import numpy as np

import matplotlib.pyplot as plt

# --- Parameters ---

grid\_size = 140

timesteps = 150

dt = 0.18

width = 5

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

dr = r[1] - r[0]

# Two superconducting "wells"

well1, well2 = 40, 100

ψ1 = np.exp(-((r - well1)\*\*2) / (2\*width\*\*2)) \* np.exp(1j \* 0.2 \* r)

ψ2 = np.exp(-((r - well2)\*\*2) / (2\*width\*\*2)) \* np.exp(1j \* 0.0 \* r)

ψ = np.array([ψ1, ψ2])

n = len(ψ)

# Barrier (Josephson junction)

barrier\_center = 70

barrier\_width = 6

barrier\_height = 0.06

V = np.zeros\_like(r)

V[(r > barrier\_center - barrier\_width/2) & (r < barrier\_center + barrier\_width/2)] = barrier\_height

# Storage

currents, phases, t\_axis = [], [], []

for t in range(timesteps):

# MBT: Keep phase memory!

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 \* (0.7 \* lap - 0.55 \* V \* ψ[i])

norm = np.linalg.norm(ψ[i])

if norm != 0: ψ[i] /= norm

# "Current" = phase gradient in the barrier region

total = np.sum(ψ, axis=0)

phase = np.angle(total)

current = np.mean(np.gradient(phase)[(r > barrier\_center - 2) & (r < barrier\_center + 2)])

currents.append(current)

phases.append(phase)

t\_axis.append(t \* dt)

# --- Plot ---

plt.figure(figsize=(8,5))

plt.plot(t\_axis, currents, label="MBT Supercurrent")

plt.xlabel("Time")

plt.ylabel("Current (phase gradient)")

plt.title("MBT Josephson Junction (Supercurrent Across Barrier)")

plt.legend()

plt.tight\_layout()

plt.show()