Godframe Simulation Code: Collision Detonation Model

Date: June 19, 2025

# Description

This document contains the full Python code for the Godframe scalar field simulation based on the Collision Detonation model. This version includes a high-energy Ξ field that sharply drops, triggering scalar field φ ignition via a hard Heaviside activation. The field receives an explicit initial acceleration to simulate a superluminal-to-frictional impact event and successfully demonstrates scalar ignition through asymmetry and curvature.

# Python Simulation Code

# Godframe Flashpoint: COLLISION DETONATION MODEL  
import numpy as np  
import matplotlib.pyplot as plt  
import matplotlib.animation as animation  
  
# Constants  
c, G = 1, 1  
phi\_0 = 1  
lambda\_ = 10  
Xi\_c = c\*\*5 / G  
  
# Grid  
nx, nt = 200, 300  
dx, dt = 1.0, 0.1  
x = np.linspace(-nx//2, nx//2, nx)  
  
# Fields  
phi = np.zeros((nt, nx))  
Xi = np.zeros((nt, nx))  
  
# Ξ: superluminal state that crashes into resistance  
for t in range(nt):  
 Xi[t] = 50 \* Xi\_c \* np.exp(-((x / 2)\*\*2)) \* (1 - np.tanh(5 \* (t - nt//4)))  
  
# φ: still + sharp asymmetric kick + explosive acceleration  
phi[0] = 0  
phi[1] = 0.5 \* np.exp(-((x + 3) / 0.5)\*\*2)  
phi[2] = phi[1] + dt\*\*2 \* 5.0 \* np.exp(-((x + 3) / 0.5)\*\*2) # raw φ\_tt injection  
  
# Heaviside trigger  
def theta(Xi): return np.heaviside(Xi - Xi\_c, 0)  
  
# Potential  
def V\_prime(phi): return lambda\_ \* (phi\*\*3 + 0.5 \* phi\*\*5) - lambda\_ \* phi \* phi\_0\*\*2  
  
# Leapfrog evolution  
for t in range(2, nt - 1):  
 th = theta(Xi[t])  
 lap = (np.roll(phi[t-1], -1) - 2 \* phi[t-1] + np.roll(phi[t-1], 1)) / dx\*\*2  
 phi\_tt = th \* (lap - V\_prime(phi[t-1]))  
 phi[t] = 2 \* phi[t-1] - phi[t-2] + dt\*\*2 \* phi\_tt  
  
# Visualization  
fig, ax = plt.subplots()  
im = ax.imshow(phi.T, cmap='plasma', aspect='auto', origin='lower', vmin=0, vmax=0.001)  
ax.set\_title('GODFRAME COLLISION DETONATION: φ(x, t)')  
ax.set\_xlabel('Time')  
ax.set\_ylabel('Space')  
  
def animate(i): im.set\_array(phi[:i+1].T); return [im]  
ani = animation.FuncAnimation(fig, animate, frames=nt, interval=40, blit=True)  
ani.save("Godframe\_COLLISION\_DETONATION.gif", writer='pillow', fps=30)  
  
# Output  
print("Max φ:", np.max(phi))  
print("Final φ center:", phi[-1, nx//2])