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]

# --- Wavepacket Parameters ---

x0 = 25

k0 = 1.2

# --- Frequency Ranges ---

seed\_freqs = np.linspace(0.01, 0.12, 12)

barrier\_freqs = np.linspace(0.01, 0.12, 12)

# --- Breathing seed function (internal MBT curvature seed) ---

def mbt\_seed(center, freq, base\_phase, t):

phase = base\_phase + 0.5 \* np.sin(freq \* t)

return np.exp(-((r - center)\*\*2)/(2\*width\*\*2)) \* np.exp(1j \* phase)

# --- Evolve system with both breathing barrier and internal seed ---

def evolve\_mbt\_breathing(x0, k0, seed\_center, seed\_freq, barrier\_center, barrier\_width, barrier\_base\_height, barrier\_freq):

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

for t in range(timesteps):

# Breathing barrier height

barrier\_height = barrier\_base\_height \* (1 + 0.5 \* np.sin(barrier\_freq \* t))

V = np.zeros\_like(r)

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

# Internal MBT seed

ψ\_int = mbt\_seed(center=seed\_center, freq=seed\_freq, base\_phase=0, t=t)

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

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

# Evolve ψ with both Laplacians

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 + 0.3 \* lap\_int)

# Normalize

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

# --- Scan over frequencies ---

barrier\_center = 60

barrier\_width = 6

barrier\_base\_height = 0.08

seed\_center = 53

trans\_map = np.zeros((len(seed\_freqs), len(barrier\_freqs)))

for i, s\_freq in enumerate(seed\_freqs):

for j, b\_freq in enumerate(barrier\_freqs):

P\_trans = evolve\_mbt\_breathing(

x0=x0, k0=k0, seed\_center=seed\_center, seed\_freq=s\_freq,

barrier\_center=barrier\_center, barrier\_width=barrier\_width,

barrier\_base\_height=barrier\_base\_height, barrier\_freq=b\_freq

)

trans\_map[i, j] = P\_trans

# --- Plot resonance map ---

plt.figure(figsize=(11, 9))

plt.imshow(trans\_map, origin='lower', aspect='auto', cmap='viridis',

extent=[barrier\_freqs[0], barrier\_freqs[-1], seed\_freqs[0], seed\_freqs[-1]])

plt.xlabel("Breathing Barrier Frequency (rad/step)")

plt.ylabel("MBT Seed Frequency (rad/step)")

plt.title("MBT Dispatch: Double-Resonance Coherence Map\n(Transmission Probability)")

plt.colorbar(label="Transmission Probability")

plt.show()