Découverte d'une loi tout extraordinaire des nombres pairs par rapport à l'un de leur décomposant de Goldbach

Denise Vella-Chemla

Février 2023

On présente ici un résultat expérimental, qu'on a obtenu par hasard : on s'interrogeait sur la possibilité d'une relation entre tout décomposant de Goldbach d'un nombre pair et la moitié de ce nombre pair.

On n'a pas mis au jour une telle relation pour tous les décomposants de Goldbach, mais pour au moins l'un d'entre eux, et ce jusqu'à 10^9.

Pour les nombres pairs de la forme $4k$, on réussit toujours à trouver, à partir de $n = 8$ un nombre premier p qui est un décomposant de Goldbach de n, i.e. tel que p et $n - p$ sont premiers, mais également tel que $(n/2) + 2 - p$ est premier aussi.

Pour les nombres pairs de la forme $4k + 2$, on réussit toujours à trouver, à partir de $n = 10$ un nombre premier p qui est un décomposant de Goldbach de n, i.e. tel que p et $n - p$ sont premiers, mais également tel que $(n/2) + 1 - p$ est premier aussi.

On écrit dans deux tableaux (l'un contenant les nombres pairs de la forme $4k$, l'autre les nombres pairs de la forme $4k + 2$), à la manière d'Euler dans son article *Découverte d'une loi tout extraordinaire des nombres par rapport à la somme de leurs diviseurs*, les quadruplets de nombres, de la forme $(n, p_1, p_2, n - p_1)$ tels que p_1, $p_2 = (n/2) + \varepsilon - p_1$, $n - p_1$ sont tous les trois premiers, avec ε qui est égal à 2 ou 1 selon que n est le double d'un nombre pair (i.e. de la forme $4k$) ou le double d'un nombre impair (i.e. de la forme $4k + 2$).

Pour l'instant, on reste surprise. Le risque, lorsqu'on s'émerveille du résultat d'un programme, c'est de s'émerveiller alors que cela n'en vaut pas la peine[1].

On peut interpréter le tableau de gauche ci-dessous de la façon suivante : il semblerait (jusqu'à 10 milliards) qu'un nombre pair n de la forme $4k$ partage toujours l'un de ses décomposants de Goldbach avec sa moitié à laquelle on ajoute 2. Les premières lignes du tableau se lisent ainsi : 8 partage avec 6 le décomposant de Goldbach 3, puis (deuxième ligne), 12 partage avec 8 le décomposant de Goldbach 5, puis (troisième ligne), 16 partage avec 10 le décomposant de Goldbach 3, puis 20 partage avec 12 le décomposant de Goldbach 7...

Le tableau de droite s'interprète de la façon suivante : il semblerait (jusqu'à 40 milliards) qu'un nombre pair n de la forme $4k + 2$ partage toujours l'un de ses décomposants de Goldbach avec sa moitié à laquelle on ajoute 1. Les premières lignes du tableau se lisent ainsi : 10 partage avec 6 le

[1] Lire ces belles phrases de Maryam Mirzakhani pour décrire ce sentiment, à la page 16 de http://denisevellachemla.eu/trad-Maryam-Secrets-de-la-Surface.pdf, en espérant que la traduction ne trahit pas ce que dit d'elle son mari, Jan Vondrák : "Parfois, vous voyez, elle était heureuse lorsqu'elle disait "Oh j'ai cette idée, et j'espère qu'elle est correcte. En fait, permettez-moi de ne pas y penser pendant quelques heures avant de découvrir qu'elle est fausse... Mais je veux juste profiter du sentiment que cette idée *pourrait être correcte* !".

décomposant de Goldbach 3, puis (deuxième ligne), 14 partage avec 8 le décomposant de Goldbach 3, puis (troisième ligne), 18 partage avec 10 le décomposant de Goldbach 5, puis 22 partage avec 12 le décomposant de Goldbach 5...

C'est un peu comme si certaines décompositions de Goldbach étaient "envoyées" par les petits nombres pairs à des nombres pairs de plus en plus éloignés d'eux (éloignés d'eux selon la liste des nombres pairs successifs).

n	p_1	$((n/2)+2)-p_1$	$n-p_1$
8	3	$(4+2)-3=3$	5
12	5	$(6+2)-5=3$	7
16	3	$(8+2)-3=7$	13
20	7	$(10+2)-7=5$	13
24	7	$(12+2)-7=7$	17
28	5	$(14+2)-5=11$	23
32	13	$(16+2)-13=5$	19
36	7	$(18+2)-7=13$	29
40	3	$(20+2)-3=19$	37
44	7	$(22+2)-7=17$	37
48	7	$(24+2)-7=19$	41
52	5	$(26+2)-5=23$	47
56	13	$(28+2)-13=17$	43
60	13	$(30+2)-13=19$	47
64	3	$(32+2)-3=31$	61
68	7	$(34+2)-7=29$	61
72	19	$(36+2)-19=19$	53
76	3	$(38+2)-3=37$	73
80	13	$(40+2)-13=29$	67
84	13	$(42+2)-13=31$	71
88	5	$(44+2)-5=41$	83
92	19	$(46+2)-19=29$	73
96	7	$(48+2)-7=43$	89
100	11	$(50+2)-11=41$	89

Tableau 1 : Les nombres pairs de la forme $4k$

n	p_1	$((n/2)+1)-p_1$	$n-p_1$
10	3	$(5+1)-3=3$	7
14	3	$(7+1)-3=5$	11
18	5	$(9+1)-5=5$	13
22	5	$(11+1)-5=7$	17
26	3	$(13+1)-3=11$	23
30	11	$(15+1)-11=5$	19
34	5	$(17+1)-5=13$	29
38	7	$(19+1)-7=13$	31
42	5	$(21+1)-5=17$	37
46	5	$(23+1)-5=19$	41
50	3	$(25+1)-3=23$	47
54	11	$(27+1)-11=17$	43
58	11	$(29+1)-11=19$	47
62	3	$(31+1)-3=29$	59
66	5	$(33+1)-5=29$	61
70	17	$(35+1)-17=19$	53
74	7	$(37+1)-7=31$	67
78	11	$(39+1)-11=29$	67
82	11	$(41+1)-11=31$	71
86	3	$(43+1)-3=41$	83
90	17	$(45+1)-17=29$	73
94	5	$(47+1)-5=43$	79
98	19	$(49+1)-19=31$	79

Tableau 2 : Les nombres pairs de la forme $4k+2$

En annexe, est fourni le programme de détermination d'un quadruplet pour tout nombre pair. On a pu le tester (ainsi que le programme concernant les $4k+2$, dans lequel ε est égal à 1 au lieu de 2) jusqu'à un certain nombre de milliards.

On a enfin testé la même idée pour les nombres de la forme 4.10^k jusqu'à 40 milliards ou pour les nombres 10×9^k ou bien 10×11^k. (Pourquoi d'une part, des multiplications successives par 9 et d'autre part, des multiplications successives par 11 ? Simplement pour "rester dans les $4k+2$" (des multiplications successives par 10 feraient passer une fois sur deux d'un nombre de la forme $4k$ à un nombre de la forme $4k+2$ et inversement les fois intermédiaires).

Multiplications par 10 (à partir de 4) :

n	p_1	$((n/2)+2)-p_1$	$n-p_1$
40	3	$(20+2)-3=19$	37
400	3	$(200+2)-3=199$	397
4 000	53	$(2\,000+2)-53=1\,949$	3 947
40 000	11	$(20\,000+2)-11=19\,991$	39 989
400 000	113	$(200\,000+2)-113=199\,889$	399 887
4 000 000	149	$(2\,000\,000+2)-149=1\,999\,853$	3 999 951
40 000 000	563	$(20\,000\,000+2)-563=19\,999\,439$	39 999 437
400 000 000	53	$(200\,000\,000+2)-53=199\,999\,949$	399 999 947
4 000 000 000	89	$(2\,000\,000\,000+2)-89=19\,999\,999\,913$	3 999 999 911
40 000 000 000	641	$(20\,000\,000\,000+2)-641=19\,999\,999\,361$	39 999 999 359

Multiplications par 9 (à partir de 10) :

n	p_1	$((n/2)+1)-p_1$	$n-p_1$
10	3	$(5+1)-3=3$	7
90	17	$(45+1)-17=29$	73
810	23	$(405+1)-23=383$	787
7 290	53	$(3\,645+1)-53=3\,593$	7 237
65 610	23	$(32\,805+1)-23=32\,783$	65 587
590 490	59	$(295\,245+1)-59=295\,187$	590 431
5 314 410	659	$(2\,657\,205+1)-659=2\,656\,547$	5 313 751
47 829 690	47	$(23\,914\,845+1)-47=23\,914\,799$	47 829 643
430 467 210	1103	$(215\,233\,605+1)-1\,103=215\,232\,503$	430 466 107
3 874 204 890	467	$(1\,937\,102\,445+1)-467=1\,937\,101\,979$	3 874 204 423

Multiplications par 11 (à partir de 10) :

n	p_1	$((n/2)+1)-p_1$	$n-p_1$
10	3	$(5+1)-3=3$	7
110	3	$(55+1)-3=53$	107
1 210	29	$(605+1)-29=577$	1 181
13 310	19	$(6\,655+1)-19=6\,637$	13 291
146 410	197	$(73\,205+1)-197=73\,009$	146 213
1 610 510	37	$(805\,255+1)-37=805\,219$	1610473
17 715 610	53	$(8\,857\,805+1)-53=8\,857\,753$	17 715 557
194 871 710	127	$(97\,435\,855+1)-127=97\,435\,729$	194 871 583
2 143 588 810	269	$(1\,071\,794\,405+1)-269=1\,071\,794\,137$	2 143 588 541

Y aurait-il une loi de récurrence, derrière certaines décompositions de Goldbach du moins ?

Annexe : Programme python de découverte de ces relations surprenantes

```
def prime(atester):
    k = 2
    if atester in [0, 1]: return False
    if atester in [2, 3, 5, 7]: return True
    while True:
        if k * k ¿ atester: return True
        else:
            if atester % k == 0: return False
            else: k = k + 1

    def reverseList(L):
    lReverse = []
    for item in L:
        lReverse.insert(0, item)
    return lReverse

ok = []
fichier = open(”fic.txt”, ”a”)
for n in range(4,1000,4):
    trouve = False
    for p in range(3,int(n/2),2):
        if not trouve:
            if prime(p) and prime(n-p):
                if prime((n/2)-p+2):
                    print(’youpi’, n, p)
                    fichier.write(str(n))
                    fichier.write(”)
                    fichier.write(str(p))
                    fichier.write(”)
                    fichier.write(str(int((n/2)-p+2)))
                    fichier.write(”)
                    fichier.write(”)
                    trouve = True
                    ok.append(True)
    if not trouve:
        ok.append(False)
print(ok)
okrev = reverseList(ok)
print(okrev.index(False))
fichier.close()
```