Phase 9 – Part 3  
Thermodynamic Flows and Entropic Currents in ψ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

🎯 Goal  
In this part, I extend the ψ-thermodynamic picture into the language of nonequilibrium thermodynamics.  
The aim is to show how ψ not only encodes equilibrium distributions (Part 1) and fluctuation–dissipation structures (Part 2), but also supports transport of entropy and irreversible flows.  
This establishes ψ as a thermodynamic fluid: currents push ψ out of equilibrium, entropy gradients drive flows, and dissipation ensures eventual relaxation.

🏜 Desert Analogy Extension

* The desert floor (ψ) now channels thermodynamic rivers: entropic currents sliding down irregularities.
* The wind (current²) injects constant disturbance, like heating the desert surface.
* Sand (space) drifts toward low “thermodynamic potential” regions, much like mass flows downhill.
* Dunes (force) are no longer static but continually reshaped by entropic winds.

The desert becomes a dynamic terrain where ψ mediates flows of entropy and free energy.

⚖️ ψ-Entropy Current

Define an entropy density s(x,t) associated with ψ:

Plain text:  
s(x,t) = − ψ(x,t) ln ψ(x,t)

The entropy current Jₛ(x,t) is given by:

Plain text:  
Js(x,t) = − Ds ∇ s(x,t)

Where Dₛ is an entropic diffusivity constant.  
Entropy flows downhill along its gradient, seeking equilibrium.

🔹 Continuity Equation for Entropy

Entropy satisfies a conservation-like relation:

Plain text:  
∂s/∂t + ∇·Js = σ(x,t)

Here:

* σ(x,t) ≥ 0 is the entropy production rate.
* This reflects the irreversible tendency of ψ to generate entropy under agitation.

🔹 Entropy Production

Inspired by nonequilibrium thermodynamics, entropy production is quadratic in currents:

Plain text:  
σ(x,t) = Js² / (Ds s)

Entropy is always produced (nonnegative), consistent with the second law of thermodynamics.

🔹 Thermodynamic Force in ψ

The thermodynamic force driving ψ dynamics is the gradient of chemical potential-like quantity μψ:

Plain text:  
μψ = δF/δψ

Transport law:

Plain text:  
Jψ = − Lψ ∇ μψ

Where Lψ is a transport coefficient.  
This is the ψ-gravity analogue of Onsager’s linear response laws.

🔬 Example: Entropic Drift of Gaussian ψ

Suppose ψ initially Gaussian:

Plain text:  
ψ(x,0) = (1 / sqrt(2πσ²)) exp(−x² / (2σ²))

Entropy density is:

Plain text:  
s(x,0) = − ψ(x,0) ln ψ(x,0)

Entropy flux evolves as diffusion of s, leading to spreading and eventual flattening.

🖥 Python Demonstration

# simulations/phase9\_part3\_entropy\_currents.py  
import numpy as np  
import matplotlib.pyplot as plt  
  
# parameters  
L = 50  
N = 400  
dx = L / N  
dt = 0.01  
steps = 1000  
Ds = 0.5  
  
# grid  
x = np.linspace(-L/2, L/2, N)  
psi = (1.0 / np.sqrt(2\*np.pi\*2.0\*\*2)) \* np.exp(-x\*\*2 / (2\*2.0\*\*2))  
  
def entropy\_density(psi):  
 return -psi \* np.log(psi + 1e-12)  
  
# evolve entropy density via diffusion  
s = entropy\_density(psi)  
history = []  
for step in range(steps):  
 laplacian = (np.roll(s, -1) - 2\*s + np.roll(s, 1)) / dx\*\*2  
 s += dt \* Ds \* laplacian  
 if step % 200 == 0:  
 history.append(s.copy())  
  
# plot snapshots  
for i, snapshot in enumerate(history):  
 plt.plot(x, snapshot, label=f"t={i\*200\*dt:.1f}")  
plt.legend()  
plt.title("ψ Entropy Density Evolution")  
plt.xlabel("x")  
plt.ylabel("s(x,t)")  
plt.show()