# MBT Topological Superconductor Simulation: Edge/Surface States

import numpy as np

import matplotlib.pyplot as plt

N = 32 # Grid size

timesteps = 150

noise = 0.5

# --- Phase Field (bulk + edges) ---

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

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

order = []

# Define MBT update (bulk + edge special rule)

def update\_mbt(phase, memory, noise, edge\_factor=3.0):

new\_phase = np.copy(phase)

for x in range(N):

for y in range(N):

neigh = []

for dx,dy in [(-1,0),(1,0),(0,-1),(0,1)]:

nx, ny = (x+dx)%N, (y+dy)%N

neigh.append(phase[nx, ny])

mean\_n = np.mean(neigh)

if x in [0, N-1] or y in [0, N-1]: # Edge rule: amplify memory order

mboost = edge\_factor

else:

mboost = 1.0

dphi = (mean\_n - phase[x,y]) + mboost\*memory[x,y]

new\_phase[x,y] += 0.13 \* dphi + noise \* np.random.randn()

return new\_phase

# --- Main loop ---

for t in range(timesteps):

phase = update\_mbt(phase, memory, noise)

memory = 0.98 \* memory + 0.10 \* np.cos(phase)

# Order: edge coherence vs. bulk

edge\_mask = np.zeros((N,N), bool)

edge\_mask[0,:] = edge\_mask[-1,:] = edge\_mask[:,0] = edge\_mask[:,-1] = True

edge\_order = np.abs(np.mean(np.exp(1j\*phase[edge\_mask])))

bulk\_order = np.abs(np.mean(np.exp(1j\*phase[~edge\_mask])))

order.append([edge\_order, bulk\_order])

# --- Plot ---

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)')

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

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

order = np.array(order)

axs[2].plot(order[:,0], label="Edge Order")

axs[2].plot(order[:,1], label="Bulk Order")

axs[2].set\_title("Edge vs. Bulk Coherence (Topological Order)")

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

axs[2].set\_ylabel("Order parameter")

axs[2].legend()

plt.tight\_layout()

plt.show()