import numpy as np

import matplotlib.pyplot as plt

# Simulation parameters

N = 32 # grid size

T = 80 # timesteps

braid\_steps = T // 2

# Initialize phase field and memory field

phase = np.random.uniform(-np.pi, np.pi, (N, N))

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

# Define Majorana pair (non-Abelian anyons) positions

fixed = (N//2, N//4)

moving\_path = [(N//2 + int((N//4)\*np.sin(2\*np.pi\*t/braid\_steps)),

N//4 + int((N//2)\*t/braid\_steps)) for t in range(braid\_steps)]

moving\_path += [(N//2, N//4 + N//2 - int((N//2)\*(t-braid\_steps)/braid\_steps)) for t in range(braid\_steps, T)]

# Braiding logic

order\_memory = []

for t in range(T):

# Place Majorana anyons

p = phase.copy()

fx, fy = fixed

if t < len(moving\_path):

mx, my = moving\_path[t]

else:

mx, my = moving\_path[-1]

p[fx, fy] += np.pi/2 # fixed Majorana: add pi/2 phase

p[mx, my] -= np.pi/2 # moving Majorana: subtract pi/2 phase

# Evolve memory as overlap of local phase (simulating non-Abelian memory channel)

memory = 0.98\*memory + 0.02\*np.cos(p)

# "Order" is the overlap (global coherence) between the two anyon sites

overlap = np.abs(memory[fx, fy] \* memory[mx, my])

order\_memory.append(overlap)

# Plot results

plt.figure(figsize=(14,4))

plt.subplot(1,3,1)

plt.imshow(np.cos(phase), cmap='twilight')

plt.title("Final MBT Phase Field (cos, non-Abelian)")

plt.subplot(1,3,2)

plt.imshow(memory, cmap='inferno')

plt.title("Final MBT Memory Field (non-Abelian)")

plt.subplot(1,3,3)

plt.plot(order\_memory, label='Non-Abelian Order')

plt.axvline(braid\_steps, color='k', linestyle='--', label='Braid Complete')

plt.xlabel("Timestep")

plt.ylabel("Non-Abelian Overlap (memory)")

plt.legend()

plt.title("MBT Non-Abelian Braiding Memory")

plt.tight\_layout()

plt.show()