import numpy as np

import matplotlib.pyplot as plt

from scipy.integrate import trapezoid

# --- Simulation Parameters ---

grid\_size = 120

timesteps = 180

dt = 0.12

width = 6

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

dr = r[1] - r[0]

# --- Wave Packet Parameters ---

x0 = 25

k0 = 1.1

# --- Barrier Locations ---

b1 = 50

b2 = 60

b3 = 72

barrier\_width = 3

barrier\_height = 0.11

# --- Breathing Frequencies ---

freqs = np.linspace(0.015, 0.12, 22) # Finer grid

trans\_map = np.zeros((len(freqs), len(freqs), len(freqs)))

# --- Breathing Seed Function ---

def barrier\_profile(center, width, amp, t, freq, phase=0):

osc = amp \* (1.0 + 0.85 \* np.sin(freq \* t + phase))

profile = np.zeros\_like(r)

idx = (r > center - width/2) & (r < center + width/2)

profile[idx] = osc

return profile

# --- Simulation Function ---

def run\_dispatch(freq1, freq2, freq3):

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

for t in range(timesteps):

V = (

barrier\_profile(b1, barrier\_width, barrier\_height, t, freq1)

+ barrier\_profile(b2, barrier\_width, barrier\_height, t, freq2)

+ barrier\_profile(b3, barrier\_width, barrier\_height, t, freq3)

)

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

T = trapezoid(final[r > 95], r[r > 95])

return T

# --- Run the resonance sweep (takes a few minutes) ---

for i, f1 in enumerate(freqs):

for j, f2 in enumerate(freqs):

for k, f3 in enumerate(freqs):

trans\_map[i, j, k] = run\_dispatch(f1, f2, f3)

# --- Find max resonance point ---

max\_val = np.max(trans\_map)

max\_idx = np.unravel\_index(np.argmax(trans\_map), trans\_map.shape)

best\_f1, best\_f2, best\_f3 = freqs[max\_idx[0]], freqs[max\_idx[1]], freqs[max\_idx[2]]

print(f"Max tunneling found: {max\_val:.2e} at (f1={best\_f1:.3f}, f2={best\_f2:.3f}, f3={best\_f3:.3f})")

# --- Visualize a few 2D slices ---

plt.figure(figsize=(18,5))

for n, plane in enumerate([0, len(freqs)//4, len(freqs)//2, -1]):

plt.subplot(1,4,n+1)

plt.imshow(trans\_map[plane,:,:], origin='lower', aspect='auto',

extent=[freqs[0], freqs[-1], freqs[0], freqs[-1]], cmap='viridis')

plt.colorbar(label="Transmission Probability")

plt.xlabel("Barrier 2 Frequency")

plt.ylabel("Barrier 3 Frequency")

plt.title(f"Barrier 1 freq = {freqs[plane]:.3f}")

plt.suptitle("MBT Triple-Barrier Resonance: Tunneling Probability Slices")

plt.tight\_layout(rect=[0,0,1,0.94])

plt.show()