# MBT Defect Braiding Simulation (First Principles)

import numpy as np

import matplotlib.pyplot as plt

np.random.seed(42)

N, T = 32, 100

field = np.random.rand(N, N) \* 2 \* np.pi

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

vx, vy = [8, 24], [8, 24] # Defect initial positions

# Pin vortex and antivortex (opposite phase windings)

def pin\_defect(field, x, y, sign=1):

yy, xx = np.ogrid[:N, :N]

angle = np.arctan2(yy-y, xx-x)

field += sign \* angle

return field

# Trajectory for braiding: spiral swap paths

def traj(t, i):

phase = np.pi \* t / (T-1)

r = 6 + 2\*np.sin(2\*phase + i\*np.pi)

xc = 8 if i==0 else 24

yc = 24 if i==0 else 8

return int(xc + r\*np.cos(phase)), int(yc + r\*np.sin(phase))

order\_trace = []

for t in range(T):

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

# Move defects along spiral trajectories (braid)

for i, sign in enumerate([1, -1]):

x, y = traj(t, i)

f = pin\_defect(f, x, y, sign)

# MBT update: weighted memory + phase smoothing

field = 0.90 \* field + 0.10 \* f + 0.07 \* (np.roll(field,1,0) + np.roll(field,-1,0) +

np.roll(field,1,1) + np.roll(field,-1,1) - 4\*field)

memory += np.cos(field)

# Order: overlap of memory with initial state

order\_trace.append(np.mean(np.cos(memory - memory[0,0])))

# Final plots

plt.figure(figsize=(15,4))

plt.subplot(1,3,1)

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

plt.title("Final MBT Phase Field (braided)")

plt.axis('off')

plt.subplot(1,3,2)

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

plt.title("Final MBT Memory Field (braided)")

plt.axis('off')

plt.subplot(1,3,3)

plt.plot(order\_trace)

plt.axvline(T//2, color='gray', ls='--', label='Halfway (braid complete)')

plt.title("MBT Topological Memory (Braid Order)")

plt.ylabel("Order (topological memory)")

plt.xlabel("Timestep")

plt.legend()

plt.tight\_layout()

plt.show()