Phase 12 – Quantization & Coupling  
Part 3: ψ Quantization Framework

Canonical quantization rules  
I promote fields and conjugate momenta to operators with equal-time commutators:

Plain text:  
[ψ(x,t), π(y,t)] = iħ δ³(x−y); [ψ,ψ]=0; [π,π]=0.

Quantum Hamiltonian operator  
The Hamiltonian operator is

Plain text:  
H = ∫ d³x [ 1/2 π² + 1/2 (∇ψ)² + Vψ(x) ψ² ].

ψ field expansion (free-mode decomposition)  
In a region where is approximately constant (or translationally symmetric), expand ψ in Fourier modes:

Plain text:  
ψ(x,t) = ∫ d³k/(2π)³ 1/√(2ωk) [ a\_k e^{i(k·x−ωk t)} + a†\_k e^{−i(k·x−ωk t)} ].

with commutators

Plain text:  
[a\_k, a†\_{k’}] = (2π)³ δ³(k−k’).

The dispersion relation follows from the ψ equation of motion:

Plain text:  
ωk² = |k|² + mψ²(x), with mψ²(x) = 2 Vψ(x).

Path integral formulation  
The generating functional is

Plain text:  
Z[J] = ∫ Dψ exp{ i/ħ ∫ d⁴x [ 1/2 (∂μ ψ)² − Vψ(x) ψ² + J(x) ψ ] }.

The ψ propagator in position space satisfies

Plain text:  
(∂t² − ∇² − 2Vψ(x)) G(x,y) = iħ δ⁴(x−y).

Quantum vacuum structure  
The vacuum is defined by

Plain text:  
a\_k |0⟩ = 0.

Expectation value of ψ fluctuations:

Plain text:  
⟨0| ψ(x,t) ψ(y,t) |0⟩ = G(x,y).

Numerical test: ψ quantum fluctuation spectrum (stochastic sampling)  
A discrete approximation using Fourier modes with random Gaussian amplitudes to mimic vacuum fluctuations.

# simulations/phase12\_part3\_quantization\_modes.py  
import numpy as np  
  
# Grid setup  
Nx = 256  
Lx = 20.0  
dx = Lx / Nx  
x = np.linspace(-Lx/2, Lx/2, Nx, endpoint=False)  
  
# Momentum modes  
dk = 2\*np.pi / Lx  
k = np.fft.fftfreq(Nx, d=dx) \* 2\*np.pi  
  
# Define effective mass term (from Vψ, here set as constant for test)  
mpsi = 1.0  
omega\_k = np.sqrt(k\*\*2 + mpsi\*\*2)  
  
# Random Gaussian amplitudes with variance ~ ħ/2ωk  
np.random.seed(0)  
a\_real = np.random.normal(0, 1, Nx)  
a\_imag = np.random.normal(0, 1, Nx)  
a\_k = (a\_real + 1j\*a\_imag) / np.sqrt(2\*omega\_k)  
  
# Construct ψ(x) from inverse Fourier transform  
psi\_fluct = np.fft.ifft(a\_k).real  
  
# Power spectrum |a\_k|^2  
power = np.abs(a\_k)\*\*2  
  
print("Sample ψ fluctuation field (first 10 points):", psi\_fluct[:10])  
print("Sample power spectrum (first 10 modes):", power[:10])