Probabilité d'obtenir une décomposition de Goldbach d'un nombre pair (Denise Vella-Chemla, août 2022)

1. Un exemple illustratif

Prenons l'exemple de la recherche des décomposants de Goldbach de l'entier pair $n = 98$.

$$S_{98} = \begin{cases} 98 \equiv 0 \pmod 2 \\ 98 \equiv 2 \pmod 3 \\ 98 \equiv 3 \pmod 5 \\ 98 \equiv 0 \pmod 7 \end{cases}$$

Appelons d_{98} un décomposant de Goldbach potentiel de $n = 98$. d_{98} peut être congru, hormis 0, à tout ce à quoi $n = 98$ n'est pas congru. Le signe $\vee$ dans le système ci-dessous est à lire comme un ou exclusif, son emploi étendu est à comprendre comme le fait de vérifier autant de systèmes de congruences que la combinatoire le permet.

$$S_{d_{98}} = \begin{cases} d_{98} \equiv 1 \pmod 2 \\ d_{98} \equiv 1 \pmod 3 \\ d_{98} \equiv 1 \vee 2 \vee 4 \pmod 5 \\ d_{98} \equiv 1 \vee 2 \vee 3 \vee 4 \vee 5 \vee 6 \pmod 7 \end{cases}$$

Remarque : on note que le fait de respecter le système de systèmes de congruences ci-dessus est une condition suffisante mais non nécessaire pour obtenir un décomposant de Goldbach de n. La démonstration de la validité de cette caractérisation des décomposants de Goldbach d'un nombre pair n qui sont supérieurs à la racine carrée de n est fournie en section 2.

Comme on le comprend aisément, les modules qui ne divisent pas n "éliminent davantage de classes de congruences" (au nombre de 2 par module premier inférieur à $\sqrt{n}$) que les modules qui divisent n. Plaçons-nous dans le pire des cas, où l'on élimine deux classes de congruences par module premier inférieur à $\sqrt{n}$, on trouve tout de même

$$\prod_{\substack{p \text{ premier} \\ 3 \leq p \leq \sqrt{n}}} (p-2)$$

classes de congruences *différentes* par l'application du théorème des restes chinois à chacun des systèmes de congruences combinatoirement trouvé (voir $S_{d_{98}}$ ci-dessus). Ces solutions sont inférieures à $D = \prod_{\substack{p \text{ premier} \\ 3 \leq p \leq \sqrt{n}}} p$.

Serait-il possible de "rater l'intervalle visé", i.e. que toutes les solutions soient supérieures à n, comprises entre n et D ? À la section 3, on verra que la probabilité d'obtenir au moins une solution inférieure à n tend très vite vers 1.

2. Caractérisation des décomposants de Goldbach de n supérieurs à $\sqrt{n}$[1]

Soit $n \in 2\mathbb{N} + 6$ un entier pair supérieur à 6.

[1] Leila Schneps a démontré que la caractérisation de certains décomposants de Goldbach proposée était valide.

Pour tout $p \in \mathbb{P}^*$ premier impair inférieur à $\sqrt{n}$ (*i.e.* $3 \leq p \leq \sqrt{n}$), on définit l'ensemble :

$$F_n(p) = \{m \in 2\mathbb{N}+1 : 3 \leq m \leq n/2,\ m \not\equiv 0\ [p],\ m \not\equiv n\ [p]\}$$

L'intersection des ensembles $F_n(p)$ pour tout p premier compris entre 3 et $\sqrt{n}$ est notée :

$$D_n = \bigcap_{\substack{p \in \mathbb{P} \\ 3 \leq p \leq \sqrt{n}}} F_n(p)$$

Nous allons montrer que D_n et son complémentaire $n - D_n$ ne contiennent que des nombres premiers.
Lemme 1 : Soit $m \in 2\mathbb{N}+1$ un entier impair. Si m n'est divisible par aucun nombre premier compris entre 3 et $\sqrt{m}$, alors il est premier.

Démonstration : Si m est composé, on a $m = pq$, où p est le plus petit nombre premier intervenant dans la factorisation de m en nombres premiers et où q est le produit de tous les autres facteurs. Puisque m est impair, $p \geq 3$, et puisque $q \geq p$ (q étant le produit d'entiers $\geq p$), $m = pq \geq pp = p^2$ et donc $\sqrt{m} \geq p$ (la fonction racine carrée étant croissante). On a ainsi montré que si m impair est composé, il est divisible par un premier compris entre 3 et $\sqrt{m}$. Le lemme s'obtient par contraposition. □

Lemme 2 : $D_n \subseteq \mathbb{P}$

Démonstration : Soit $m \in D_n$. Alors $m \in F_n(p)$ pour tout p premier compris entre 3 et $\sqrt{n}$. Par conséquent, m est impair et m n'est divisible par aucun nombre premier p compris entre 3 et $\sqrt{n}$ (puisque $m \not\equiv 0\ [p]$), et donc *a fortiori* par aucun premier compris entre 3 et $\sqrt{m}$ (car $m \leq n/2 \implies m \leq n \implies \sqrt{m} \leq \sqrt{n}$). D'après le lemme 1, m est donc premier. □

Lemme 3 : $n - D_n \subseteq \mathbb{P}$

Démonstration : Soit $m \in D_n$. Alors $m \in F_n(p)$ pour tout p premier compris entre 3 et $\sqrt{n}$. Par conséquent, $n-m$ est impair (car m est impair et n pair) et $n-m$ n'est divisible par aucun nombre premier p compris entre 3 et $\sqrt{n}$ (puisque $m \not\equiv n\ [p]$), et donc *a fortiori* par aucun premier compris entre 3 et $\sqrt{n-m}$ (car $n - m \leq n \implies \sqrt{n-m} \leq \sqrt{n}$). D'après le lemme 1, $n - m$ est donc premier. □

Les ensembles D_n ne contiennent que des décomposants de Goldbach de n.

Lemme 4 : Soit $n \in 2\mathbb{N}+6$. Si $D_n \neq \varnothing$, alors n vérifie la conjecture de Goldbach.

Démonstration : Si $D_n \neq \varnothing$, il contient un entier p nécessairement premier (d'après le lemme 1), tel que $q = n-p$ est également premier (d'après le lemme 2), et donc $n = p+q$ vérifie la conjecture de Goldbach.

3. Probabilité $P(n,k,p)$ de tirer un nombre inférieur ou égal à k, sans remise, quand on tire uniformément p entiers parmi les n premiers entiers.

La probabilité[2] $P(n,k,p)$ de tirer tirer un nombre inférieur ou égal à k, sans remise, quand on tire uniformément p entiers parmi les n premiers entiers se calcule par la formule suivante :

$$P = \frac{k}{n} + \frac{n-k}{n}\left(\frac{k}{n-1} + \frac{n-k-1}{n-1}\left(\frac{k}{n-2} + \frac{n-k-2}{n-2}\left(\dots\left(\frac{k}{n-p+1}\right)\dots\right)\right)\right)$$

Le premier terme de la somme correspond au fait de trouver un nombre inférieur à k dès le premier tirage. Le deuxième terme de la somme correspond au fait d'avoir tiré un nombre supérieur à k lors du premier tirage, de ne pas avoir la possibilité de tirer à nouveau ce nombre, et de tenter sa chance sur les nombres restant, la probabilité restant uniforme sur les nombres restant, etc.

On calcule cette probabilité pour

$$p = \prod_{\substack{x \text{ premier} \\ 3 \leqslant x \leqslant \sqrt{k}}} (x-2)$$

et

$$n = \prod_{\substack{x \text{ premier} \\ 3 \leqslant x \leqslant \sqrt{k}}} x.$$

Le programme python utilisé est le suivant :

```python
import math

def P(n, k, p):
    assert(1 <= p and p <= n and k <= n-p)
    s, t = 0, 1
    for i in range(p):
        s += t*(k/(n-i))
        t *= (n-k-i)/(n-i)
    return s

for n, k, p in [(30, 26, 3),
                (210, 50, 15),
                (2310, 122, 135),
                (30030, 170, 1485),
                (510510, 290, 22275),
                (9699690, 362, 378675),
                (223092870, 530 , 7952175),
                (6469693230, 842, 214708725),
                (200560490130, 962, 6226553025)]:
    print(f'n = {n}, k = {k}, p = {p} : P_n(k,p) = {P(n, k, p)}')
```

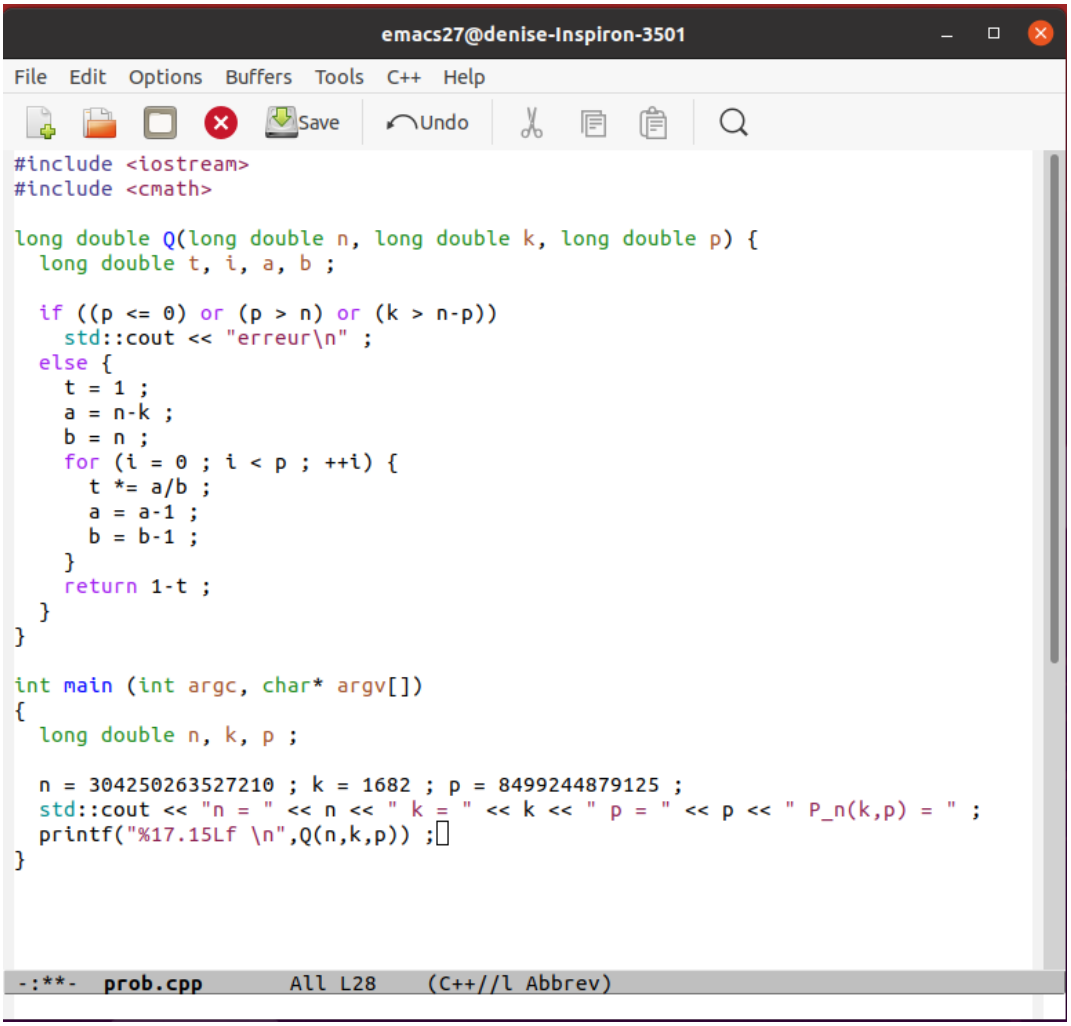
```
n = 30, k = 26, p = 3 : P_n(k,p) = 0.9990147783251231
n = 210, k = 50, p = 15 : P_n(k,p) = 0.9856514594832753
n = 2310, k = 122, p = 135 : P_n(k,p) = 0.9994752040784769
n = 30030, k = 170, p = 1485 : P_n(k,p) = 0.999824267526177
n = 510510, k = 290, p = 22275 : P_n(k,p) = 0.9999976037996607
n = 9699690, k = 362, p = 378675 : P_n(k,p) = 0.9999994514468453
n = 223092870, k = 530, p = 7952175 : P_n(k,p) = 0.9999999955788792
n = 6469693230, k = 842, p = 214708725 : P_n(k,p) = 0.9999999997119475
n = 200560490130, k = 962, p = 6226553025 : P_n(k,p) = 0.9999999921336346
```

[2] Merci Jacques.

Fournissons ses résultats dans le tableau ci-dessous :

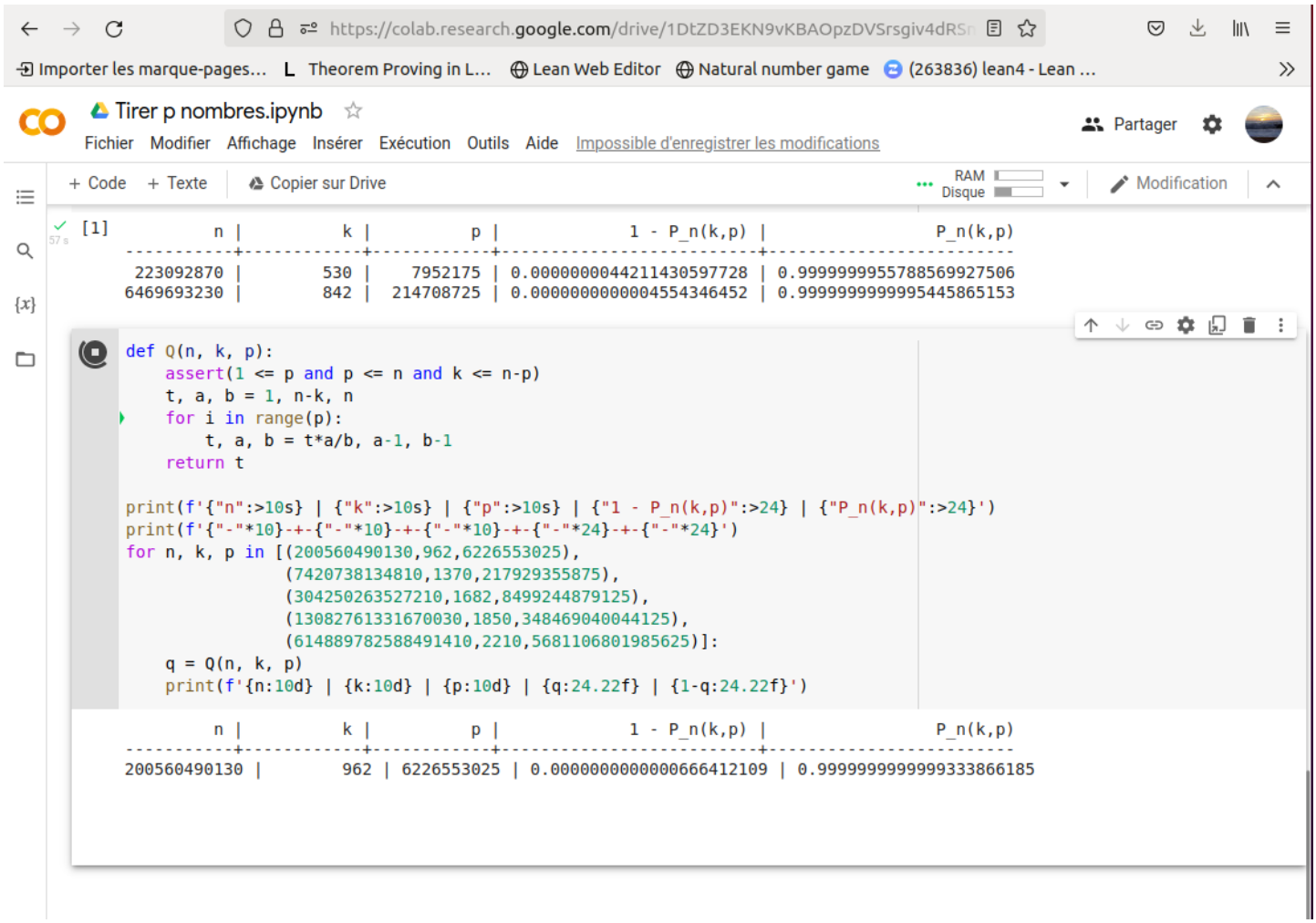
k	n	k^2+1	p	$P(n,k,p)$
5	30	26	3	0.9990147783251231
7	210	50	15	0.9856514594832753
11	2310	122	135	0.9994752040784769
13	30030	170	1485	0.999824267526177
17	510510	290	22275	0.9999976037996607
19	9699690	362	378675	0.9999994514468453
23	223092870	530	7952175	0.9999999955788792
29	6469693230	842	214708725	0.9999999997119475
31	200560490130	962	6226553025	0.9999999921336346

Pour confirmer les résultats du programme, on utilise une fonction qui calcule la probabilité des événements complémentaires, i.e. la probabilité qu'au cours des p tirages sans remise réalisés selon la loi uniforme discrète dans l'intervalle $1..n$, tous les entiers tirés soient strictement supérieurs à k ; on programme aussi ces fonctions en C++ pour comparer les résultats et essayer de gagner en rapidité. Malheureusement, la taille des nombres et donc le temps d'exécution nécessaire oblige à se contenter des probabilités jusqu'au nombre premier 37. Un compilateur C++ écrit une probabilité de 1 à partir du nombre premier 37.



```python
def Q(n, p, k):
    assert(1 <= p and p <= n and k <= n-p)
    t, a, b = 1, n-k, n
    for i in range(p):
        t, a, b = t*a/b, a-1, b-1
    return t

print(f'{"n":>10s} | {"p":>10s} | {"k":>10s} | {"1 - P(n,p,k)":>24} | {"P(n,p,k)":>24}')
print(f'{"-"*10}-+-{"-"*10}-+-{"-"*10}-+-{"-"*24}-+-{"-"*24}')
for n, p, k in [(30, 3, 26), (210, 15, 50), (2310, 135, 122), (30030, 1485, 170), (510510, 22275, 290), (9699690, 378675, 362)]:
    q = Q(n, p, k)
    print(f'{n:10d} | {p:10d} | {k:10d} | {q:24.22f} | {1-q:24.22f}')
```



```
denise@denise-Inspiron-3501:~/Bureau$ ./probasansdeux.exe
n = 30 k = 26 p = 3 P_n(k,p) = 0.999014778325123
n = 210 k = 50 p = 15 P_n(k,p) = 0.985651459483275
n = 210 k = 66 p = 15 P_n(k,p) = 0.997267239090670
n = 210 k = 80 p = 15 P_n(k,p) = 0.999458511995887
n = 210 k = 100 p = 15 P_n(k,p) = 0.999962342308524
n = 2310 k = 122 p = 135 P_n(k,p) = 0.999475204078478
n = 2310 k = 140 p = 135 P_n(k,p) = 0.999833990944990
n = 2310 k = 160 p = 135 P_n(k,p) = 0.999954324106302
n = 30030 k = 170 p = 1485 P_n(k,p) = 0.999824267526178
n = 30030 k = 220 p = 1485 P_n(k,p) = 0.999986315722772
n = 30030 k = 260 p = 1485 P_n(k,p) = 0.999998230122287
n = 510510 k = 290 p = 22275 P_n(k,p) = 0.999997603799665
n = 9.69969e+06 k = 362 p = 378675 P_n(k,p) = 0.999999451446880
n = 2.23093e+08 k = 530 p = 7.95218e+06 P_n(k,p) = 0.999999995578857
n = 6.46969e+09 k = 842 p = 2.14709e+08 P_n(k,p) = 0.999999999999545
```

```
denise@denise-Inspiron-3501:~/Bureau$ ./probasansdeux.exe
n = 7.42074e+12 k = 1370 p = 2.17929e+11 P_n(k,p) = 1.000000000000000
```

```
denise@denise-Inspiron-3501:~/Bureau$ ./probasansdeux.exe
n = 2.0056e+11 k = 962 p = 6.22655e+09 P_n(k,p) = 0.999999999999933
```

```
denise@denise-Inspiron-3501:~/Bureau$ ./probasansdeux.exe
n = 7.42074e+12 k = 1370 p = 2.17929e+11 P_n(k,p) = 1.000000000000000
```