import numpy as np

import matplotlib.pyplot as plt

from scipy.integrate import trapezoid

# PARAMETERS

grid\_size = 100

timesteps = 140

dt = 0.25

width = 3

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

dr = r[1] - r[0]

# INIT 4 STATIC PARTICLES (central tetra-phase structure)

centers = [35, 50, 65, 45]

phases = [0, 2\*np.pi/3, 4\*np.pi/3, np.pi]

ψ\_static = [np.exp(-((r - c)\*\*2)/(2\*width\*\*2)) \* np.exp(1j \* p) for c, p in zip(centers, phases)]

# GLUEBALL DUET — ψ₅ and ψ₆, same phase and speed

xL, xR = 20, 80

k = +0.55 # symmetric magnitude, opposite direction

ψ5 = np.exp(-((r - xL)\*\*2)/(2\*width\*\*2)) \* np.exp(1j \* k \* r)

ψ6 = np.exp(-((r - xR)\*\*2)/(2\*width\*\*2)) \* np.exp(1j \* -k \* r)

# Combine all wavefunctions

ψ = np.array(ψ\_static + [ψ5, ψ6])

n = len(ψ)

# STORAGE

ψ\_total = []

overlap = []

CoMs = []

# TIME EVOLUTION

for t in range(timesteps):

V = -np.exp(-((r - grid\_size/2)/10)\*\*2) # MBT dispatch well

for i in range(n):

lap = np.zeros\_like(ψ[i], dtype=complex)

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

ψ[i] += dt \* (0.8 \* lap - 0.6 \* V \* ψ[i])

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

if norm != 0:

ψ[i] /= norm

ψ\_total.append(np.sum(ψ, axis=0))

ov = sum(trapezoid(np.abs(ψ[i]\*np.conj(ψ[j])), r)

for i in range(n) for j in range(i+1, n))

overlap.append(ov)

CoMs.append([trapezoid(r \* np.abs(ψ[i])\*\*2, r)/trapezoid(np.abs(ψ[i])\*\*2, r) for i in range(n)])

# PLOT RESULTS

ψ\_total = np.array(ψ\_total)

CoMs = np.array(CoMs)

colors = ['skyblue', 'orange', 'lime', 'crimson', 'gold', 'hotpink']

fig, axs = plt.subplots(3, 1, figsize=(10,10))

for i in range(n):

axs[0].plot(r, np.abs(ψ[i])\*\*2, label=f'ψ{i+1}', color=colors[i])

axs[0].plot(r, np.abs(ψ\_total[-1])\*\*2, color='black', lw=2, label='Final Total')

axs[0].set\_title("Tetra Composite + Synchronized Dual Glueball")

axs[0].legend(ncol=3)

axs[1].plot(np.linspace(0, dt\*timesteps, timesteps), overlap, color='mediumvioletred')

axs[1].set\_ylabel("Total Overlap")

axs[1].set\_title("Resonance Fusion Monitor")

for i in range(n):

axs[2].plot(np.linspace(0, dt\*timesteps, timesteps), CoMs[:, i], label=f'ψ{i+1} CoM', color=colors[i])

axs[2].set\_title("Center of Mass Drift — Duet Convergence")

axs[2].set\_xlabel("Time")

axs[2].legend(ncol=3)

plt.tight\_layout()

plt.show()