# MBT Double Slit Experiment – Curvature Pulse Interference

This simulation models the Motion = Being Theory (MBT) version of the classic Double Slit Experiment.  
Instead of photons or particles, a single curvature pulse (tension disturbance) propagates through a field   
and interacts with a wall containing two slits.  
  
The result is an interference pattern on the far side of the slits —   
not because a particle passed through both, but because tension geometry flows along all available paths.  
  
No particles. No observers. No collapse. Only motion and resistance.

# MBT Double Slit Experiment – Curvature Pulse with Slits  
import numpy as np  
import matplotlib.pyplot as plt  
from matplotlib.animation import FuncAnimation  
  
# --- Parameters ---  
grid\_size = 120  
timesteps = 200  
dt = 0.1  
pulse\_amplitude = 1.0  
decay\_factor = 0.98  
  
# Slit barrier setup  
slit\_y = grid\_size // 3  
slit\_width = 3  
gap\_size = 10  
  
# Initialize fields  
field = np.zeros((grid\_size, grid\_size))  
velocity = np.zeros\_like(field)  
barrier = np.ones((grid\_size, grid\_size), dtype=bool)  
  
# Create barrier with two slits  
barrier[slit\_y, :] = False  
center = grid\_size // 2  
barrier[slit\_y, center - gap\_size - slit\_width:center - gap\_size] = True  
barrier[slit\_y, center + gap\_size:center + gap\_size + slit\_width] = True  
  
# Inject a pulse just before the slits  
field[slit\_y - 2, center] = pulse\_amplitude  
  
# Visualization setup  
fig, ax = plt.subplots(figsize=(7, 6))  
im = ax.imshow(field, cmap='plasma', vmin=-1, vmax=1, animated=True)  
ax.set\_title("MBT Double Slit Experiment (Curvature Pulse Interference)")  
  
# Laplacian  
def laplacian(Z):  
 return (  
 -4 \* Z  
 + np.roll(Z, 1, axis=0) + np.roll(Z, -1, axis=0)  
 + np.roll(Z, 1, axis=1) + np.roll(Z, -1, axis=1)  
 )  
  
# Update function  
def update(frame):  
 global field, velocity  
 lap = laplacian(field)  
 velocity += lap \* dt  
 velocity \*= decay\_factor  
 field += velocity \* dt  
 field \*= barrier # apply slit mask  
 im.set\_array(field)  
 return [im]  
  
ani = FuncAnimation(fig, update, frames=timesteps, interval=50, blit=True)  
plt.show()