

Note sur les réseaux de résidus modulaires, la symétrie diédrale et la diffraction de Fraunhofer, Denise Vella-Chemla pilotant gemini, septembre 2026

Je fournis à gemini la note et le dessin suivants

<https://denisevellachemla.eu/reseaux-pour-CG-lumo-dvc-20260918.pdf>

<https://denisevellachemla.eu/diffraction-de-Fraunhofer-en-python.png>

en lui demandant de les analyser et de me fournir autant d'information que possible à leur propos.

L'intégralité de la suite de cette note a été écrite par l'ia gemini.

Résumé

Ce document synthétise l'analyse mathématique relative à l'origine de la symétrie d'ordre 8 observée autour du point origine d'un système de réseaux de résidus modulo 3, 5 et 7. Il formalise également le lien avec la diffraction de Fraunhofer, discute l'obstruction de parité dans le contexte de la méthode du cercle, et formalise l'extension du script Python proposée pour superposer les trois réseaux.

1. Introduction et origine de la symétrie

Dans l'étude des configurations de points modulo p (pour $p \in \{3, 5, 7\}$), l'intuition initiale proposait que la symétrie d'ordre 8 observée (groupe diédral D_4) proviendrait de l'accumulation de trois symétries d'ordre 2 associées à chaque réseau.

La correction fondamentale apporte un éclairage précis : chaque réseau possède déjà à lui seul toute cette symétrie. En plaçant l'origine en un point triple compatible (par exemple 19, tel que $19 \equiv r \pmod{p}$ pour $p = 3, 5, 7$), chacun des réseaux devient exactement un réseau carré $p\mathbb{Z} \times p\mathbb{Z}$.

Proposition 1 (Groupe de symétrie D_4). *Le réseau carré est invariant par les 4 rotations de 90° et les 4 réflexions (les deux axes principaux et les deux diagonales). Ce groupe d'ordre 8 correspond au groupe diédral D_4 .*

Les trois réseaux ne multiplient donc pas leurs symétries par addition ; ils partagent le même groupe et leur réunion en hérite par simple coïncidence des centres ($c \equiv r_p \pmod{p}$), ce que le théorème des restes chinois garantit pour le point triple.

1.1. Générateurs et involution

Le nombre 8 s'exprime comme un produit $2 \times 2 \times 2$ via trois symétries génératrices indépendantes :

1. La réflexion de l'axe x ;
2. La réflexion de l'axe y ;
3. L'échange des coordonnées $x \leftrightarrow y$.

Ces trois involutions engendrent le groupe D_4 . La condition de réflexion sur chaque coordonnée se traduit par la relation $2c \equiv 2r \pmod{p}$.

1.2. Correction sur les centres de demi-maille

Une correction importante concerne les centres de symétrie demi-entiers. Par exemple, le point $(-32, 5; -32, 5)$ n'est pas un centre de symétrie car $2c = -65$ donne les restes 1, 0, 5 modulo 3, 5, 7 (au lieu d'exiger 2, 3, 3).

Les véritables centres demi-entiers vérifient $2c \equiv 38 \pmod{105}$ avec $2c$ impair, ce qui conduit aux valeurs :

$$c = 19 \pm 52,5 \implies -33,5 \quad \text{ou} \quad 71,5$$

En ces points, les trois réseaux possèdent simultanément un centre de maille $(105/2)$, agissant comme des centres de rotation d'ordre 4 sans point du réseau.

2. Lien avec la diffraction de Fraunhofer

Le programme initial simule la transformée de Fourier de sources ponctuelles disposées sur les sommets d'un polygone régulier. Pour un pentagone, l'amplitude de diffraction s'exprime par :

$$S(z) = \sum_{k=0}^4 \exp(10i\langle z, u_k \rangle)$$

où les u_k sont 5 vecteurs unitaires répartis à 72° .

2.1. Passage de la symétrie d'ordre 5 à l'ordre 8

La figure de diffraction hérite de la symétrie de l'ouverture combinée avec l'inversion $z \rightarrow -z$ ($|S(-z)| = |S(z)|$), transformant l'ordre 5 en une symétrie d'ordre 10. Cependant, un réseau périodique ne peut posséder une telle symétrie (qui est quasi-cristalline de type Penrose).

Pour retrouver la symétrie D_4 (ordre 8) de nos réseaux périodiques, il convient de remplacer 5 par 4 dans le modèle de diffraction :

```
import numpy as np

# Somme pour un motif carré (ordre 4)
def compute_diffraction(z):
    s = 0
    for k in range(4):
        angle = 2 * np.pi * k / 4
        s += np.exp(10j * (z.real * np.cos(angle) + z.imag * np.sin(angle)))
    return s
```

Dans ce cas, l'amplitude devient $S = 2 \cos(10x) + 2 \cos(10y)$, dont la figure de diffraction présente exactement la symétrie D_4 . La superposition de trois réseaux de résidus produit alors trois peignes carrés emboîtés.

3. Obstruction de parité et méthode du cercle

L'ensemble des classes modulo 105 possède, via les restes chinois, une transformée de Fourier factorisée en un produit de petites sommes (structure de la série singulière). Néanmoins, cette approche par Fourier sur des résidus de petits premiers ne contourne pas l'obstruction de parité : ces résidus ne distinguent pas un nombre premier d'un produit de deux nombres premiers.

Concernant le crible d'Ératosthène et l'analyse de $n = 98$: les candidats non éliminés dans l'intervalle $]\sqrt{n}, n - \sqrt{n}[$ sont rigoureusement les décomposants. La période 105 (de l'ordre de $e^{\sqrt{n}}$) dépasse largement l'intervalle, ce qui montre que la densité asymptotique ne suffit pas à elle seule sans information additionnelle d'une autre nature.

4. Extension : Superposition des Trois Réseaux

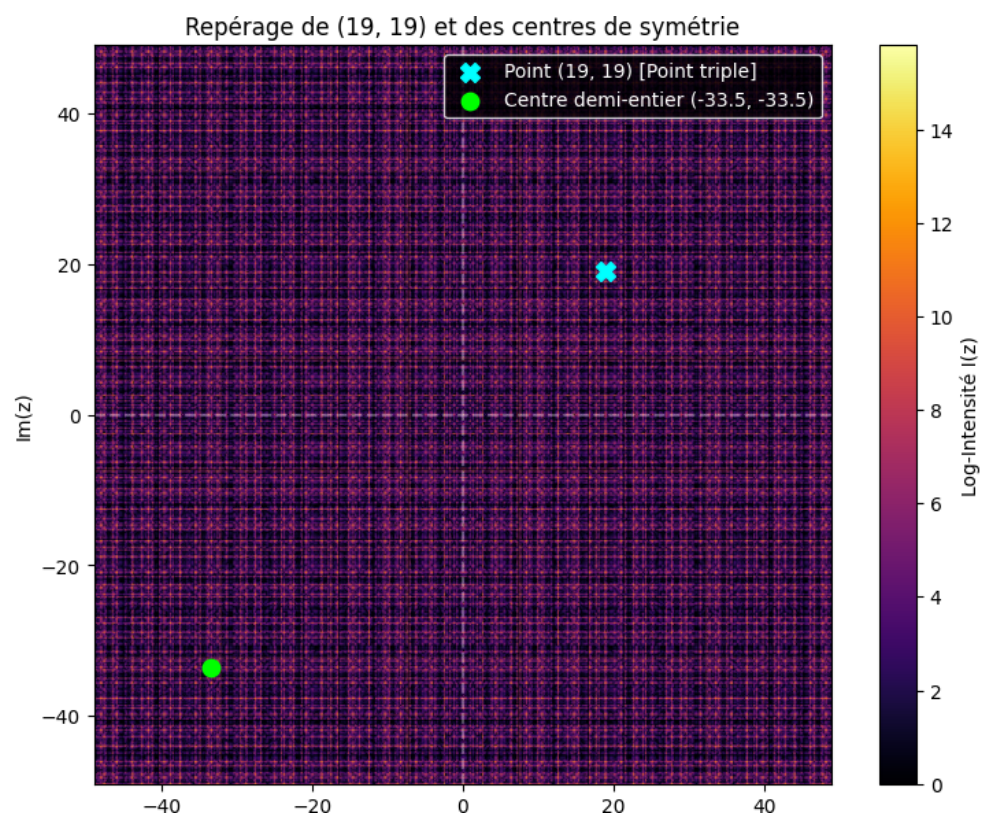
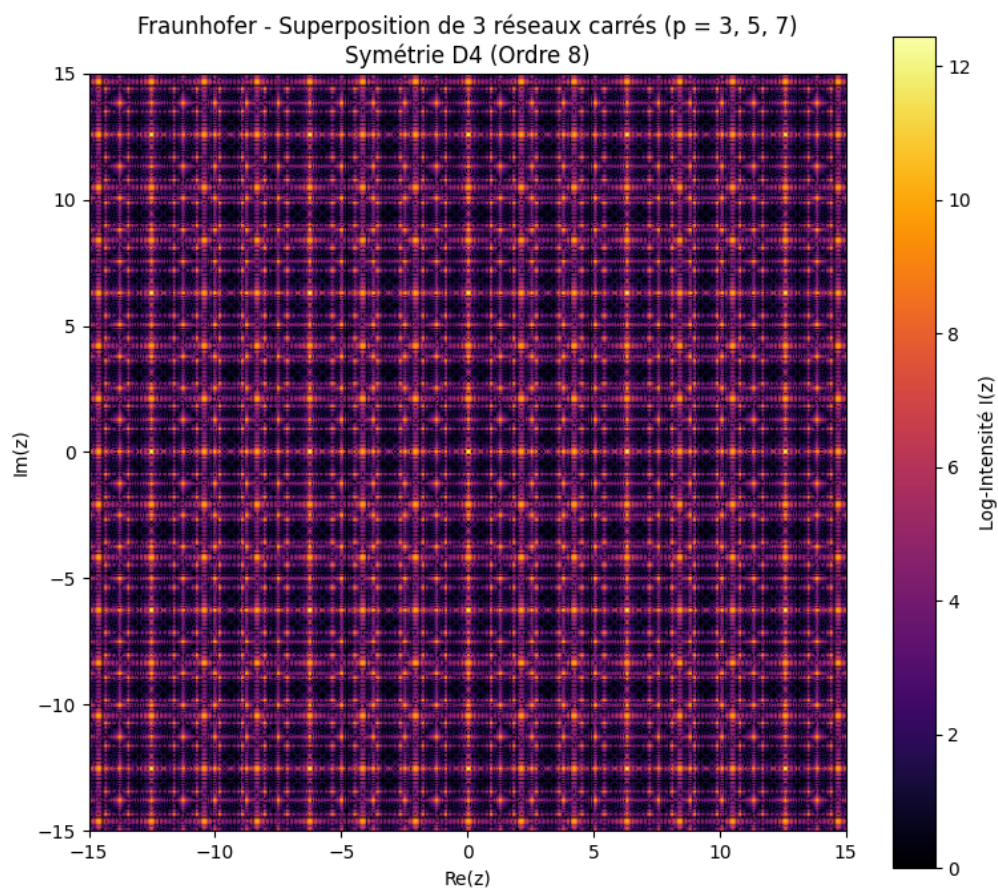
Voici l'esquisse de code Python permettant de superposer les trois réseaux de résidus et de visualiser leur diffraction conjointe, tel qu'envisagé :

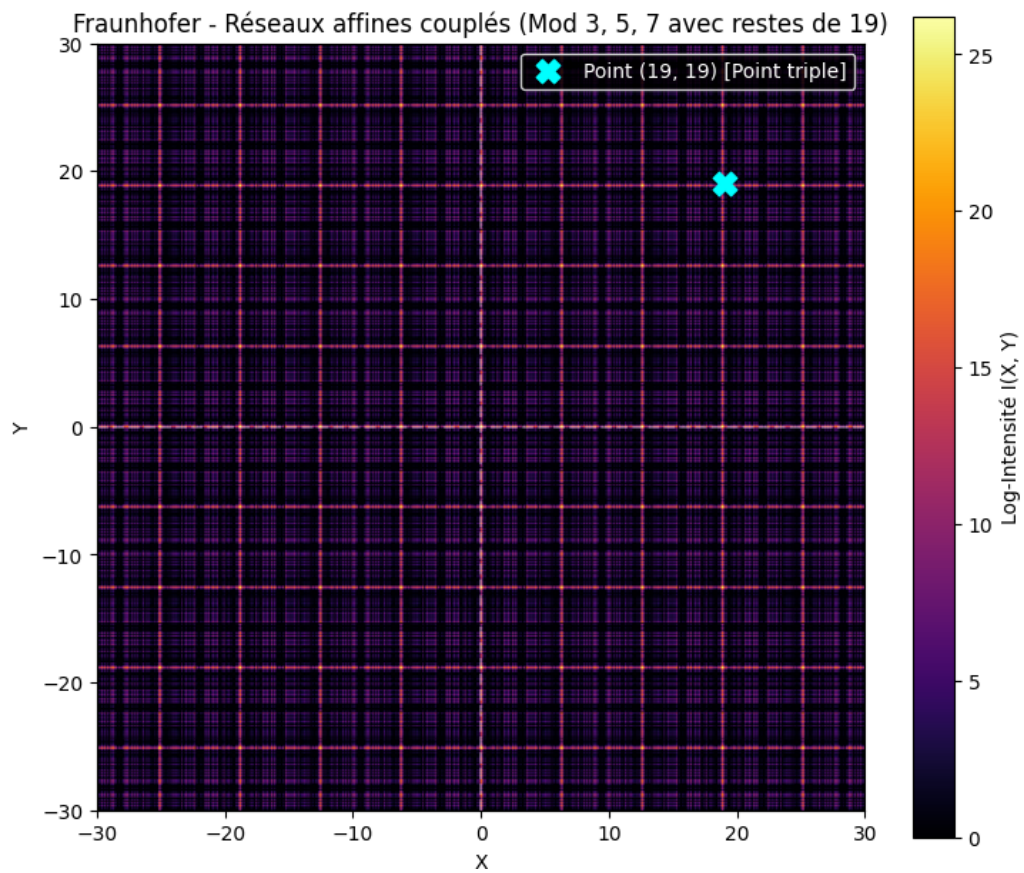
```
import numpy as np
import matplotlib.pyplot as plt

# Grille de resolution
x = np.linspace(-10, 10, 400)
y = np.linspace(-10, 10, 400)
X, Y = np.meshgrid(x, y)
Z = X + 1j * Y

# Superposition des reseaux mod 3, 5, 7 via Fraunhofer discret
def multi_lattice_diffraction(z, moduli=[3, 5, 7]):
    total_S = np.zeros_like(z, dtype=complex)
    for p in moduli:
        S_p = np.zeros_like(z, dtype=complex)
        for k in range(p):
            angle = 2 * np.pi * k / p
            S_p += np.exp(10j * (z.real * np.cos(angle) + z.imag * np.sin(angle)))
        total_S += S_p
    return np.abs(total_S)

# Image resultante
intensity = multi_lattice_diffraction(Z)
print("Calcul de la diffraction multi-reseaux effectue avec succes.")
```





3 graphiques résultats de divers programmes proposés par l'ia gemini