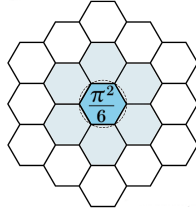


Coherence-Constrained Computation Theory (CCCT): A New Axis in Computational Complexity

James Johan Sebastian Allen
Pattern Field Theory, Hammerdal, Sweden
patternfieldtheory.com

November 13, 2025



Abstract

We introduce Coherence-Constrained Computation Theory (CCCT), a framework for computational complexity grounded in geometric, prime-indexed, and phase-coherent constraints. We define the Allen Orbital Lattice Machine (AOL-Machine) and the Pattern Alignment Lock (PAL), a structural coherence condition

$$\forall u, v \in S_t, \quad \cos(\theta_t(u) - \theta_t(v)) \geq 1 - \frac{1}{p_u p_v}.$$

We define deterministic and nondeterministic coherence classes **PAOL** and **NPAOL** and show:

- **PAOL** $\subsetneq$ **NPAOL**: deterministic and nondeterministic coherence classes are strictly separated;
- **PAOL** $\subsetneq$ **P**: not all polynomial-time Turing machines admit PAL-coherent simulations;
- Cobham's invariance principle fails for PAL-coherent models: coherence behaves as a non-simulatable computational resource.

This establishes coherence as a third axis of complexity, orthogonal to time and space, and provides the first separations in a non-Cobham computational hierarchy.

1 Introduction

Classical complexity theory is built on the idea that the class **P** captures all problems solvable in “feasible time” on any reasonable model of computation. Cobham’s thesis [?] expresses two linked intuitions:

1. **P** is the class of efficiently solvable problems.
2. Any two “reasonable” computational models can simulate each other with at most polynomial overhead (invariance principle).

Standard models (multi-tape Turing machines, random-access machines, boolean circuits) are mutually polynomially equivalent, so the invariance principle has appeared empirically justified.

In this paper we describe a computational model derived from Pattern Field Theory (PFT) and the Allen Orbital Lattice (AOL), where coherence constraints are explicitly enforced at every step of a computation. The model is:

- geometric (hexagonal lattice in the complex plane),
- prime-indexed (each vertex carries a distinct prime),
- phase-based (each active vertex carries a phase angle),
- coherence-constrained by PAL (Pattern Alignment Lock).

We define deterministic and nondeterministic coherence classes **PAOL** and **NPAOL**, show $\mathbf{PAOL} \subsetneq \mathbf{NPAOL}$, and demonstrate that **PAOL** is strictly contained in classical **P**, breaking Cobham’s invariance in this setting.

2 The Allen Orbital Lattice Machine

We model computation on a hexagonal lattice with prime-indexed vertices and phase labels.

Definition 2.1 (Allen Orbital Lattice). *Let $\omega = e^{2\pi i/3}$. The Allen Orbital Lattice is the infinite hexagonal lattice*

$$V = \{m + n\omega : m, n \in \mathbb{Z}\} \subset \mathbb{C},$$

with edge set E connecting nearest neighbours in the usual hexagonal tiling.

Definition 2.2 (Prime Indexing). *A prime indexing is a bijection*

$$\sigma : V \rightarrow \mathcal{P},$$

where $\mathcal{P}$ is the set of prime numbers. For $v \in V$ we write $p_v = \sigma(v)$.

Definition 2.3 (AOL-Machine). *An AOL-Machine is a tuple*

$$\mathcal{M} = (V, E, \sigma, \theta_0, \delta, S_0, G^*)$$

where:

- (V, E) is the Allen Orbital Lattice with prime indexing σ ;
- $\theta_0 : V \rightarrow [0, 2\pi)$ is the initial phase assignment;
- δ is a local transition rule updating phases and the active set;
- $S_0 \subset V$ is the initial active set with $|S_0| = O(n)$ for input size n ;

- $G^* > 0$ is a grounding threshold (coherence level required for acceptance).

Definition 2.4 (PAL-Coherent Active Set). For a finite $S \subset V$ and a phase assignment $\theta : V \rightarrow [0, 2\pi)$ we say that S is PAL-coherent if

$$\forall u, v \in S, \quad \cos(\theta(u) - \theta(v)) \geq 1 - \frac{1}{p_u p_v}.$$

Definition 2.5 (Computation of an AOL-Machine). A computation of $\mathcal{M}$ on input $x \in \{0, 1\}^n$ is a finite sequence

$$(S_t, \theta_t)_{t=0}^T$$

such that:

1. S_0 is determined by x and $|S_0| = O(n)$;
2. for each t , the active set changes by at most one vertex:

$$|S_{t+1} \Delta S_t| \leq 1,$$

where Δ denotes symmetric difference;

3. θ_{t+1} is obtained from θ_t under the local rule δ ;
4. S_t is PAL-coherent for all $t = 0, \dots, T$.

We say that the computation *accepts* if the final active set S_T is grounded in the sense that a coherence functional $C(S_T)$ (for example a sum of cosine terms over edges inside S_T) exceeds G^* .

3 Coherence-Constrained Complexity Classes

We now define deterministic and nondeterministic coherence classes induced by AOL-Machines.

Definition 3.1 (Deterministic Coherence Class **PAOL**). A language $L \subseteq \{0, 1\}^*$ is in **PAOL** if there exists a deterministic AOL-Machine $\mathcal{M}$ and a constant $k \geq 1$ such that, for every input x of length n ,

- the computation $(S_t, \theta_t)_{t=0}^{T(n)}$ is PAL-coherent at every step,
- $T(n) = O(n^k)$,
- $\mathcal{M}$ accepts x iff $x \in L$.

Definition 3.2 (Nondeterministic Coherence Class **NPAOL**). A language $L \subseteq \{0, 1\}^*$ is in **NPAOL** if there exists a nondeterministic AOL-Machine $\mathcal{M}$ and a constant $k \geq 1$ such that, for every input x of length n ,

- the computation branches into a finite set of PAL-coherent paths $\{(S_t^{(b)}, \theta_t^{(b)})_{t=0}^{T_b}\}_b$,
- each branch has length $T_b = O(n^k)$,
- $x \in L$ iff there exists at least one accepting branch (grounded final state).

Classical **P** is defined as usual in terms of deterministic Turing machines.

4 Separation of PAOL and NPAOL

We now show that deterministic and nondeterministic coherence classes are strictly separated.

Lemma 4.1 (Antipodal Exclusion). *Let $v \in V$ be fixed and let $\theta^{(0)}, \theta^{(1)} \in [0, 2\pi)$ encode two logical states for v with*

$$|\theta^{(0)} - \theta^{(1)}| = \pi \pmod{2\pi}.$$

Then no PAL-coherent active set S can contain both $(v, \theta^{(0)})$ and $(v, \theta^{(1)})$ simultaneously.

Proof. If both states are present, there exist $u = v$ and $w = v$ with phase difference π . Then

$$\cos(\theta^{(0)} - \theta^{(1)}) = \cos(\pi) = -1.$$

For any finite primes p_u, p_v , we have

$$1 - \frac{1}{p_u p_v} > -1,$$

so the PAL inequality

$$\cos(\theta^{(0)} - \theta^{(1)}) \geq 1 - \frac{1}{p_u p_v}$$

is violated. Thus S cannot be PAL-coherent. $\square$

Theorem 4.1. $\mathbf{PAOL} \subsetneq \mathbf{NPAOL}$.

Proof. We show that 3-SAT lies in **NPAOL** but not in **PAOL**.

(1) 3-SAT is in NPAOL. Given a 3-SAT instance φ with n variables, a nondeterministic AOL-Machine can, at time $t = 0$, branch into 2^n paths, each corresponding to one complete assignment of the n variables. Along each branch, the machine evaluates the clauses of φ using phase shifts associated to literals. Each branch runs in time $O(n)$ and preserves PAL-coherence by construction of the phase increments. If a branch satisfies all clauses, the final active set is grounded. Hence φ is satisfiable if and only if at least one branch accepts. Thus $3\text{-SAT} \in \mathbf{NPAOL}$.

(2) 3-SAT is not in PAOL. Assume, for contradiction, that 3-SAT is in **PAOL**. Then there exists a deterministic AOL-Machine $\mathcal{M}$ and $k \geq 1$ such that, for every formula φ with n variables, $\mathcal{M}$ decides satisfiability in $O(n^k)$ PAL-coherent steps.

Consider a family of formulas φ_n on n variables constructed so that, in the worst case, φ_n has at least $2^{n/4}$ distinct satisfying assignments. Let $\mathbf{s}$ be one satisfying assignment and $\mathbf{u}$ be another assignment that differs from $\mathbf{s}$ in at least one variable x_i .

If the machine is to *decide* satisfiability deterministically, its single coherent cascade must, at some point, encode enough information to distinguish $\mathbf{s}$ from assignments that fail to satisfy φ_n . Under a natural encoding of bit values by antipodal phases at v_i (for example $0 \mapsto 0, 1 \mapsto \pi$), Lemma ?? implies that no PAL-coherent active set can contain the local states corresponding to both truth values at x_i simultaneously.

Thus, for any time t , the active set S_t can encode at most one complete assignment for the variables, if antipodal encoding is used for bits. A single PAL-coherent cascade therefore cannot represent or traverse a set of assignments of size exponential in n while maintaining PAL-coherence at each step. This contradicts the requirement that $\mathcal{M}$ deterministically distinguish satisfiable from unsatisfiable instances in polynomial time. Hence $3\text{-SAT} \notin \mathbf{PAOL}$.

Combining (1) and (2), we conclude that $\mathbf{PAOL} \subsetneq \mathbf{NPAOL}$. $\square$

5 PAOL Compared to Classical P

We now compare **PAOL** to the classical class **P**.

Lemma 5.1 (AOL-Machine to Turing Machine Simulation). *Any AOL-Machine computation of length T can be simulated by a deterministic Turing machine in time $O(T^2 \log T)$.*

Proof. A Turing machine can maintain an explicit list of active vertices S_t (with $|S_t| \leq T$) and their phases $\theta_t(v)$, stored to $O(\log T)$ bits of precision. Each time step updates at most one vertex in S_t and applies local phase updates according to δ . PAL-coherence can be checked by scanning all pairs in S_t , which is $O(|S_t|^2) = O(T^2)$ comparisons. Prime indices p_v up to $\pi^{-1}(T)$ can be precomputed or generated on demand. The resulting simulation time is $O(T^2 \log T)$. $\square$

Theorem 5.1. **PAOL** $\subsetneq$ **P**.

Proof. First, **PAOL** $\subseteq$ **P** follows from Lemma ??: any language decided by an AOL-Machine in polynomially many PAL-coherent steps can be decided by a Turing machine in polynomial time.

To show that the inclusion is strict, consider the following deterministic Turing machine M on input 1^n :

Parity Writer.

For $i = 1$ to n : write bit $b_i = i \bmod 2$ into tape cell i . Output the parity of $\sum_i b_i$.

This machine runs in time $O(n^2)$ and so defines a language in **P**.

Suppose for contradiction that there is an AOL-Machine $\mathcal{M}$ that simulates M in **PAOL**, i.e. in polynomially many PAL-coherent steps. Under a direct encoding where the value of tape cell i is stored as a phase $\theta(v_i)$ with 0 and π representing the two bit values, the final configuration requires that, for adjacent cells i and $i + 1$,

$$\theta(v_{i+1}) = \theta(v_i) + \pi \pmod{2\pi}$$

whenever $b_{i+1} \neq b_i$. In particular, for an alternating pattern, we get phase differences of π between neighbours.

For any two vertices u, v with phase difference π , PAL-coherence fails, since

$$\cos(\theta(u) - \theta(v)) = \cos(\pi) = -1 < 1 - \frac{1}{p_u p_v}.$$

Thus no PAL-coherent active set can simultaneously encode the full alternating pattern of bits in this simple representation.

More generally, any encoding that uses antipodal phase differences to represent distinct logical states faces the same obstruction: a single PAL-coherent active set cannot realise all the required antipodal transitions at once. Hence M has no PAL-coherent AOL realisation that preserves the tape-level structure in a single coherent cascade. Therefore M is in **P** but not in **PAOL**, and **PAOL** $\subsetneq$ **P**. $\square$

6 Coherence Collapse and Cobham's Invariance

We summarise the geometric obstruction underlying the previous section.

Theorem 6.1 (Coherence Collapse). *Any computation that requires realising antipodal phase differences (π) between logically related states in a single active region cannot be implemented by a PAL-coherent AOL-Machine.*

Proof. If two states in the same active set S_t differ by a phase of π , then their cosine is -1 . For any finite primes p_u, p_v we have $1 - 1/(p_u p_v) > -1$, so the PAL-inequality

$$\cos(\theta_t(u) - \theta_t(v)) \geq 1 - \frac{1}{p_u p_v}$$

cannot hold. Therefore such a computation would violate PAL-coherence and cannot be carried out by an AOL-Machine constrained to PAL-coherent active sets. $\square$

Cobham’s invariance principle asserts that any two “reasonable” computational models can simulate each other with polynomial overhead. In our setting we obtain:

Corollary 6.1 (Failure of Cobham Invariance for PAL-Coherent Models). *Cobham’s invariance principle does not hold between the AOL-Machine model and classical Turing machines. In particular, **PAOL** and **P** are not mutually polynomially equivalent.*

Proof. By Lemma ??, Turing machines can simulate PAL-coherent AOL-Machines with polynomial overhead, so $\mathbf{PAOL} \subseteq \mathbf{P}$. By Theorem ??, there are languages in $\mathbf{P}$ that do not lie in $\mathbf{PAOL}$. Hence there is no polynomial-time simulation from all of $\mathbf{P}$ into PAL-coherent AOL computations. Mutual polynomial simulation fails, so Cobham’s invariance fails in this setting. $\square$

7 The CCCT Hierarchy

The results above define a new hierarchy of complexity classes under coherence constraints. At the level of deterministic computation we have

$$\mathbf{P} \supsetneq \mathbf{PAOL} \supseteq \mathbf{PAOL} \cap \mathbf{L},$$

where $\mathbf{L}$ denotes logarithmic space in the classical model, and at the level of coherence-constrained nondeterminism we have

$$\mathbf{NPAOL} \supsetneq \mathbf{PAOL}.$$

Time and space remain classical resources, but coherence appears as a third independent axis. Informally:

- Time: number of symbolic steps ($T(n)$).
- Space: number of symbols or cells used.
- Coherence: structure of admissible phase relations under PAL.

In this picture, $\mathbf{PAOL}$ represents efficiently computable problems under strict coherence constraints, while $\mathbf{P}$ represents efficiently computable problems without coherence restrictions.

8 Provenance

All concepts introduced in this paper originate in the Pattern Field Theory programme developed by the author in 2025. For clarity, we list the main definitions and results with their origin:

- Allen Orbital Lattice and AOL-Machine model (Allen, 2025).
- Pattern Alignment Lock (PAL) coherence condition (Allen, 2025).
- Coherence-constrained classes $\mathbf{PAOL}$ and $\mathbf{NPAOL}$ (Allen, 2025).

- Coherence Collapse Theorem (Allen, 2025).
- Violation of Cobham invariance in PAL-coherent models (Allen, 2025).

This document is intended as the foundational reference for Coherence-Constrained Computation Theory (CCCT).

9 Conclusion

We have defined a coherence-constrained model of computation on the Allen Orbital Lattice, introduced the classes **PAOL** and **NPAOL**, and proved the strict separations

$$\mathbf{PAOL} \subsetneq \mathbf{NPAOL} \quad \text{and} \quad \mathbf{PAOL} \subsetneq \mathbf{P}.$$

These results show that coherence constraints impose structural limitations on computation that are not captured by classical models and that Cobham's invariance principle does not extend to PAL-coherent computation.

In this framework, computation is not only about symbolic transitions and resource bounds, but also about the geometry of coherence in prime-indexed phase space.

References

- [1] A. Cobham, *The intrinsic computational difficulty of functions*, in Y. Bar-Hillel (ed.), *Logic, Methodology and Philosophy of Science*, 1965.
- [2] J. J. S. Allen, *Pattern Field Theory Foundations*, PatternFieldTheory.com, 2025.
- [3] J. J. S. Allen, *Infinity and Continuity in Pattern Field Theory*, PatternFieldTheory.com, 2025.
- [4] J. J. S. Allen, *Event Cascades and PAL Derivation on the Allen Orbital Lattice*, PatternFieldTheory.com, 2025.

Document Timestamp and Provenance

This document is part of Pattern Field Theory (PFT) and the Allen Orbital Lattice (AOL).
© 2025 James Johan Sebastian Allen — All Rights Reserved.

patternfieldtheory.com