# MBT Superconductor Simulation: Disorder-Order (Critical Transition)

import numpy as np

import matplotlib.pyplot as plt

# Parameters

N = 32 # Grid size

timesteps = 100 # Simulation steps per temperature

temps = np.linspace(2.5, 0.1, 18) # Temperature (noise) steps: high to low

J = 1.0 # Coupling (strong order at low T)

# Helper: MBT Coupling function

def mbt\_coupling(phi\_grid):

"""Returns field after MBT-style interaction with nearest neighbors"""

lap = (

np.roll(phi\_grid, 1, 0) + np.roll(phi\_grid, -1, 0) +

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

)

return J \* lap

# Helper: Order parameter (coherence)

def order\_param(phi):

vec = np.exp(1j \* phi)

return np.abs(np.mean(vec))

# Storage

orders = []

last\_phi, last\_mem = None, None

# Main: Step through decreasing temperature

phi = np.random.uniform(0, 2 \* np.pi, (N, N)) # Initial random phases

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

for T in temps:

for t in range(timesteps):

# Noise (temperature)

noise = np.random.normal(0, T, (N, N))

# MBT update

dphi = mbt\_coupling(phi) + noise

phi = (phi + dphi) % (2 \* np.pi)

memory = 0.93 \* memory + 0.07 \* np.cos(phi) # Trace memory

last\_phi = np.cos(phi) # Store for later visualization

last\_mem = memory

orders.append(order\_param(phi))

# Plot

fig, axes = plt.subplots(1, 3, figsize=(16, 4))

axes[0].imshow(last\_phi, cmap='twilight', interpolation='nearest')

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

axes[1].imshow(last\_mem, cmap='inferno', interpolation='nearest')

axes[1].set\_title("Final MBT Memory Field")

axes[2].plot(np.flip(temps), orders, label='Coherence (Order)')

axes[2].axvline(x=temps[np.argmax(np.diff(orders))], color='k', linestyle='--', label='Critical Point?')

axes[2].set\_xlabel("Temperature (noise, ↓)")

axes[2].set\_ylabel("Order")

axes[2].set\_title("Disorder → Order (MBT Superconductivity)")

axes[2].legend()

plt.tight\_layout()

plt.show()