Mesh Field Theory – Lecture 08:
Simon's Problem as Causal Symmetry Detection

From First Principles: Redundancy in the Coherence Field

1. Introduction

Simon's problem asks: Given a function $f : \{0,1\}^n \to \{0,1\}^n$ with a hidden bitstring s, such that $f(x) = f(x \oplus s)$, can we recover s using fewer queries than classically allowed?

In Mesh, this redundancy is not encoded in a function. It is embedded in the coherence field itself — via real spatial repetition of field structure and shared twist states.

—

2. Physical Encoding of Redundancy

Let the Mesh coherence field $\vec{C}_f(x,t)$ be structured such that:

$$\vec{C}_f(x,t) = \vec{C}_f(x \oplus s, t)$$

That is, the same coherence structure exists at input x and $x \oplus s$. This is a **physical field-level symmetry**, not a symbolic property.

This redundancy arises from shared field evolution — e.g. phase locking, identical twist, or causal linkage.

—

3. Input Register Initialization

Initialize 2^n Mesh qubits Q_x at spatial positions $x \in \{0,1\}^n$, each with identical phase and coherence:

$$\vec{C}_x(x,t) = \vec{C}_0 \quad \text{for all } x$$

This replaces the symbolic uniform superposition of quantum computation.

—

4. Mesh Oracle: Coherence Linking

The Mesh oracle exposes the input coherence field to the symmetry structure:

$$\vec{C}_x(x,t) \Rightarrow \vec{C}_f(x,t) \quad \text{where} \quad \vec{C}_f(x) = \vec{C}_f(x \oplus s)$$

This oracle action is not a function call — it is a **real twist or phase field alignment** applied to linked inputs.

—

5. Interference via Overlap and Collapse

Once field structure is applied, interference begins between matched and unmatched regions.

Collapse occurs when divergence builds:

$$\Gamma(x,t) = \nabla \cdot \vec{C}_f(x,t) > \Gamma_{\text{crit}}$$

Each collapse region x_k encodes a **bitstring z_k** such that:

$$z_k \cdot s = 0 \mod 2$$

Because each collapse arises from a distinct alignment of linked pairs, the output forms a linear constraint on s.

—

6. Statistical Recovery of s

Repeat Mesh collapse across varying initial conditions:

- Slight changes in $\phi_0(x)$, coherence boundaries, or timing - Cause collapse at different symmetry-aligned regions - Yield distinct bitstrings $z_1, z_2, \ldots, z_{n-1}$

Collecting $n-1$ linearly independent such z_i, we solve:

$$z_i \cdot s = 0 \mod 2 \quad \text{to recover } s$$

This is the same solution method as in standard Simon's algorithm — but the source of symmetry is real.

—

7. Worked Example: $n = 3$, $s = 101$

Oracle structure:

$$\vec{C}_{000} = \vec{C}_{101}, \quad \vec{C}_{001} = \vec{C}_{100}, \quad \vec{C}_{010} = \vec{C}_{111}, \quad \vec{C}_{011} = \vec{C}_{110}$$

Mesh collapse occurs at regions aligned to:

$$z_1 = 011, \quad z_2 = 110 \Rightarrow \begin{cases} 011 \cdot 101 = 0 \\ 110 \cdot 101 = 0 \end{cases} \Rightarrow s = 101$$

—

8. Comparison to Quantum Simon Algorithm

— Feature — Quantum — Mesh — —————————————— — Oracle — $f(x) = f(x \oplus s)$ — $\vec{C}_f(x) = \vec{C}_f(x \oplus s)$ — — Initialization — Uniform superposition — Uniform coherence initialization — — Interference — Amplitudes cancel/reinforce — Coherence vectors cancel/reinforce — — Collapse — Probabilistic measurement — Divergence-triggered causal collapse — — Recovery — Solve $z_i \cdot s = 0$ — Solve $z_i \cdot s = 0$ —

—

9. Summary

Mesh reconstructs Simon's algorithm causally:

- Redundancy is built into the field — not abstracted as a function - Collapse generates constraints deterministically - Ensemble variation yields multiple solutions - Symmetry is not symbolic — it is geometric

This is not a simulation of logic. This is logic formed from field structure.

Next: Mesh modularity extended to Shor's algorithm.