

Formulaire résumé, Denise Vella-Chemla, 4 décembre 2024

1) Fonction qui s'annule sur les nombres premiers

$$f_1(x) = \sum_{b=2}^{x-1} \sum_{o=1}^b \cos\left(\frac{2\pi x o}{b}\right).$$

J'ai trouvé cette formule en juillet 2014 lien 1.

Victor Varyn a démontré qu'elle s'annule bien sur les nombres premiers et eux seuls en juillet 2017 lien 2.

On peut utiliser à la place des sommes de cosinus des produits de sinus lien 3 (mai 2009).

J'ai retrouvé "mon" produit de sinus dans un article de 1951 de Redheffer lien 4 (je l'ai retrouvé en septembre 2021).

2) Fonction qui s'annule sur les zéros de ζ

La fonction de van der Pol ¹

$$f_2(t) = \left| \left(\int_0^t -\exp\left(-\frac{x}{2}\right) (\exp(x) \bmod 1) \cos(-x t) dx \right)^2 + \left(\int_0^t -\exp\left(-\frac{x}{2}\right) (\exp(x) \bmod 1) \sin(-x t) dx \right)^2 \right|.$$

3) Fonction qui fournit les valeurs des parties imaginaires des zéros non triviaux de ζ

La fonction de França, LeClair et Mussardo (W est la fonction de Lambert, i.e. la réciproque de la fonction : $W^{\leftarrow}(z) = z e^z$, avec z complexe.)

$$f_3(t) = \frac{2\pi(t - \frac{11}{8})}{W\left(\frac{t - \frac{11}{8}}{e}\right)}$$

Pour des tests de programmation de cette fonction, se reporter à lien 10.

4) Fonction qui compte les nombres premiers

Dans Around Wilson's Theorem, d'A. Connes, on trouve par exemple

$$f_4(x) = \pi(x) = \left\lfloor \sum_{k=1}^x \sin^2\left(\frac{\pi\Gamma(k)}{2k}\right) \right\rfloor.$$

¹voir lien 5 , lien 6 , lien 7 , lien 8 et lien 9.

5) Fonctions qui comptent les zéros non triviaux de ζ sur la droite critique

a) Formule de Riemann ^{2 3}, à laquelle on ajoute $\frac{7}{8}$.

$$f_5(T) = \frac{T}{2\pi} \left(\log \left(\frac{T}{2\pi} - 1 \right) \right) + \frac{7}{8}.$$

b) Formule de Backlund qui compte les zéros non triviaux sur la droite critique

$$f_6(t) = \frac{1}{\pi} \arg \Gamma \left(\frac{1}{2} + \frac{t}{2}i \right) + 1 + \frac{1}{\pi} \arg \zeta \left(\frac{1}{2} + it \right)$$

Pour des tests de programmation de ces fonctions, se reporter à lien 10.

6) Opérateur dont le spectre serait l'ensemble des nombres premiers

J'ai cherché à programmer l'opérateur dont il est question à la page 7/47 (ou bien 417/457 de la publication originale) de l'article *Hecke algebras, Type III factors and Phase transitions with spontaneous symmetry breaking in number theory* de J.-B. Bost et A. Connes (lien 12), mais je n'ai pas compris / trouvé la définition de l'opérateur utilisé ⁴.

7) Opérateur dont le spectre serait l'ensemble des (parties imaginaires des) zéros de ζ

J'ai également cherché à programmer l'opérateur dont il est question dans l'article *The UV Prolate spectrum matches the zeros of zeta* de H. Moscovici et A. Connes mais je n'ai pas compris / trouvé la définition de l'opérateur utilisé ⁵.

8) Opérateur qui a un comportement particulier qui permet de distinguer les nombres premiers des nombres composés

Voir la matrice multi-circulante à multiples matrices de permutations sur sa diagonale lien 16 .

9) petite remarque en passant sur l'utilité des exponentielles et des logarithmes pour calculer "directement" les carrés, les cubes, les puissances

$$\exp(x)^{\ln(y)} = y^x.$$

²Voir lien 11, extrait de lien 17, et se reporter à la page de l'Institut Clay concernant le manuscrit de Riemann lien 18.

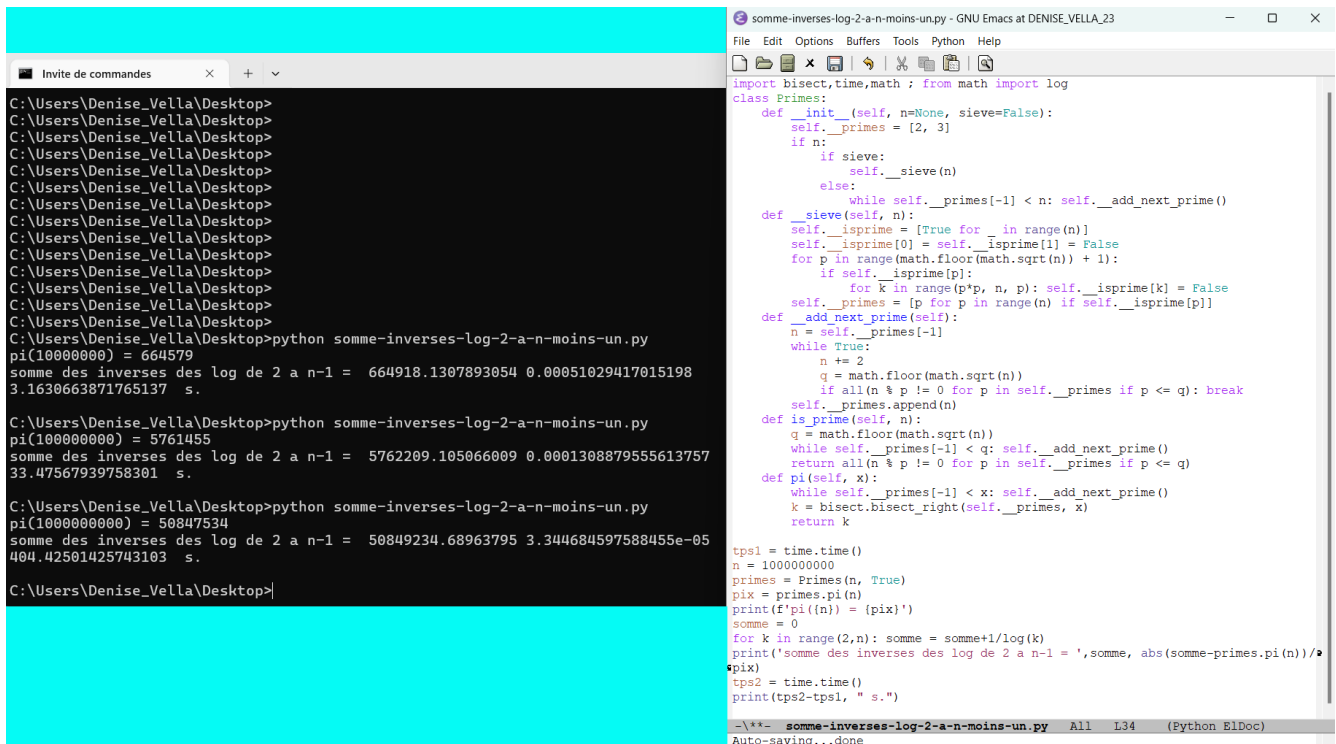
³On trouve parfois $\log \left(\frac{T}{2\pi} - 1 \right)$ abrégé en $\log \left(\frac{T}{2\pi e} \right)$.

⁴voir p. 6 de cette traduction lien 13 .

⁵voir lien 14 et traduction d'une version postée sur arxiv (et peut-être modifiée depuis) lien 15.

10) enfin comprendre un peu quel est le problème

La formule de Riemann pour calculer le nombre de nombres premiers inférieurs à un certain nombre utilise le logarithme intégral, qui est, pour le dire rapidement, une intégrale de l'inverse du logarithme. Quand on n'aime pas trop les réels et les intégrales, on remplace cette intégrale par une somme pour les entiers de 2 à $x - 1$ de l'inverse de leur logarithme (à la recherche de $\pi(x)$, le nombre de nombres premiers jusqu'à x), et on obtient sensiblement le même résultat qu'avec la fonction li . Quand on observe la différence entre cette somme des inverses des logarithmes des entiers (de 2 à $x - 1$) et le nombre de nombres premiers (jusqu'à x), on voit que cette différence entre eux deux devient de plus en plus petite, vraiment, mais il faudrait être sûr qu'elle ne va pas se remettre à grandir, loin, très loin, très très loin.



```
Invite de commandes
C:\Users\Denise_Vella\Desktop>
C:\Users\Denise_Vella\Desktop>
C:\Users\Denise_Vella\Desktop>
C:\Users\Denise_Vella\Desktop>
C:\Users\Denise_Vella\Desktop>
C:\Users\Denise_Vella\Desktop>
C:\Users\Denise_Vella\Desktop>
C:\Users\Denise_Vella\Desktop>
C:\Users\Denise_Vella\Desktop>
C:\Users\Denise_Vella\Desktop>
C:\Users\Denise_Vella\Desktop>
C:\Users\Denise_Vella\Desktop>
C:\Users\Denise_Vella\Desktop>
C:\Users\Denise_Vella\Desktop>
C:\Users\Denise_Vella\Desktop>
C:\Users\Denise_Vella\Desktop>
C:\Users\Denise_Vella\Desktop>
C:\Users\Denise_Vella\Desktop>python somme-inverses-log-2-a-n-moins-un.py
pi(1000000) = 664579
somme des inverses des log de 2 a n-1 = 664918.1307893054 0.00051029417015198
3.1630663871765137 s.

C:\Users\Denise_Vella\Desktop>python somme-inverses-log-2-a-n-moins-un.py
pi(10000000) = 5761455
somme des inverses des log de 2 a n-1 = 5762209.105066009 0.0001308879555613757
33.47567939758301 s.

C:\Users\Denise_Vella\Desktop>python somme-inverses-log-2-a-n-moins-un.py
pi(100000000) = 50847534
somme des inverses des log de 2 a n-1 = 50849234.68963795 3.344684597588455e-05
404.42501425743103 s.

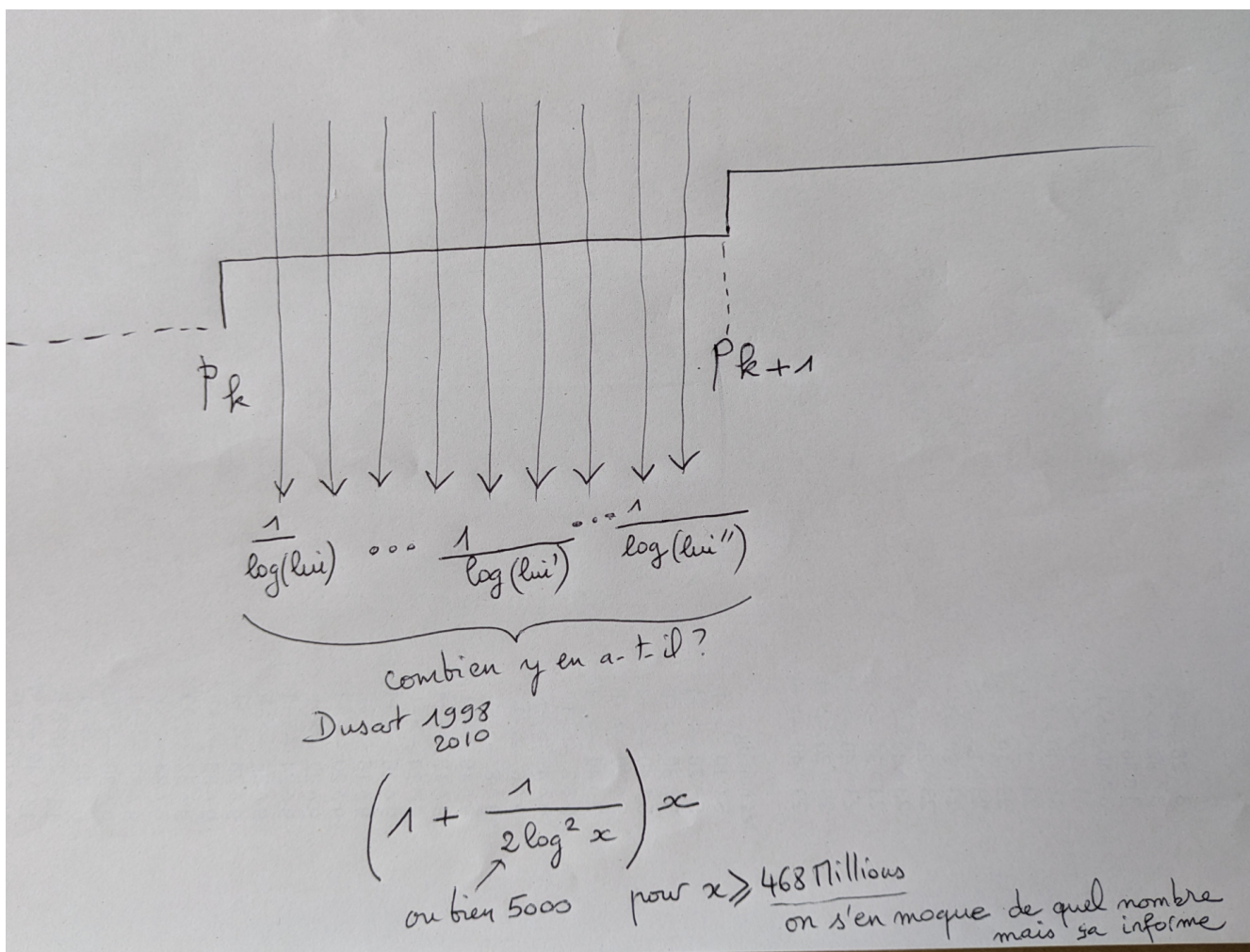
C:\Users\Denise_Vella\Desktop>

somme-inverses-log-2-a-n-moins-un.py - GNU Emacs at DENISE_VELLA_23
File Edit Options Buffers Tools Python Help
import bisect,time,math ; from math import log
class Primes:
    def __init__(self, n=None, sieve=False):
        self.__primes = [2, 3]
        if n:
            if sieve:
                self.__sieve(n)
            else:
                while self.__primes[-1] < n: self.__add_next_prime()
    def __sieve(self, n):
        self.__isprime = [True for _ in range(n)]
        self.__isprime[0] = self.__isprime[1] = False
        for p in range(math.floor(math.sqrt(n)) + 1):
            if self.__isprime[p]:
                for k in range(p*p, n, p): self.__isprime[k] = False
        self.__primes = [p for p in range(n) if self.__isprime[p]]
    def __add_next_prime(self):
        n = self.__primes[-1]
        while True:
            n += 2
            q = math.floor(math.sqrt(n))
            if all(n % p != 0 for p in self.__primes if p <= q): break
        self.__primes.append(n)
    def is_prime(self, n):
        q = math.floor(math.sqrt(n))
        while self.__primes[-1] < q: self.__add_next_prime()
        return all(n % p != 0 for p in self.__primes if p <= q)
    def pi(self, x):
        while self.__primes[-1] < x: self.__add_next_prime()
        k = bisect.bisect_right(self.__primes, x)
        return k

tps1 = time.time()
n = 1000000000
primes = Primes(n, True)
pix = primes.pi(n)
print(f'pi({n}) = {pix}')
somme = 0
for k in range(2,n): somme = somme+1/log(k)
print('somme des inverses des log de 2 a n-1 = ',somme, abs(somme-primes.pi(n))/
pix)
tps2 = time.time()
print(tps2-tps1, " s.")

-**- somme-inverses-log-2-a-n-moins-un.py All L34 (Python ElDoc)
Auto-saving...done
```

Tant qu'on n'avait pas compris de quoi il retournait, on se disait "c'est simple, il suffit de montrer qu'une somme d'inverses de logarithmes entre deux nombres premiers est toujours inférieure à 1" (puisque c'est de 1 que $\pi(x)$ augmente lors du passage d'un nombre premier au nombre premier suivant), en ayant à l'esprit le dessin ci-dessous.



Dusart en 1998 puis en 2010 ⁶ a démontré que le nombre premier suivant immédiatement x est au maximum à distance $\left(\frac{x}{2 \log^2 x}\right)$ de x .

Alors on remplace la somme cumulée des $\frac{1}{\log k}$ par le plus grand d'entre ces ratios sur l'intervalle (qui est le premier d'entre eux, puisque l'inverse du logarithme décroît), multiplié par ce nombre de nombres qu'a fourni Dusart, pour voir comment cette majoration de la somme cumulée se conduit, et pour voir si elle est très inférieure à 1, ce qui tranquilliserait.

Mais pas du tout : de 2 à 20, elle décroît de 0.001201112282828627621 à 0.00014878220705519027, mais ensuite elle croît, très très lentement, c'est sûr (elle vaut à peine 0.1 et quelques pour 1 421 822, 0.2 pour 3 401 984, etc., et n'a pas encore atteint 0.5 à 10 000 000), mais elle croît : ça coince donc.

⁶où le 2 de la formule devant le log au dénominateur est devenu 5000.