import numpy as np

import matplotlib.pyplot as plt

from scipy.integrate import trapezoid

# --- Parameters ---

grid\_size = 120

timesteps = 180

dt = 0.12

width = 6

r = np.linspace(0, grid\_size, grid\_size)

dr = r[1] - r[0]

# --- Packet Parameters ---

x0 = 25

k0 = 1.2

# --- Barrier positions ---

barrier1\_center = 55

barrier2\_center = 65

barrier\_width = 5

barrier\_height = 0.07

# --- Frequency grid ---

freqs1 = np.linspace(0.01, 0.12, 16)

freqs2 = np.linspace(0.01, 0.12, 16)

results = np.zeros((len(freqs1), len(freqs2)))

# --- Evolution Function for Two Breathing Barriers ---

def run\_dispatch(freq1, freq2):

ψ = np.exp(-((r - x0)\*\*2)/(2\*width\*\*2)) \* np.exp(1j \* k0 \* r)

for t in range(timesteps):

# Moving barriers (breathing)

V = np.zeros\_like(r)

V[(r > barrier1\_center - barrier\_width/2) & (r < barrier1\_center + barrier\_width/2)] = (

barrier\_height \* (1 + 0.4\*np.sin(freq1 \* t))

)

V[(r > barrier2\_center - barrier\_width/2) & (r < barrier2\_center + barrier\_width/2)] = (

barrier\_height \* (1 + 0.4\*np.cos(freq2 \* t))

)

lap = np.zeros\_like(ψ, dtype=complex)

lap[1:-1] = (ψ[2:] - 2\*ψ[1:-1] + ψ[:-2]) / dr\*\*2

ψ += dt \* (0.65 \* lap - 0.5 \* V \* ψ)

norm = np.sqrt(trapezoid(np.abs(ψ)\*\*2, r))

if norm != 0:

ψ /= norm

final = np.abs(ψ)\*\*2

P\_trans = trapezoid(final[r > 85], r[r > 85])

return P\_trans

# --- Compute 2D Map ---

for i, f1 in enumerate(freqs1):

for j, f2 in enumerate(freqs2):

results[i, j] = run\_dispatch(f1, f2)

# --- Plotting ---

fig, ax = plt.subplots(figsize=(9,8))

pcm = ax.imshow(results, origin='lower', aspect='auto',

extent=[freqs2[0], freqs2[-1], freqs1[0], freqs1[-1]],

cmap='viridis')

plt.colorbar(pcm, ax=ax, label='Transmission Probability')

ax.set\_xlabel('Barrier 2 Breathing Frequency (rad/step)')

ax.set\_ylabel('Barrier 1 Breathing Frequency (rad/step)')

ax.set\_title("MBT Dispatch: Twin Breathing Barriers Coherence Map")

plt.tight\_layout()

plt.show()