import numpy as np

import matplotlib.pyplot as plt

# Simulation parameters

L = 30 # Lattice size (L x L)

timesteps = 150 # Number of time steps

thermal\_noise = 0.6 # "Temperature" level, 0 = no noise, >1 = hot

echo\_coupling = 0.23 # Strength of memory echo

neighbor\_coupling = 0.7 # Normal spin-spin coupling

echo\_pulse\_step = 50 # When to pulse the echo field (simulate MBT healing)

# Initialize phase and memory echo fields

phase = np.random.uniform(0, 2\*np.pi, (L, L))

echo\_memory = np.zeros((L, L))

# Track global coherence and echo

coherence\_history = []

echo\_history = []

def update\_lattice(phase, echo\_memory, echo\_pulse=False):

new\_phase = np.zeros\_like(phase)

for i in range(L):

for j in range(L):

# Sum neighbor phases

neighbors = [phase[(i+1)%L, j], phase[(i-1)%L, j], phase[i, (j+1)%L], phase[i, (j-1)%L]]

neighbor\_mean = np.angle(np.exp(1j \* np.array(neighbors)).mean())

# Memory echo

echo\_effect = echo\_coupling \* echo\_memory[i, j]

# Thermal noise

noise = np.random.normal(0, thermal\_noise)

# Update rule: align to neighbors + echo + noise

new\_phase[i, j] = (1-neighbor\_coupling) \* phase[i, j] + neighbor\_coupling \* neighbor\_mean + echo\_effect + noise

# Echo memory is updated with a fraction of the current phase (decays slowly)

if echo\_pulse:

echo\_memory += np.cos(phase) \* 0.4 # Pulse MBT memory field!

else:

echo\_memory = 0.95 \* echo\_memory + 0.05 \* np.cos(phase)

return new\_phase, echo\_memory

# Main simulation loop

for t in range(timesteps):

pulse = (t == echo\_pulse\_step)

phase, echo\_memory = update\_lattice(phase, echo\_memory, echo\_pulse=pulse)

# Calculate coherence (how phase-aligned are all sites?)

global\_order = np.abs(np.mean(np.exp(1j \* phase)))

coherence\_history.append(global\_order)

echo\_history.append(echo\_memory.copy())

# Plot results

fig, axs = plt.subplots(1, 3, figsize=(18,6))

axs[0].imshow(np.cos(phase), cmap='twilight', vmin=-1, vmax=1)

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

axs[1].imshow(echo\_memory, cmap='inferno')

axs[1].set\_title('Final Echo Memory Field')

axs[2].plot(coherence\_history)

axs[2].set\_title('Global Phase Coherence (order)')

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

plt.tight\_layout()

plt.show()