import numpy as np

import matplotlib.pyplot as plt

# Parameters

L = 32

timesteps = 120

noise = 0.10

fraction = 1/3 # Fractional twist

# MBT Phase field: initialize with a fractional vortex at (center)

y, x = np.ogrid[:L, :L]

cx, cy = L // 2, L // 2

theta = np.arctan2(y - cy, x - cx)

phase = np.mod(fraction \* theta, 2\*np.pi) + np.random.uniform(-noise, noise, (L, L))

memory = np.zeros((L, L))

# MBT dynamics (simplified)

for t in range(timesteps):

# Simple local averaging + "echo"

avg = (

np.roll(phase, 1, axis=0) + np.roll(phase, -1, axis=0) +

np.roll(phase, 1, axis=1) + np.roll(phase, -1, axis=1)

) / 4

memory = 0.92 \* memory + 0.08 \* np.cos(phase - avg)

phase += 0.06 \* np.sin(avg - phase) + np.random.uniform(-noise, noise, (L, L))

# --- Fractional winding measurement

r = L//3 # Measurement ring radius

ring\_angles = np.linspace(0, 2\*np.pi, 200)

xs = (cx + r \* np.cos(ring\_angles)).astype(int)

ys = (cy + r \* np.sin(ring\_angles)).astype(int)

phs = phase[ys, xs]

dphi = np.diff(np.unwrap(phs))

winding = np.sum(dphi) / (2\*np.pi) # Should approach fraction

# --- Plotting

fig, axs = plt.subplots(1, 3, figsize=(15, 4))

axs[0].imshow(np.cos(phase), cmap='twilight', origin='lower')

axs[0].set\_title("Final MBT Phase Field (cos, anyon)")

axs[1].imshow(memory, cmap='inferno', origin='lower')

axs[1].set\_title("Final MBT Memory Field")

axs[2].bar([0], [winding], width=0.3, color='orchid', label="Measured winding")

axs[2].axhline(fraction, color='k', linestyle='--', label="Fractional input")

axs[2].set\_ylim(0, 1)

axs[2].set\_xticks([])

axs[2].set\_ylabel("Winding / Topological Charge")

axs[2].legend()

axs[2].set\_title("MBT Anyon: Fractional Charge")

plt.tight\_layout()

plt.show()