import numpy as np

import matplotlib.pyplot as plt

# Parameters

N = 32 # Grid size

timesteps = 150

edge\_bias = 0.8 # Boost at edge to simulate boundary field

noise\_level = 0.20 # "Temperature" (tune lower for more order)

# Fields

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

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

edge\_order = []

bulk\_order = []

def laplacian(f):

return (np.roll(f, +1, 0) + np.roll(f, -1, 0) +

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

# Time evolution

for t in range(timesteps):

# Edge bias (simulate "magnetic field" at edge)

mask = np.zeros\_like(phase)

mask[0, :] = mask[-1, :] = mask[:, 0] = mask[:, -1] = 1

phase += edge\_bias \* mask

# Local MBT update: geometry + neighbor influence + noise

lap = laplacian(phase)

noise = noise\_level \* np.random.randn(N, N)

dphase = 0.15 \* lap + noise

phase = (phase + dphase) % (2 \* np.pi)

memory = 0.90 \* memory + 0.10 \* np.cos(phase)

# Order parameter (coherence)

edge\_indices = (mask == 1)

bulk\_indices = (mask == 0)

edge\_coherence = np.abs(np.mean(np.exp(1j \* phase[edge\_indices])))

bulk\_coherence = np.abs(np.mean(np.exp(1j \* phase[bulk\_indices])))

edge\_order.append(edge\_coherence)

bulk\_order.append(bulk\_coherence)

# Plot: Final phase field, memory, and edge/bulk order

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

plt.subplot(1, 3, 1)

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

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

plt.axis('off')

plt.subplot(1, 3, 2)

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

plt.title("Final MBT Memory Field")

plt.axis('off')

plt.subplot(1, 3, 3)

plt.plot(edge\_order, label="Edge Order")

plt.plot(bulk\_order, label="Bulk Order", alpha=0.7)

plt.axvline(x=20, linestyle='--', color='k', alpha=0.3, label="Flux Inserted")

plt.title("MBT Quantum Hall Edge State (Emergence)")

plt.xlabel("Timestep")

plt.ylabel("Order Parameter")

plt.legend()

plt.tight\_layout()

plt.show()