import numpy as np

import matplotlib.pyplot as plt

from scipy.integrate import trapezoid

# --- Grid & Time ---

r = np.linspace(0, 60, 600)

dr = r[1] - r[0]

dt = 0.05

timesteps = 100 # Keep it reasonable for a plot grid

frames\_to\_plot = [0, 20, 40, 60, 80, 99] # Chosen time steps to visualize

# --- Curvature Parameters ---

omega0 = 8000 # rad/s

tau = 80 # s

sigma = 8 # basin sharpness

def gaussian(r, r0, width):

return np.exp(-((r - r0) \*\* 2) / (2 \* width \*\* 2))

pulse\_width = 1.5

ψ1 = gaussian(r, 27, pulse\_width)

ψ2 = gaussian(r, 33, pulse\_width)

ψ\_total = ψ1 + ψ2

# --- Inward Velocity Nudge ---

v = np.zeros\_like(r)

v += -0.04 \* np.exp(-((r - 27)\*\*2) / 4)

v += 0.04 \* np.exp(-((r - 33)\*\*2) / 4)

# --- Curvature Field ---

def curvature(r, omega):

return omega\*\*2 \* np.exp(-((r - 30)\*\*2) / sigma\*\*2)

ψ1\_snapshots = []

ψ2\_snapshots = []

ψ\_total\_snapshots = []

times = []

# --- Main Simulation Loop (no animation, just snapshot data) ---

for frame in range(timesteps):

t = frame \* dt

omega = omega0 \* np.exp(-t / tau)

C = curvature(r, omega)

potential = 1 + C \* 0.0005

ψ\_total = ψ1 + ψ2

ψ\_total /= np.max(np.abs(ψ\_total))

laplacian = np.zeros\_like(ψ\_total)

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

accel = laplacian / potential

v += accel \* dt

ψ\_total += v \* dt

ψ1 = ψ\_total \* np.exp(-((r - 27)\*\*2) / 50)

ψ2 = ψ\_total \* np.exp(-((r - 33)\*\*2) / 50)

if frame in frames\_to\_plot:

ψ1\_snapshots.append(ψ1.copy())

ψ2\_snapshots.append(ψ2.copy())

ψ\_total\_snapshots.append(ψ\_total.copy())

times.append(t)

# --- Plotting Snapshots ---

plt.figure(figsize=(14, 8))

for i, (t, ψ1\_s, ψ2\_s, ψt\_s) in enumerate(zip(times, ψ1\_snapshots, ψ2\_snapshots, ψ\_total\_snapshots)):

plt.subplot(2, 3, i+1)

plt.plot(r, ψ1\_s, color='gold', alpha=0.6, label='ψ₁')

plt.plot(r, ψ2\_s, color='teal', alpha=0.6, label='ψ₂')

plt.plot(r, ψt\_s, color='crimson', label='ψ₁+ψ₂')

plt.title(f"t = {t:.2f} s")

plt.ylim(-0.2, 1.2)

plt.xlim(15, 45)

if i == 0:

plt.legend()

plt.suptitle("Snapshots: MBT Dispatch Node – Wave Packet Merging")

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

plt.show()