Spectre lumineux (Denise Vella-Chemla, 5.3.2019)

On utilise des pages très didactiques ici :

https://www.courspython.com/fft-introduction.html

ou là

http://www.f-legrand.fr/scidoc/docmml/numerique/tfd/tfdimage/tfdimage.html

pour essayer de comprendre un peu la notion de transformée de Fourier puis pour dessiner le spectre de la fonction somme de sommes de cosinus, qui coïncide avec l'identité pour les nombres premiers et pour eux-seuls, fonction qu'on a définie ainsi (par exemple pour connaître les nombres premiers jusqu'à t) :

$$signal(t) = \sum_{b=2}^{t} \sum_{o=1}^{b} \cos \frac{2\pi\ t\ o}{b}$$

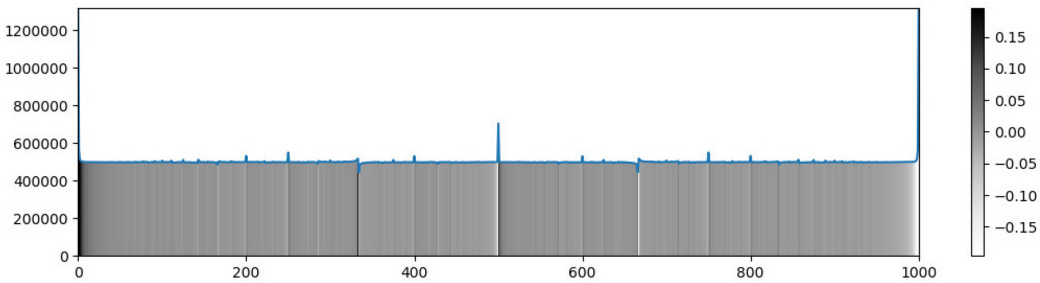
Voici le programme en python qui permet de calculer le spectre de la fonction *signal* [1] :

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

dt = 0.1
nmax = 1000
t = range(nmax)
signal = [np.sum([sum([np.cos(2*np.pi*n*o/b)
      for o in range(1,b+1)]) for b in range(2,nmax)])
      for n in range(nmax)]
print(signal)
plt.subplot(211)
plt.plot(t,signal)
fourier = np.fft.fft(signal)
freq = np.fft.fftfreq(nmax, d=dt)
plt.subplot(212)
k = np.arange(nmax)
print(str(k[-1])+"␣␣"+str(k[1])+"␣␣"+str(k[0]))
x = np.append(k, k[-1]+k[1]-k[0])
z = np.append(fourier, fourier[0])
y = np.abs(z)
plt.plot(x,y)
X = np.array([x,x])
Z = np.array([z,z])
y0 = np.zeros(len(x))
print(y0)
Y = np.array([y0,y])
C = np.angle(Z)
plt.pcolormesh(X, Y, C, cmap=Greys,
               vmin=-np.pi/16.0, vmax=np.pi/16.0)
plt.colorbar()
plt.show()
```

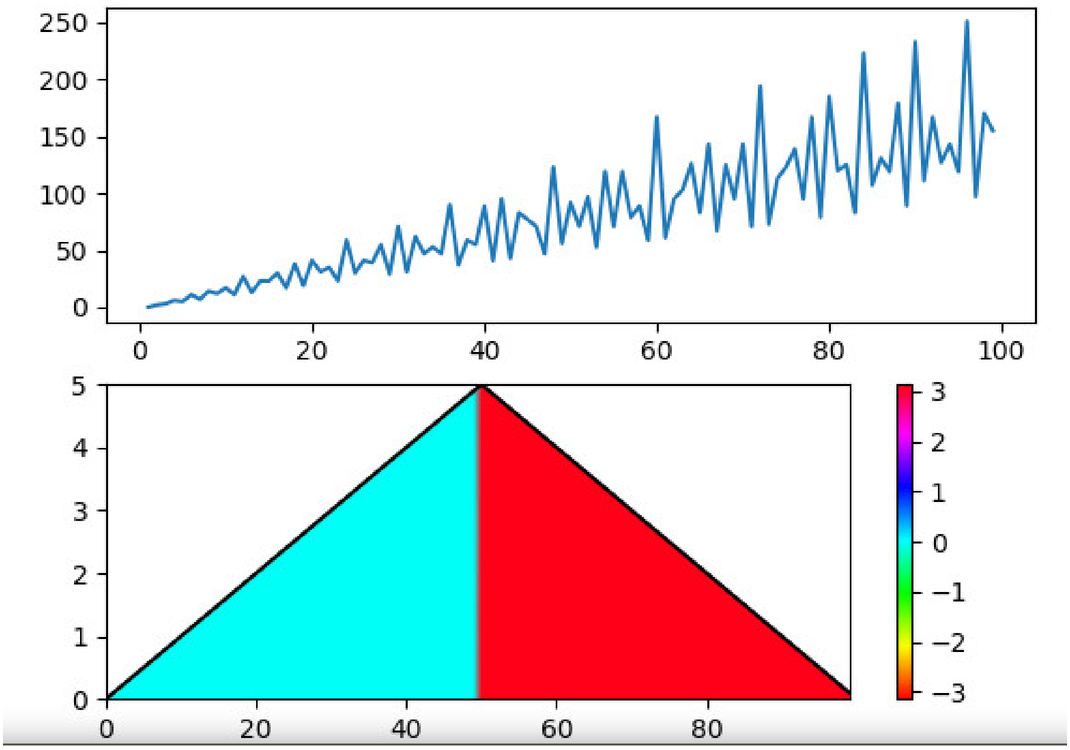
1. jusqu'à 1000.

Voici le spectre en question :



Les raies de ce spectre apparaissent très clairement aux positions 500, 333, 666, 200, 400, 600, 800, etc., c'est à dire aux positions $\frac{1}{k}$ pour k compris entre 2 et $p-1$ avec p premier si l'on considère 1000 comme l'unité.

On peut également calculer le spectre en fréquence de la fonction en question :



Notre problème, récurrent, est qu'on ne sait toujours pas si, en "découvrant" cette propriété du spectre, on a découvert un diamant ou de la simple verroterie mathématique.