

**Sophie Germain et la moyenne des courbures
pour trouver un décomposant de Goldbach
Denise Vella-Chemla
décembre 2024**

Malheureusement (ajouté le 29.12.2024), l'idée et le programme ci-dessous échouent à trouver un décomposant de Goldbach dès le nombre 128, pour lequel 37, le décomposant potentiel de courbure maximale, n'est pas un décomposant de Goldbach. C'est normal, cette idée mélange tout : il faut plutôt considérer le produit des restes, et les nombres qui ont leur produit des restes non nuls (on peut même considérer ces restes "modulo tous les nombres de 2 à \sqrt{n} ") sont les décomposants de Goldbach de n ; mais ça, c'est la définition triviale de la notion-même de décomposant de Goldbach et cela n'apporte rien.

Suite à l'écoute d'une émission de France Culture consacrée à Sophie Germain ¹, on a l'idée d'utiliser les moyennes des courbures, une idée dont Etienne Ghys dit qu'elle est à attribuer à Sophie Germain, pour trouver comme décomposant de Goldbach de n le nombre compris entre 3 et $n/2$ qui a la "moyenne de ses courbures qui est maximum".

Voici l'idée : on sait qu'un décomposant de Goldbach de n doit être aussi éloigné qu'il est possible en terme de restes modulaires (modulo les nombres premiers inférieurs à la racine de n) à la fois de 0 et de n . On voit les restes comme représentant des pentes et la pente est maximum, soit l'éloignement, disons du niveau qui nous intéresse pour alléger le propos, lorsque la moyenne des courbures en x et $n - x$ est maximum. Ce qui semble le plus correspondre à la notion de courbure ici, c'est l'inverse du reste modulaire.

On définit la fonction courbure² ainsi :

$$\text{courbure}(x, n) = \frac{\sum_{\substack{p \text{ premier} ; 3 \leq p \leq \sqrt{n} \\ x \bmod p \neq 0 ; n-x \bmod p \neq 0}} \frac{1}{x \bmod p} + \frac{1}{(n-x) \bmod p}}{\pi(\sqrt{n})}.$$

On trouve un décomposant de Goldbach de n en prenant le nombre impair x qui est de courbure maximale

$$\text{dg}_1(n) = x \text{ avec } \text{courbure}(x, n) = \max\{\text{courbure}(k, n) \mid 3 \leq k \leq n/2\}.$$

On teste le programme suivant :

¹lien vers l'émission.

²Il semblerait que la recherche d'un décomposant de Goldbach maximisant la moyenne des courbures fonctionne également en supprimant la condition de primalité sur p mais en ne prenant cependant que des diviseurs impairs (sauf 1).

```

import math ; from math import floor,sqrt

def premier(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

for n in range(6,100,2):
    print('= ',n)
    courburemax = 0
    for x in range(3,n//2+1,2):
        courburex = 0
        nbp = 0
        for p in range(3,floor(sqrt(n))+1,2):
            if premier(p):
                nbp = nbp+1
                if x%p != 0 and (n-x)%p != 0:
                    courburex = courburex+1/(x%p)+1/((n-x)%p)
        if nbp != 0:
            courburex = courburex/nbp
            if courburex > courburemax:
                courburemax = courburex
            print(x, ' --> ',courburex,end='')
        if premier(x) and premier(n-x):
            print(' dg')
        else:
            print('')
    print('courburemax = ',courburemax)

```

Voici son résultat.

```

n = 16
3 --> 0.0 dg
5 --> 1.0 dg
7 --> 0.0
courburemax = 1.0

n = 18
3 --> 0.0
5 --> 1.5 dg
7 --> 1.5 dg
9 --> 0.0
courburemax = 1.5

n = 20
3 --> 0.0 dg
5 --> 0.0
7 --> 2.0 dg
9 --> 0.0
courburemax = 2.0

```

```
n = 22
3 --> 0.0 dg
5 --> 1.0 dg
7 --> 0.0
9 --> 0.0
11 --> 1.0 dg
courburemax = 1.0

n = 24
3 --> 0.0
5 --> 1.5 dg
7 --> 1.5 dg
9 --> 0.0
11 --> 1.5 dg
courburemax = 1.5

n = 26
3 --> 0.3333333333333333 dg
5 --> 0.0
7 --> 1.375 dg
9 --> 0.375
11 --> 0.0
13 --> 1.3333333333333335 dg
courburemax = 1.375

n = 28
3 --> 0.0
5 --> 0.5 dg
7 --> 0.75
9 --> 0.25
11 --> 1.25 dg
13 --> 0.0
courburemax = 1.25

n = 30
3 --> 0.41666666666666663
5 --> 0.75
7 --> 1.1666666666666667 dg
9 --> 0.625
11 --> 1.375 dg
13 --> 1.1666666666666665 dg
15 --> 0.0
courburemax = 1.375

n = 32
3 --> 0.29166666666666663 dg
5 --> 0.0
7 --> 1.0
9 --> 0.29166666666666663
11 --> 1.0
13 --> 1.2916666666666667 dg
15 --> 0.0
courburemax = 1.2916666666666667
```

```
n = 34
3 --> 0.6666666666666666 dg
5 --> 0.5 dg
7 --> 0.5
9 --> 0.0
11 --> 1.1666666666666667 dg
13 --> 0.6666666666666666
15 --> 0.0
17 --> 1.0 dg
courburemax = 1.1666666666666667
```

```
n = 36
3 --> 0.3333333333333333
5 --> 0.75 dg
7 --> 1.125 dg
9 --> 0.375
11 --> 0.75
13 --> 1.0833333333333333 dg
15 --> 0.0
17 --> 1.125 dg
courburemax = 1.125
```

```
n = 38
3 --> 0.0
5 --> 0.0
7 --> 1.75 dg
9 --> 0.25
11 --> 0.75
13 --> 1.0
15 --> 0.0
17 --> 0.75
19 --> 1.25 dg
courburemax = 1.75
```

```
n = 40
3 --> 0.41666666666666663 dg
5 --> 0.5
7 --> 0.41666666666666663
9 --> 0.625
11 --> 1.125 dg
13 --> 0.41666666666666663
15 --> 0.0
17 --> 0.9166666666666666 dg
19 --> 0.625
courburemax = 1.125
```

```
n = 42
3 --> 0.29166666666666663
5 --> 0.75 dg
7 --> 0.75
9 --> 0.29166666666666663
11 --> 1.75 dg
13 --> 1.0416666666666665 dg
15 --> 0.0
17 --> 0.75
19 --> 1.0416666666666667 dg
21 --> 1.0
courburemax = 1.75
```

```
n = 44
3 --> 0.6666666666666666 dg
5 --> 0.0
7 --> 1.5 dg
9 --> 0.0
11 --> 0.6666666666666666
13 --> 1.6666666666666667 dg
15 --> 0.0
17 --> 0.5
19 --> 1.0
21 --> 0.6666666666666666
courburemax = 1.6666666666666667
```

```
n = 46
3 --> 0.3333333333333333 dg
5 --> 0.5 dg
7 --> 0.375
9 --> 0.375
11 --> 0.5
13 --> 0.3333333333333333
15 --> 0.0
17 --> 0.875 dg
19 --> 0.375
21 --> 0.0
23 --> 0.8333333333333333 dg
courburemax = 0.875
```

```
n = 48
3 --> 0.0
5 --> 0.75 dg
7 --> 1.5 dg
9 --> 0.25
11 --> 1.5 dg
13 --> 0.75
15 --> 0.0
17 --> 1.5 dg
19 --> 1.0 dg
21 --> 0.75
23 --> 0.75
courburemax = 1.5
```

```
n = 50
3 --> 0.4555555555555555 dg
5 --> 0.1777777777777778
7 --> 0.9444444444444445 dg
9 --> 0.6388888888888889
11 --> 0.5833333333333334
13 --> 1.1666666666666667 dg
15 --> 0.0
17 --> 0.4555555555555555
19 --> 1.2611111111111113 dg
21 --> 0.4166666666666667
23 --> 0.5
25 --> 0.8333333333333334
courburemax = 1.2611111111111113
```

```
n = 52
3 --> 0.19444444444444442
5 --> 0.4666666666666666 dg
7 --> 0.0
9 --> 0.6944444444444443
11 --> 1.1388888888888888 dg
13 --> 0.3333333333333333
15 --> 0.5
17 --> 0.3333333333333333
19 --> 0.3277777777777777
21 --> 0.6666666666666666
23 --> 1.0277777777777777 dg
25 --> 0.1388888888888887
courburemax = 1.1388888888888888
```

```
n = 54
3 --> 0.7222222222222222
5 --> 0.5
7 --> 0.8333333333333334 dg
9 --> 0.2777777777777773
11 --> 1.3611111111111114 dg
13 --> 1.0555555555555554 dg
15 --> 0.4166666666666667
17 --> 1.1111111111111112 dg
19 --> 0.5
21 --> 0.4444444444444444
23 --> 1.2222222222222222 dg
25 --> 0.9166666666666666
27 --> 0.4444444444444445
courburemax = 1.3611111111111114
```

```
n = 56
3 --> 0.4166666666666667 dg
5 --> 0.2333333333333333
7 --> 0.9166666666666666
9 --> 0.4833333333333334
11 --> 0.19444444444444442
13 --> 1.2777777777777778 dg
15 --> 0.3888888888888889
17 --> 0.4444444444444444
19 --> 1.1500000000000001 dg
21 --> 0.0
23 --> 0.4555555555555555
25 --> 0.8611111111111112
27 --> 0.6388888888888888
courburemax = 1.2777777777777778
```

```
n = 58
3 --> 0.1666666666666666
5 --> 0.4833333333333334 dg
7 --> 0.5
9 --> 0.1666666666666666
11 --> 0.9833333333333334 dg
13 --> 0.1666666666666666
15 --> 0.6666666666666666
17 --> 1.0 dg
19 --> 0.3166666666666665
21 --> 0.5
23 --> 0.3333333333333333
25 --> 0.15
27 --> 0.6666666666666666
29 --> 1.166666666666667 dg
courburemax = 1.166666666666667
```

```
n = 60
3 --> 0.7222222222222222
5 --> 0.6222222222222222
7 --> 0.7777777777777778 dg
9 --> 0.75
11 --> 0.9166666666666666
13 --> 0.8999999999999999 dg
15 --> 0.4444444444444444
17 --> 1.2222222222222223 dg
19 --> 1.038888888888889 dg
21 --> 0.4166666666666667
23 --> 1.111111111111111 dg
25 --> 0.5
27 --> 0.3999999999999997
29 --> 1.361111111111111 dg
courburemax = 1.361111111111111
```

```
n = 62
3 --> 0.4166666666666666 dg
5 --> 0.3999999999999997
7 --> 0.6666666666666666
9 --> 0.4444444444444444
11 --> 0.9166666666666666
13 --> 0.8611111111111112
15 --> 0.3999999999999997
17 --> 0.2222222222222222
19 --> 1.2611111111111113 dg
21 --> 0.6666666666666666
23 --> 0.4444444444444444
25 --> 0.9166666666666666
27 --> 0.0
29 --> 0.5944444444444444
31 --> 1.5555555555555554 dg
courburemax = 1.5555555555555554
```

```
n = 64
3 --> 0.622222222222221 dg
5 --> 0.511111111111111 dg
7 --> 0.333333333333333
9 --> 0.222222222222222
11 --> 0.9444444444444445 dg
13 --> 0.666666666666666
15 --> 0.0
17 --> 0.8444444444444446 dg
19 --> 0.177777777777778
21 --> 0.444444444444444
23 --> 0.999999999999999 dg
25 --> 0.166666666666666
27 --> 0.555555555555556
29 --> 0.333333333333333
31 --> 0.622222222222221
courburemax = 0.999999999999999
```

```
n = 66
3 --> 0.222222222222222
5 --> 0.633333333333333 dg
7 --> 0.75 dg
9 --> 0.75
11 --> 0.638888888888889
13 --> 0.861111111111111 dg
15 --> 0.5
17 --> 0.75
19 --> 0.883333333333334 dg
21 --> 0.0
23 --> 1.222222222222222 dg
25 --> 0.638888888888889
27 --> 0.3888888888888884
29 --> 1.25 dg
31 --> 0.5
33 --> 0.355555555555557
courburemax = 1.25
```

```
n = 68
3 --> 0.2777777777777773
5 --> 0.0
7 --> 1.166666666666667 dg
9 --> 0.444444444444444
11 --> 0.916666666666666
13 --> 0.777777777777777
15 --> 0.416666666666667
17 --> 0.777777777777777
19 --> 0.833333333333334
21 --> 0.5
23 --> 0.2777777777777773
25 --> 1.083333333333333
27 --> 0.611111111111112
29 --> 0.583333333333334
31 --> 1.444444444444446 dg
33 --> 0.0
courburemax = 1.444444444444446
```



```
n = 70
3 --> 0.4722222222222215 dg
5 --> 0.5666666666666667
7 --> 0.2777777777777773
9 --> 0.65
11 --> 0.9444444444444445 dg
13 --> 0.6666666666666666
15 --> 0.3888888888888889
17 --> 0.8055555555555555 dg
19 --> 0.65
21 --> 0.4166666666666667
23 --> 0.8444444444444444 dg
25 --> 0.1944444444444442
27 --> 0.6666666666666666
29 --> 1.1388888888888888 dg
31 --> 0.6111111111111111
33 --> 0.5111111111111111
35 --> 0.3333333333333333
courburemax = 1.1388888888888888
```

```
n = 72
3 --> 0.3611111111111111
5 --> 0.65 dg
7 --> 0.5
9 --> 0.1944444444444442
11 --> 1.3166666666666667 dg
13 --> 0.8611111111111111 dg
15 --> 0.6666666666666666
17 --> 0.6666666666666666
19 --> 0.8444444444444446 dg
21 --> 0.6666666666666666
23 --> 0.6944444444444443
25 --> 0.65
27 --> 0.1666666666666666
29 --> 1.3611111111111114 dg
31 --> 1.3333333333333333 dg
33 --> 0.3444444444444444
35 --> 0.5
courburemax = 1.3611111111111114
```

```
n = 74
3 --> 0.8888888888888888 dg
5 --> 0.1222222222222223
7 --> 1.0 dg
9 --> 0.3333333333333333
11 --> 0.4444444444444444
13 --> 1.2333333333333334 dg
15 --> 0.4444444444444444
17 --> 0.7777777777777777
19 --> 0.7888888888888889
21 --> 0.4444444444444444
23 --> 0.7777777777777777
25 --> 0.6666666666666666
27 --> 0.4555555555555555
29 --> 0.4444444444444444
```

```
31 --> 1.5555555555555556 dg
33 --> 0.5666666666666667
35 --> 0.0
37 --> 1.3333333333333333 dg
courburemax = 1.5555555555555556
```

```
n = 76
```

```
3 --> 0.4444444444444444 dg
5 --> 0.7333333333333334 dg
7 --> 0.25
9 --> 0.5
11 --> 0.5833333333333334
13 --> 0.2222222222222222
15 --> 0.3999999999999997
17 --> 0.8055555555555557 dg
19 --> 0.65
21 --> 0.0
23 --> 0.8055555555555555 dg
25 --> 0.25
27 --> 0.25
29 --> 0.9833333333333334 dg
31 --> 0.2222222222222222
33 --> 0.6222222222222222
35 --> 0.3333333333333333
37 --> 0.5
courburemax = 0.9833333333333334
```

```
n = 78
```

```
3 --> 0.1777777777777778
5 --> 0.6777777777777777 dg
7 --> 1.0 dg
9 --> 0.3888888888888889
11 --> 1.1666666666666667 dg
13 --> 0.7222222222222223
15 --> 0.0
17 --> 1.1777777777777778 dg
19 --> 0.8444444444444446 dg
21 --> 0.5
23 --> 0.7222222222222222
25 --> 0.6666666666666666
27 --> 0.7222222222222223
29 --> 0.6666666666666666
31 --> 1.1777777777777778 dg
33 --> 0.1777777777777778
35 --> 0.5
37 --> 1.2222222222222222 dg
39 --> 0.3333333333333333
courburemax = 1.2222222222222222
```

```
n = 80
3 --> 0.2777777777777773
5 --> 0.1333333333333333
7 --> 0.9444444444444445 dg
9 --> 0.9166666666666666
11 --> 0.5555555555555556
13 --> 1.0833333333333333 dg
15 --> 0.5
17 --> 0.2777777777777773
19 --> 1.2166666666666668 dg
21 --> 0.4166666666666667
23 --> 0.7777777777777777
25 --> 0.8055555555555555
27 --> 0.4166666666666667
29 --> 0.9166666666666666
31 --> 1.0833333333333333
33 --> 0.4111111111111104
35 --> 0.0
37 --> 1.4444444444444446 dg
39 --> 0.5555555555555556
courburemax = 1.4444444444444446
```

```
n = 82
3 --> 0.4722222222222215 dg
5 --> 0.3333333333333333
7 --> 0.0
9 --> 0.4722222222222215
11 --> 1.4166666666666667 dg
13 --> 0.3055555555555555
15 --> 0.4166666666666667
17 --> 0.6111111111111111
19 --> 0.1944444444444442
21 --> 0.6666666666666666
23 --> 0.8055555555555555 dg
25 --> 0.4166666666666667
27 --> 0.1111111111111111
29 --> 0.9444444444444443 dg
31 --> 0.9444444444444445
33 --> 0.1944444444444442
35 --> 0.3333333333333333
37 --> 0.2777777777777773
39 --> 0.6111111111111111
41 --> 1.111111111111111 dg
courburemax = 1.4166666666666667
```

```
n = 84
3 --> 0.6388888888888888
5 --> 0.7333333333333334 dg
7 --> 0.8333333333333334
9 --> 0.2333333333333333
11 --> 1.138888888888889 dg
13 --> 1.3333333333333333 dg
```

```
15 --> 0.3888888888888889
17 --> 1.027777777777778 dg
19 --> 0.7333333333333334
21 --> 0.4444444444444444
23 --> 1.177777777777778 dg
25 --> 0.6944444444444445
27 --> 0.7222222222222223
29 --> 0.8888888888888888
31 --> 1.138888888888889 dg
33 --> 0.6777777777777777
35 --> 0.5
37 --> 1.066666666666667 dg
39 --> 0.1944444444444442
41 --> 1.333333333333333 dg
courburemax = 1.333333333333333
```

n = 86

```
3 --> 0.3888888888888889 dg
5 --> 0.15
7 --> 0.916666666666666 dg
9 --> 0.25
11 --> 0.15
13 --> 1.055555555555556 dg
15 --> 0.6666666666666666
17 --> 0.4166666666666667
19 --> 1.066666666666667 dg
21 --> 0.0
23 --> 0.2222222222222222
25 --> 0.8166666666666668
27 --> 0.4166666666666667
29 --> 0.9166666666666666
31 --> 0.8333333333333334
33 --> 0.3722222222222223
35 --> 0.0
37 --> 0.9166666666666666
39 --> 0.3999999999999997
41 --> 0.1666666666666666
43 --> 1.555555555555556 dg
courburemax = 1.555555555555556
```

n = 88

```
3 --> 0.4444444444444444
5 --> 0.4555555555555555 dg
7 --> 0.5
9 --> 0.5
11 --> 0.8333333333333334
13 --> 0.1222222222222223
15 --> 0.4444444444444444
17 --> 1.277777777777778 dg
19 --> 0.2888888888888886
21 --> 0.5
23 --> 0.6666666666666666
```

```

25 --> 0.0
27 --> 0.6222222222222222
29 --> 0.9444444444444445 dg
31 --> 0.9444444444444443
33 --> 0.1222222222222223
35 --> 0.3333333333333333
37 --> 0.8333333333333334
39 --> 0.1666666666666666
41 --> 0.9555555555555556 dg
43 --> 0.4444444444444444
courburemax = 1.277777777777778

n = 90
3 --> 0.4999999999999994
5 --> 0.9
7 --> 0.7777777777777778 dg
9 --> 0.6666666666666666
11 --> 1.1666666666666667 dg
13 --> 0.7777777777777777
15 --> 0.3999999999999997
17 --> 1.0000000000000002 dg
19 --> 1.3166666666666667 dg
21 --> 0.4166666666666667
23 --> 1.0277777777777777 dg
25 --> 0.75
27 --> 0.2777777777777773
29 --> 1.3166666666666667 dg
31 --> 1.138888888888889 dg
33 --> 0.6777777777777777
35 --> 0.5
37 --> 1.027777777777778 dg
39 --> 0.6666666666666666
41 --> 0.9166666666666666
43 --> 1.177777777777778 dg
45 --> 0.2222222222222222
courburemax = 1.3166666666666667

n = 92
3 --> 0.3722222222222222 dg
5 --> 0.1777777777777778
7 --> 0.6666666666666666
9 --> 0.4166666666666667
11 --> 0.8333333333333334
13 --> 1.0833333333333333 dg
15 --> 0.0
17 --> 0.1777777777777778
19 --> 1.038888888888889 dg
21 --> 0.6666666666666666
23 --> 0.4166666666666667
25 --> 0.8333333333333334
27 --> 0.2222222222222222
29 --> 0.1944444444444442
31 --> 1.511111111111111 dg
33 --> 0.3722222222222222
35 --> 0.0

```

```
37 --> 0.8888888888888888
39 --> 0.3611111111111111
41 --> 0.8888888888888888
43 --> 0.8611111111111112
45 --> 0.1777777777777778
courburemax = 1.511111111111111
```

```
n = 94
3 --> 0.4444444444444444
5 --> 0.4666666666666666 dg
7 --> 0.3333333333333333
9 --> 0.5
11 --> 0.9166666666666666 dg
13 --> 0.5833333333333334
15 --> 0.5
17 --> 0.6666666666666666
19 --> 0.1333333333333333
21 --> 0.4444444444444444
23 --> 1.2777777777777777 dg
25 --> 0.1388888888888887
27 --> 0.4722222222222227
29 --> 0.8333333333333334
31 --> 0.4444444444444444
33 --> 0.5777777777777777
35 --> 0.3333333333333333
37 --> 0.8333333333333334
39 --> 0.1388888888888887
41 --> 0.9166666666666666 dg
43 --> 0.9444444444444443
45 --> 0.0
47 --> 0.8000000000000002 dg
courburemax = 1.2777777777777777
```

```
n = 96
3 --> 0.5
5 --> 0.5
7 --> 0.75 dg
9 --> 0.5277777777777778
11 --> 0.9166666666666666
13 --> 0.8333333333333331 dg
15 --> 0.4166666666666667
17 --> 1.0277777777777778 dg
19 --> 0.75
21 --> 0.0
23 --> 1.0 dg
25 --> 0.9166666666666666
27 --> 0.3611111111111111
29 --> 1.1666666666666667 dg
31 --> 0.7777777777777777
33 --> 0.2222222222222222
35 --> 0.5
37 --> 1.0277777777777778 dg
39 --> 0.6666666666666666
41 --> 0.6111111111111112
43 --> 1.138888888888888 dg
45 --> 0.2777777777777773
47 --> 0.75
courburemax = 1.1666666666666667
```

```

n = 98
3 --> 0.19444444444444442
5 --> 0.23333333333333333
7 --> 1.1666666666666667
9 --> 0.39999999999999997
11 --> 0.69444444444444445
13 --> 1.0555555555555556
15 --> 0.3888888888888889
17 --> 0.69444444444444443
19 --> 1.0666666666666667 dg
21 --> 0.5
23 --> 0.23333333333333333
25 --> 0.8611111111111112
27 --> 0.888888888888889
29 --> 0.5555555555555556
31 --> 1.3611111111111114 dg
33 --> 0.23333333333333333
35 --> 0.0
37 --> 1.4000000000000001 dg
39 --> 0.3611111111111111
41 --> 0.888888888888889
43 --> 1.0555555555555556
45 --> 0.19444444444444442
47 --> 0.7333333333333334
49 --> 0.8333333333333334
courburemax = 1.4000000000000001

```

Comme attendu, le nombre dont la moyenne des courbures en x et $n - x$ est maximum est toujours un décomposant de Goldbach de n .

Cette idée a longtemps été un objectif à atteindre et c'est cette phrase de l'émission : "il faut prendre la moyenne des courbures" qui a enfin permis de la mettre en oeuvre. Voici une surface bosselée qui montre les trous correspondant aux décompositions de Goldbach des nombres : dans ce cas, qui ne correspond pas aux courbures telles qu'on les a calculées ci-dessus, la troisième coordonnée, l'altitude de la surface en un point entier (x, y) du maillage, est la somme de leur somme des diviseurs $\sigma(x) + \sigma(y)$. Les décompositions concernant un nombre n sont à trouver sur la diagonale d'équation $x + y = n$. La pente est ici aussi maximale pour les décompositions de Goldbach (i.e. sur la diagonale correspondant à n) car la somme des diviseurs d'un nombre premier est un minimum local ($\sigma(x) = x + 1$ pour x premier et $\sigma(x) > x + 1$ pour x composé).

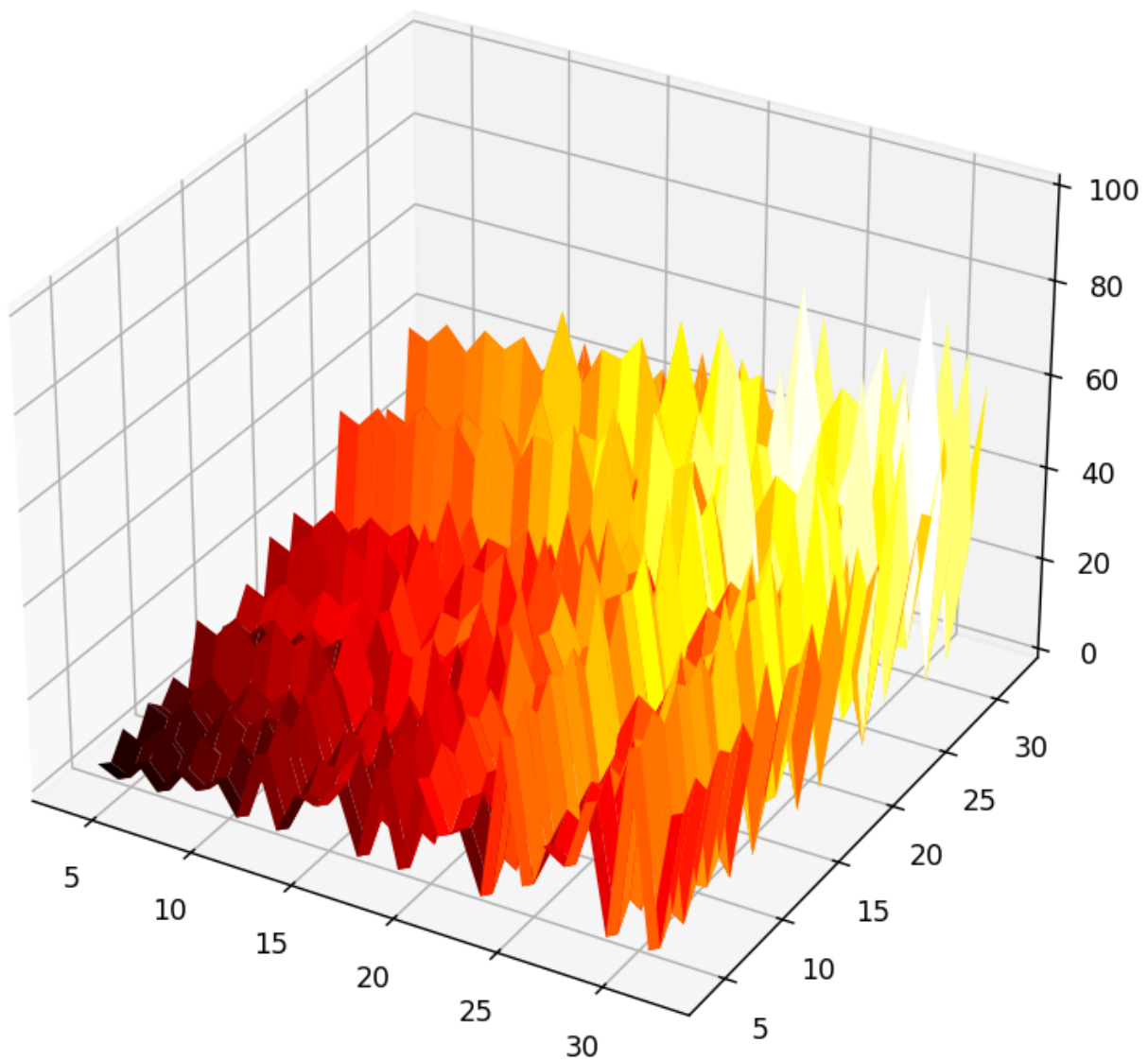


FIGURE 1 : Surface bosselée avec trous des décompositions de Goldbach vue de 3/4.

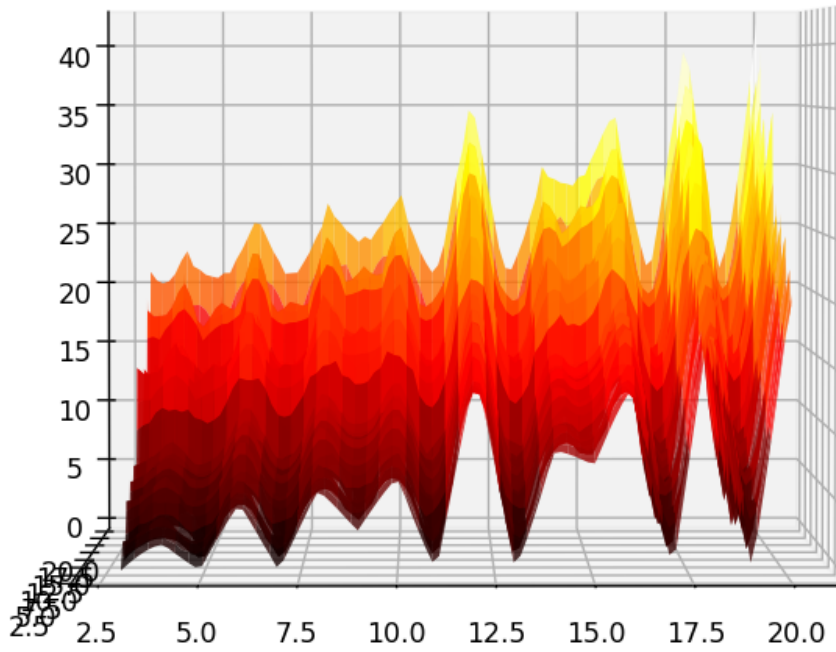


FIGURE 2 : Surface bosselée avec trous des décompositions de Goldbach vue de profil.

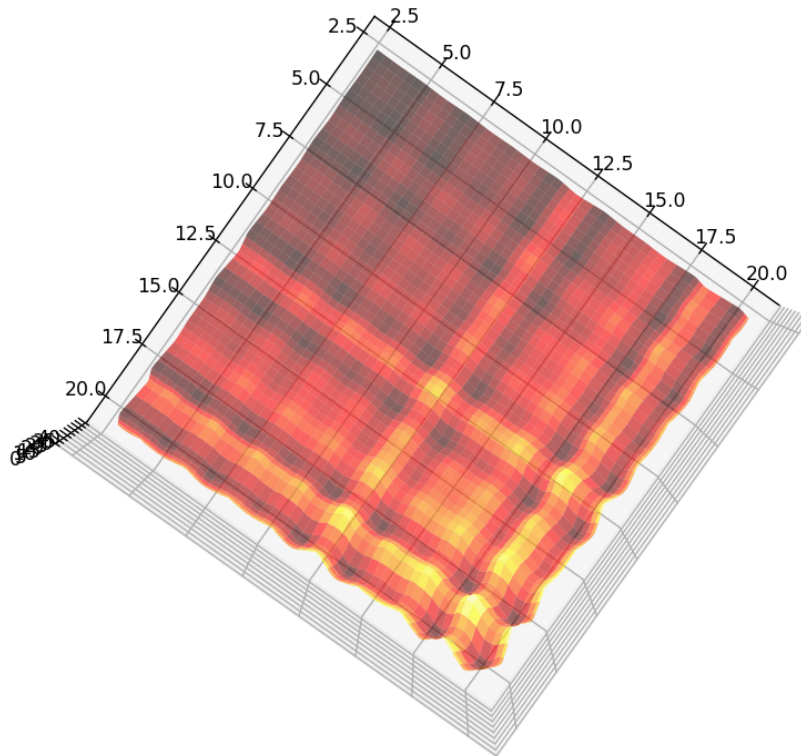
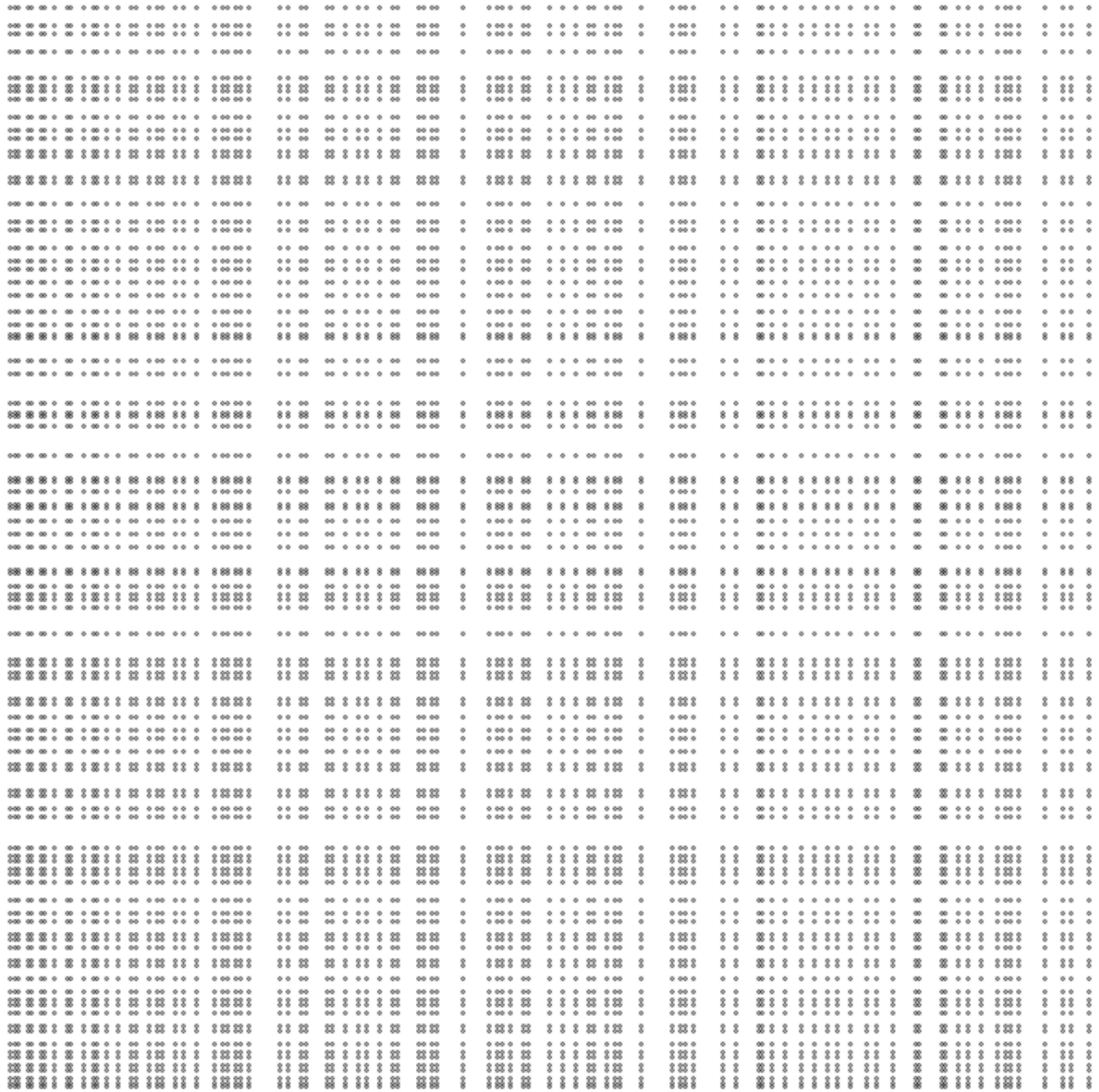


FIGURE 3 : Surface bosselée avec trous des décompositions de Goldbach vue de dessus.



0

200

400

FIGURE 4 : Surface bosselée vue de dessus jusqu'à $n = 500$.