# MBT Vortex/Defect Superconductivity Demo

import numpy as np

import matplotlib.pyplot as plt

# Parameters

N = 32 # grid size

timesteps = 150

noise\_level = 0.09 # MBT "temperature" - keep low for order

coupling = 1.7 # MBT sheet tension

inject\_vortex = True

# MBT Phase field (angle in radians), memory, and coherence tracker

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

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

coherence = []

def add\_vortex(phase, x0, y0, charge=1):

"""Injects a vortex defect at (x0, y0) with given charge."""

Y, X = np.meshgrid(np.arange(N), np.arange(N), indexing='ij')

angle = np.arctan2(Y-y0, X-x0)

phase += charge \* angle

return np.mod(phase, 2\*np.pi)

# Inject a +1 vortex in center

if inject\_vortex:

cx, cy = N//2, N//2

phase = add\_vortex(phase, cx, cy, charge=1)

for t in range(timesteps):

# MBT sheet coupling (Kuramoto)

neighbors = np.roll(phase, 1, 0) + np.roll(phase, -1, 0) \

+ np.roll(phase, 1, 1) + np.roll(phase, -1, 1)

dtheta = coupling \* np.sin(neighbors/4 - phase) + noise\_level \* np.random.randn(N,N)

phase = np.mod(phase + dtheta, 2\*np.pi)

# MBT memory: running average of |cos(phase diff from global mean)|

avg\_phase = np.angle(np.mean(np.exp(1j\*phase)))

memory = 0.95 \* memory + 0.05 \* np.abs(np.cos(phase - avg\_phase))

# Global order parameter (like "superconductivity" order)

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

coherence.append(r)

# Plot 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, vortex)')

axs[1].imshow(memory, cmap='inferno', interpolation='nearest')

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

axs[2].plot(coherence, label='Coherence')

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

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

axs[2].legend()

plt.tight\_layout()

plt.show()