# MBT Majorana Edge Mode (Topological Qubit Emergence)

import numpy as np

import matplotlib.pyplot as plt

np.random.seed(42)

N = 32

timesteps = 160

# Phase field: random initial phase

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

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

# Edge “twist” for topological effect

phase[0, :] += np.pi # one edge is phase-flipped

order\_hist = []

zero\_mode\_profile = []

for t in range(timesteps):

# Laplacian-like coupling (nearest-neighbor sync)

laplacian = (

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

np.roll(phase, 1, axis=1) + np.roll(phase, -1, axis=1) - 4 \* phase

)

# Apply “MBT” update: order from tension, with edge “twist”

phase += 0.14 \* laplacian + 0.018 \* np.random.randn(N, N)

# Edge phase-flip persists

phase[0, :] = (phase[0, :] + np.pi) % (2\*np.pi) - np.pi

# “Majorana” zero mode: edge amplitude contrast (absolute value)

zero\_mode = np.abs(np.cos(phase[0, :]))

zero\_mode\_profile.append(zero\_mode)

# Bulk order parameter (coherence)

order\_hist.append(np.abs(np.mean(np.exp(1j\*phase))))

# Optional: build up “memory” field (not strictly necessary)

memory += np.cos(phase)

zero\_mode\_profile = np.array(zero\_mode\_profile)

# ---- Plot ----

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

axs[0].imshow(np.cos(phase), 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(zero\_mode\_profile[-1], label="Edge Zero Mode Profile")

axs[2].plot([0, N-1], [1, 1], 'k--', alpha=0.3, label="Protected Value")

axs[2].set\_title("MBT Majorana Edge Mode")

axs[2].set\_xlabel("Edge Position")

axs[2].set\_ylabel("Zero Mode Amplitude")

axs[2].legend()

plt.tight\_layout()

plt.show()