import numpy as np

import matplotlib.pyplot as plt

from scipy.integrate import trapezoid

# --- PARAMETERS ---

grid\_size = 120

timesteps = 140

dt = 0.12

width = 6

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

dr = r[1] - r[0]

# --- Quantum Packet ---

def initial\_packet(x0, k0):

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

# --- Breathing triple barrier potential ---

def triple\_breathing\_barrier(t, freq1, freq2, freq3):

V = np.zeros\_like(r)

# First barrier

V[(r > 40) & (r < 46)] = 0.04 + 0.03 \* np.sin(freq1 \* t)

# Second (middle) barrier

V[(r > 57) & (r < 63)] = 0.06 + 0.04 \* np.sin(freq2 \* t)

# Third barrier

V[(r > 75) & (r < 81)] = 0.03 + 0.025 \* np.sin(freq3 \* t)

return V

# --- Simulation function ---

def run\_sim(freq1, freq2, freq3, x0=25, k0=1.1):

ψ = initial\_packet(x0, k0)

for t in range(timesteps):

V = triple\_breathing\_barrier(t, freq1, freq2, 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)

# Normalize

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

if norm != 0:

ψ /= norm

# Measure transmission (rightmost) and reflection (leftmost)

final = np.abs(ψ)\*\*2

T = trapezoid(final[r > 100], r[r > 100])

R = trapezoid(final[r < 15], r[r < 15])

return T, R

# --- Scan resonance space (small grid for mobile) ---

freqs1 = np.linspace(0.02, 0.12, 8) # Barrier 1 frequency

freqs2 = np.linspace(0.02, 0.12, 8) # Barrier 2 frequency

freqs3 = np.linspace(0.02, 0.12, 8) # Barrier 3 frequency

results = np.zeros((len(freqs1), len(freqs2), len(freqs3)))

for i, f1 in enumerate(freqs1):

for j, f2 in enumerate(freqs2):

for k, f3 in enumerate(freqs3):

T, \_ = run\_sim(f1, f2, f3)

results[i, j, k] = T

# --- Visualize as "slices" (keep it readable on mobile) ---

from matplotlib import cm

fig, axs = plt.subplots(2, 4, figsize=(18, 8))

for idx, i in enumerate(range(0, 8)):

ax = axs.flat[idx]

im = ax.imshow(results[i], origin='lower', extent=[freqs2[0],freqs2[-1],freqs3[0],freqs3[-1]],

aspect='auto', cmap=cm.viridis)

ax.set\_title(f"Barrier 1 freq: {freqs1[i]:.2f}")

ax.set\_xlabel('Barrier 2 freq')

ax.set\_ylabel('Barrier 3 freq')

plt.colorbar(im, ax=ax, fraction=0.05)

fig.suptitle("MBT Dispatch: Triple Breathing Barriers\n(Tunneling Probability vs All Resonances)")

plt.tight\_layout(rect=[0, 0, 1, 0.97])

plt.show()