

Unités (Denise Vella-Chemla, 14.1.2022)

On voudrait fournir ici les conclusions obtenues à partir de l'exécution d'un programme informatique qui montre certaines relations entre des listes de nombres, en lien avec la conjecture de Goldbach.

Le programme calcule des intersections d'ensembles de nombres, pour les nombres pairs de 6 à 100. Il s'agit de considérer la liste de nombres intersection de deux listes de nombres (on omet le nombre 1 lorsqu'il appartient à l'intersection de listes en question), pour $n \geq 46$.

Pour les nombres pairs n compris entre 46 et 100, on constate que les nombres x qui sont à la fois premiers à n et premiers au produit des nombres premiers inférieurs à \sqrt{n} , que l'on notera *Prodprem* dans la suite¹, et dont le complémentaire à n (i.e. le nombre $n - x$) est aussi premier à *Prodprem*, sont des décomposants de Goldbach de n (sauf 1, lorsque ce nombre appartient à L3). On ne sait pas démontrer que l'intersection (notée L3 ici) des listes considérées n'est jamais vide.

Tout x qui est une unité à n (un nombre premier à n) vérifie l'équation modulaire

$$x^{\varphi(n)} \equiv 1 \pmod{n}$$

On vérifie par exemple, puisque $\varphi(98) = 42$ et $\varphi(210) = 48$, que 19, décomposant de Goldbach de 98, vérifie le système de trois équations modulaires :

$$\begin{cases} 19^{42} \equiv 1 \pmod{98} \\ 19^{48} \equiv 1 \pmod{210} \\ 79^{48} \equiv 1 \pmod{210} \end{cases}$$

tandis que 17 n'est pas un décomposant de Goldbach de 98 parce que $17^{48} \not\equiv 1 \pmod{210}$.

Un décomposant de Goldbach x du nombre pair n doit donc rendre vraies les 3 équations modulaires suivantes

$$\begin{cases} x^{\varphi(n)} & \equiv 1 \pmod{n} \\ x^{\varphi(\text{Prodprem})} & \equiv 1 \pmod{\text{Prodprem}} \\ (n - x)^{\varphi(\text{Prodprem})} & \equiv 1 \pmod{\text{Prodprem}} \end{cases}$$

avec

$$\varphi(\text{Prodprem}) = \prod_{\substack{p \text{ premier} \\ 3 \leq p \leq \sqrt{n}}} (p - 1)$$

et

$$(n - x)^{\varphi(\text{Prodprem})} = \sum_{k=0}^{\varphi(\text{Prodprem})} \binom{\varphi(\text{Prodprem})}{k} n^{\varphi(\text{Prodprem})-k} (-x)^k$$

La notation $\binom{\varphi(\text{Prodprem})}{k}$ représente le coefficient binomial qui intervient dans le calcul du développement de la puissance de $(n - x)$.

¹Un nombre a est premier à un nombre b (différent de lui) si et seulement si leur pgcd est égal à 1 (i.e. si et seulement si a et b n'ont aucun facteur commun autre que 1).

Il faudrait comprendre pourquoi un tel système de trois équations modulaires a toujours une solution comprise entre 3 et $n/2$.

On écrit le programme suivant pour vérifier notre découverte.

```
import math

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

def pgcd(m, n):
    while (m != 0):
        r = n % m
        n = m
        m = r
    return(n)

def indic(n):
    phi = 0
    for k in range(n):
        if pgcd(n,k) == 1:
            phi = phi+1
    print(' ',n, 'indic ', phi)
    return phi

def premiersa(n):
    liste = []
    for k in range(n+1):
        if pgcd(n,k) == 1:
            liste.append(k)
    return liste
```

```

def puissance(x, k, n):
    expo = 1
    puiss = 1
    while expo != k:
        puiss = (puiss*x)%n
        expo = expo+1
    print(x, '**', k, '=', puiss, '(mod ',n,')')
    return puiss

for n in range(6,102,2):
    print('=', n, '-> ')
    moitie = int(n/2)
    rac = int(math.sqrt(n))
    prodprem = 1
    for x in range(rac+1):
        if prime(x):
            prodprem = prodprem * x
    varphi1 = indic(n)
    varphi2 = indic(prodprem)
    listexverifiantlestroisegalites = []
    for x in range(moitie):
        if puissance(x,varphi1,n) == 1 and puissance(x,varphi2,prodprem) ==
1 and puissance(n-x,varphi2,prodprem) == 1:
            print(' ',x, 'verifie les 3 egalites.')
            listexverifiantlestroisegalites.append(x)
    print(' ',listexverifiantlestroisegalites)
    nbsol = 0
    listedg = []
    for x in range(3,moitie+1,2):
        if prime(x) and prime(n-x):
            nbsol += 1
            listedg.append(x)
    print(' Liste des decomposants de Goldbach de ',n,' -> ', listedg)
    print(' card = ', nbsol)

```

Voici le résultat du programme ci-dessus.

n = 6 ->

6 indic 2

2 indic 1

1 verifie les 3 egalites.

[1]

Liste des decomposants de Goldbach de 6 —> [3]

card = 1

n = 8 ->

8 indic 4

2 indic 1

1 verifie les 3 egalites.

3 verifie les 3 egalites.

[1, 3]

Liste des decomposants de Goldbach de 8 —> [3]

card = 1

n = 10 ->

10 indic 4

6 indic 2

[]

Liste des decomposants de Goldbach de 10 —> [3, 5]

card = 2

n = 12 ->

12 indic 4

6 indic 2

1 verifie les 3 egalites.

5 verifie les 3 egalites.

[1, 5]

Liste des decomposants de Goldbach de 12 —> [5]

card = 1

n = 14 ->

14 indic 6

6 indic 2

1 verifie les 3 egalites.

[1]

Liste des decomposants de Goldbach de 14 —> [3, 7]

card = 2

n = 16 ->

16 indic 8

6 indic 2

5 verifie les 3 egalites.

[5]

Liste des decomposants de Goldbach de 16 —> [3, 5]

card = 2

n = 18 ->

18 indic 6

6 indic 2

1 verifie les 3 egalites.

5 verifie les 3 egalites.

7 verifie les 3 egalites.

[1, 5, 7]

Liste des decomposants de Goldbach de 18 —> [5, 7]

card = 2

n = 20 ->

20 indic 8

6 indic 2

1 verifie les 3 egalites.

7 verifie les 3 egalites.

[1, 7]

Liste des decomposants de Goldbach de 20 —> [3, 7]

card = 2

n = 22 ->

22 indic 10

6 indic 2

5 verifie les 3 egalites.

[5]

Liste des decomposants de Goldbach de 22 —> [3, 5, 11]

card = 3

n = 24 ->

24 indic 8

6 indic 2

1 verifie les 3 egalites.

5 verifie les 3 egalites.

7 verifie les 3 egalites.

11 verifie les 3 egalites.

[1, 5, 7, 11]

Liste des decomposants de Goldbach de 24 —> [5, 7, 11]

card = 3

n = 26 ->

26 indic 12

30 indic 8

7 verifie les 3 egalites.

[7]

Liste des decomposants de Goldbach de 26 —> [3, 7, 13]

card = 3

n = 28 ->

28 indic 12

30 indic 8

11 verifie les 3 egalites.

[11]

Liste des decomposants de Goldbach de 28 —> [5, 11]

card = 2

n = 30 ->

30 indic 8

30 indic 8

1 verifie les 3 egalites.

7 verifie les 3 egalites.

11 verifie les 3 egalites.

13 verifie les 3 egalites.

[1, 7, 11, 13]

Liste des decomposants de Goldbach de 30 —> [7, 11, 13]

card = 3

n = 32 ->

32 indic 16

30 indic 8

1 verifie les 3 egalites.

13 verifie les 3 egalites.

[1, 13]

Liste des decomposants de Goldbach de 32 —> [3, 13]

card = 2

n = 34 ->

34 indic 16

30 indic 8

11 verifie les 3 egalites.

[11]

Liste des decomposants de Goldbach de 34 —> [3, 5, 11, 17]

card = 4

n = 36 ->

36 indic 12

30 indic 8

7 verifie les 3 egalites.

13 verifie les 3 egalites.

17 verifie les 3 egalites.

[7, 13, 17]

Liste des decomposants de Goldbach de 36 —> [5, 7, 13, 17]

card = 4

n = 38 ->

38 indic 18

30 indic 8

1 verifie les 3 egalites.

7 verifie les 3 egalites.

[1, 7]

Liste des decomposants de Goldbach de 38 —> [7, 19]

card = 2

n = 40 ->

40 indic 16

30 indic 8

11 verifie les 3 egalites.

17 verifie les 3 egalites.

[11, 17]

Liste des decomposants de Goldbach de 40 —> [3, 11, 17]

card = 3

n = 42 ->

42 indic 12

30 indic 8

1 verifie les 3 egalites.

11 verifie les 3 egalites.

13 verifie les 3 egalites.

19 verifie les 3 egalites.

[1, 11, 13, 19]

Liste des decomposants de Goldbach de 42 —> [5, 11, 13, 19]

card = 4

n = 44 ->

44 indic 20

30 indic 8

1 verifie les 3 egalites.

7 verifie les 3 egalites.

13 verifie les 3 egalites.

[1, 7, 13]

Liste des decomposants de Goldbach de 44 —> [3, 7, 13]

card = 3

n = 46 ->

46 indic 22

30 indic 8

17 verifie les 3 egalites.

[17]

Liste des decomposants de Goldbach de 46 —> [3, 5, 17, 23]

card = 4

n = 48 ->

48 indic 16

30 indic 8

1 verifie les 3 egalites.

7 verifie les 3 egalites.

11 verifie les 3 egalites.

17 verifie les 3 egalites.

19 verifie les 3 egalites.

[1, 7, 11, 17, 19]

Liste des decomposants de Goldbach de 48 —> [5, 7, 11, 17, 19]

card = 5

n = 50 ->

50 indic 20

210 indic 48

13 verifie les 3 egalites.

19 verifie les 3 egalites.

[13, 19]

Liste des decomposants de Goldbach de 50 —> [3, 7, 13, 19]

card = 4

n = 52 ->

52 indic 24

210 indic 48

11 verifie les 3 egalites.

23 verifie les 3 egalites.

[11, 23]

Liste des decomposants de Goldbach de 52 —> [5, 11, 23]

card = 3

n = 54 ->

54 indic 18

210 indic 48

1 verifie les 3 egalites.

11 verifie les 3 egalites.

13 verifie les 3 egalites.

17 verifie les 3 egalites.

23 verifie les 3 egalites.

[1, 11, 13, 17, 23]

Liste des decomposants de Goldbach de 54 —> [7, 11, 13, 17, 23]

card = 5

n = 56 ->

56 indic 24

210 indic 48

13 verifie les 3 egalites.

19 verifie les 3 egalites.

[13, 19]

Liste des decomposants de Goldbach de 56 —> [3, 13, 19]

card = 3

n = 58 ->

58 indic 28

210 indic 48

11 verifie les 3 egalites.

17 verifie les 3 egalites.

[11, 17]

Liste des decomposants de Goldbach de 58 —> [5, 11, 17, 29]

card = 4

n = 60 ->

60 indic 16

210 indic 48

1 verifie les 3 egalites.

13 verifie les 3 egalites.

17 verifie les 3 egalites.

19 verifie les 3 egalites.

23 verifie les 3 egalites.

29 verifie les 3 egalites.

[1, 13, 17, 19, 23, 29]

Liste des decomposants de Goldbach de 60 —> [7, 13, 17, 19, 23, 29]

card = 6

n = 62 ->

62 indic 30

210 indic 48

1 verifie les 3 egalites.

19 verifie les 3 egalites.

[1, 19]

Liste des decomposants de Goldbach de 62 —> [3, 19, 31]

card = 3

n = 64 ->

64 indic 32

210 indic 48

11 verifie les 3 egalites.

17 verifie les 3 egalites.

23 verifie les 3 egalites.

[11, 17, 23]

Liste des decomposants de Goldbach de 64 —> [3, 5, 11, 17, 23]

card = 5

n = 66 ->

66 indic 20

210 indic 48

13 verifie les 3 egalites.

19 verifie les 3 egalites.

23 verifie les 3 egalites.

29 verifie les 3 egalites.

[13, 19, 23, 29]

Liste des decomposants de Goldbach de 66 —> [5, 7, 13, 19, 23, 29]

card = 6

n = 68 ->

68 indic 32

210 indic 48

1 verifie les 3 egalites.

31 verifie les 3 egalites.

[1, 31]

Liste des decomposants de Goldbach de 68 —> [7, 31]

card = 2

n = 70 ->

70 indic 24

210 indic 48

11 verifie les 3 egalites.

17 verifie les 3 egalites.

23 verifie les 3 egalites.

29 verifie les 3 egalites.

[11, 17, 23, 29]

Liste des decomposants de Goldbach de 70 —> [3, 11, 17, 23, 29]

card = 5

n = 72 ->

72 indic 24

210 indic 48

1 verifie les 3 egalites.

11 verifie les 3 egalites.

13 verifie les 3 egalites.

19 verifie les 3 egalites.

29 verifie les 3 egalites.

31 verifie les 3 egalites.

[1, 11, 13, 19, 29, 31]

Liste des decomposants de Goldbach de 72 —> [5, 11, 13, 19, 29, 31]

card = 6

n = 74 ->

74 indic 36

210 indic 48

1 verifie les 3 egalites.

13 verifie les 3 egalites.

31 verifie les 3 egalites.

[1, 13, 31]

Liste des decomposants de Goldbach de 74 —> [3, 7, 13, 31, 37]

card = 5

n = 76 ->

76 indic 36

210 indic 48

17 verifie les 3 egalites.

23 verifie les 3 egalites.

29 verifie les 3 egalites.

[17, 23, 29]

Liste des decomposants de Goldbach de 76 —> [3, 5, 17, 23, 29]

card = 5

n = 78 ->

78 indic 24

210 indic 48

11 verifie les 3 egalites.

17 verifie les 3 egalites.

19 verifie les 3 egalites.

31 verifie les 3 egalites.

37 verifie les 3 egalites.

[11, 17, 19, 31, 37]

Liste des decomposants de Goldbach de 78 —> [5, 7, 11, 17, 19, 31, 37]

card = 7

n = 80 ->

80 indic 32

210 indic 48

1 verifie les 3 egalites.

13 verifie les 3 egalites.

19 verifie les 3 egalites.

37 verifie les 3 egalites.

[1, 13, 19, 37]

Liste des decomposants de Goldbach de 80 —> [7, 13, 19, 37]

card = 4

n = 82 ->

82 indic 40

210 indic 48

11 verifie les 3 egalites.

23 verifie les 3 egalites.

29 verifie les 3 egalites.

[11, 23, 29]

Liste des decomposants de Goldbach de 82 —> [3, 11, 23, 29, 41]

card = 5

n = 84 ->

84 indic 24

210 indic 48

1 verifie les 3 egalites.

11 verifie les 3 egalites.

13 verifie les 3 egalites.

17 verifie les 3 egalites.

23 verifie les 3 egalites.

31 verifie les 3 egalites.

37 verifie les 3 egalites.

41 verifie les 3 egalites.

[1, 11, 13, 17, 23, 31, 37, 41]

Liste des decomposants de Goldbach de 84 —> [5, 11, 13, 17, 23, 31, 37, 41]

card = 8

n = 86 ->

86 indic 42

210 indic 48

13 verifie les 3 egalites.

19 verifie les 3 egalites.

[13, 19]

Liste des decomposants de Goldbach de 86 —> [3, 7, 13, 19, 43]

card = 5

n = 88 ->

88 indic 40

210 indic 48

17 verifie les 3 egalites.

29 verifie les 3 egalites.

41 verifie les 3 egalites.

[17, 29, 41]

Liste des decomposants de Goldbach de 88 —> [5, 17, 29, 41]

card = 4

n = 90 ->

90 indic 24

210 indic 48

1 verifie les 3 egalites.

11 verifie les 3 egalites.

17 verifie les 3 egalites.

19 verifie les 3 egalites.

23 verifie les 3 egalites.

29 verifie les 3 egalites.

31 verifie les 3 egalites.

37 verifie les 3 egalites.

43 verifie les 3 egalites.

[1, 11, 17, 19, 23, 29, 31, 37, 43]

Liste des decomposants de Goldbach de 90 —> [7, 11, 17, 19, 23, 29, 31, 37, 43]

card = 9

n = 92 ->

92 indic 44

210 indic 48

13 verifie les 3 egalites.

19 verifie les 3 egalites.

31 verifie les 3 egalites.

[13, 19, 31]

Liste des decomposants de Goldbach de 92 —> [3, 13, 19, 31]

card = 4

n = 94 ->

94 indic 46

210 indic 48

11 verifie les 3 egalites.

23 verifie les 3 egalites.

41 verifie les 3 egalites.

[11, 23, 41]

Liste des decomposants de Goldbach de 94 —> [5, 11, 23, 41, 47]

card = 5

n = 96 ->

96 indic 32

210 indic 48

13 verifie les 3 egalites.

17 verifie les 3 egalites.

23 verifie les 3 egalites.

29 verifie les 3 egalites.

37 verifie les 3 egalites.

43 verifie les 3 egalites.

[13, 17, 23, 29, 37, 43]

Liste des decomposants de Goldbach de 96 —> [7, 13, 17, 23, 29, 37, 43]

card = 7

n = 98 ->

98 indic 42

210 indic 48

1 verifie les 3 egalites.

19 verifie les 3 egalites.

31 verifie les 3 egalites.

37 verifie les 3 egalites.

[1, 19, 31, 37]

Liste des decomposants de Goldbach de 98 —> [19, 31, 37]

card = 3

n = 100 ->

100 indic 40

210 indic 48

11 verifie les 3 egalites.

17 verifie les 3 egalites.

29 verifie les 3 egalites.

41 verifie les 3 egalites.

47 verifie les 3 egalites.

[11, 17, 29, 41, 47]

Liste des decomposants de Goldbach de 100 —> [3, 11, 17, 29, 41, 47]

card = 6