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]

x0 = 25

k0 = 1.1

def tunneling\_for\_frequencies(freqs, n\_barriers):

freq\_vals = np.linspace(0.02, 0.14, 8)

combos = np.array(np.meshgrid(\*([freq\_vals]\*n\_barriers))).reshape(n\_barriers, -1).T

barrier\_centers = np.linspace(50, 70, n\_barriers)

T\_map = np.zeros([len(freq\_vals)]\*n\_barriers)

best\_T = 0

best\_key = None

for idx in np.ndindex(\*([len(freq\_vals)]\*n\_barriers)):

fs = [freq\_vals[i] for i in idx]

V = np.zeros\_like(r)

for c, f in zip(barrier\_centers, fs):

V += np.exp(-((r-c)\*\*2)/(2\*2.3\*\*2)) \* 0.12 \* (1 + 0.7 \* np.sin(f \* 5)) # t=5 snapshot

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

for t in range(timesteps):

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

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

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

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

if norm != 0:

psi /= norm

prob = np.abs(psi)\*\*2

T = trapezoid(prob[r > 85], r[r > 85])

T\_map[idx] = T

if T > best\_T:

best\_T = T

best\_key = fs

return T\_map, freq\_vals, best\_T, best\_key

# --- Plot Quantum Lockpick Gallery ---

n\_setups = 5

fig, axs = plt.subplots(1, n\_setups, figsize=(6\*n\_setups, 5))

for n\_barriers in range(1, n\_setups+1):

T\_map, freq\_vals, best\_T, best\_key = tunneling\_for\_frequencies(freqs=None, n\_barriers=n\_barriers)

ax = axs[n\_barriers-1]

if n\_barriers == 1:

ax.plot(freq\_vals, T\_map, marker='o', color='blue', label='Tunneling')

ax.scatter(freq\_vals[np.argmax(T\_map)], T\_map.max(), color='cyan', s=80, label='Best Key')

ax.set\_xlabel("f1")

ax.set\_ylabel("Transmission Probability")

ax.set\_title(f"1 Barrier\nBest T: {best\_T:.2e}\nKey: {best\_key}")

ax.legend()

else:

# 2D heatmap slice (fix higher dims at center)

slice\_idx = [slice(None)]\*2 + [len(freq\_vals)//2]\*(n\_barriers-2)

T2d = T\_map[tuple(slice\_idx)]

im = ax.imshow(T2d, origin='lower', aspect='auto',

extent=[freq\_vals[0], freq\_vals[-1], freq\_vals[0], freq\_vals[-1]])

ax.set\_xlabel("f1")

ax.set\_ylabel("f2")

ax.set\_title(f"{n\_barriers} Barriers\nBest T: {best\_T:.2e}\nKey: {np.round(best\_key,3)}")

ax.scatter(best\_key[0], best\_key[1], color='cyan', s=80, label='Best Key')

ax.legend()

fig.colorbar(im, ax=ax, shrink=0.8, label='Transmission Probability')

fig.suptitle("MBT Quantum Lockpick Gallery: 1–5 Breathing Barriers\n(Tunneling Probability Heatmaps & Best Keys)", fontsize=20)

plt.tight\_layout()

plt.show()