import numpy as np

import matplotlib.pyplot as plt

from scipy.ndimage import laplace

# --- Parameters ---

N = 60 # grid size

timesteps = 220 # how long to run

barrier\_prob = 0.22

barrier\_refresh = 30 # how often to scramble maze

# --- Initialize barrier maze ---

def random\_barriers():

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

mask = np.random.rand(N, N) < barrier\_prob

V[mask] = 1.0 + 0.5 \* np.random.rand(mask.sum())

return V

V = random\_barriers()

# --- Initialize two wavepackets ---

def make\_wave(x0, y0, kx=0, ky=0, width=4):

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

psi = np.exp(-((X-x0)\*\*2 + (Y-y0)\*\*2) / (2\*width\*\*2)) \* np.exp(1j \* (kx\*X + ky\*Y))

return psi / np.sqrt(np.sum(np.abs(psi)\*\*2))

psi\_A = make\_wave(x0=6, y0=N//2, kx=0.20, ky=0.00, width=4) # From left

psi\_B = make\_wave(x0=N-7, y0=N//2, kx=-0.20, ky=0.00, width=4) # From right

# --- Evolve ---

results = []

for t in range(timesteps):

# Occasionally scramble maze

if t % barrier\_refresh == 0 and t > 0:

V = random\_barriers()

# Evolve both waves

for psi in [psi\_A, psi\_B]:

lap = laplace(psi)

psi += 0.19 \* (0.35 \* lap - 0.8 \* V \* psi)

norm = np.sqrt(np.sum(np.abs(psi)\*\*2))

if norm != 0:

psi /= norm

# Measure overlap, edge transmission

overlap = np.sum(np.abs(psi\_A \* np.conj(psi\_B)))

trans\_A = np.sum(np.abs(psi\_A[-1, :]))

trans\_B = np.sum(np.abs(psi\_B[0, :]))

results.append((overlap, trans\_A, trans\_B))

# Optionally visualize every 50 steps

if t in [0, timesteps//4, timesteps//2, 3\*timesteps//4, timesteps-1]:

plt.figure(figsize=(9, 4.5))

plt.subplot(1,2,1)

plt.imshow(np.abs(psi\_A)\*\*2, origin='lower', cmap='Blues', vmin=0, vmax=0.04)

plt.scatter(\*np.where(V>0), color='k', s=6, label='Barrier')

plt.title(f'Wave A | t={t}')

plt.axis('off')

plt.subplot(1,2,2)

plt.imshow(np.abs(psi\_B)\*\*2, origin='lower', cmap='Reds', vmin=0, vmax=0.04)

plt.scatter(\*np.where(V>0), color='k', s=6)

plt.title(f'Wave B | t={t}')

plt.axis('off')

plt.tight\_layout()

plt.show()

# --- Plot summary ---

results = np.array(results)

plt.figure(figsize=(8,4))

plt.plot(results[:,0], label='Wave Overlap')

plt.plot(results[:,1], label='A transmission (right edge)')

plt.plot(results[:,2], label='B transmission (left edge)')

plt.title('Wild MBT Quantum Maze: Double Waves, Scrambling Maze')

plt.xlabel('Time step')

plt.ylabel('Value')

plt.legend()

plt.tight\_layout()

plt.show()