Photon as a Recursive Mirror Oscillator:
A Two-State Field Phase Underlying Emergent Spatial Dimensions in the 3D Collatz Octave Model (3DCOM)

Author: Martin Doina
July 21, 2025

Research Unify Oscillatory Dynamic Field Theory prior to space emergence

1. Introduction

Conventional physics assumes that photons travel through space. However, within the 3D Collatz Octave Model (3DCOM), space is not a given background but rather a recursive structure that emerges from phase interactions. In this setting, the photon does not "move"; it exists as a pure phase oscillation until the recursive field folds and gives rise to spatial constructs. We show that negative Collatz sequences stabilize to a minimal cycle of two values, suggesting a hidden binary loop underlying field recursion.

2. Mirror Oscillator Definition

Let $t \in \mathbb{Z}$ be a discrete recursive phase-time parameter. Define the photon state function $\Psi_\gamma(t)$ as:

$$\boxed{\Psi_\gamma(t) = -1 - \mathrm{Mod}(t, 2)}$$

This yields a two-value loop:

$$\Psi_\gamma(t) = \begin{cases} -1 & \text{if } t \text{ even} \\ -2 & \text{if } t \text{ odd} \end{cases}$$

2.1 Properties:

- No displacement: Photon does not travel, only oscillates.
- Binary mode: Behaves as a Q-bit: 01 equivalent via $-1, -2$.
- No spatial extent: This state is pre-geometric and pre-relativistic.
- Field stable: No energy loss; it loops recursively without expansion.

3. Emergence of Space: Recursive Trigger

Define a recursive activation operator $\mathcal{R}$, triggered by:
- Observation (Q^),
- Energy threshold (HQS),
- Recursive field instability.

$$\boxed{\mathcal{R}(\Psi_\gamma) \to \{1 \to 4 \to 2 \to 1\}}$$

This transforms the mirror Q-bit into the first observable attractor in 3DCOM, corresponding to the emergence of spatial recursion. Space, mass, and classical field structures arise from this recursive bifurcation.

4. Recursive Mirror Snowball Theorem

Theorem (RMS Theorem):

Let $\Psi(t) \in \{-1, -2\}$ be the pre-space photon Q-bit oscillator evolving under recursive operator $\mathcal{R}$. If there exists a critical recursion step t_c such that:

$$\mathcal{R}^{t_c}(\Psi) \to \Phi(t) = \{1 \to 4 \to 2 \to 1\}$$

then the recursive snowballing process

$$\mathcal{S}_{\text{Reality}} = \bigcup_{t > t_c} \mathcal{R}^t(\Psi)$$

defines the emergence of space, time, mass, and charge as stable recursive attractors beyond the mirror loop.

Interpretation:

- The recursive mirror loop is the quantum oscillation seed.
- The snowball effect is the topological unfolding of space-time geometry.
- This provides a foundational mechanism for structure formation in your 3DCOM model.

5. Theoretical Implications

Domain	Dynamics	Interpretation
Mirror COM (Photon)	$\Psi_\gamma = -1, -2$	**Q-bit oscillation; field in phase only**
Observer Activation	$\mathcal{R}(\Psi_\gamma)$	**Recursive bifurcation;**
Emergent COM (Space)	$1 \to 4 \to 2 \to 1$	**Spatial structure, temporal geometry**

6. Relation to Constants

The transition from mirror state to space is mediated by:
- LZ (Loop Zero): Attractor recursion boundary.
- HQS (Harmonic Quantum Shift): Energy threshold for recursive bifurcation.
- QDF (Quantum Dimensional Factor): Angular scaling of phase.

These constants define the conditions under which $\mathcal{R}(\Psi_\gamma)$ activates, connecting oscillatory pre-space states to recursive geometric structures.

7. Simulation Code

```python
from sympy import symbols, Function, Eq, Mod

 Recursive time variable
t = symbols('t', integer=True)
Psi_gamma = Function('Psi_gamma')(t)

 Photon Q-bit loop in negative Collatz domain
photon_qbit = Eq(Psi_gamma, -1 - Mod(t, 2))
print(photon_qbit)

 Recursive transition operator: sample logic for snowball trigger
def recursive_transition(t, threshold):
   if t < threshold:
        Mirror loop oscillation
       return -1 - (t % 2)
   else:
        Emergent 3-cycle fold (simplified)
       cycle = [1, 4, 2]
       return cycle[(t - threshold) % 3]

 Example: simulate for 15 steps with trigger at t=6
for step in range(15):
   state = recursive_transition(step, 6)
   print(f"t={step}: State={state}")
```