# MBT Superconductivity Demo — Colab Ready

import numpy as np

import matplotlib.pyplot as plt

# Parameters

N = 32 # grid size

steps = 150

initial\_coupling = 0.03 # MBT "tension" start

final\_coupling = 0.25 # MBT "tension" end

noise = 0.10 # random 'motion', like atomic vibration

# Initialize random phase field (values in [0, 2pi])

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

coherence\_history = []

coupling\_history = []

# Simulate

for t in range(steps):

coupling = initial\_coupling + (final\_coupling-initial\_coupling) \* t/(steps-1)

coupling\_history.append(coupling)

new\_phase = phase.copy()

for i in range(N):

for j in range(N):

# Get neighbors (periodic boundary)

neighbors = [

phase[(i-1)%N, j], phase[(i+1)%N, j],

phase[i, (j-1)%N], phase[i, (j+1)%N]

]

avg\_neighbor = np.angle(np.mean(np.exp(1j\*np.array(neighbors))))

# Update phase with MBT-like "sheet tension"

delta = coupling \* np.sin(avg\_neighbor - phase[i,j])

new\_phase[i,j] += delta + np.random.normal(0, noise)

phase = np.mod(new\_phase, 2\*np.pi)

# MBT Memory field: running average

if t == 0:

memory = np.cos(phase)

else:

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

# Global phase coherence (order parameter)

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

coherence\_history.append(coherence)

# Plot results

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

plt.subplot(1,3,1)

plt.imshow(np.cos(phase), cmap="twilight", interpolation="nearest")

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

plt.axis("off")

plt.subplot(1,3,2)

plt.imshow(memory, cmap="inferno", interpolation="nearest")

plt.title("Final MBT Memory Field")

plt.axis("off")

plt.subplot(1,3,3)

plt.plot(coherence\_history, label="Coherence")

plt.plot(np.array(coupling\_history)/np.max(coupling\_history), '--', label="Norm. Coupling")

plt.xlabel("Timestep")

plt.ylabel("Order / Coupling")

plt.title("Global Phase Coherence (MBT Order)")

plt.legend()

plt.tight\_layout()

plt.show()