

## 1. Introduction et traitement du signal

Le document intitulé “*Spectre lumineux*” <https://denisevellachemla.eu/spectrenb.pdf> (mars 2019), introduit une approche remarquable de traitement du signal appliquée à l’arithmétique élémentaire. L’auteur y définit une fonction oscillatoire combinant des sommes finies de cosinus :

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

## 2. Propriétés harmoniques et résonances

L’évaluation analytique de la somme interne révèle une fonction indicatrice de divisibilité. En effet, la somme des racines de l’unité orthogonales redonne une distribution de Dirac discrète. La résonance harmonique maximale (la sommation constructive des ondes) se produit précisément lorsque  $t$  évite les harmoniques fractionnaires des nombres composés. Le signal agit comme un filtre spectral mettant en relief la distribution des nombres premiers sous forme de pics d’amplitudes spécifiques dans le domaine fréquentiel, analysés par Transformée de Fourier Discrète (TFD/FFT).

## 3. Exploration locale

Le programme Python ci-dessous associé reproduit ce calcul de spectre fréquentiel pour analyser l’empreinte digitale harmonique des nombres premiers.

**Programme python**

```
# -*- coding: utf-8 -*-
import numpy as np

def calculer_signal_premier(nmax=100):
    print(f"—— Piste 7 : Signal harmonique pour nmax = {nmax} ——")
    signal = []
    for n in range(nmax):
        val_t = 0
        for b in range(2, nmax):
            val_t += sum(np.cos(2 * np.pi * n * o / b) for o in range(1, b + 1))
        signal.append(val_t)
    print("Premieres valeurs du signal arithmetique :")
    print([round(x, 2) for x in signal[:15]])

if __name__ == "__main__":
    calculer_signal_premier(40)
```

— Piste 7 : Signal harmonique pour  $n_{\max} = 40$  —

Premières valeurs du signal arithmétique :

```
[np.float64(779.0), np.float64(-0.0), np.float64(2.0), np.float64(3.0),  
 np.float64(6.0), np.float64(5.0), np.float64(11.0), np.float64(7.0),  
 np.float64(14.0), np.float64(12.0), np.float64(17.0), np.float64(11.0),  
 np.float64(27.0), np.float64(13.0), np.float64(23.0)]
```