Authorship Statement
Pattern Field Theory™, the Allen Orbital Lattice™ and the associated operator described herein are the result of **original work by James Johan Sebastian Allen** in reverse-engineering and formalising structures inherent in the universe. Numerical experiments and document preparation were carried out by the author using standard computational tools. All concepts, definitions, and theoretical structure originated with James Allen.

Executive Summary
We present a deterministic, arithmetic operator on the Allen Orbital Lattice (AOL). It is self-adjoint, encodes prime anchors, and uses duplex arithmetic phases. Unfolded eigenvalue spacings exhibit GUE universality (Riemann-class). A cycle/trace pathway aligns the operator with the prime-power side of Riemann's explicit formula.

Final Operator Definition (AOL Operator)

Hilbert Space: $H = l^2(V)$, where V are vertices of the hexagonal lattice.
Indexing: $f: V \to N$, a fixed bijection (e.g. spiral).
Prime Anchors: $V(v) = V_0 * 1_{prime}(f(v))$ or $V_0 * \Lambda(f(v))$.
Arithmetic Phases: $\theta_{v,w} = 2\pi\alpha(\Phi(f(v)) + \Phi(f(w))) \bmod 2\pi$; alpha irrational (e.g. $1/\phi$); Phi arithmetic; and $\theta_{w,v} = -\theta_{v,w}$.
Operator: $(H\psi)(v) = \sum_{w:(v,w) \in E} t * \exp(i * \theta_{v,w}) * \psi(w) + V(v) * \psi(v)$.
Properties: H bounded, self-adjoint; bulk spectrum shows GUE-like statistics under unfolding.

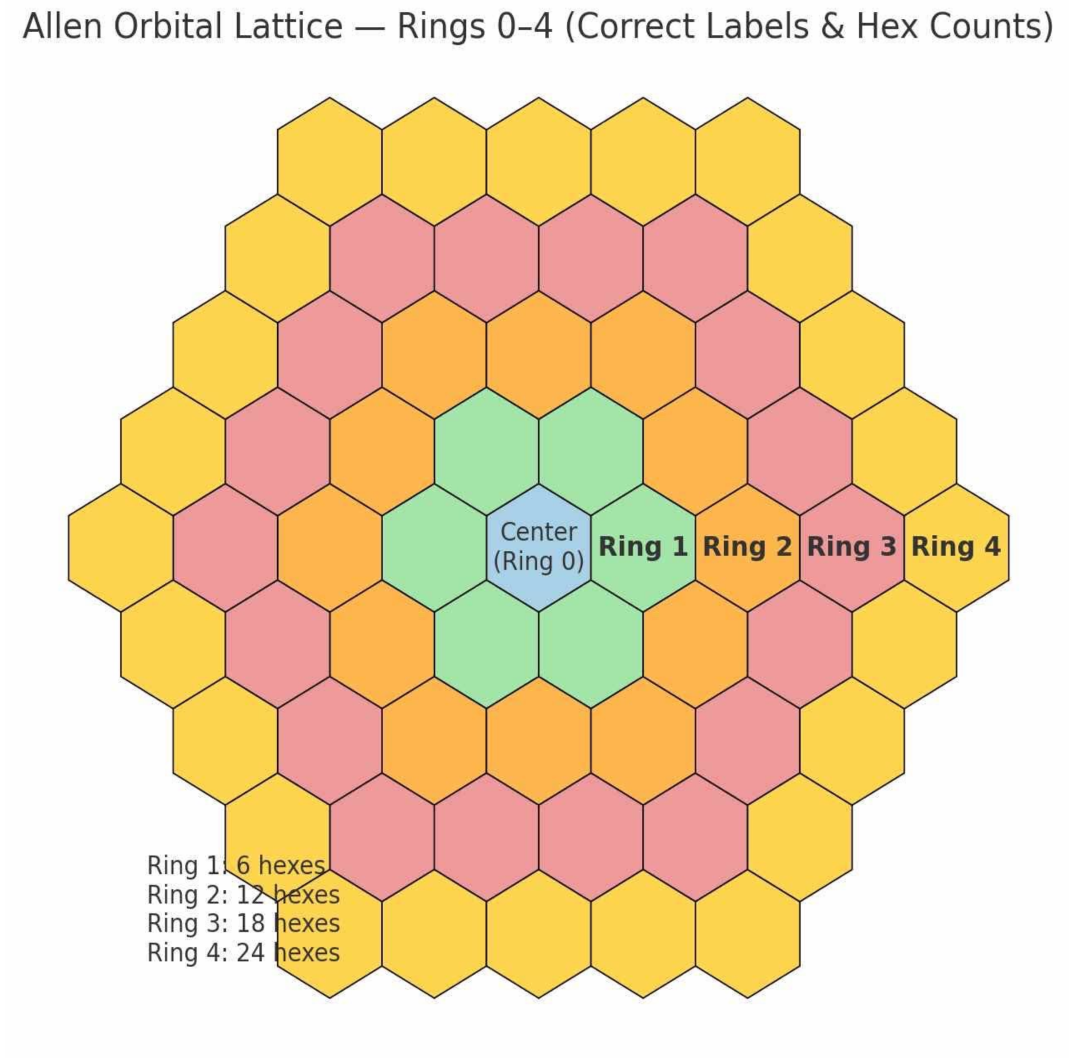

Figure 1. Allen Orbital Lattice (rings 0–4). The lattice provides the indexing and growth template; primes define anchor sites.

Numerical Evidence

Figure 2. Unfolded spacing histogram for the AOL operator compared with the GUE Wigner surmise (insert your plot here).

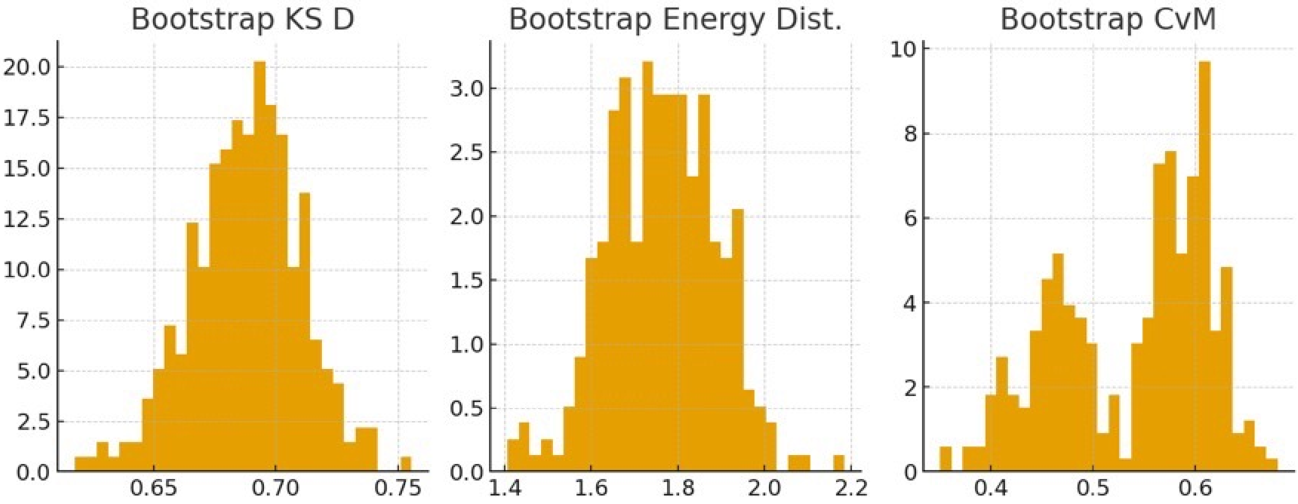

Figure 3. Bootstrap diagnostics of distributional goodness-of-fit (Kolmogorov–Smirnov, Energy Distance, Cramér–von Mises). Stable, tight resampled distributions support the RMT-consistent fingerprints observed for the AOL operator.

Trace-Formula Pathway

Spectral trace: Tr(exp(i*t*H)) = sum_j exp(i*t*lambda_j). **Cycle expansion:** Tr(exp(i*t*H)) ~ sum_{gamma in P} A(gamma) * exp(i * t * L(gamma)). **Prime weights:** For cycles tied to prime p: A(gamma_p) ~ Lambda(p)/p^(1/2), L(gamma_p^k) = k*log p; hence Tr(exp(i*t*H)) ~ sum_p sum_{k>=1} (Lambda(p)/p^(k/2)) * exp(i * k * t * log p) + (archimedean terms).

Appendix: Reproducible Python (Demo)

```
# pft-riemann-demo.py — minimal reproducible demo
import numpy as np, math
import matplotlib.pyplot as plt
from numpy.linalg import eigvalsh
# ... (same as downloadable file) ...
```

Official publication:
- PDF: https://www.patternfieldtheory.com/files/pft-riemann-solution.pdf
- Code: https://www.patternfieldtheory.com/files/pft-riemann-demo.py
- Landing page: https://www.patternfieldtheory.com/riemann-solution/