# MBT Josephson Junction Simulation (Colab/Mobile Ready)

import numpy as np

import matplotlib.pyplot as plt

np.random.seed(42)

N, steps = 32, 200

junction\_width = 2 # Width of the weak link

coupling\_strong = 1.0 # Normal MBT coupling

coupling\_weak = 0.07 # Weak link coupling

# --- Lattice: two "superconductors" with weak junction in center ---

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

memory = np.zeros\_like(theta)

coupling = np.full((N, N), coupling\_strong)

coupling[:, N//2-junction\_width:N//2+junction\_width] = coupling\_weak

order\_history = []

phase\_diff\_history = []

for t in range(steps):

# Local phase update with MBT rules

new\_theta = np.copy(theta)

for i in range(N):

for j in range(N):

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

n\_sum = 0

for di, dj in [(-1,0),(1,0),(0,-1),(0,1)]:

ni, nj = (i+di)%N, (j+dj)%N

n\_sum += np.sin(theta[ni, nj] - theta[i, j])

new\_theta[i, j] += coupling[i, j] \* n\_sum \* 0.13

theta = new\_theta + np.random.normal(0, 0.01, theta.shape)

memory += np.cos(theta)

# Track coherence (global order parameter)

order = np.abs(np.mean(np.exp(1j\*theta)))

order\_history.append(order)

# Track mean phase left vs. right of junction

left\_mean = np.angle(np.mean(np.exp(1j\*theta[:, :N//2-junction\_width])))

right\_mean = np.angle(np.mean(np.exp(1j\*theta[:, N//2+junction\_width:])))

phase\_diff = np.angle(np.exp(1j\*(right\_mean - left\_mean)))

phase\_diff\_history.append(phase\_diff)

# --- Plot Results ---

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

axs[0].imshow(np.cos(theta), cmap='twilight', aspect='auto')

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

axs[1].imshow(memory, cmap='inferno', aspect='auto')

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

axs[2].plot(phase\_diff\_history, label='Phase Difference (Junction)')

axs[2].plot(order\_history, '--', label='Coherence')

axs[2].set\_title('MBT Josephson Junction')

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

axs[2].set\_ylabel('Phase/Order')

axs[2].legend()

plt.tight\_layout()

plt.show()