import numpy as np

import matplotlib.pyplot as plt

# Parameters

N = 32

timesteps = 150

temperature = 0.4

pair\_strength = 2.0 # This is our MBT analogue of pairing force

# Initialize phase field and 'pair field' (each cell tries to pair with a neighbor)

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

pair\_field = np.zeros((N, N))

order\_list = []

for t in range(timesteps):

# Choose neighbor direction for pairing (up/down/left/right)

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

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

neighbor\_avg = np.mean(neighbors, axis=0)

# MBT phase update: pair interaction plus noise

dphase = pair\_strength \* np.sin(neighbor\_avg - phase) + temperature \* np.random.randn(N, N) \* 0.1

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

# Calculate local pair 'order' (how much each cell matches its neighbor)

pair\_field = np.cos(phase - neighbor\_avg)

# Global order parameter (how phase-locked the whole field is)

order\_list.append(np.abs(np.mean(np.exp(1j\*phase))))

# Plot final results

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

axs[0].imshow(np.cos(phase), cmap="twilight", interpolation="nearest")

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

axs[1].imshow(pair\_field, cmap="inferno", interpolation="nearest")

axs[1].set\_title("Final Pair Field (MBT Cooper Pairs)")

axs[2].plot(order\_list, label="Pairing Order (MBT Cooper Pairs)")

axs[2].set\_title("Global Pairing Order (Emergence)")

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

axs[2].legend()

plt.tight\_layout()

plt.show()