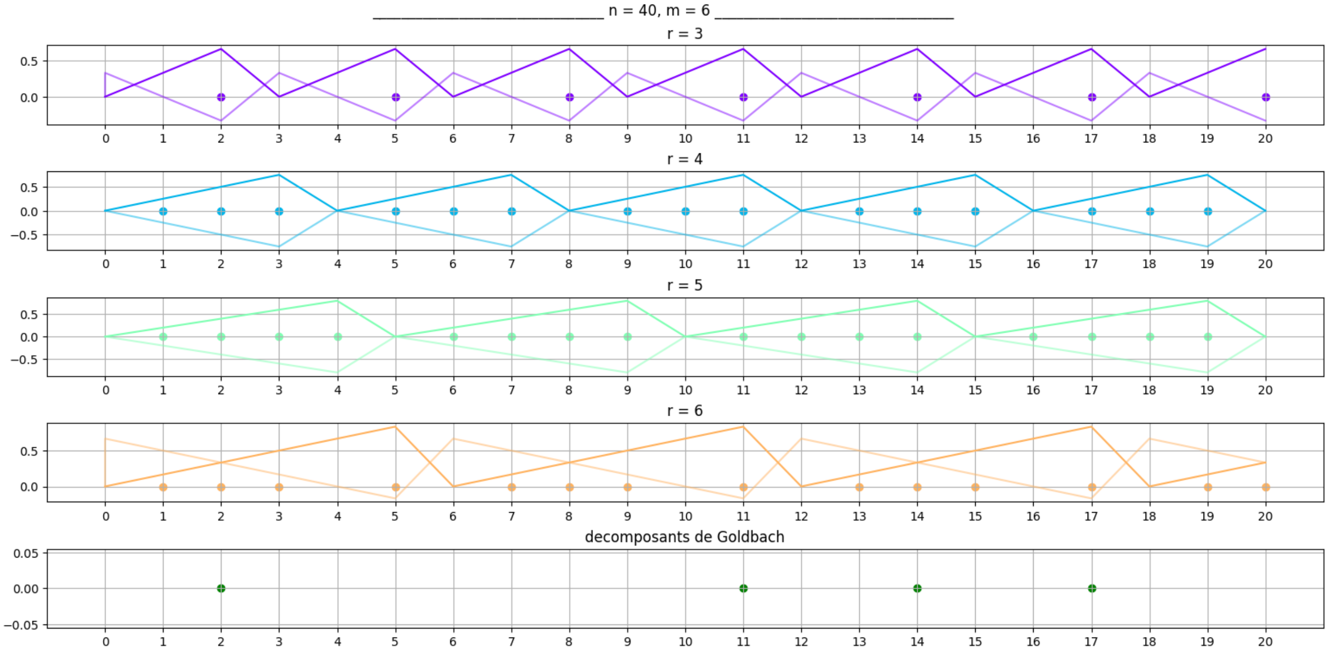
Papillons et sapins (Denise Vella-Chemla, 4.8.2024)

On cherche à démontrer que tout nombre pair n supérieur à 4 admet un décomposant de Goldbach, i.e. un nombre premier x dont le complémentaire à n, $n - x$, est premier également.

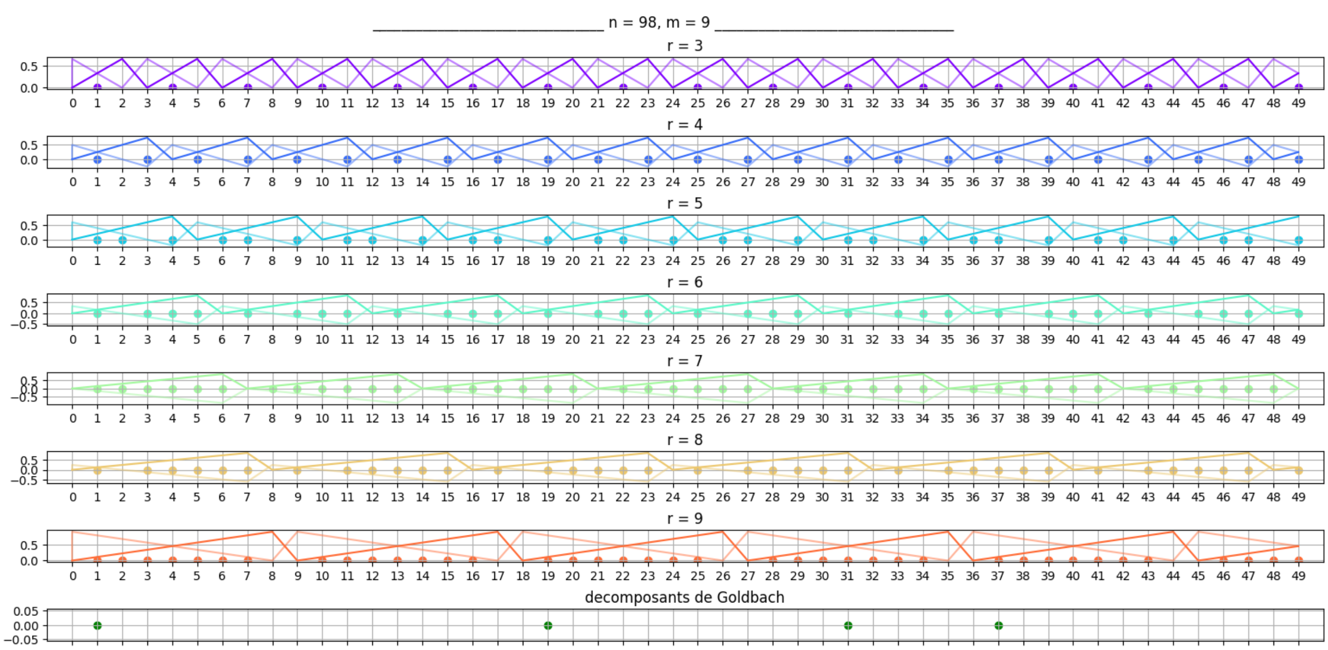
Suite à des travaux récents qui permettaient de caractériser les décomposants de Goldbach en effectuant un petit détour par le plan complexe, on est finalement amenée à utiliser le programme suivant, très simple, qui exprime des conditions booléennes (x est premier, $n - x$ est premier) en utilisant les parties décimales de nombres rationnels.

```
import math
from math import floor, sqrt
import matplotlib.pyplot as plt, matplotlib.cm as cm

def partiedecimale(x):
    return x-int(x)

epsilon = 1e-6
for n in range(2024, 2026, 2):
    milieu = n//2 if n%2 == 0 else n//2+1
    m = int(floor(sqrt(n)))
    fig, axs = plt.subplots(nrows=m+1-3+1, layout='constrained')
    fig.set_figwidth(18)
    fig.set_figheight(12)
    produitglobal = [True for x in range(milieu+1)]
    for r in range(3, m+1):
        c = cm.rainbow((r-3)/(m+1-3))
        xs, y1, y2 = [0], [0], [0]
        for x in range(milieu+1):
            cond1 = partiedecimale(x/r)
            cond2 = partiedecimale(n/r)-partiedecimale(x/r)
            produit = abs(cond1)>epsilon and abs(cond2)>epsilon
            if produit:
                axs[r-3].scatter(x, 0, color=c)
            produitglobal[x] &= produit
            xs += [x]
            y1 += [cond1]
            y2 += [cond2]
        axs[r-3].plot(xs, y1, color=c)
        axs[r-3].plot(xs, y2, color=c, alpha=0.5)
        axs[r-3].set_title(f'r = r')
        axs[r-3].set_xticks(range(0, milieu+1))
        axs[r-3].set_xlim(-1, milieu+1)
        axs[r-3].grid()
    xs = [k for k, p in enumerate(produitglobal) if p]
    ys = [0 for x in xs]
    axs[-1].scatter(xs, ys, color='green')
    axs[-1].set_title(f'decomposants de Goldbach')
    axs[-1].set_xticks(range(0, milieu+1))
    axs[-1].set_xlim(-1, milieu+1)
    axs[-1].grid()
    fig.suptitle(f' n = n, m = m '.center(80, '_'))
    plt.show()
```

Ci-dessous le dessin des différentes courbes, indicées par l'entier r, r variant entre 3 et $\lfloor\sqrt{n}\rfloor$, pour $n = 40$, puis pour $n = 98$.



L'une des courbes, selon un certain entier r, montre la variation de la partie décimale de $\dfrac{x}{r}$, qu'on peut écrire $\dfrac{x}{r}$ mod 1, et l'autre courbe (de même couleur, mais un peu plus claire) montre la variation de la différence des parties décimales de $\dfrac{n}{r}$ et de $\dfrac{x}{r}$, qu'on peut écrire $\dfrac{n}{r}$ mod $1 - \dfrac{x}{r}$ mod 1. Les courbes sont en miroir l'une de l'autre, autour d'un axe de symétrie. On marque, pour chaque r, les points d'images non nulles. On s'est ainsi placé, en étudiant les restes modulo 1, dans $\mathbb{R}/\mathbb{Z}$, dans lequel deux nombres appartiennent à la même classe d'équivalence s'ils ont même partie décimale.



La dernière courbe, en bas de chaque graphique, montre les décomposants de Goldbach de n. Ce sont les nombres dont les images selon tout r sont non nulles.

On exécute le programme ci-dessus, pour les nombres pairs de 10 à 102, son résultat d'exécution est consigné dans le fichier https://denisevellachemla.eu/harmonies-colorees.pdf (on a omis les courbes pour le nombre premier 2 pour gagner de la place ; celles-ci font éliminer les décomposants pairs).

L'étude des différentes courbes montre que :

- chaque courbe est périodique, de période r ;

- lorsque r divise n, un seul nombre est d'image nulle par période de longueur r ;

- lorsque r ne divise pas n, deux nombres sont d'image nulle par période de longueur r ; on va donc considérer qu'on élimine au maximum deux nombres par période pour tout r compris entre 2 et $\sqrt{n}$ (on cherche une borne qui maximise le nombre de nombres éliminés - i.e. qui ne peuvent pas être des décomposants de Goldbach de n, ce qui permettra de trouver une borne qui minimise le nombre de nombres conservés - i.e. ce qui permettra de dire une phrase comme “Tout nombre pair a forcément, au moins, tant de décompositions de Goldbach”) [1].

- il y a $\dfrac{n}{2r}$ périodes de longueur r sur l'intervalle $\left[0, \dfrac{n}{2}\right]$;

- le nombre de nombres d'image nulle sur chaque graphique (on appelle aussi cela le cardinal de l'intersection des noyaux des deux fonctions présentées par graphique) est ainsi inférieur à $\left\lceil \dfrac{n}{2r} \right\rceil \times 2$;

- si l'on recense dans une liste, en les ajoutant à la liste au fur et à mesure, les nombres d'image nulle, on voit que le nombre de nombres supplémentaires ajoutés à cette liste pour le niveau r ne peut excéder $\left\lceil \dfrac{n}{2r} \right\rceil \times 2 \times \dfrac{1}{(r-1)!}$. Voyons un exemple : pour $n = 98$, les nombres d'image nulle pour $r = 3$ sont les nombres de la liste

$$\begin{aligned} L_3 = \{&2, 3, 5, 6, 8, 9, 11, 12, 14, 15, 17, 18, 20, 21, 23, 24,\\ &26, 27, 29, 30, 32, 33, 35, 36, 38, 39, 41, 42, 44, 45, 47, 48\} \end{aligned}$$

tandis que ceux d'image nulle pour $r = 5$ sont les nombres de la liste

$$L_5 = \{3, 5, 8, 10, 13, 15, 18, 20, 23, 25, 28, 30, 33, 35, 38, 40, 43, 45, 48\}.$$

Si on barre dans L_5 les nombres qui appartiennent à L_3, il est clair que 2 “paquets de 2 nombres” sur 3 sont à barrer (i.e. appartenaient déjà à L_3) quand 1 paquet de 2 nombres sur 3 va être à considérer en plus, car ces nombres n'appartiennent pas à L_3.

$$L_5 = \{\cancel{3}, \cancel{5}, \cancel{8}, 10, 13, \cancel{15}, \cancel{18}, \cancel{20}, \cancel{23}, 25, 28, \cancel{30}, \cancel{33}, \cancel{35}, \cancel{38}, 40, 43, \cancel{45}, \cancel{48}\}$$

[1] La démonstration, par Leila Schneps, du fait que les nombres trouvés ici sont bien des décompositions de Goldbach de n (i.e. des sommes de deux nombres premiers) est consultable à cette adresse https://denisevellachemla.eu/jade1.pdf.

C'est cette petite découverte récente qui permet de maximiser le nombre de nombres d'image nulle par l'expression :

$$\sum_{\substack{2 \leq r \leq \lfloor \sqrt{n} \rfloor \\ r \in \mathbb{N}}} \left\lceil \frac{n}{2r} \right\rceil \times \frac{2}{(r-1)!} \simeq \sum_{\substack{2 \leq r \leq \lfloor \sqrt{n} \rfloor \\ r \in \mathbb{N}}} \frac{n}{r(r-1)!} \simeq \sum_{\substack{2 \leq r \leq \lfloor \sqrt{n} \rfloor \\ r \in \mathbb{N}}} \frac{n}{r!}$$

On utilise le programme ci-dessous (qui distingue les diviseurs de n pour lesquels un nombre est éliminé par période, des non-diviseurs pour lesquels 2 nombres sont éliminés par période) :

```
import math
from math import sqrt,floor,factorial

for n in range(10,10002,2):
     print('n = ',n,' ---> ',end='')
     racine = int(floor(sqrt(n)))
     somme = 0
     for r in range(2,racine+1):
          if n%r == 0:
               somme = somme+n/(2*factorial(r))
          else:
               somme = somme+n/factorial(r)
     print(somme, ' a comparer a ',n/2)
```

dont le résultat est :

```
n = 10 ---> 4.166666666666667 a comparer a 5.0
n = 12 ---> 4.0 a comparer a 6.0
n = 14 ---> 5.833333333333334 a comparer a 7.0
n = 16 ---> 6.999999999999999 a comparer a 8.0
n = 18 ---> 6.75 a comparer a 9.0
n = 20 ---> 8.75 a comparer a 10.0
n = 22 ---> 10.083333333333332 a comparer a 11.0
n = 24 ---> 8.5 a comparer a 12.0
n = 26 ---> 12.133333333333333 a comparer a 13.0
n = 28 ---> 12.483333333333334 a comparer a 14.0
n = 30 ---> 11.375 a comparer a 15.0
n = 32 ---> 14.266666666666666 a comparer a 16.0
n = 34 ---> 15.866666666666667 a comparer a 17.0
n = 36 ---> 13.075000000000001 a comparer a 18.0
n = 38 ---> 17.786111111111108 a comparer a 19.0
n = 40 ---> 17.722222222222225 a comparer a 20.0
n = 42 ---> 16.129166666666666 a comparer a 21.0
n = 44 ---> 19.677777777777777 a comparer a 22.0
n = 46 ---> 21.53055555555556 a comparer a 23.0
n = 48 ---> 17.433333333333334 a comparer a 24.0
n = 50 ---> 23.20436507936508 a comparer a 25.0
n = 52 ---> 23.265873015873012 a comparer a 26.0
n = 54 ---> 20.748214285714287 a comparer a 27.0
n = 56 ---> 25.050000000000004 a comparer a 28.0
n = 58 ---> 27.158730158730158 a comparer a 29.0
```

```
n = 60 ---> 21.55357142857143 a comparer a 30.0
n = 62 ---> 29.03174603174603 a comparer a 31.0
n = 64 ---> 28.635714285714283 a comparer a 32.0
n = 66 ---> 25.36056547619048 a comparer a 33.0
n = 68 ---> 30.426289682539686 a comparer a 34.0
n = 70 ---> 32.48090277777777 a comparer a 35.0
n = 72 ---> 26.165178571428577 a comparer a 36.0
n = 74 ---> 34.65262896825397 a comparer a 37.0
n = 76 ---> 34.005853174603175 a comparer a 38.0
n = 78 ---> 29.97157738095238 a comparer a 39.0
n = 80 ---> 35.46130952380953 a comparer a 40.0
n = 82 ---> 38.39908509700175 a comparer a 41.0
n = 84 ---> 30.518981481481482 a comparer a 42.0
n = 86 ---> 40.27221119929454 a comparer a 43.0
n = 88 ---> 39.374349647266314 a comparer a 44.0
n = 90 ---> 34.2077132936508 a comparer a 45.0
n = 92 ---> 41.16523368606702 a comparer a 46.0
n = 94 ---> 44.01846340388006 a comparer a 47.0
n = 96 ---> 34.88716931216931 a comparer a 48.0
n = 98 ---> 45.88186728395062 a comparer a 49.0
n = 100 ---> 44.32816633597884 a comparer a 50.0
...
n = 1000 ---> 443.26928987879825 a comparer a 500.0
n = 10000 ---> 4432.692898787745 a comparer a 5000.0
n = 100000 ---> 44326.92898787744 a comparer a 50000.0
n = 1000000 ---> 443269.28987877443 a comparer a 500000.0
```

Ces résultats montrent que le nombre maximum de nombres éliminés (nombre de nombres d'image nulle) est toujours inférieur à $\frac{n}{2}$.