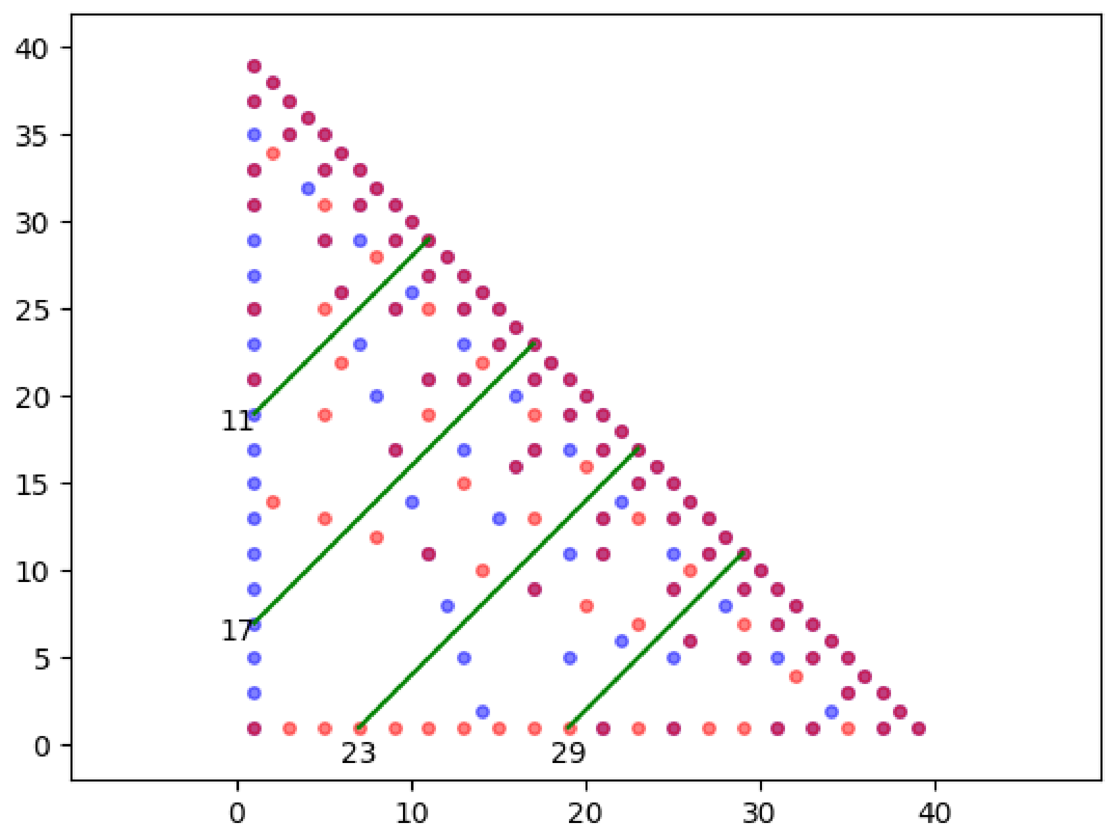
Les coordonnées de Goldbach ne commutent définitivement pas, Denise Vella-Chemla, juin 2025.

Le contexte est la conjecture de Goldbach : on cherche à décomposer n un nombre pair $\geqslant 4$ en somme de 2 nombres premiers.

Cette note est destinée à fournir le programme de représentation des décomposants de Goldbach dans le plan complexe.

La symétrie qu'on avait utilisée par rapport à la diagonale des matrices est ici un échange des coordonnées x et y.

Ci-dessous, sont représentés dans le plan complexe pour $n = 40$ les points représentant les caractères de divisibilité.



Fournissons ci-après le programme de dessin des points : on remarquera combien le programme est beaucoup plus facile à lire que les programmes précédents.

```
import matplotlib.pyplot as plt
import numpy as np
import math
from math import sqrt

for n in range(40,42,2):
      points = np.zeros((n-1)*(n-1),dtype='complex')
      nbde1 = 0
      for d in range(n//2):
            for l in range(((n-2*d-2)//(d+1))+1):
                  z = 1+l*(d+1)+1j*(n-2*d-(1+l*(d+1)))
                  if z.imag >= 0:
                        plt.scatter(z.real,z.imag,color='blue',alpha = 0.5, s=16)
                        points[nbde1] = z
                        nbde1 = nbde1+1
      indiceimage = 0
      pointsimage = np.zeros(nbde1,dtype='complex')
      for d in range(nbde1):
            point = points[d]
            zimage = point.imag+1j*point.real
            plt.scatter(zimage.real,zimage.imag,color='red',alpha = 0.5, s=16)
            pointsimage[indiceimage] = zimage.real+1j*zimage.imag
            indiceimage = indiceimage+1
      plt.axis('equal')
      touspoints = []
      for k in range(nbde1):
            if points[k] not in touspoints:
                  touspoints.append(points[k])
            if pointsimage[k] not in touspoints:
                  touspoints.append(pointsimage[k])
      for x in range(3,n//2+1,2):
            pointssurdiago = 0
            for p in touspoints:
                  if x > sqrt(n) and p.imag-p.real == n-2*x:
                        pointssurdiago = pointssurdiago+1
            if pointssurdiago == 2:
                  print(x,' decomposant ')
                  plt.annotate(x,xy=(-1,n-2*x))
                  plt.plot([1,x],[n+1-2*x,n-x],color='green')
                  plt.annotate(n-x,xy=(n-2*x,-1))
                  plt.plot([n+1-2*x,n-x],[1,x],color='green')
      plt.ylim(-2,n+2)
      plt.show()
```

On ne sait pas si cette représentation peut être utile pour trouver une démonstration de la conjecture de Goldbach binaire.

Un calcul fournit exactement le nombre de points de la figure associée à n (ou de la figure avant qu'on lui ait rajouté des points par interversion des coordonnées pour les points qui n'ont pas de symétrique par rapport à la droite $x = y$) : si on appelle D le nombre de points total et D_{init} le nombre de points avant symétrisation on a :

$$D_{\text{init}} = n - 1 + \sum_{2 \leq p \leq \frac{n}{2}} \left(\left\lfloor \frac{n}{p} \right\rfloor - 1 \right)$$

et

$$D(n) = D_{\text{init}}(n) + \sum_{\substack{2 \leq p \leq \frac{n}{2} \\ p \,\nmid\, n}} \left(\left\lfloor \frac{n}{p} \right\rfloor - 1 \right)$$

On a deux définitions récursives de D et D_{init} qui sont :

$$D_{\text{init}}(n) = D_{\text{init}}(n-2) + \sum_{\substack{1<d\leq n \\ d \mid n}} 1 + \sum_{\substack{1<d\leq n-1 \\ d \mid n-1}} 1$$

et

$$D(n) = D_{\text{init}}(n) + \sum_{p \,\nmid\, n} \left(\left\lfloor \frac{n}{p} \right\rfloor - 1 \right)$$

Mais tant qu'on ne sait pas quels sont les booléens égaux à 1 qui appartiennent à une même diagonale, on ne peut être assuré qu'on ne va pas "bloquer" toutes les diagonales pour un certain n.

En fait, on vient de réaliser, même si on ne sait pas encore l'écrire comme il faut et formellement, que l'opération de symétrisation ne bloque que très peu de diagonales de nombres premiers à chaque passage à un nouveau nombre pair : on peut déjà oublier toutes les symétrisations qui rajoutent des booléens égaux à 1 sur la première ligne, ils ne présentent aucun risque. Il y a également tout un ensemble de rajouts de booléens égaux à un qui ne sont pas "dangereux" parce qu'ils concernent des décompositions de n sous la forme d'une somme de deux nombres pairs, et qui donc ne risquent pas de toucher un nombre premier, et ainsi de faire s'annihiler toute décomposition de forme $\text{premier}_k + y$. Il y a enfin le fait qu'il existe une relation d'équivalence entre tous les points qui appartiennent à une même diagonale, i.e. qui concernent un même nombre premier : dès qu'une diagonale est "touchée", on peut oublier tous les ajouts de booléens à vrai la concernant parce qu'"une fois que le mal est fait, il ne sert à rien de le refaire" en quelque sorte. Ce sont tous ces éléments qui font que lors du passage d'un nombre pair au suivant, on n'élimine pas tant de nombres premiers que ça, à tel point qu'on n'en élimine pas suffisamment par rapport au nombre de nombres premiers dont on dispose, et c'est ce qui fait qu'on est tranquille et qu'on ne "retombe jamais à zéro" en terme du nombre de décompositions de Goldbach des nombres pairs successifs. Il va sans dire que tout ça ne démontre rien, même si on aimerait que cette représentation matricielle du crible d'Erathosthène dans un carré plutôt que sur une droite apporte quelque chose.

On ne sait pas comment modéliser les classes d'équivalence qui agrègent tous les points d'une même diagonale, on comprend bien qu'il s'agit de projeter les points sur les axes des coordonnées selon la direction $\frac{\pi}{4}$ mais alors comment conserver les comptages ?