# Phase 5.11 — Perturbation Analysis

In this phase, I investigate the response of the ψ field to small perturbations. The goal is to study stability: whether fluctuations grow (instability), decay (damping), or persist (neutral stability). This connects ψ-gravity dynamics to wave stability and soliton theory.

## Core ψ-Gravity Equation (Restated)

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

Corresponding force:

Plain text:  
Force(x) = −∇[Gravity(x)]

## Linearization Around Background ψ

Let the field be decomposed as a background ψ plus a small perturbation δψ:

Plain text:  
ψ(x,t) = ψ₀(x) + δψ(x,t), with δψ ≪ ψ₀.

Substituting into the ψ-gravity equation, I expand:

Splitting terms:

Thus δψ evolves linearly under the same Laplacian-weighted operator as ψ₀.

## Effective Perturbation Equation

I define the perturbation evolution operator:

Plain text:  
L[δψ] = (∇²[space + current²]) × δψ

The growth rate of perturbations depends on the eigenvalues of this operator. Positive eigenvalues → growth (instability), negative → decay (damping).

## Energy Shift from Perturbations

The energy density defined in Phase 5.10 modifies under perturbation:

Expanding to linear order in δψ:

Plain text:  
E ≈ E₀ + ∇ψ₀·∇δψ + 0.5 (∇²[space+current²]) ψ₀ δψ

This shows how perturbations shift the energy landscape locally.

## Numerical Simulation: Perturbation Evolution in 2D

I simulate ψ in 2D as in Phase 5.10, but add a small Gaussian perturbation δψ to test stability.

# -----------------------------  
# simulations/phase5\_part5\_11\_ψ-Perturbation\_Analysis.py  
# Phase 5.11 — Perturbation stability analysis  
# -----------------------------  
  
import numpy as np  
import matplotlib.pyplot as plt  
  
# Grid setup  
N = 100  
x = np.linspace(0, 2\*np.pi, N)  
y = np.linspace(0, 2\*np.pi, N)  
X, Y = np.meshgrid(x, y)  
  
# Base ψ field (background)  
psi0 = np.exp(-((X-np.pi)\*\*2 + (Y-np.pi)\*\*2) / 0.3)  
  
# Small perturbation δψ (Gaussian ripple off-center)  
delta\_psi = 0.05 \* np.exp(-((X-1.5\*np.pi)\*\*2 + (Y-1.2\*np.pi)\*\*2) / 0.2)  
  
# Total ψ = ψ0 + δψ  
psi = psi0 + delta\_psi  
  
# Define space(x,y) and current(x,y)  
space = np.sin(X) \* np.cos(Y)  
current = np.cos(X) \* np.sin(Y)  
  
# Laplacian operator  
def laplacian(Z, dx):  
 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)  
 ) / dx\*\*2  
  
dx = x[1] - x[0]  
  
# Gravity field  
gravity = laplacian(space + current\*\*2, dx) \* psi  
  
# Energy density  
grad\_x, grad\_y = np.gradient(psi, dx, dx)  
energy = 0.5\*(grad\_x\*\*2 + grad\_y\*\*2) + 0.5\*gravity\*psi  
  
# Visualization  
fig, axs = plt.subplots(1, 3, figsize=(15,5))  
axs[0].imshow(psi, extent=[0,2\*np.pi,0,2\*np.pi], origin='lower')  
axs[0].set\_title("Perturbed ψ Field")  
  
axs[1].imshow(gravity, extent=[0,2\*np.pi,0,2\*np.pi], origin='lower')  
axs[1].set\_title("Gravity with Perturbation")  
  
axs[2].imshow(energy, extent=[0,2\*np.pi,0,2\*np.pi], origin='lower')  
axs[2].set\_title("Energy Density (Perturbed)")  
  
plt.show()