⚙️ Phase 7 – Part 6: Observable Predictions & Empirical Handles

This section explores the observable consequences of the ψ-gravity framework with the upgraded equation:

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

## 1. Light Bending and Gravitational Lensing

In General Relativity, light follows null geodesics in curved spacetime, producing lensing effects. In ψ-gravity, the Laplacian of , modulated by , acts as an effective refractive index. Thus light bending will depend not only on mass/curvature but also on current² contributions.

Key prediction: Environments with strong current flows (plasma streams, rotating halos, or cosmic flows) will produce lensing anomalies even if mass density is low.

**Python Simulation: 2D Ray-Tracing**

# 2D Ray Tracing Simulation in ψ-gravity  
import numpy as np  
import matplotlib.pyplot as plt  
  
# Grid setup  
N = 200  
x = np.linspace(-5, 5, N)  
y = np.linspace(-5, 5, N)  
X, Y = np.meshgrid(x, y)  
  
def space\_field(x, y):  
 return -1.0/np.sqrt(x\*\*2 + y\*\*2 + 0.1)  
  
def current\_field(x, y):  
 return 0.5\*np.exp(-(x\*\*2 + y\*\*2))  
  
field = space\_field(X, Y) + current\_field(X, Y)\*\*2  
  
# Laplacian (finite-difference)  
dx = x[1] - x[0]  
laplacian = (  
 np.roll(field, 1, axis=0) + np.roll(field, -1, axis=0) +  
 np.roll(field, 1, axis=1) + np.roll(field, -1, axis=1) - 4\*field  
) / dx\*\*2  
  
# Rays  
rays = []  
for y0 in np.linspace(-2, 2, 5):  
 pos = np.array([-5.0, y0])  
 vel = np.array([0.05, 0.0])  
 traj = []  
 for \_ in range(400):  
 ix = np.argmin(np.abs(x - pos[0]))  
 iy = np.argmin(np.abs(y - pos[1]))  
 gx, gy = np.gradient(laplacian)  
 acc = np.array([gx[iy, ix], gy[iy, ix]]) \* 0.01  
 vel += acc  
 pos += vel  
 traj.append(pos.copy())  
 rays.append(np.array(traj))  
  
plt.figure(figsize=(6, 6))  
plt.contourf(X, Y, laplacian, levels=50, cmap='inferno')  
for traj in rays:  
 plt.plot(traj[:, 0], traj[:, 1], 'w-')  
plt.title("2D Ray-trace in ψ-gravity field")  
plt.xlabel("x")  
plt.ylabel("y")  
plt.show()

## 2. Test Particle Orbits and Precession

Test particles in ψ-gravity experience forces from both curvature (space) and dynamic flows (current²). I expect measurable deviations in orbital precession compared to Newtonian and GR predictions in regimes where the current² contribution or ψ inhomogeneity is non-negligible. Numerical orbit integrations (using the Force = −∇[Gravity(x)]) provide concrete precession rates to compare with observation.

## 3. Wave Propagation in ψ-gravity

ψ-gravity may permit wave-like modulations of the ψ-field itself. This can be tested by propagating a wave packet through a medium where space+current² vary. Observable effects include dispersion or distortion of signals traveling through gravitational environments (e.g., pulses from pulsars or fast radio bursts traversing regions with strong flows).

## Observational signatures & empirical handles (summary)

* **Lensing anomalies:** Strong current regions (rotating halos, jets, accretion flows) produce lensing deviation from mass-only models.
* **Orbital precession departures:** Test bodies in regions with high ∇²(current²) or sharp ψ-gradients should show anomalous perihelion/precession signals.
* **Signal dispersion/distortion:** Wave packets traveling through inhomogeneous ψ-modulated curvature show frequency-dependent delays or scattering distinct from plasma dispersion.
* **Parameter diagnostics:** I use the dimensionless ratio
* R(x) = current(x)² / space(x)  
  R\_tilde(x) = | ∇² (current(x)²) | / | ∇² (space(x)) |

to identify candidate regions where ψ-gravity effects are observationally accessible.

Plain text:  
R(x) = current(x)² / space(x)  
R\_tilde(x) = | ∇² (current(x)²) | / | ∇² (space(x)) |

I include the above simulation code for initial exploration and archival; I executed calculations and example code with the AI to generate diagnostic trajectories and visual intuition for lensing and ray-deflection in ψ-gravity.