import numpy as np

import matplotlib.pyplot as plt

N = 32

center = N // 2

# Initialize a phase field with a "winding" at the center (like a vortex)

y, x = np.ogrid[-center:N-center, -center:N-center]

angle = np.arctan2(y, x)

phase = angle # Phase wraps around the center (vortex)

# Simulate a basic MBT evolution (just for noise, not essential)

for t in range(15):

noise = np.random.randn(N, N) \* 0.01

phase = (phase + noise) % (2 \* np.pi)

# Calculate phase winding around the vortex (measuring quantized "flux")

circle\_r = 10

theta = np.linspace(0, 2 \* np.pi, 100)

xc = (center + circle\_r \* np.cos(theta)).astype(int)

yc = (center + circle\_r \* np.sin(theta)).astype(int)

winding = np.diff(np.unwrap(phase[yc, xc]))

flux\_quant = np.sum(winding) / (2 \* np.pi)

fig, axs = plt.subplots(1, 2, figsize=(10, 4))

im = axs[0].imshow(np.cos(phase), cmap="twilight", interpolation="nearest")

axs[0].set\_title("MBT Phase Field (Vortex Winding)")

axs[1].plot(theta[:-1], np.cumsum(winding), label="Phase Winding (MBT flux)")

axs[1].axhline(0, color='k', linestyle='--')

axs[1].set\_title(f"Flux Quantization: {flux\_quant:.1f} × 2π")

axs[1].set\_xlabel("Angle around vortex")

axs[1].legend()

plt.tight\_layout()

plt.show()