## Reprise des produits de sinus (ou produits de restes), Denise Vella-Chemla, janvier 2025

On revient au calcul des produits de sinus pour trouver les décomposants de Goldbach d'un nombre pair $n$ : un décomposant de Goldbach de $n$ maximise la fonction $P(x)$ ci-dessous, valeur absolue d'un produit de sinus, dans presque tous les cas. On note $\mathcal{M}$ l'ensemble $\{2\} \cup \{2k+1 \mid k \in [1, .., \lfloor \sqrt{n}/2 \rfloor]\}$.

$$P(x) = \left| \prod_{p \in \mathcal{M}} \sin\left(\frac{x\pi}{p}\right) \sin\left(\frac{(n-x)\pi}{p}\right) \right|$$

On utilise le programme ci-dessous :

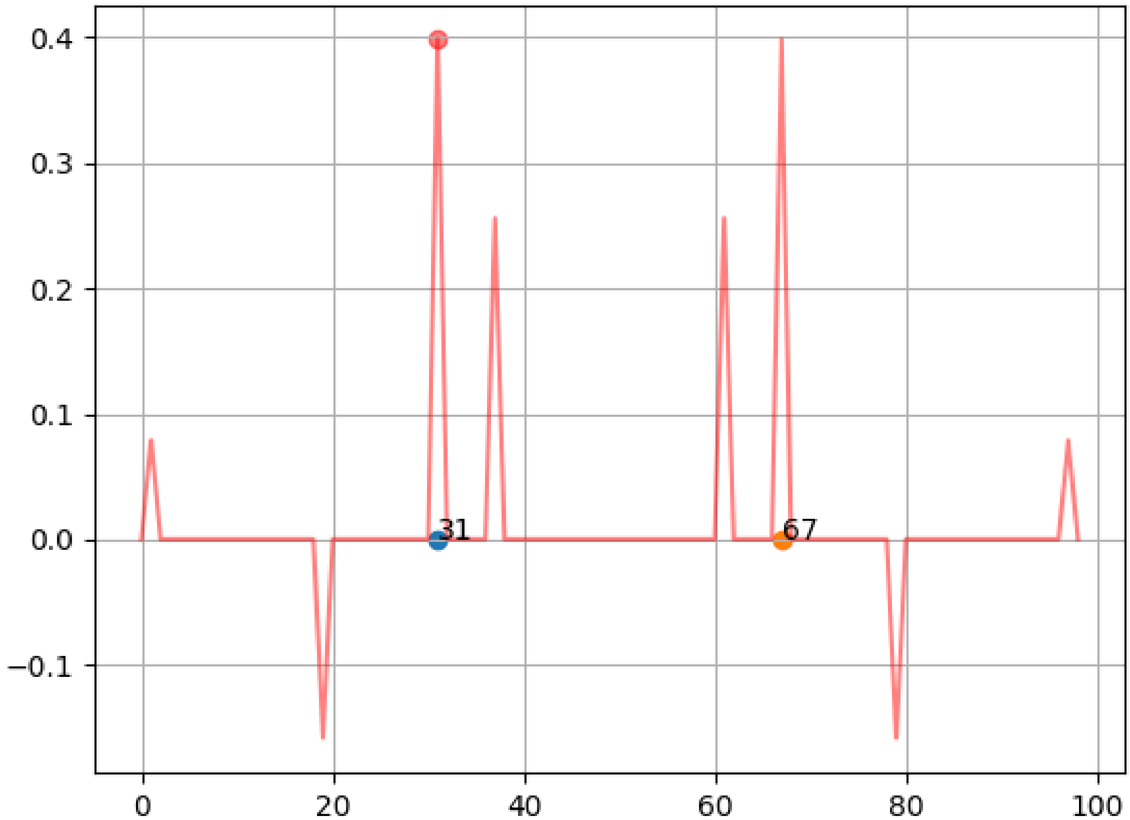
```
import numpy as np
from numpy import sin,pi
import math
from math import ceil,floor,sqrt
import matplotlib.pyplot as plt

for n in range(16,104,2):
     precision = n+1
     m = int(floor(sqrt(n)))
     x = np.linspace(0, n, precision)
     y = np.ones_like(x)
     y *= sin(x*pi/2)*sin((n-x)*pi/2)
     for p in range(3,m,2):
          y *= sin(x*pi/p)*sin((n-x)*pi/p)
     xm = np.argmax(np.abs(y))
     print(n,' --> ',ceil(x[xm]*n/precision),'+',n-ceil(x[xm]*n/precision))
     plt.plot(x, y, color='red', alpha=0.5)
     plt.scatter(x[xm], y[xm], color='red', alpha=0.5)
     plt.scatter(x[xm],0)
     plt.annotate(ceil(xm*n/precision),xy=(x[xm],0))
     plt.scatter(x[n-xm],0)
     plt.annotate(n-ceil(xm*n/precision),xy=(x[n-xm],0))
     plt.grid(True)
     plt.show()
     plt.savefig('fig'+str(n)+'.png')
     plt.clf()
```

Les graphiques de ce programme sont stockés dans un fichier à l'adresse
https://denisevellachemla.eu/toutes-figures-prodsin.pdf.

Comme on peut le constater dans le fichier, il y a quelques petites exceptions pour les nombres 24, 26, 32, 38, 48, 62, certains sinus en ayant "mangé" d'autres.

On fournit ci-dessous le graphe de $P$, pour le nombre pair $n = 98$.




On a utilisé ci-dessus des fonctions discrétisées, dont le comportement présente des "diracs" ; si on utilise à la place un pas de discrétisation inférieur à un (en affectant dans le programme la variable precision à 1000, par exemple), les graphiques deviennent, en conséquence, ceux de fonctions lisses mais le gros problème d'un tel choix est que de nombreuses décompositions d'un nombre pair $n$ comme somme de deux nombres pairs, qui ne sont évidemment pas des décompositions de Goldbach de $n$, maximisent alors la fonction produit de sinus.

On peut remplacer les produits de sinus par des produits de restes modulaires.

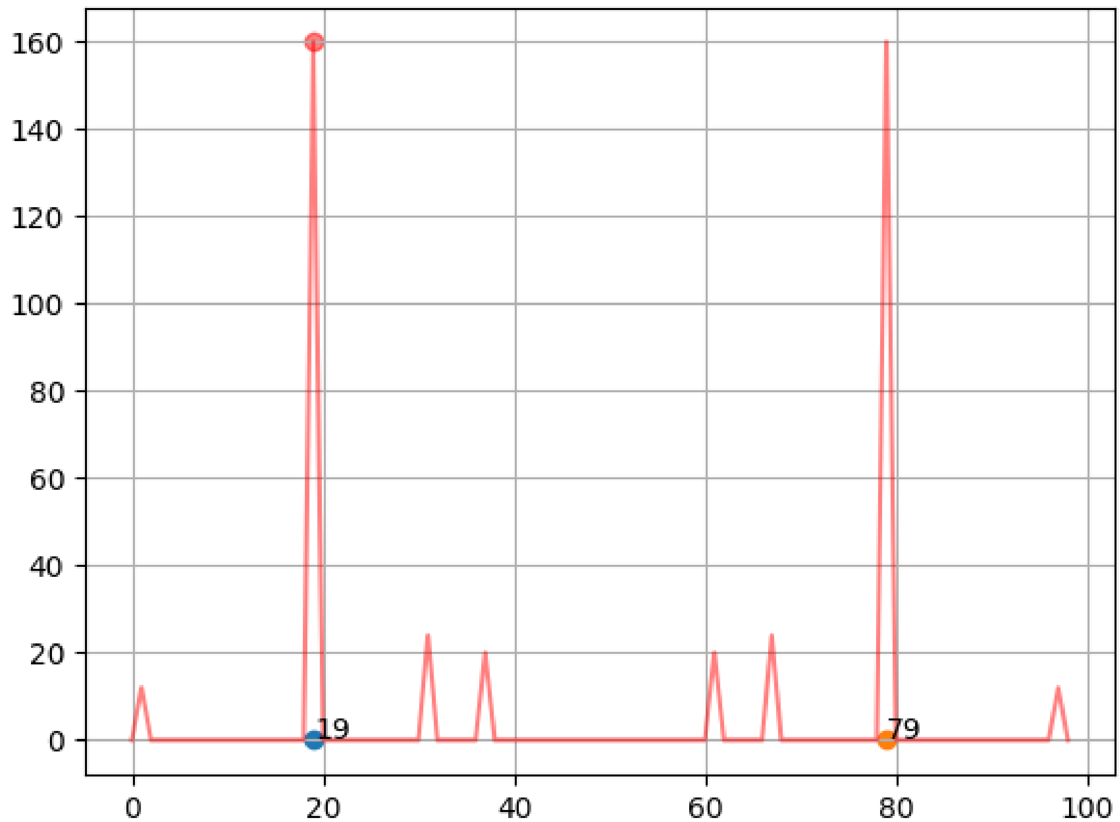
La fonction $Q$, valeur absolue d'un produit de restes, qui doit être maximisée devient :

$$Q(x) = \Big| \prod_{p \in \mathcal{M}} (x \bmod p)((n-x) \bmod p) \Big|$$

Le résultat est assez similaire, en utilisant le programme dont on a remplacé quelques lignes :

```
[...]
     y = np.ones_like(x)
     y *= (x%2)*((n-x)%2)
     for p in range(3,m,2):
          y *= (x%p)*((n-x)%p)
     xm = np.argmax(np.abs(y))
[...]
```

On fournit ci-dessous le graphe de $Q$, obtenu pour le nombre pair $n = 98$ :




Les graphiques de ce second programme sont stockés dans un fichier à l'adresse
https://denisevellachemla.eu/toutes-figures-prodrestes.pdf.

Comme on peut le constater dans le fichier, il y a quelques petites exceptions pour les nombres 18, 20, 24, 26, 30 et 32, certains restes en ayant "mangé" d'autres.