end{array}$$

Par ailleurs, on a $0 \leq \psi_k(x) \leq \frac{1}{2^{k+1}}$ et de même pour $\psi_k(x + t)$. Ainsi,

$$|\psi_k(x + t) - \psi_k(x)| \leq \frac{1}{2^k}$$

De là,

$$|\sum_{k=n+1}^{\infty}(\psi_k(x + t) - \psi_k(x))| \leq \sum_{k=n+1}^{\infty}\frac{1}{2^k} = \frac{1}{2^n} \leq 2|t| \leq 2|t|^s$$

En additionnant, on obtient donc

$$|\tau(x + t) - \tau(x)| \leq \left(2 + (n + 1)\frac{1}{2^{n(1-s)}}\right)|t|^s$$

La suite $\left(2+(n+1)\frac{1}{2^{n(1-s)}}\right)_{n\geq 0}$ tend vers 2 lorsque n tend vers l'infini. Elle est donc majorée par un réel M_s indépendant de n. Remarquons que $M_s \geq 2$ puisqu'il majore une suite qui tend vers 2. Finalement,

$$|\tau(x+t)-\tau(x)| \leq M_s|t|^s$$

Prenons maintenant $|t|>1$. Rappelons nous que pour tout réel x, $0 \leq \tau(x) \leq 1$. Ainsi,

$$|\tau(x+t)-\tau(x)| \leq 2 \leq 2|t|^s \leq M_s|t|^s$$

1.3.4 3.4 Tangente verticale

Existe-t-il des réels x en lesquels les taux d'accroissements de τ tendent vers $+\infty$ (en lesquels nous dirons abusivement que $\tau' = +\infty$) ou vers $-\infty$? Oui, et il en a même beaucoup. Nous citons sans démonstration le résultat suivant.

Théorème [Allaart, Kawamura 2010]. L'ensemble des points où $\tau' = +\infty$ et l'ensemble des points où $\tau' = -\infty$ sont denses dans $[0,1]$ et ont une dimension de Hausdorff égale à 1 (ce qui prouve qu'ils ne sont pas dénombrables). Tous les nombres dyadiques appartiennent à l'un de ces deux ensembles, mais ce ne sont pas les seuls.

1.4 4. La valeur de τ sur $\mathbb{Q}$

1.4.1 4.1 Introduction

Soit $a \in \mathbb{Q}$. Posons $a = \frac{p}{q}$ où $p, q \in \mathbb{Z}$ et $p \neq 0$. Pour tout $k \geq 0$, posons $a_k = \ll 2^k a \gg$. Soit

$$f : x \mapsto 2x \bmod 1 = 2x - \lfloor 2x \rfloor$$

Considérons la suite $(b_k)_{k\geq 0}$ définie pour tout $k \in \mathbb{N}$ par $b_k = (2^k a) \bmod 1$. On a pour tout entier k, $b_{k+1} = f(b_k)$. Les b_k étant des rationnels compris entre 0 et 1 dont le dénominateur divise q, la suite (b_k) ne peut prendre qu'un nombre fini de valeurs. Étant définie par une relation de récurrence, cette suite est ultimement périodique. Comme $a_k = \ll b_k \gg$, la suite $(a_k)_{k\geq 0}$ est elle aussi périodique à partir d'un certain rang.

Pour fixer les idées, supposons que cette suite est périodique de période $T > 0$ à partir du rang N. Réécrivons $\tau(x)$.

$$\begin{array}{rcl}
\tau(x) & = & \sum_{k=0}^{\infty} \frac{a_k}{2^k} \\
& = & \sum_{k=0}^{N-1} \frac{a_k}{2^k} + \sum_{k=N}^{\infty} \frac{a_k}{2^k} \\
& = & \sum_{k=0}^{N-1} \frac{a_k}{2^k} + \sum_{j=0}^{T-1} \sum_{k=0}^{\infty} \frac{a_{N+kT+j}}{2^{N+kT+j}} \\
& = & \sum_{k=0}^{N-1} \frac{a_k}{2^k} + \sum_{j=0}^{T-1} \frac{a_{N+j}}{2^{N+j}} \sum_{k=0}^{\infty} \frac{1}{2^{kT}} \\
& = & \sum_{k=0}^{N-1} \frac{a_k}{2^k} + \frac{2^T-1}{2^T} \sum_{j=0}^{T-1} \frac{a_{N+j}}{2^{N+j}}
\end{array}$$

Connaissant T et N, il est ainsi possible d'écrire $\tau(x)$ comme une somme finie, et donc d'obtenir la valeur exacte de ce réel. Au passage, nous venons de prouver :

Proposition. Pour tout $x \in \mathbb{Q}$, $\tau(x) \in \mathbb{Q}$.

1.4.2 4.2 L'algorithme de Floyd

Soit $f : E \to E$ une fonction où E est un ensemble fini. Soit $a \in E$. La suite définie par $x_0 = a$ et $x_{n+1} = f(x_n)$ est ultimement périodique : il existe un entier n_0 et un entier $T > 0$ tels que pour tout $n \geq n_0$ on ait $x_{n+T} = x_n$. Plus précisément, les valeurs $a, f(a), f(f(a))$, etc. sont distinctes, puis on tombe à l'indice n_0 sur un cycle. Si l'on relie graphiquement les valeurs successives de la suite, on ne forme pas un rond (la suite ne boucle pas sur son premier terme) mais un ρ.

L'algorithme de Floyd permet de trouver une période ultime de f (la longueur de la boucle d'un ρ) à partir de la valeur a en espace constant et dans un temps de l'ordre de cette période. On initialise une variable x à la valeur $f(a)$ et une variable y à la valeur $f(f(a))$. Puis, tant que $x \neq y$, on remplace x par $f(x)$ et y par $f(f(y))$. Le point y se déplace deux fois plus vite que x et finira par « rattraper » x le long d'un cycle.

On montre facilement que le nombre d'itérations effectuées par l'algorithme est inférieur à la longueur du ρ.

La fonction `floyd` ci-dessous prend en paramètres la fonction f et un élément a de l'ensemble de départ (et d'arrivée !) de f. Elle renvoie un triplet (x, N, T) où $x = x_N$ est un point cyclique de la suite (x_n) (un point qui se trouve sur la boucle du ρ), et T est une période ultime de (x_n). L'entier T n'est pas nécessairement la plus petite période possible.

[18]:
```
def floyd(f, a):
    x = f(a)
    y = f(f(a))
    N = 1
    q = 2
    while y != x:
        x = f(x)
        y = f(f(y))
        N = N + 1
        q = q + 2
    return (x, N, q - N)
```

Prenons un exemple. Soit $f : \mathbb{Z}/10403\mathbb{Z} \to \mathbb{Z}/10403\mathbb{Z}$ définie par $f(x) = x^2 + 1$.

[19]:
```
def f(x):
    return (x * x + 1) % 10403
```

[20]:
```
floyd(f, 6)
```

[20]:
```
(9288, 126, 126)
```

1.4.3 4.3 Application à la fonction de Takagi

En reprenant les notations du 4.1, nous allons appliquer l'algorithme de Floyd à la suite récurrente $(b_k)_{k\geq 0}$. Il nous faut donc légèrement modifier le test d'arrêt « $y \neq x$ » en « $\ll y \gg \neq \ll x \gg$ ».

[21]:
```
def mod1(x):
    p = x.numerator
    q = x.denominator
    return Fraction(p % q, q)

def f_takagi(x): return mod1(2 * x)
```

[22]:
```
def floyd_takagi(a):
    x = f_takagi(a)
    y = f_takagi(f_takagi(a))
    N = 1
    q = 2
    while delta(y) != delta(x):
        x = f_takagi(x)
        y = f_takagi(f_takagi(y))
        N = N + 1
        q = q + 2
    return (x, N, q - N)
```

La fonction `tau_exact` prend un rationnel a en paramètre et renvoie la valeur exacte de $\tau(a)$. Elle utilise la formule vue en 4.1.

[23]:
```
def tau_exact(a):
    (x, N, T) = floyd_takagi(a)
    s1 = 0
    for k in range(N):
        s1 = s1 + Fraction(delta(a), 2 ** k)
        a = f_takagi(a)
    s2 = 0
    for j in range(T):
        s2 = s2 + Fraction(delta(a), 2 ** (N + j))
        a = f_takagi(a)
    return s1 + Fraction(2 ** T, 2 ** T - 1) * s2
```

Faisons quelques tests.

[24]:
```
print(tau_exact(Fraction(1, 3)))
```

```
2/3
```

[25]:
```
print(tau_exact(Fraction(1, 5)))
```

```
8/15
```

[26]:
```
for k in range(18):
    a = Fraction(k, 17)
    print('%10s%30s' % (a, tau_exact(a)))
```

```
         0                             0
      1/17                        64/255
      2/17                        98/255
      3/17                         26/51
      4/17                          8/15
      5/17                         32/51
      6/17                           2/3
      7/17                           2/3
      8/17                       152/255
      9/17                       152/255
     10/17                           2/3
     11/17                           2/3
     12/17                         32/51
     13/17                          8/15
     14/17                         26/51
     15/17                        98/255
     16/17                        64/255
         1                             0
```

Bien évidemment, il nous faut être conscients que les fractions renvoyées par `tau_exact` peuvent avoir des numérateurs et des dénominateurs énormes …

[27]:
```
a = Fraction(12345, 678910)
y = tau_exact(a)
print(y)
```

```
74726777827878507043263301074069896278358782319027492110503067189658748645015392
15770578855408838551201931800354524225454089781486716616366522876295352180229007
98360899399829134363098562401578672452511538334133571919321225331936105989046123
90398000389616853186154717183584572934463191415510733402075072152215471144602639
34982763659595597108274720068970066323732426602055354412674473858879075418589228
78289444178718592901876212504371755151200179340055183540299368086880980491758417
17996766204368820964479564482418226014981114510874391363004614372261778412004252
34829888664189358590163158734863584668365240112587382194958018382689754590944656
72951573299652072700961581457837555209557453733622813216154340530287945969076761
24742131066056385468864109865839164350393766665889602398591505642481126845990388
75637051778014659757493388123266131157990871675463657251760657561136517766317556
09251442749152969154123199304284302113047176948612318375383467446308898704679813
69876327428740955901353992276222235225046944260455211957600446388113173472781299
14846722941611202968146649477563977837870166149637832754507172790046753901673612
99653743484213399162928949973896118656868454226304664496360457226068881880813569
48379935800286695103412864276113950526313713773732199838808197312614077562978065
93014135203687043689625603049613178246074760831000957472600527653690279842639259
42261176265743115137587129744468356534934516124305316200181760047985569006215187
```

0628399507833356592217683263175884067148556428621274151852897177927860637571202928264755656580737297753496733320833513410091456161975227469325125120961816654254708585277264652303756350927838193816117516236279113285389472817953474316682836329769002435386001775126190821885857977756182023176322874123518086961866780854786049076369597880842843779170415390250630172491188522806999258544674743970902361394292216271126240144577538484676001086042563027170422869755070705161249938863471938369604356790062141637120018957509808489659730024550373666096894567881882532179681424748858340735845931798230908365729122210265/669087265354629282517162599626849549315332043873054808596304692958379238194462425283883261438044222725299364297888248939904141997344146649260097901759929210461509865733534151171073407332314309510629199216416396356386112751547275279917767605288286033706068378808128531540589707960232058920498177738661333791389117541680828908885872586105152192725165893616915497945156027961232252583020990922684390058226994643087409257533346084571741886251684564037886767198540789197653598842660869017229840885526983605749890812954080084754938927561315004303302915808495928824612633131076662736918819498443960894754909630134297876357001189399577763891367079009462965064979172942710426641527496329727084692469050076778669791753708998966074240679234743439807446997301325584920760887944852311629339058060403558712056580158164074945923352856231303778284112110094620130218759762448000043457218200533856233831263388123102441993818555412959827247594349105117271512557321077412693152698106496707999656809614479967502536621145418026871843917377375854430493118792458817576643245422923685813319126126221564909553365961098102260933062353709552170981756199469154170081871674693649564226621660497876442872952363404148074844353901186481892329728476421357572989291197732955728846072960610670232531800282368740247228641803094571892200787251103327939994827815383178726031856939296369838075351605701769295136349666492120482465956241168699336130473104218550478801677515556552238508588540563679999312973734361221244180381191483877747646037033511386710653796828260158425522189678878718916158307641965197334142915958197211886470145570723345548493041404639878473352638977299467811078150953618279923850046157328956199345561148858291763173847972850028515670941026676456666454614580418619844507846901477243005717725337546990392261911062284336351453971491745585542122499302343115836946480698184994295709963910283102238472515932109342585995642463498039594125360905659665789351359785474983176435515668985721106313492892048987327771481839982057707740004862337153865398047472444875488954383428438025.

1.4.4 4.4 Quelques vérifications

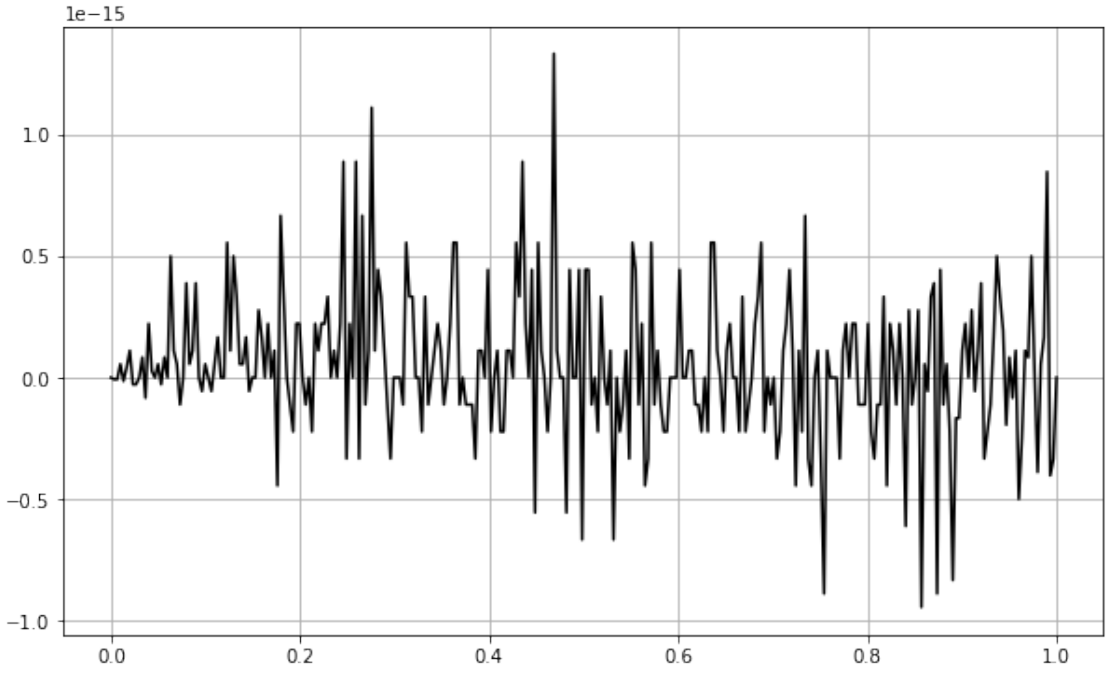
En imaginant que nous ne sommes pas encore convaincus, comment nous convaicre de la correction de la fonction `tau_exact` ? Pour tout rationnel a, nous disposons maintenant de deux fonctions

- La fonction `tau(n, a)` qui renvoie `tau_n(a)`, une valeur approchée de `tau(a)`.
- La fonction `tau_exact(a)`.

La fonction `test_tau` ci-dessous prend en paramètre un entier n. Pour $k = 0, \ldots, n$ elle évalue $\tau_n(\frac{k}{n})$ et calcule également la valeur exacte de $\tau(\frac{k}{n})$. La fonction affiche ensuite les différences de ces nombres, qui devraient être nulles aux erreurs d'arrondi près.

[28]:
```
def test_tau(n):
    fs = [Fraction(k, n) for k in range(n + 1)]
    ts = [tau_exact(a) for a in fs]
    xs = [float(a) for a in fs]
    ys = [float(tau_exact(a)) - tau(60, float(a)) for a in fs]
    plt.plot(xs, ys, 'k')
    plt.grid()
```

[29]:
```
test_tau(301)
```



Les différences sont de l'ordre de 10^{-15}, et donc conformes à nos espérances.

1.5 Références

- J. Lagarias, The Takagi Function and Its Properties (2012)
- J. Lagarias, The Takagi Function (2011)
- P. Allaart, K. Kawamura, The Takagi Function: a Survey (2011)
- A. Shidfar, K. Sabetfakhri - On the Continuity of Van der Waerden's Function in the Hölder Sense (2015)
- Y. Baba, On Maxima of Takagi-Van der Waerden Functions (1984)