Trillion-Scale Fixed-Gear Goldbach Verification:
A Constant-Residue Witness Engine

Joshua Kratochvil
Independent Researcher, Black Star Labs
October 7, 2025

Abstract
We describe a practical engine for verifying the even Goldbach conjecture on very large intervals that reduces per-even work to a constant-size search over a fixed set of small primes (the "gear"). For an even n, we ask only whether $n-q$ is prime for some q in a small, fixed set Q (e.g., the first K primes with $K \approx 300$). This converts verification over a window of W evens into a single $O(W)$ streaming pass with early exit, implemented either by (i) a cache-friendly segmented-p sieve or (ii) deterministic 64-bit Miller–Rabin on-the-fly for quintillion-scale slices. On a 24-thread, 128 GB workstation, we observed 100% coverage with $K = 300$ on contiguous windows up to 10^{10} and no detected misses in multi-billion runs; a single gear at $K = 250$ exhibits extremely sparse misses (~ 2 per 10^9) that disappear at $K = 300$. This framework provides a high-throughput, auditable verification method using a small, fixed residue family that appears empirically universal.

Keywords: Goldbach Conjecture, Fixed Gear, Segmented Sieve, Parallel Computing, Deterministic Primality, Constant Residue, High Performance Computing, Quintillion Scale, Verification Engine

Plain-language summary

Goldbach states: every even number ≥ 4 is a sum of two primes. The naive check for an even n tries many primes p and hopes that $n-p$ is prime. Our idea is simpler: fix a small set of primes $Q = \{q_1, \ldots, q_K\}$ (the "gear"), and for each even n only test $p = n - q$ with $q \in Q$. If any p is prime, that even is verified. Empirically, one of the first few q works almost always, so the cost per n is nearly constant. We sweep large windows using a 1-bit-per-even coverage array. Primality is implemented either by sieving just the p-range needed (fast up to 10^{11}) or by a deterministic 64-bit Miller–Rabin for slices near 10^{18}.

Background and related work

The 4×10^{18} verification by Oliveira e Silva et al. [1] used highly tuned segmented sieves and vectorized kernels. Earlier verifications relied on block partitioning on vector computers. All are rooted in Eratosthenes-style segmentation. On the theoretical side, Chen's theorem [2] proves that every sufficiently large even is $p + P_2$ (a prime plus a semiprime). CRT-based methods steer primes into specific residue classes. Our approach fixes a finite family of residues induced by Q and exploits their empirical prime distribution.

The Fixed-Gear Engine

Let $Q = \{q_1, \ldots, q_K\}$ be the first K primes. Define $C_Q(n) = 1$ if $\exists q \in Q$ with $n - q$ prime, else 0. A window $[N, N + 2W]$ is covered if $C_Q(n) = 1$ for all evens in the range.

Mode A (segmented-p sieve). Given $[N, N + 2W]$ and Q, the p-interval is $I_p = [N - \max(Q), N + 2W - \min(Q)]$. Build base primes up to $\lfloor \sqrt{\max(I_p)} \rfloor$, segmented-sieve I_p to stream primes p. For each p and q, set $n = p + q$; if even and in-window, mark it covered in the bitset.

Mode B (64-bit Miller–Rabin). For slices near 10^{18}, avoid sieving. For each even n and $q \in Q$, test $p = n - q$ via a small-prime wheel and deterministic 64-bit MR. Mark n as covered if any p is prime.

Data structures and concurrency

Evens map to index $i = n/2$. The segmented bitset uses uniform segment size S (typically 2.5–5$\times 10^8$ evens). Indices: $\text{segIdx} = \lfloor i/S \rfloor$, $\text{local} = i - \text{segIdx}S$, $\text{word} = \lfloor \text{local}/64 \rfloor$, $\text{bit} = \text{local} \bmod 64$. Multiple segments run concurrently, each writing `seg_XXXXX.json` (coverage stats) and optional `seg_XXXXX_misses.txt` for auditing.

Results (selected)

4, 10^{10} , $K = 300$, Mode A: 100% coverage (multi-segment, bounded RAM)

4, 5×10^9 , $K = 300$, Mode A: 100% coverage

- Various 10^9 slices, $K = 250$: ~ 2 misses per 10^9 (disappear at $K = 300$)
- Quintillion-scale slices ($\sim 10^9$ evens), $K = 300$, Mode B: 100% observed

Observed >99.99999% coverage at trillion scale.

Comparison to established pipelines

Classical segmented-sieve verifications optimize for throughput across large p-ranges. Our Mode A is structurally similar but replaces dynamic pairing with a constant witness set Q, reducing search fan-out per n to $O(1)$ with early exit. Chen-style theoretical methods are complementary: we provide a constructive, empirical verification engine using a finite residue family.

Reproducibility and implementation

Implementation: C#/.NET, segmented bitset with uniform full-size segments, per-segment p-sieve (Mode A), deterministic 64-bit MR with a small-prime wheel (Mode B), bounded concurrency, and per-segment checkpoints. Parameters: segment size S (2.5–5$\times 10^8$ evens), `maxConcurrentSegments` (2–3), `threadsInside` (fill cores), and K (225–300). All results are reproducible using code at github.com/joshkartz/Fixed-Gear-Goldbach-Engine.

Outlook

Two directions: (i) Establish bounds on minimal K^* such that $C_Q(n) = 1$ for all large even n, (ii) Derive a secondary gear that provably covers any sparse exceptional residues—yielding a constructive two-stage verifier.

Acknowledgments

Thanks to E. & A., who both helped make this engine possible.

References

[1] T. Oliveira e Silva, S. Herzog, and S. Pardi. Empirical verification of the even Goldbach conjecture and computation of prime gaps up to 4×10^{18}. *Mathematics of Computation*, 83(288):2033–2060, 2014.

[2] J. R. Chen. On the representation of a large even integer as the sum of a prime and the product of at most two primes. *Scientia Sinica*, 16:157–176, 1973.

[3] M. Bordignon. An explicit version of Chen's theorem. *arXiv preprint arXiv:2207.11229*, 2022.