Phase 8 – Part 7  
Stability, Attractors, and Chaos in ψ–Metric Feedback  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

🎯 Goal  
To study the long-term dynamics of the ψ–geometry feedback system developed in Part 6.  
We want to determine:

* Are there stable attractor states where ψ settles?
* Do oscillatory states arise, analogous to standing waves?
* Are there chaotic regimes, where ψ–metric feedback never stabilizes?

This part explores ψ-gravity’s nonlinear phase space.

🔧 Setup  
We start from the coupled feedback law (from Part 6):

Plain text: □ψ - m² ψ = β R ψ

With approximate Ricci scalar:

Plain text: R(x,t) ≈ γ \* ∂²\_x ψ(x,t)

So in 1D:

Plain text: ∂²\_t ψ - c² ∂²\_x ψ - m² ψ = βγ (∂²\_x ψ) ψ

🌊 Desert Analogy Extension

* Stable attractor: dunes settle into repeating shapes; the desert floor achieves equilibrium.
* Oscillatory state: dunes “breathe,” shifting back and forth but staying bounded.
* Chaos: dunes collapse and reform endlessly, no stable pattern emerges.

📐 Dynamical Systems Framing  
We can rewrite the evolution as:

Plain text: ψ¨ = c² ψ’’ + m² ψ + βγ ψ ψ’’

Where overdots = time derivatives, primes = space derivatives.

Key parameter regimes:

* Small βγ → weak feedback → nearly linear waves.
* Moderate βγ → nonlinear but bounded oscillations.
* Large βγ → chaos emerges.

🔬 Sample Case: Nonlinear Oscillations

We test three cases for βγ:

* Stable (βγ small): ψ relaxes to equilibrium.
* Oscillatory (βγ moderate): ψ oscillates around a trench.
* Chaotic (βγ large): ψ fluctuates wildly without repeating pattern.

🖥️ Python Simulation

# simulations/phase8\_part7\_stability.py  
import numpy as np  
import matplotlib.pyplot as plt  
  
# Parameters  
L = 10.0  
N = 200  
dx = L / N  
dt = 0.01  
steps = 800  
c = 1.0  
m = 0.1  
gamma = 1.0  
  
x = np.linspace(-L/2, L/2, N)  
  
def evolve(beta, steps=steps):  
 psi = np.exp(-x\*\*2) # initial Gaussian  
 psi\_old = np.exp(-x\*\*2)  
 snapshots = []  
  
 for step in range(steps):  
 laplacian = (np.roll(psi, -1) - 2\*psi + np.roll(psi, 1)) / dx\*\*2  
 R = gamma \* laplacian  
 psi\_new = (2\*psi - psi\_old +  
 dt\*\*2 \* (c\*\*2 \* laplacian - m\*\*2 \* psi + beta \* R \* psi))  
 psi\_old, psi = psi, psi\_new  
  
 if step % 100 == 0:  
 snapshots.append(psi.copy())  
 return snapshots  
  
# Three regimes  
betas = [0.1, 1.0, 5.0]  
plt.figure(figsize=(12,6))  
  
for i, beta in enumerate(betas, 1):  
 snapshots = evolve(beta)  
 plt.subplot(1,3,i)  
 for j, snap in enumerate(snapshots):  
 plt.plot(x, snap, label=f"t={j\*100\*dt:.2f}")  
 plt.title(f"β = {beta}")  
 plt.xlabel("x")  
 if i == 1:  
 plt.ylabel("ψ")  
 plt.legend()  
  
plt.suptitle("ψ Evolution: Stable, Oscillatory, Chaotic")  
plt.tight\_layout()  
plt.show()