Phase 8 – Part 13  
Hamiltonian / Canonical Formulation of ψ-Gravity

🎯 Goal  
In this part, I aim to recast ψ-gravity into a Hamiltonian (canonical) framework, where ψ and its conjugate momentum form the basis of phase space. This allows me to ask whether ψ-gravity supports a well-defined Hamiltonian structure, opening the pathway toward quantization in Phase 8 Part 14.

The central question:  
Can I write ψ-gravity in the form

Plain text:  
H[ψ, πψ] = ∫ d^d x 𝓗(ψ, πψ, ∇ψ, …)

where the evolution equations follow from Hamilton’s equations.

🏜 Desert Analogy

* ψ = desert floor (substrate).
* πψ = “momentum of the desert floor” (how fast the ground shifts beneath the dunes).
* Gravity = pressure from sand + wind² interacting with ψ.
* Force = dunes evolving across the desert surface.

In this analogy, introducing πψ means the desert floor is not static but dynamically coupled to currents (wind energy).

⚖️ Canonical Setup  
The upgraded ψ-gravity equation is:

Plain text:  
Gravity(x) = (∇²[space(x) + current(x)²]) ψ(x)

and the force is

Plain text:  
F(x) = −∇[Gravity(x)]

To transition into Hamiltonian form, I introduce:

* ψ(x,t) = field coordinate.
* πψ(x,t) = conjugate momentum, defined through the Lagrangian.

📐 Effective Lagrangian Ansatz  
I propose a Lagrangian density inspired by Klein–Gordon-like dynamics:

Plain text:  
𝓛 = 1/2 (∂t ψ)² − (c²/2)(∇ψ)² − V(ψ, x)

with potential

Plain text:  
V(ψ, x) = 1/2 M(x) ψ(x)²  
M(x) = ∇²[space(x) + current(x)²]

This matches the ψ-gravity definition: curvature + wind² set the effective “mass term.”

🔑 Conjugate Momentum  
From the Lagrangian:

Plain text:  
πψ(x,t) = ∂t ψ(x,t)

🧮 Hamiltonian Density  
The Hamiltonian density is

Substitute:

Plain text:  
𝓗 = 1/2 πψ² + (c²/2)(∇ψ)² + 1/2 M(x) ψ²

Thus the full Hamiltonian:

Plain text:  
H[ψ, πψ] = ∫ d^d x [ 1/2 πψ² + (c²/2)(∇ψ)² + 1/2 M(x) ψ² ]

📜 Hamilton’s Equations  
The evolution equations are:

Plain text:  
∂t ψ = δH/δπψ = πψ

Plain text:  
∂t πψ = −δH/δψ = c² ∇²ψ − M(x) ψ

These are wave-like equations with an effective mass term set by ψ-gravity curvature.

🌊 Interpretation

* ψ = field coordinate: shape of the desert floor.
* πψ = canonical momentum: how quickly the floor shifts.
* Hamiltonian = total energy: dune energy + ground deformation + curvature-induced mass.
* M(x) = Laplacian of (space + wind²): encodes how the desert’s texture dictates motion.

This formulation gives ψ-gravity a canonical structure, paving the way for quantization.

🖥️ Python Simulation — Canonical Evolution

# simulations/phase8\_part13\_hamiltonian.py  
import numpy as np  
import matplotlib.pyplot as plt  
  
# Grid setup  
L = 20.0  
N = 512  
dx = L / N  
x = np.linspace(-L/2, L/2, N, endpoint=False)  
  
# Parameters  
c = 1.0  
dt = 0.01  
steps = 1000  
  
# Define space and current  
space = np.exp(-x\*\*2 / 10.0) # localized bump  
current = 0.5 \* np.sin(2\*np.pi\*x / L)  
  
# Effective mass term: Laplacian of (space + current^2)  
def laplacian(f, dx):  
 return (np.roll(f, -1) - 2\*f + np.roll(f, 1)) / dx\*\*2  
  
M = laplacian(space + current\*\*2, dx)  
  
# Initial conditions  
psi = np.exp(-x\*\*2) # Gaussian profile  
pi = np.zeros\_like(x) # conjugate momentum  
  
# Storage  
psi\_history = []  
pi\_history = []  
  
for step in range(steps):  
 # Hamilton's equations  
 dpsi\_dt = pi  
 dpi\_dt = c\*\*2 \* laplacian(psi, dx) - M \* psi  
  
 # Leapfrog integration  
 pi += dpi\_dt \* dt  
 psi += dpsi\_dt \* dt  
  
 if step % 10 == 0:  
 psi\_history.append(psi.copy())  
 pi\_history.append(pi.copy())  
  
# Plot result  
plt.figure(figsize=(8, 4))  
plt.imshow(psi\_history, aspect='auto', extent=[x[0], x[-1], steps\*dt, 0],  
 cmap='RdBu', interpolation='bilinear')  
plt.colorbar(label="ψ(x,t)")  
plt.xlabel("x")  
plt.ylabel("time")  
plt.title("Phase 8 Part 13: Hamiltonian ψ-Gravity Evolution")  
plt.show()