

```

# -*- coding: utf-8 -*-

import matplotlib.pyplot as plt
import numpy as np

phi = (1 + 5 ** 0.5) / 2      # nombre d'or, racine de  $X^2 - X - 1$ 
phi_bar = (1 - 5 ** 0.5) / 2 # son conjugué de Galois (~ -0.618)

# --- Le reseau vu par les deux plongements reels de  $Z[\phi]$  -----
# Un element  $x + y\phi$  (x,y entiers) donne le point (a,b) = (x+y*phi,
x+y*phi_bar).
xs, ys = [], []
for x in range(-10, 11):
    for y in range(-10, 11):
        a = x + phi * y
        b = x + phi_bar * y
        if -8 <= a <= 8 and -8 <= b <= 8:
            xs.append(a)
            ys.append(b)
xs = np.array(xs)
ys = np.array(ys)

plt.figure(figsize=(6.5, 6.5))
plt.scatter(xs, ys, color='red', s=10, zorder=1, label='reseau (t=0)')

# --- Le flot hyperbolique -----
#  $g_t(a,b) = (a \cdot e^t, b \cdot e^{-t})$ . Comme  $(a \cdot e^t) \cdot (b \cdot e^{-t}) = a \cdot b$ , CHAQUE point
# du reseau glisse sur SA PROPRE hyperbole  $a \cdot b = \text{const}$  quand t varie.
#
# Erreur de la version originale : on prenait seulement t dans
# {-2,-1,1,2} (4 valeurs isolees, formule fausse ( $x \cdot t, y \cdot (-t)$ )) et on
# transformait tout le nuage d'un coup -> ca ne peut donner que 4 nuages
# separees, jamais une courbe.
#
# Ce qu'il faut faire : balayer une plage DENSE de t (des centaines de
# valeurs proches les unes des autres) et superposer TOUTES les images
# du reseau entier sur le meme dessin, sans effacer les precedentes.
# C'est cette accumulation de "snapshots" tres rapproches qui dessine
# la courbe (comme une trainee), exactement comme dans l'animation de
# la video.
for t in np.linspace(0.05, 1.6, 5):
    e = np.exp(t)
    a_t, b_t = xs * e, ys / e      # sens "avant" du flot
    m = (a_t >= -8) & (a_t <= 8) & (b_t >= -8) & (b_t <= 8)
    print('t = ', t)
    if t == 0.4375:
        plt.scatter(a_t[m], b_t[m], color='blue', s=25, zorder=2)
    else:
        plt.scatter(a_t[m], b_t[m], color='blue', s=6, zorder=2)
    a_t, b_t = xs / e, ys * e      # sens "arriere" du flot
    m = (a_t >= -8) & (a_t <= 8) & (b_t >= -8) & (b_t <= 8)
    plt.scatter(a_t[m], b_t[m], color='purple', s=6, zorder=2)

plt.axhline(0, color='black', linewidth=0.8)
plt.axvline(0, color='black', linewidth=0.8)
plt.grid(True)
plt.xlim(-8, 8)
plt.ylim(-8, 8)
plt.gca().set_aspect('equal')
plt.title("Flot hyperbolique  $g_t(a,b)=(a \cdot e^t, b \cdot e^{-t})$  sur  $Z[\phi]$ ")
plt.show()

```