import numpy as np

import matplotlib.pyplot as plt

# --- Parameters ---

N = 32 # Grid size

timesteps = 100 # Number of simulation steps

vortex\_pos = [(8, 8), (24, 24)] # Fixed (no braid) vortex positions

braid\_halfway = timesteps // 2 # Just for reference on plot

# --- Initialize MBT fields ---

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

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

# --- Dynamics ---

def add\_vortex(phase, center, strength=1):

"""Imprint a static vortex into the phase field."""

y, x = np.indices(phase.shape)

angle = np.arctan2(y - center[0], x - center[1])

return (phase + strength \* angle) % (2\*np.pi)

topological\_memory = []

for t in range(timesteps):

# Static vortices only

for pos in vortex\_pos:

phase = add\_vortex(phase, pos)

# Evolve phase a little

phase += 0.05 \* (np.roll(phase, 1, 0) + np.roll(phase, -1, 0) +

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

phase %= 2\*np.pi

# MBT memory: log global phase winding

winding = np.mean(np.cos(phase))

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

topological\_memory.append(np.abs(winding))

# --- Plot ---

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

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

axs[0].set\_title("Final MBT Phase Field (no braid)")

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

axs[1].set\_title("Final MBT Memory Field (no braid)")

axs[2].plot(topological\_memory, label="Order (no braid)")

axs[2].axvline(braid\_halfway, color='k', linestyle='--', alpha=0.4, label="Halfway")

axs[2].set\_title("Topological Memory (No Braiding)")

axs[2].set\_xlabel("Timestep")

axs[2].set\_ylabel("Order (memory)")

axs[2].legend()

plt.tight\_layout()

plt.show()