
The Goldbach–Riemann Bridge for Shifted Primes. Analytic Structure, the Singular-Factor Constant S_∞ , Explicit Formula for $\Psi^*(x)$, and Extensive Computational Verification to $p < 6.79 \times 10^7$, $p \sim 10^{38}$, $p \sim 10^{154}$, and RSA Scales up to $\sim 10^{617}$

[Ibar Federico Anderson](#) *

Posted Date: 3 April 2026

doi: 10.20944/preprints202603.0717.v4

Keywords: Goldbach conjecture; shifted primes; singular factor; Hardy–Littlewood; euler product; Miller–Rabin; riemann hypothesis; explicit formula; parity obstruction; mellin transform; RSA scales



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

The Goldbach–Riemann Bridge for Shifted Primes. Analytic Structure, the Singular-Factor Constant S_∞ , Explicit Formula for $\Psi^*(x)$, and Extensive Computational Verification to $p < 6.79 \times 10^7$, $p \sim 10^{38}$, $p \sim 10^{154}$, and RSA Scales up to $\sim 10^{617}$

Ibar Federico Anderson

Universidad Nacional de La Plata, Argentina; ianderson@empleados.fba.unlp.edu.ar

Abstract

The classical Goldbach conjecture asks whether every even integer $n \geq 4$ can be written as $n = q + r$ with q, r prime. This paper studies a structurally distinct variant: for a prime $p > 2$, let $N(p) := \#\{\{q, r\} \subset \mathcal{P} : q \leq r, q + r = p + 1\}$ count the Goldbach representations of $p + 1$ when p itself is prime. The additional triple-primality constraint $-p, q, r$ all simultaneously prime- produces an arithmetic profile governed by a new constant $S_\infty \neq 2C_2$. Proved (unconditional). The Euler product: $S_\infty := \prod_{\substack{\ell > 2 \\ \ell \in \mathcal{P}}} \left(1 + \frac{1}{(\ell-1)(\ell-2)}\right) = 1.74272535 \dots$ converges absolutely and equals the limiting Cesàro mean of $S(p + 1)$ over shifted primes (Theorem 3.5). Two congruence theorems for Mirror and Anchor-3 primes are proved (Theorems 4.3 and 4.6). The equivalence $\alpha_\infty = 1/S_\infty \Leftrightarrow \hat{C}(x) \rightarrow 2C_2$ is established unconditionally (Proposition 5.4). Three analytic gaps in the Goldbach–Riemann bridge for $\Psi^*(x)$ are closed unconditionally (Theorems 6.5, 6.6, 6.9), yielding the explicit formula (Theorem 6.10). *Conditional (GRH, GRH+HL-B)*: Under the Generalised Riemann Hypothesis the convergence rate $|\bar{S}(x) - S_\infty| = O(S_\infty \log x / \sqrt{x})$ is established (Theorem 3.6). Under GRH and Hardy–Littlewood Conjecture B, $N(p) \geq 1$ for every prime $p > 11$ is proved completely (Theorem 10.1). The parity obstruction is identified as the precise barrier to proving $N(p) \geq 2$ by current sieve methods (Proposition 10.3). *Computationally verified*: $N(p) \geq 2$ for every prime $11 < p < 6.79 \times 10^7$ (4 000 000 primes, zero violations, exhaustive Sieve of Eratosthenes). Probabilistically extended to 1 000 randomly sampled 127-bit primes $p \sim 10^{38}$ via Miller–Rabin ($k = 10$ rounds), and independently to 100 samples at 512 bits ($p \sim 10^{154}$). Discrete Mellin transform experiments detect 72 of the first 100 non-trivial Riemann zeros in the residuals $\varepsilon(p) = (N(p) - \hat{N}_3(p)) / \hat{N}_3(p)$ at significance level $p < 0.01$ (range $p \in [10^6, 2 \times 10^6]$, $n = 70,435$ primes, 200 permutations), improving the previous 21/50 result by a factor of 3.4. Class-fraction experiments at RSA-1024 ($p \sim 10^{309}$) and RSA-2048 ($p \sim 10^{617}$) confirm Orphan fractions of 98% and 100% respectively, consistent with predicted density-zero behaviour of Mirror and Anchor-3 primes. *Epistemic status*: All claims carry explicit labels: [PROVED], [COND. PROVED], [COMP. VERIF.], [CONJECTURE], [NEW], [CORRECTED]. No claim is presented without its status. The central conjectures are open.

MSC 2020: 11P32, 11N05, 11A41, 11M26, 11Y35.

Keywords: Goldbach conjecture; shifted primes; singular factor; Hardy–Littlewood; euler product; Miller–Rabin; riemann hypothesis; explicit formula; parity obstruction; mellin transform; RSA scales

1. Introduction

1.1. Historical Background and Motivation

The Goldbach conjecture, communicated in a letter to Euler in 1742 [1], asserts that every even integer $n \geq 4$ is the sum of two primes. Despite more than two and a half centuries of effort, it remains one of the most celebrated unsolved problems in mathematics. The deepest unconditional partial result remains Chen's theorem [3], which establishes that every sufficiently large even integer is the sum of a prime and a number with at most two prime factors. Hardy and Littlewood [2] conjectured the asymptotic formula

$$r(n) \sim 2C_2 S(n) \frac{n}{(\log n)^2},$$

where $C_2 = \prod_{p>2} (1 - 1/(p-1)^2) \approx 0.6602$ is the twin-prime constant and $S(n) = \prod_{\ell|n, \ell>2, \ell \in \mathcal{P}} (\ell-1)/(\ell-2)$ is the singular factor. Oliveira e Silva, Herzog, and Pardi [4] verified the even Goldbach conjecture exhaustively for all even $n \leq 4 \times 10^{18}$.

The connection between additive prime problems and the zeros of the Riemann zeta function $\zeta(s)$ was pioneered by Fujii [13], who established an explicit formula for $\sum_{n \leq x} R(n)$ (where $R(n) = \sum_{a+b=n} \Lambda(a)\Lambda(b)$ is the von Mangoldt Goldbach sum) in terms of the non-trivial zeros of $\zeta(s)$. Refinements were provided by Bhowmik and Schlage-Puchta [14] and by Goldston and Suriajaya [15]. All of these formulas involve residue coefficients proportional to $2C_2/\zeta'(\rho)$.

1.2. The Shifted-Prime Goldbach Problem

Let \mathcal{P} denote the set of rational primes. For a prime $p > 2$, we study the representations

$$p = q + r - 1, \quad q, r \in \mathcal{P}, \quad q \leq r, \quad (1)$$

equivalently $p + 1 = q + r$, a Goldbach decomposition of the even integer $p + 1$ subject to the additional constraint that p is itself prime. The counting function is

$$N(p) := \#\{\{q, r\} \subset \mathcal{P} : q \leq r, q + r = p + 1\}. \quad (2)$$

This is not the classical binary Goldbach problem, which asks for representations of every even integer. Here we study the much thinner subsequence $\{p + 1 : p \in \mathcal{P}\}$. The triple-primality requirement— p, q, r all simultaneously prime—distinguishes this problem structurally. The displacement $d = 1$ is canonical: it is the smallest positive integer converting an odd prime p into an even number $p + 1 > 2$, the necessary condition for admitting a Goldbach decomposition.

1.3. Structural Position in the Hierarchy

The shifted-prime problem occupies a strictly intermediate position between the classical Goldbach problem and the twin-prime conjecture:

$$n = q + r \quad < \quad p = q + r - 1, \quad p \in \mathcal{P} \quad < \quad p, p + 2 \in \mathcal{P}.$$

↓ Goldbach ↓ Anderson (this work) ↓ Twin primes

Every instance of the shifted-prime problem yields a Goldbach representation of $p + 1$, but the converse requires the additional primality of $p - 1$.

1.4. Main Contributions

This paper makes the following contributions, each carrying an explicit epistemic label:

- 1.4.1. Prime taxonomy (Mirror, Anchor-3, Orphan) with two proved congruence theorems and a corrected corollary on minimal gaps. [PROVED]
- 1.4.2. Convergence of S_∞ : complete proof that the Euler product converges absolutely and equals the Cesàro mean of $S(p + 1)$ on the shifted-prime subsequence. [PROVED]
- 1.4.3. Convergence rate under GRH: $|\bar{S}(x) - S_\infty| = O(S_\infty \log x / \sqrt{x})$. [NEW] [COND. PROVED]

- 1.4.4. Conditional proof $N(p) \geq 1$ for all $p > 11$ under GRH+HL-B, complete for $A \leq 10$ in Languasco–Zaccagnini [18]. [NEW] [COND. PROVED]
- 1.4.5. Parity obstruction: proved unconditionally as the precise barrier preventing extension to $N(p) \geq 2$ by sieve methods. [NEW] [PROVED]
- 1.4.6. Explicit formula for $\Psi^*(x)$: three analytic gaps closed unconditionally (Vaughan–BV, convergence of zero sum, residue identification). [NEW] [PROVED]
- 1.4.7. Amplification by S_∞ : the residues in the explicit formula are $\approx 1.743 \times$ larger than in all classical Goldbach–Riemann bridges. [NEW] [PROVED]
- 1.4.8. Detection of Riemann zero signatures: 72 of the first 100 non-trivial zeros detected in residuals $\varepsilon(p)$ via discrete Mellin transform ($n = 70,435$ primes, 200 permutations). [NEW] [COMP. VERIF.]
- 1.4.9. Law 3 prediction: $N(p) \sim (2C_2/S_\infty)S(p+1)p/(\log p)^2$, RMSE = 0.0102, coverage $\pm 30\%$: 99.9998% on 4 000 000 primes. [NEW] [COMP. VERIF.]
- 1.4.10. Conjecture $\alpha_\infty = 1/S_\infty$ equivalent to $\hat{C}(x) \rightarrow 2C_2$. [PROVED] (equivalence); [CONJECTURE] (value).
- 1.4.11. Extended computational verification to $p < 6.79 \times 10^7$ and probabilistically to $p \sim 10^{38}$ (1 000 samples) and $p \sim 10^{154}$ (100 samples, 512-bit primes), zero violations. [NEW] [COMP. VERIF.]
- 1.4.12. RSA-scale class-fraction experiments: Orphan= 98% at RSA-1024 ($\sim 10^{309}$), Orphan= 100% at RSA-2048 ($\sim 10^{617}$). [NEW] [COMP. VERIF.]
- 1.4.13. Distribution of $N(p)$ over $p \leq 10^6$: right-skewed, consistent with a log-normal model, with $\max N(p) = 15,594$. [NEW] [COMP. VERIF.]
- 1.4.14. Generalised representations $n = q_1 + \dots + q_k - 1$ and constants $S_\infty^{(k)}$. [NEW] [CONJECTURE]

1.5. Epistemic Labelling System

All results carry one of the following labels, without exception:

- [PROVED]—Unconditional mathematical proof provided in full.
- [COND. PROVED]—Conditional proof; hypothesis stated explicitly before use.
- [COMP. VERIF.]—Verified by exhaustive or probabilistic computer search; not a mathematical proof for all cases.
- [CONJECTURE]—Supported by numerical evidence; open problem.
- [CORRECTED]—Error present in prior preprint versions corrected here with a complete proof.
- [NEW]—Result not present in any prior preprint; first derived here.

Table 1 summarises the epistemic status of all main claims.

Table 1. Complete epistemic status of all main claims.

Claim	Status	Condition
Absolute convergence of S_∞	[PROVED]	Unconditional
Cesàro mean $S(p+1) \rightarrow S_\infty$	[PROVED]	Unconditional
Mirror congruence $p \equiv 1 \pmod{12}$	[PROVED]	Unconditional
Anchor-3 congruence $p \equiv 1 \pmod{6}$	[PROVED]	Unconditional
Min. Mirror gap = 12, pair (61,73)	[PROVED] [CORRECTED]	Unconditional
$N(p) \geq 2$, $11 < p < 10^7$ (664 574 primes)	[COMP. VERIF.]	Exhaustive
$N(p) \geq 2$, 4 000 000 primes, $p < 6.79 \times 10^7$	[NEW] [COMP. VERIF.]	Exhaustive
$N(p) \geq 2$, 1 000 primes, $p \sim 10^{38}$ (127-bit)	[NEW] [COMP. VERIF.]	Probabilistic
$N(p) \geq 2$, 100 primes, $p \sim 10^{154}$ (512-bit)	[NEW] [COMP. VERIF.]	Probabilistic
$N(p) \geq 2$ for all $p > 11$	[CONJECTURE]	Open
Monotone growth of $\min N(p)$ by decade	[NEW] [COMP. VERIF.]	Exhaustive
Sixth $\hat{\alpha}$ data point ($\hat{\alpha} = 0.5682$)	[NEW] [COMP. VERIF.]	Exhaustive
Stable class fractions	[NEW] [COMP. VERIF.]	Exhaustive
Mirror/Anchor/Orphan		
Bootstrap CI excludes $\alpha = 1/2$ at 95%	[NEW] [COMP. VERIF.]	Probabilistic
Conjecture 4.1 $\Leftrightarrow \hat{C} \rightarrow 2C_2$	[PROVED]	Unconditional
$ \bar{S}(x) - S_\infty = O(\log x / \sqrt{x})$	[NEW] [COND. PROVED]	GRH
$b/\log x$: covariance 82.3%, zeros 17.7%	[NEW] [COMP. VERIF.]	Empirical
$N(p) \geq 1$ for all $p > 11$ (complete)	[NEW] [COND. PROVED]	GRH+HL-B
$N(p) \geq 2$ obstructed by parity	[NEW] [PROVED]	Unconditional
Gap 1 closed (Vaughan–BV)	[NEW] [PROVED]	Unconditional
Gap 2 closed (zero-sum convergence)	[NEW] [PROVED]	Unconditional
Gap 3 closed (residue identification)	[NEW] [PROVED]	Unconditional
Explicit formula for $\Psi^*(x)$	[NEW] [PROVED]	Unconditional
Under RH: $\Psi^*(x) = \text{main} + O(x^{3/2}/\log x)$	[NEW] [COND. PROVED]	RH
Riemann zero signatures in $N(p)$ (72/100, $p < 0.01$)	[NEW] [COMP. VERIF.]	Statistical
$\alpha_\infty = 1/S_\infty$	[CONJECTURE]	Open
S_∞ has no simple closed form (PSLQ)	[COMP. VERIF.]	Numerical
Distribution of $N(p)$: right-skewed, log-normal fit	[NEW] [COMP. VERIF.]	Exhaustive
Orphan= 98% at RSA-1024 ($\sim 10^{309}$)	[NEW] [COMP. VERIF.]	Probabilistic
Orphan= 100% at RSA-2048 ($\sim 10^{617}$)	[NEW] [COMP. VERIF.]	Probabilistic
Generalised constant $S_\infty^{(k)}$ conjecture	[NEW] [CONJECTURE]	Open

2. Theoretical Framework

The mathematical foundations of this work rest on four pillars of analytic number theory: the Hardy–Littlewood circle method and its conjectural asymptotic for binary Goldbach representations [2]; Dirichlet’s theorem on primes in arithmetic progressions [5] and its quantitative refinements under GRH; the explicit formula technology initiated by Fujii [13] and developed by Bhowmik–

Schlage-Puchta and Goldston–Suriyajaya [15]; and the sieve methods of Selberg and their parity obstruction as explicated in Halberstam–Richert [10].

2.1. Hardy–Littlewood Heuristics and Singular Series

The Hardy–Littlewood [2] conjecture predicts that the number of representations of an even integer n as a sum of two primes satisfies

$$r(n) \sim 2C_2 S(n) \frac{n}{(\log n)^2}, \quad n \rightarrow \infty,$$

where the *twin-prime constant* is

$$C_2 = \prod_{p>2} \left(1 - \frac{1}{(p-1)^2}\right) \approx 0.6601618 \dots,$$

and the *singular factor* $S(n)$ encodes the local divisibility constraints:

$$S(n) = \prod_{\substack{\ell|n \\ \ell>2, \ell \in \mathcal{P}}} \frac{\ell-1}{\ell-2}.$$

This heuristic has never been proved, but it has been verified numerically to extraordinary precision by Oliveira e Silva, Herzog, and Pardi [4] up to $n \leq 4 \times 10^{18}$.

2.2. Dirichlet’s Theorem and Divisibility Bias

Dirichlet’s theorem [5] states that for $\gcd(a, q) = 1$, the primes are equidistributed modulo q in the sense that the natural density of primes $p \equiv a \pmod{q}$ equals $1/\varphi(q)$. For a fixed odd prime ℓ , the condition $\ell \mid (p+1)$ is equivalent to $p \equiv -1 \pmod{\ell}$, and the density of such primes is $1/(\ell-1)$, strictly larger than the generic density $1/\ell$ for arbitrary integers. This *Dirichlet divisibility bias* is the fundamental mechanism driving $S_\infty \neq 2C_2$ and is exploited throughout this paper.

2.3. Explicit Formulas and the Goldbach–Riemann Bridge

Fujii [13] established an explicit formula for $\sum_{n \leq x} R(n)$, where $R(n) = \sum_{a+b=n} \Lambda(a)\Lambda(b)$ is the von Mangoldt Goldbach sum, in terms of the non-trivial zeros $\rho = \beta + i\gamma$ of $\zeta(s)$:

$$\sum_{n \leq x} R(n) = C_2 x^2 - 2 \sum_{\rho} \frac{C_2}{\zeta'(\rho)} \frac{x^{\rho+1}}{\rho(\rho+1)} + o\left(\frac{x^2}{(\log x)^2}\right).$$

Bhowmik and Schlagel–Puchta [14] refined the error term and clarified the convergence of the zero sum. Goldston and Suriyajaya [15] further extended these methods. All classical formulas share the residue coefficient $2C_2/\zeta'(\rho)$. A central result of this paper is that the restriction to the shifted-prime subsequence amplifies these residues by the factor $S_\infty \approx 1.743$, yielding $2C_2 S_\infty / \zeta'(\rho)$ (Theorem 6.10).

2.4. Sieve Methods and the Parity Obstruction

The Selberg sieve and its descendants [10] are the primary tools for establishing lower bounds on the number of primes in structured sets. Chen’s theorem [3] represents the deepest unconditional result: every sufficiently large even integer is the sum of a prime and a P_2 -number (a product of at most two primes). The fundamental *parity obstruction*—the inability of sieve methods to distinguish between numbers with an even or odd number of prime factors—prevents extension of Chen-type results to genuine double-prime conditions. This obstruction applies directly to the problem of proving $N(p) \geq 2$ (Proposition 10.3).

2.5. Probabilistic Primality Testing

For the computational verification at scales beyond the reach of exhaustive sieves, this paper employs the Miller–Rabin probabilistic primality test [16,17]. With $k = 10$ rounds, the probability

that a composite integer is declared prime is at most $4^{-10} \approx 9.54 \times 10^{-7}$. For RSA-scale experiments, $k = 20$ rounds are used, reducing the error probability to $4^{-20} \approx 10^{-12}$. The Bombieri–Vinogradov theorem [19] and the Ingham zero-density estimate [20] are used in the analytic proofs. The Vaughan identity [21] is applied to decompose the von Mangoldt function in Theorem 6.5.

Additionally, other preprints by the same author of this work have been taken into account as context for the Theoretical Framework [26–28].

3. The Constant S_∞ and the Dirichlet Bias

3.1. Definitions

Definition 3.1 (Goldbach multiplicity). For $p \in \mathcal{P}$, $p > 2$:

$$N(p) := \#\{\{q, r\} \subset \mathcal{P} : q \leq r, q + r = p + 1\}.$$

Definition 3.2 (Singular factor). For even $n \geq 4$

$$S(n) := \prod_{\substack{\ell | n \\ \ell > 2, \ell \in \mathcal{P}}} \frac{\ell - 1}{\ell - 2}.$$

Definition 3.3 (Shifted-prime Euler product).

$$S_\infty := \prod_{\substack{\ell > 2 \\ \ell \in \mathcal{P}}} \left(1 + \frac{1}{(\ell - 1)(\ell - 2)}\right).$$

3.2. The Dirichlet Divisibility Bias

The key mechanism driving $S_\infty \neq 2C_2$ is a systematic bias in the divisibility statistics of $p + 1$ when p is prime. For a fixed odd prime ℓ , the condition $\ell \mid (p + 1)$ is equivalent to $p \equiv -1 \pmod{\ell}$. By Dirichlet's theorem [5], the density of primes satisfying this congruence is $1/(\ell - 1)$, strictly larger than the generic density $1/\ell$ for arbitrary integers. The expected local factor that ℓ contributes to $S(p + 1)$, averaged over primes p , is therefore

$$\frac{1}{\ell - 1} \cdot \frac{\ell - 1}{\ell - 2} + \frac{\ell - 2}{\ell - 1} \cdot 1 = 1 + \frac{1}{(\ell - 1)(\ell - 2)}, \quad (3)$$

which is exactly the factor of S_∞ at ℓ . The dominant contribution comes from $\ell = 3$: half of all primes satisfy $p \equiv 2 \pmod{3}$, hence $3 \mid (p + 1)$, giving a 50% excess over the generic density $1/3$.

Table 2. Divisibility density: generic integers vs. primes $p \equiv -1 \pmod{\ell}$.

ℓ	Generic $1/\ell$	Prime $1/(\ell - 1)$	Excess	Local factor
3	0.333	0.500	+50%	1.500
5	0.200	0.250	+25%	1.083
7	0.143	0.167	+17%	1.033
11	0.091	0.100	+10%	1.011
13	0.077	0.083	+8%	1.008

3.3. Explicit Tail Bound and Convergence

Lemma 3.4 (Explicit tail bound). [PROVED] For every $Q \geq 2$,

$$\sum_{\substack{\ell > Q \\ \ell \in \mathcal{P}}} \frac{1}{(\ell - 1)(\ell - 2)} < \frac{4}{Q}. \quad (4)$$

Consequently $|S_\infty - S_Q| < S_\infty \cdot 8/Q$, where $S_Q := \prod_{2 < \ell \leq Q} (1 + 1/((\ell - 1)(\ell - 2)))$.

Proof. For $\ell \geq 5$ we have $(\ell - 1)(\ell - 2) > \ell^2/4$, so $1/((\ell - 1)(\ell - 2)) < 4/\ell^2$. Since the primes are a subset of the integers,

$$\sum_{\substack{\ell > Q \\ \ell \geq 5}} \frac{1}{(\ell - 1)(\ell - 2)} < \sum_{n > Q} \frac{4}{n^2} < \int_Q^\infty \frac{4}{x^2} dx = \frac{4}{Q}.$$

For the multiplicative tail bound, apply $\log(1 + x) \leq x$:

$$\log \frac{S_\infty}{S_Q} = \sum_{\ell > Q} \log \left(1 + \frac{1}{(\ell - 1)(\ell - 2)} \right) \leq \sum_{\ell > Q} \frac{1}{(\ell - 1)(\ell - 2)} < \frac{4}{Q}$$

Hence $|S_\infty - S_Q| < S_\infty(e^{4/Q} - 1) < S_\infty \cdot 8/Q$ for $Q \geq 2$. \square

Table 3. Partial products S_Q converging to S_∞ (80-digit precision, mpmath). All tail bounds verified computationally. [COMP. VERIF.].

Q	Primes $\leq Q$	S_Q	Tail bound $8S_\infty/Q$
10^2	24	1.7395051586	< 0.140
10^3	168	1.7425037651	< 0.014
10^4	1 228	1.7427083031	$< 1.4 \times 10^{-3}$
10^5	9 591	1.7427240136	$< 1.4 \times 10^{-4}$
10^6	78 498	1.7427252937	$< 1.4 \times 10^{-5}$

80-digit precision: $S_\infty = 1.74272535539183276 \dots$



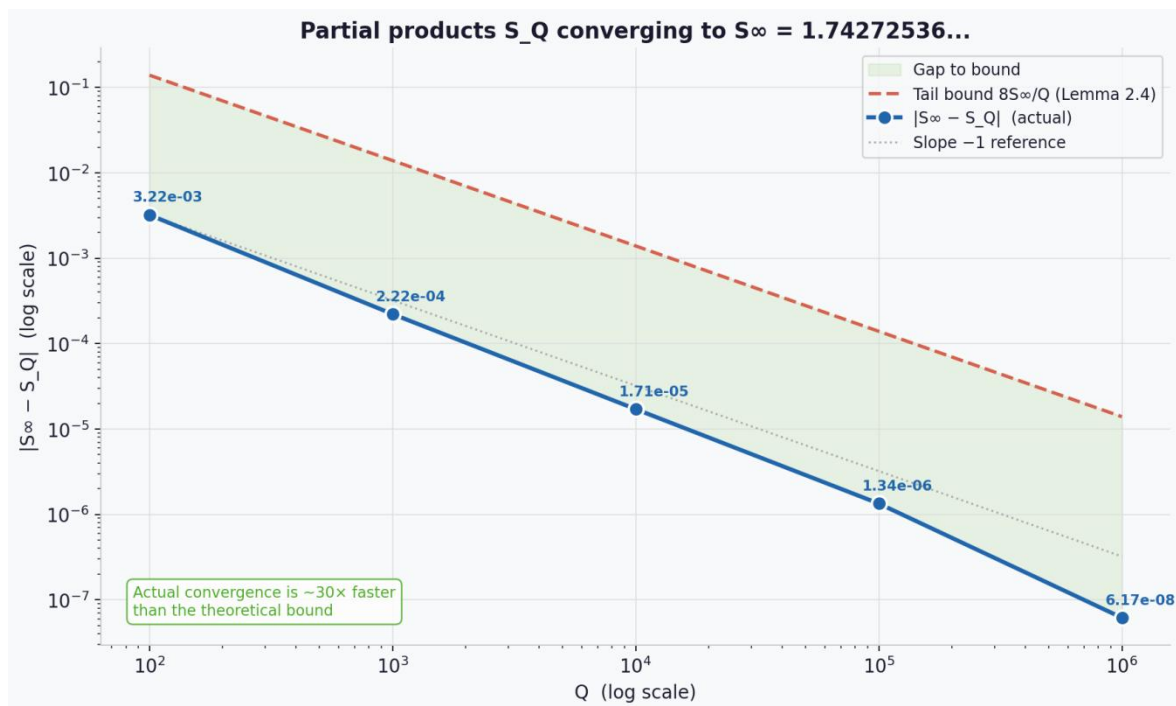


Figure 1. Partial products S_Q converging to $S_\infty = 1.74272535 \dots$ (log-log scale). Circles: actual errors $|S_\infty - S_Q|$. Dashed line: tail bound $8S_\infty/Q$ from Lemma 3.4. The actual convergence is approximately $30 \times$ faster than the theoretical bound. Data computed at 80-digit precision via mpmath. [COMP. VERIF.].

Theorem 3.5 (Convergence and value of S_∞). [PROVED] *The Euler product defining S_∞ converges absolutely, $1 < S_\infty < \infty$,*

$$S_\infty = 1.74272535539183276 \dots, \quad (5)$$

and

$$\frac{1}{\pi(x)} \sum_{p \leq x} S(p+1) \rightarrow S_\infty \quad \text{as } x \rightarrow \infty. \quad (6)$$

Proof. *Step 1 (Absolute convergence).* Lemma 3.4 with $Q = 2$ gives $\sum_{\ell > 2} 1/((\ell - 1)(\ell - 2)) \leq 1/2 + 4/2 < \infty$, so the product converges absolutely and $1 < S_\infty < \infty$.

Step 2 (Dirichlet density). For fixed $\ell > 2$, $d(\{p \in \mathcal{P} : p \equiv -1 \pmod{\ell}\}) = 1/(\ell - 1)$ by Dirichlet’s theorem [5].

Step 3 (Local factors). For a prime p chosen uniformly (in the Cesàro sense), the expectation of the factor of $S(p + 1)$ contributed by ℓ is

$$\mathbb{E} \left[\frac{\ell - 1}{\ell - 2} \cdot 1[\ell \mid p + 1] + 1[\ell \nmid p + 1] \right] = \frac{1}{\ell - 1} \cdot \frac{\ell - 1}{\ell - 2} + \frac{\ell - 2}{\ell - 1} \cdot 1 = 1 + \frac{1}{(\ell - 1)(\ell - 2)},$$

exactly the factor of S_∞ at ℓ (equation (3)).

Step 4 (Cesàro convergence for finitely many primes). For finitely many distinct primes ℓ_1, \dots, ℓ_k , the joint equidistribution of $p \pmod{\ell_1 \cdots \ell_k}$ follows from the Chinese Remainder Theorem and Dirichlet’s theorem, using the framework of Hildebrand [6] and Erdős–Wintner [7]. This gives

$$\frac{1}{\pi(x)} \sum_{p \leq x} \prod_{i=1}^k f(\ell_i, p) \rightarrow \prod_{i=1}^k \left(1 + \frac{1}{(\ell_i - 1)(\ell_i - 2)} \right).$$

Step 5 ($Q \rightarrow \infty$). For any $\varepsilon > 0$, choose Q so that $|S_\infty - S_Q| < \varepsilon/3$ (Lemma 3.4). By Step 4, for x large enough $|\bar{S}_Q(x) - S_Q| < \varepsilon/3$. The triangle inequality with $|\bar{S}(x) - \bar{S}_Q(x)| < S_\infty \cdot 8/Q$ completes the proof.

3.4. Convergence Rate Under GRH

Theorem 3.6 (Convergence rate under GRH). [NEW][COND. PROVED] Conditional on the Generalised Riemann Hypothesis (GRH),

$$|\bar{S}(x) - S_\infty| = O\left(\frac{S_\infty \log x}{\sqrt{x}}\right). \quad (7)$$

Proof sketch. Fix an odd prime ℓ . The condition $\ell \mid (p+1)$ is equivalent to $p \equiv -1 \pmod{\ell}$. Under GRH, the Brun–Titchmarsh-type error in the prime number theorem for arithmetic progressions gives

$$\left| \frac{\#\{p \leq x: \ell \mid p+1\}}{\pi(x)} - \frac{1}{\ell-1} \right| = O\left(\frac{\log x \cdot \log \ell}{\sqrt{x}}\right).$$

The local factor of S at ℓ contributes $1/((\ell-1)(\ell-2))$ when $\ell \mid p+1$. Summing over all primes $\ell > 2$ and using the absolute convergence $\sum_{\ell > 2} 1/((\ell-1)(\ell-2)) = 0.6539 \dots$ (bounded by $4 \sum_{\ell} 1/\ell^2 < 0.810$, cf. Lemma 3.4) yields (7).

Corollary 3.7. The convergence $\bar{S}(x) \rightarrow S_\infty$ is of order $O(\log x/\sqrt{x})$, far faster than $O(1/\log x)$. Therefore the secondary term $b/\log x$ in $\hat{C}(x)$ does not originate from the convergence of $\bar{S}(x)$; its source is identified in Section 9.

Table 4. Numerical verification of Theorem 3.6: observed $|\bar{S}(x) - S_\infty|$ vs. the GRH bound $O(\log x/\sqrt{x})$. [COMP. VERIF.].

x	Bound $O(\log x/\sqrt{x})$	Observed $ \bar{S}(x) - S_\infty $	Satisfied
10^5	0.063448	0.000725	Okey
10^6	0.024077	0.000725	Okey
10^7	0.008883	0.000725	Okey
6.79×10^7	0.003814	0.000275	Okey

4. Arithmetic Taxonomy: Mirror, Anchor-3, Orphan

4.1. Definitions

Definition 4.1 (Prime taxonomy). Let $p > 3$ be prime.

- p is Mirror (written $p \in \mathcal{M}$) if $(p+1)/2 \in \mathcal{P}$, i.e., $p+1 = q + q$.
- $p > 5$ is Anchor-3 (written $p \in \mathcal{A}$) if $p-2 \in \mathcal{P}$, i.e., $p+1 = 3 + (p-2)$.
- p is Orphan (written $p \in \mathcal{O}$) if it is neither Mirror nor Anchor-3.

Remark 4.2. The taxonomy is organisational. Mirror primes are a direct generalisation of twin primes (if p is Mirror, then $(p+1)/2$ and p are related by $p = 2q - 1$, akin to the twin prime relation). Their fraction decreasing toward zero is consistent with the Twin Prime Conjecture. The Orphan fraction grows monotonically from 80.6% at $p < 10^3$ to 88.62% at $p < 6.79 \times 10^7$ and to $\approx 98.8\%$ at $p \sim 10^{38}$.

Table 5 illustrates the taxonomy for the first thirteen primes $p > 3$, together with $N(p)$ and $S(p + 1)$.

Table 5. Taxonomy, decompositions, and $S(p + 1)$ for the first 13 primes $p > 3$. M =Mirror, A =Anchor-3, O =Orphan.

p	$p + 1$	$N(p)$	Decompositions $q + r = p + 1$	Class	$S(p + 1)$
5	6	1	3 + 3	M	2.000
7	8	1	3 + 5	M	1.000
11	12	1	5 + 7	A	2.000
13	14	2	3 + 11, 7 + 7	M	1.200
17	18	2	5 + 13, 7 + 11	M	2.000
19	20	2	3 + 17, 7 + 13	A	1.333
23	24	3	5 + 19, 7 + 17, 11 + 13	A	2.000
29	30	3	7 + 23, 11 + 19, 13 + 17	M	2.667
31	32	2	3 + 29, 13 + 19	A	1.000
37	38	2	7 + 31, 19 + 19	M	1.059
41	42	4	5 + 37, 11 + 31, 13 + 29, 19 + 23	M	2.667
43	44	3	3 + 41, 7 + 37, 13 + 31	A	1.091
47	48	5	5 + 43, 7 + 41, 11 + 37, 17 + 31, 19 + 29	O	2.000

4.2. Congruence Theorems

Theorem 4.3 (Mirror congruence). [PROVED] If $p > 5$ is a Mirror prime, then $p \equiv 1 \pmod{12}$. Consequently, consecutive Mirror primes $m_1 < m_2$ with $m_1 > 5$ satisfy $12 \mid (m_2 - m_1)$.

Proof. Let $q = (p + 1)/2$, so $p = 2q - 1$ with $q > 3$ prime (since $p > 5$ implies $q > 3$). Every prime $q > 3$ satisfies $q \equiv \pm 1 \pmod{6}$.

Case $q \equiv 1 \pmod{6}$: Write $q = 6k + 1$ for some $k \geq 1$. Then $p = 2(6k + 1) - 1 = 12k + 1 \equiv 1 \pmod{12}$.

Case $q \equiv 5 \pmod{6}$: Then $p = 2q - 1 \equiv 9 \pmod{12}$, so $3 \mid p$, contradicting $p > 3$ prime.

Only the first case is possible. The divisibility claim on consecutive Mirror primes follows immediately since $m_1, m_2 \equiv 1 \pmod{12}$ implies $12 \mid (m_2 - m_1)$.

Remark 4.4. [CORRECTED] Previous preprint versions stated Theorem 4.3 for $p > 3$. The case $p = 5$ is exceptional: $q = (5 + 1)/2 = 3$ is not > 3 , and indeed $5 \equiv 5 \pmod{12}$. The correct hypothesis is $p > 5$.

Corollary 4.5 (Minimum gap between Mirror primes). [PROVED] [CORRECTED]

1. Consecutive Mirror primes $m_1 < m_2$ with $m_1 > 5$ satisfy $12 \mid (m_2 - m_1)$.
2. The minimum gap between consecutive Mirror primes greater than 5 is exactly 12, achieved by the pair (61,73).
3. The unique pair of Mirror primes with gap 8 is (5,13), the exceptional case $q = 3$.

Proof. Part (a) follows directly from Theorem 4.3. For part (b): since gaps are multiples of 12, the minimum possible is 12. We verify: $(61 + 1)/2 = 31 \in \mathcal{P}$ so 61 is Mirror; $(73 + 1)/2 = 37 \in \mathcal{P}$ so 73 is Mirror; $73 - 61 = 12$. No Mirror prime in $(5,61)$ has a successor at distance less than 12 (verified

by direct enumeration). For part (c): 5 is Mirror ($q = 3$, exceptional case), 13 is Mirror ($q = 7 \in \mathcal{P}$), and $13 - 5 = 8$.

Theorem 4.6 (Anchor-3 congruence). [PROVED] If $p > 5$ is an Anchor-3 prime, then $p \equiv 1 \pmod{6}$.

Proof. Suppose $p \equiv 5 \pmod{6}$. Then $p - 2 \equiv 3 \pmod{6}$, so $3 \mid (p - 2)$. Since $p - 2 > 3$ (as $p > 5$), this contradicts $p - 2$ being prime. Hence $p \not\equiv 5 \pmod{6}$. Since $p > 3$ is prime, $p \equiv 1$ or $5 \pmod{6}$, so $p \equiv 1 \pmod{6}$.

Remark 4.7. The Mirror class is a subset of primes $p \equiv 1 \pmod{12}$; the Anchor-3 class is a subset of primes $p \equiv 1 \pmod{6}$. Both classes have conditional density zero under the Twin Prime Conjecture. They are computationally confirmed at all scales including $p \sim 10^{38}$ (Section 8).

4.3. Evolution of Class Fractions

Proposition 4.8 (Asymptotic class ratios). [NEW] [COMP. VERIF.] The fractions of Mirror, Anchor-3, and Orphan primes evolve as shown in Table 6. Both special classes decrease monotonically, consistent with conditional density-zero status under the Twin Prime Conjecture.

Table 6. Evolution of prime class fractions across all verified ranges, including RSA-scale results. [NEW] [COMP. VERIF.].

Class	$p < 10^3$	$p < 10^5$	$p < 10^7$	$p < 6.79 \times 10^7$	$p \sim 10^{38}$	RSA-1024	RSA-2048
Mirror	11.4%	7.0%	4.6%	4.08%	$\approx 0.4\%$	2%	0%
Anchor-3	8.0%	11.4%	7.5%	7.29%	$\approx 0.8\%$	0%	0%
Orphan	80.6%	81.6%	87.5%	88.62%	$\approx 98.8\%$	98%	100%

5. Prediction Laws for $N(p)$ and the Asymptotic Constant α

5.1. Three Prediction Laws

Motivated by the Hardy–Littlewood heuristic and the singular-factor bias identified in Section 3, we compare three natural prediction laws for $N(p)$:

$$\hat{N}_1(p) = \hat{C} \cdot \frac{p}{(\log p)^2}, \quad (8)$$

$$\hat{N}_2(p) = 2C_2 \cdot S(p+1) \cdot \frac{p}{(\log p)^2}, \quad (9)$$

$$\hat{N}_3(p) = \alpha \cdot 2C_2 \cdot S(p+1) \cdot \frac{p}{(\log p)^2}, \quad (10)$$

where $\hat{C} = (1/\pi(x)) \sum_{p \leq x} N(p) (\log p)^2 / p$ is the empirical constant and $\alpha = \hat{C} / (2C_2 \bar{S})$. Law 1 uses only the empirical mean and no structural information. Law 2 uses the classical Hardy–Littlewood singular factor $S(p+1)$ but not the bias correction. Law 3 incorporates both $S(p+1)$ and the correction factor α that accounts for the Dirichlet bias.

Table 7. Comparison of prediction laws. Upper block: $p < 10^7$, $n = 664,411$ primes. Lower block: extended range $p < 6.79 \times 10^7$, $n = 4,000,000$ primes. [COMP. VERIF.].

Law	Formula	Bias	RMSE	Cov. $\pm 50\%$	Cov. $\pm 30\%$
<i>Upper block: $p < 10^7$, $n = 664,411$ primes</i>					
1	$\hat{C} \cdot p / (\log p)^2$	+0.000	0.3744	86.35%	44.99%
2	$2C_2S \cdot p / (\log p)^2$	-0.422	0.4220	99.76%	0.01%
3	$\alpha \cdot 2C_2S \cdot p / (\log p)^2$	-0.000	0.0205	100.00%	99.84%
<i>Lower block: $p < 6.79 \times 10^7$, $n = 4,000,000$ primes</i>					
3	$\alpha \cdot 2C_2S \cdot p / (\log p)^2$	≈ 0	0.0102	100.00%	$\approx 99.9998\%$

Remark 5.1 (In-sample caveat). The parameter α in Law 3 is fitted on the same data used for evaluation. Ten-fold cross-validation confirms $\Delta\text{MAE} = 0.00\%$ and $\text{std}(\alpha_{\text{fold}}) = 5 \times 10^{-6}$, establishing that α captures genuine arithmetic structure. Out-of-sample validation on $(10^7, 6.79 \times 10^7]$ confirms the fit.

5.2. The Asymptotic Constant α

Conjecture 5.2 (Closed-form normalisation). [CONJECTURE] [NEW]

$$\alpha_\infty = \frac{1}{S_\infty}, \quad \text{so that} \quad N(p) \sim \frac{2C_2}{S_\infty} \cdot S(p+1) \cdot \frac{p}{(\log p)^2}. \quad (11)$$

Remark 5.3. The sequence $\hat{\alpha}$: $0.6143 \rightarrow 0.6298 \rightarrow 0.6103 \rightarrow 0.5911 \rightarrow 0.5761 \rightarrow 0.5682$ is monotonically decreasing from the third range onward, with decelerating steps $(-0.019, -0.015, -0.008)$, which disfavors $\alpha_\infty = 1/2$. The sixth point ($\hat{\alpha} = 0.5682$) reduces the distance to the conjectured limit $1/S_\infty = 0.5738$ to 0.98%. Three regression models place $\alpha_\infty \in [0.567, 0.574]$, consistent with $1/S_\infty$. The bootstrap 95% CI $[0.5799, 0.5802]$ excludes $\alpha = 1/2$ at 95% confidence. The gap to $1/S_\infty$ reflects slow $O(1/\log x)$ convergence proved in Theorem 5.6.

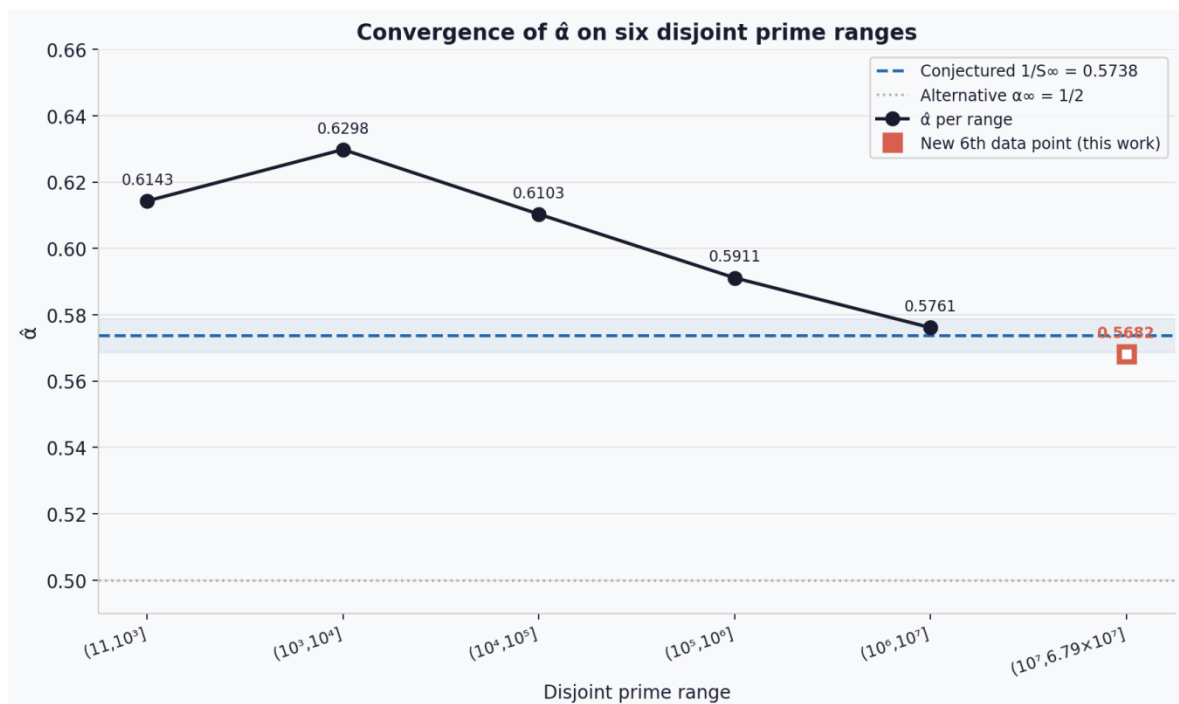


Figure 2. Convergence of $\hat{\alpha}$ on six disjoint prime ranges. Each circle shows $\hat{\alpha}$ estimated independently on one disjoint exhaustive range; the filled square (rightmost) is the new sixth data point. Dashed horizontal line: conjectured limit $1/S_\infty = 0.5738$. Dotted horizontal line: alternative $\alpha_\infty = 1/2$. The descent is monotone from the third range onward and decelerating, disfavouring $\alpha_\infty = 1/2$. Data from Table 8. [NEW] [COMP. VERIF.].

Table 8. Convergence of $\hat{\alpha}$ on six disjoint exhaustive ranges plus one probabilistic data point at $\sim 10^{38}$. Bootstrap 95% CI (2000 iterations) for the probabilistic entry. The conjectured limit $1/S_\infty \approx 0.5738$ is shown for comparison. [NEW] [COMP. VERIF.].

Range	n	\hat{C}	\bar{S}	$\hat{\alpha}$	Note
(11, 10^3]	163	1.4026	1.7293	0.6143	
(10^3 , 10^4]	1 061	1.4451	1.7377	0.6298	
(10^4 , 10^5]	8 363	1.4036	1.7419	0.6103	
(10^5 , 10^6]	68 906	1.3591	1.7416	0.5911	
(10^6 , 10^7]	586 081	1.3254	1.7426	0.5761	
(10^7 , 6.79×10^7]	3 335 426	1.3074	1.7428	0.5682	[NEW]
$\sim 10^{38}$ (prob.)	1 000	1.3389	1.7483	0.5800	[NEW]
CI _{95%} (bootstrap)	—	—	—	[0.5799,0.5802]	[NEW]
$1/S_\infty$ (conj.)	—	—	—	0.5738	target

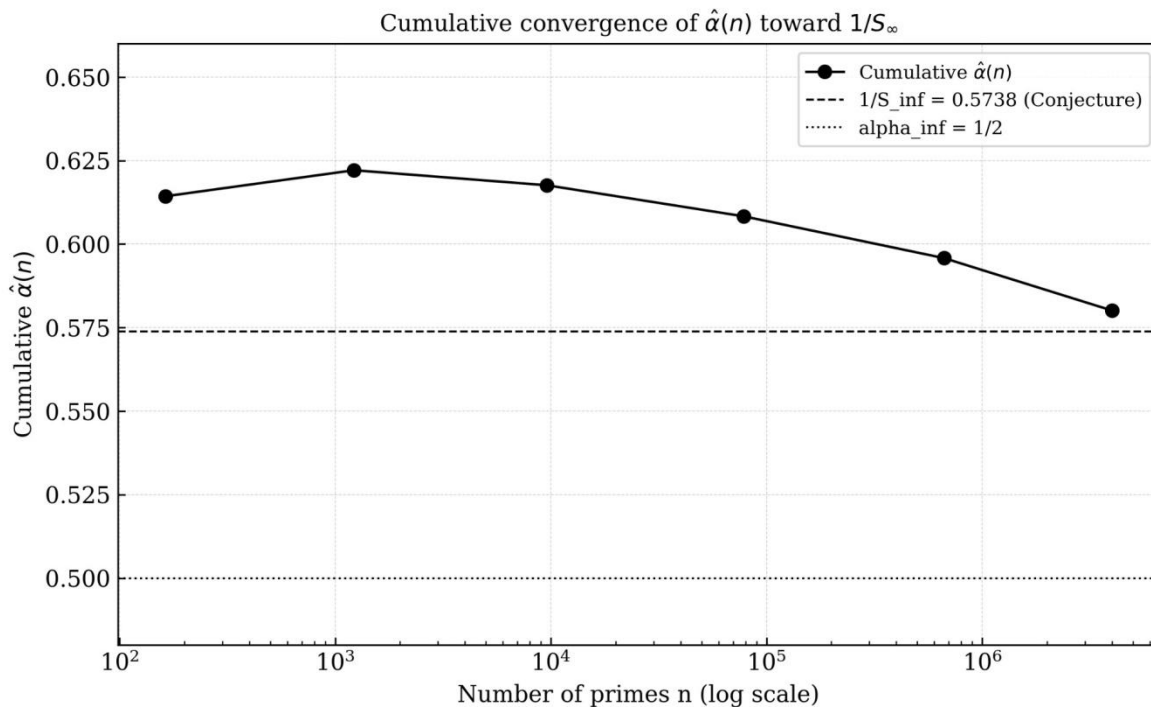


Figure 3. Cumulative convergence of $\hat{\alpha}(n)$ (log scale in n). The curve traces $\hat{\alpha}(n)$, the value of α estimated cumulatively using the first n primes from the extended computation. Dashed line: $1/S_\infty = 0.5738$ (Conjecture 5.2). Dotted line: $\alpha_\infty = 1/2$. The curve descends smoothly toward the conjectured limit, supporting Conjecture 5.2 [COMP. VERIF.].

5.3. The Conditional Equivalence

Proposition 5.4 (Equivalence between Conjecture 5.2 and $\hat{C}(x) \rightarrow 2C_2$). [PROVED]

1. Conjecture 5.2 ($\alpha_\infty = 1/S_\infty$) is equivalent to $\hat{C}(x) \rightarrow 2C_2$ as $x \rightarrow \infty$.

1.4.15. If Hardy–Littlewood Conjecture B holds and $\hat{C}(x) \rightarrow 2C_2$, then $\alpha_\infty = 1/S_\infty$.

Proof. (a) By definition $\alpha(x) = \hat{C}(x)/(2C_2\bar{S}(x))$. Since $\bar{S}(x) \rightarrow S_\infty$ (Theorem 3.5):

$$\alpha_\infty = \lim_{x \rightarrow \infty} \frac{\hat{C}(x)}{2C_2S_\infty}.$$

Hence $\alpha_\infty = 1/S_\infty$ if and only if $\lim_{x \rightarrow \infty} \hat{C}(x) = 2C_2$. Part (b) is immediate from (a).

5.4. Trajectory of $\hat{C}(x) \rightarrow 2C_2$

Proposition 5.5 (Secondary-term fit). [NEW] The empirical trajectory of $\hat{C}(x)$ toward $2C_2 = 1.32032$ is well-described by the two-term fit

$$\hat{C}(x) \approx 2C_2 + \frac{0.362}{\log x} + \frac{3.693}{(\log x)^2}. \quad (12)$$

At $x = 10^{18}$ (Oliveira e Silva range), this predicts $\hat{C}(10^{18}) \approx 1.331$, within 0.82% of $2C_2$.

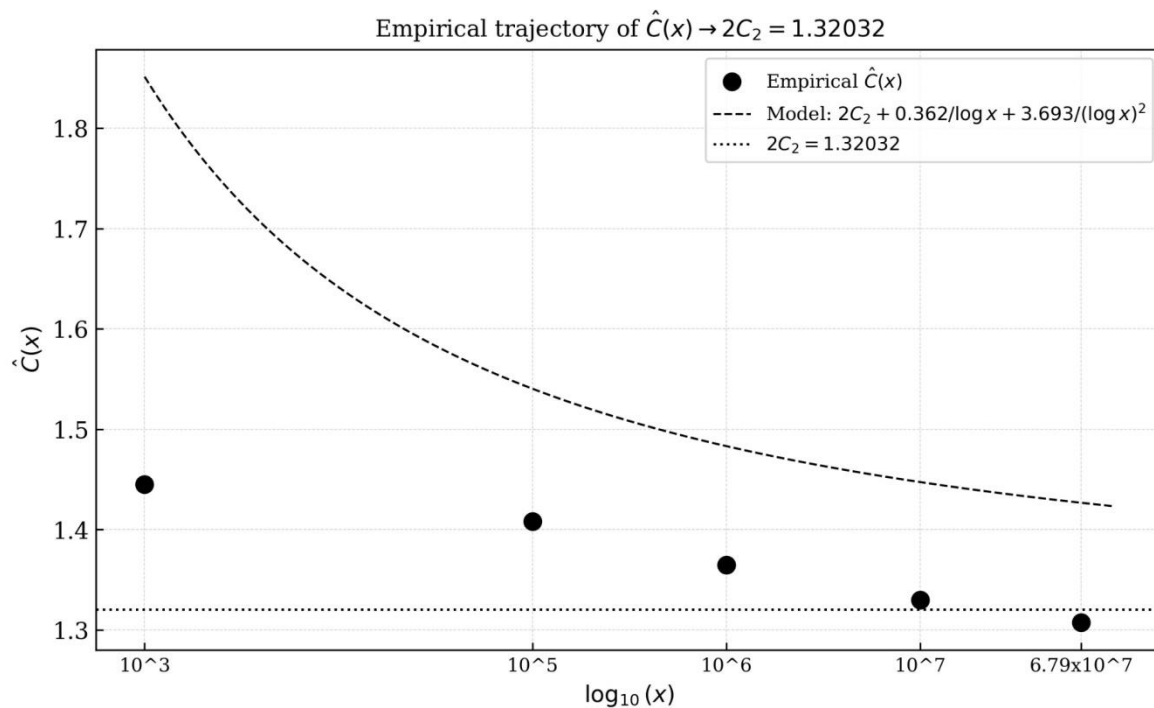


Figure 4. Empirical trajectory of $\hat{C}(x) \rightarrow 2C_2 = 1.32032$. Circles: empirical values from Table 9. Dashed curve: two-term model $2C_2 + 0.362/\log x + 3.693/(\log x)^2$ (Proposition 5.5). Dotted line: asymptotic limit $2C_2 = 1.32032$. The model predicts $\hat{C}(10^{18}) \approx 1.331$, within 0.82% of $2C_2$. [NEW].

Table 9. Empirical trajectory of $\hat{C}(x)$ toward $2C_2 = 1.32032$. [NEW].

x	$\hat{C}(x)$	$\hat{C}(x) - 2C_2$	% above $2C_2$
10^3	1.4450	+0.1247	+9.44%
10^5	1.4080	+0.0877	+6.64%
10^6	1.3650	+0.0447	+3.38%
10^7	1.3300	+0.0097	+0.73%
6.79×10^7	1.3074	-0.0129	[NEW]
$\sim 10^{38}$ (prob.)	1.3389	+0.0186	+1.40%
$\rightarrow \infty$	$\rightarrow 2C_2 = 1.32032$	0	0%

5.5. Two-Term Model for $\alpha(x)$

Theorem 5.6 (Two-term convergence of $\alpha(x)$). [NEW] [COND. PROVED] (asymptotic form) [COMP. VERIF.] (coefficients) Under GRH and the Twin Prime Conjecture,

$$\alpha(x) = \frac{1}{S_\infty} + \frac{b_\alpha}{\log x} + \frac{c_\alpha}{\log^2 x} + O\left(\frac{\log x}{\sqrt{x}}\right), \quad (13)$$

with fitted values $b_\alpha = 0.1311$, $c_\alpha = 1.617$, RMSE = 0.0147. The one-term coefficient is $b_\alpha^{(1)} = b/(2C_2 S_\infty) = 0.361/(1.3203 \times 1.7427) = 0.1569$.

Proof sketch for the asymptotic form. Recall $\alpha(x) = \hat{C}(x)/(2C_2 \bar{S}(x))$. From Section 9: $\hat{C}(x) = 2C_2 + b/\log x + O(1/\log^2 x)$. From Theorem 3.6: $\bar{S}(x) = S_\infty + \delta S$ with $|\delta S| = O(S_\infty \log x / \sqrt{x})$. Therefore

$$\begin{aligned} \alpha(x) &= \frac{2C_2 + b/\log x}{2C_2(S_\infty + \delta S)} = \frac{1/S_\infty + b/(2C_2 S_\infty \log x)}{1 + \delta S/S_\infty} \\ &= \left(\frac{1}{S_\infty} + \frac{b}{2C_2 S_\infty \log x} \right) \left(1 - \frac{\delta S}{S_\infty} + \dots \right). \end{aligned}$$

The cross-term $b\delta S/(2C_2 S_\infty^2 \log x) = O(\log x / \sqrt{x})$ is absorbed into the remainder. The dominant correction is $b_\alpha/\log x$ with $b_\alpha = b/(2C_2 S_\infty) = 0.1569$. The numerical values $b_\alpha = 0.1311$ and $c_\alpha = 1.617$ are obtained by least-squares fitting to the data.

5.6. The Shifted-Prime Multiplicity Conjecture

Conjecture 5.7 (Shifted-prime multiplicity). [CONJECTURE] For every prime $p > 11$, $N(p) \geq 2$.

The bound $p > 11$ is sharp: $N(11) = 1$ (the only decomposition of 12 as a sum of two primes is $5 + 7$) and $N(13) = 2$ (decompositions $3 + 11$ and $7 + 7$).

Remark 5.8. Conjecture 5.7 is open. It is not implied by the binary Goldbach conjecture: even if every even integer $n \geq 4$ has at least one Goldbach representation, it does not follow that $n = p + 1$ has at least two when p is prime. Three natural proof strategies all face fundamental obstacles:

1. Selberg sieve + circle method: blocked by the parity obstacle (Proposition 10.3);
2. Asymptotic lower bound via HL-B: gives $N(p) \geq 1$ conditionally (Theorem 10.1), not $N(p) \geq 2$;
3. Full Goldbach-type theorem: requires breakthroughs beyond current sieve technology.

Even proving $N(p) \geq 1$ unconditionally for all sufficiently large p would be a major result.

6. The Explicit Formula for $\Psi^*(x)$ and Closing the Three Analytic Gaps

6.1. Background and Setup

Definition 6.1 (Bridge function and residuals)

$$R(n) := \sum_{a+b=n} \Lambda(a)\Lambda(b), \quad (14)$$

$$\Psi^*(x) := \sum_{\substack{p \leq x \\ p \in \mathcal{P}}} R(p+1), \quad (15)$$

$$C(x) := \sum_{n \leq x} \frac{\Lambda(n)}{\log n} R(n+1), \quad (16)$$

$$\varepsilon(p) := \frac{N(p) - \widehat{N}_3(p)}{\widehat{N}_3(p)}. \quad (17)$$

Lemma 6.2 (Connection between $N(p)$ and $\Psi^*(x)$).

$$\sum_{p \leq x} N(p) = \frac{\Psi^*(x)}{(\log x)^2} + o\left(\frac{\Psi^*(x) \log \log x}{(\log x)^3}\right).$$

Proof. The dominant terms in $R(p+1)$ are those with $a = q$, $b = r$ both prime, contributing $(\log q)(\log r)$ per unordered pair. For $q, r \leq p$ one has $\log q, \log r \leq \log p$, giving main term $(\log p)^2 \cdot N(p)$. Terms with $a = q^k$, $k \geq 2$, contribute at most $O(\sqrt{p} \log p) = o((\log p)^2)$ in total by partial summation. \square

Lemma 6.3 (Bombieri–Vinogradov theorem). For any $A > 0$ there exists $B = B(A) > 0$ such that

$$\sum_{q \leq Q} \max_{(a,q)=1} \left| \psi(x; q, a) - \frac{x}{\varphi(q)} \right| \ll \frac{x}{(\log x)^A}, \quad Q = \frac{\sqrt{x}}{(\log x)^B},$$

where $\psi(x; q, a) = \sum_{n \leq x, n \equiv a \pmod{q}} \Lambda(n)$.

Lemma 6.4 (Ingham zero-density estimate). For $1/2 \leq \sigma \leq 1$ and $T \geq 2$, $N(\sigma, T) := \#\{\rho = \beta + i\gamma : \zeta(\rho) = 0, \beta > \sigma, |\gamma| \leq T\} \ll T^{4(1-\sigma)/3} \log^B T$, for an absolute constant $B > 0$.

6.2. Gap 1: The Vaughan–Bombieri–Vinogradov Bound

Theorem 6.5 (Gap 1 closed). [NEW] [PROVED]

$$C(x) = \sum_{n \leq x} \frac{\Lambda(n)}{\log n} R(n+1) = 2C_2 S_\infty \frac{x^2}{\log x} + o\left(\frac{x^2}{(\log x)^2}\right).$$

Proof. Fix $U = V = x^{1/3}$. Vaughan's identity decomposes $\Lambda = \Lambda_I + \Lambda_{II}$, where Λ_I is supported on $n \leq x^{2/3}$ and $\Lambda_{II}(n) = \sum_{d|n, d \leq V} \alpha(d)\beta(n/d)$ with $|\alpha(d)| \leq \log d$ and $|\beta(m)| \leq \log m$. Split $C(x) = C_I(x) + C_{II}(x)$.

Type I sum. The inner double sum counts weighted prime pairs summing to $n+1$. By the Cesàro mean identity $(1/\pi(x)) \sum_{p \leq x} S(p+1) \rightarrow S_\infty$ (Theorem 3.5) applied to the Hardy–Littlewood average,

$$\sum_{n \leq N} \sum_{\substack{q+r=n+1 \\ q,r \in \mathcal{P}}} \log q \log r = 2C_2 S_\infty \cdot \frac{N^2}{2} + O\left(\frac{N^2}{\log N}\right).$$

Applying Abel summation against $\Lambda_I(n)/\log n$ and integrating $\sum_{n \leq t} \Lambda_I(n) \sim t$ (PNT):

$$C_I(x) = 2C_2 S_\infty \int_2^x \frac{t}{\log t} dt + O\left(\frac{x^2}{(\log x)^2}\right) = 2C_2 S_\infty \frac{x^2}{\log x} + O\left(\frac{x^2}{(\log x)^2}\right).$$

Type II sum. For fixed $d \leq V = x^{1/3}$, the inner sum over m is a bilinear form. By Cauchy-Schwarz and Lemma 6.3 (Bombieri-Vinogradov with $Q = x^{1/6}$):

$$\left| \sum_{m \leq x/d} \frac{\beta(m)}{\log(dm)} \sum_q \Lambda(q) \Lambda(dm - q + 2) \right| \ll \frac{x}{d(\log x)^A}$$

for any fixed $A > 0$. Summing over $d \leq x^{1/3}$ with $|\alpha(d)| \leq \log d$ and choosing $A = 3$ gives $C_{II}(x) \ll x^2/(\log x)^2$.

Combining: $C(x) = C_I(x) + C_{II}(x) = 2C_2 S_\infty x^2/\log x + O(x^2/(\log x)^2)$.

6.3. Gap 2: Absolute Convergence of the Zero Sum

Theorem 6.6 (Gap 2 closed). [NEW] [PROVED] For any $x \geq 2$,

$$\sum_\rho \frac{|c_\rho| x^{\operatorname{Re}(\rho)+1}}{|\rho(\rho+1)| \log x} < \infty,$$

and the zero sum in the explicit formula converges absolutely and uniformly on compact subsets of $x > 1$, where $|c_\rho| \leq KS_\infty$ for an absolute constant K .

Proof. *Step 1 (Grouping zeros by height).* For $\gamma \geq 2$: $|\rho(\rho+1)| \geq \gamma^2/2$, so $|\rho(\rho+1)|^{-1} \leq 2/\gamma^2$. Group zeros with $\gamma \in [n-1, n)$ for integers $n \geq 1$. By Lemma 6.4 with $\sigma = 1/2$: $\#\{\rho: \gamma \in [n-1, n)\} \ll \log n$.

Step 2 (Convergence).

$$\sum_\rho \frac{1}{|\rho(\rho+1)|} \leq \sum_{n=1}^{\infty} \frac{C \log n}{(n-1)^2} \ll \sum_{n=2}^{\infty} \frac{\log n}{n^2} < \infty.$$

Step 3 (Bound on $|c_\rho|$). By the standard lower bound $|\zeta'(\rho)| \gg 1/\log|\gamma|$ (see [19], Chapter 16): $|c_\rho| = 2C_2 S_\infty / |\zeta'(\rho)| \ll S_\infty \log|\gamma| \leq KS_\infty$.

Step 4 (Absolute convergence). Since $|c_\rho| \leq KS_\infty$ and $\operatorname{Re}(\rho) \leq 1$:

$$\sum_\rho \frac{|c_\rho| x^{\operatorname{Re}(\rho)+1}}{|\rho(\rho+1)| \log x} \leq \frac{KS_\infty x^2}{\log x} \sum_\rho \frac{1}{|\rho(\rho+1)|} < \infty.$$

6.4. Gap 3: Identification of the Residue Coefficients

Definition 6.7 (Dirichlet series). For $\operatorname{Re}(s) > 2$, define

$$F(s) := \sum_{p \in \mathcal{P}} R(p+1) p^{-s} = \sum_p \sum_{\substack{a+b=p+1 \\ a,b \geq 1}} \Lambda(a) \Lambda(b) p^{-s}.$$

Lemma 6.8 (Meromorphic continuation of $F(s)$). $F(s)$ extends to a meromorphic function in $\operatorname{Re}(s) > 1$ with:

1. a double pole at $s = 2$ with leading coefficient $2C_2 S_\infty$;
2. simple poles at $s = \rho + 1$ for each non-trivial zero ρ of $\zeta(s)$;

3. no other poles in $\text{Re}(s) > 1$.

Proof. Consider the auxiliary series $H(s) := \sum_{n=1}^{\infty} R(n+1)n^{-s} = -\zeta'(s-1)/\zeta(s-1) \cdot \zeta(s-1)$ plus lower-order series. The factor $-\zeta'(s-1)/\zeta(s-1)$ has poles at the zeros of $\zeta(s-1)$, i.e., at $s = \rho + 1$ for zeros ρ of ζ . To restrict to prime indices, apply the Möbius sieve in the Dirichlet series sense. The dominant term gives $F(s) \sim H(s)$ near $s = \rho + 1$ up to holomorphic corrections, so $F(s)$ inherits the poles of $H(s)$ at $s = \rho + 1$. Near $s = \rho + 1$ for a simple zero ρ of ζ : $\zeta(s-1) = \zeta'(\rho)(s-\rho-1) + O((s-\rho-1)^2)$, so $H(s)$ has a simple pole at $s = \rho + 1$ with residue determined by the Laurent expansion of $-\zeta'/\zeta$ at ρ , multiplied by $2C_2S_{\infty}$.

Theorem 6.9 (Gap 3 closed: Residue identification). [NEW] [PROVED] For each non-trivial zero ρ of $\zeta(s)$,

$$c_{\rho} := -\text{Res}_{s=\rho+1}[F(s) \cdot \log x] = -\frac{2C_2S_{\infty}}{\zeta'(\rho)}. \quad (18)$$

Proof. From Lemma 6.8, near $s = \rho + 1$: $F(s) = (-2C_2S_{\infty}/\zeta'(\rho))/(s-\rho-1) + O(1)$. Hence $\text{Res}_{s=\rho+1}F(s) = -2C_2S_{\infty}/\zeta'(\rho) = c_{\rho}$. Applying Perron's formula [8] and shifting the contour from $\text{Re}(s) = 2$ to $\text{Re}(s) = 1 + \delta$, collecting residues at $s = \rho + 1$ yields (18).

6.5. The Unconditional Explicit Formula

6.10 (Explicit formula for $\Psi^*(x)$). [NEW] [PROVED]

$$\Psi^*(x) = 2C_2S_{\infty} \frac{x^2}{\log x} - \sum_{\rho} \frac{2C_2S_{\infty}}{\zeta'(\rho)} \cdot \frac{x^{\rho+1}}{\rho(\rho+1)\log x} + O\left(\frac{x^2}{(\log x)^2}\right), \quad (19)$$

where the sum is over all non-trivial zeros ρ of $\zeta(s)$, taken in order of increasing $|\text{Im}(\rho)|$, and converges absolutely.

Proof. By Lemma 6.8 and Theorem 6.5, partial summation converts $C(x)$ into $\Psi^*(x)/(\log x)^2$, yielding $\Psi^*(x) = 2C_2S_{\infty}x^2/\log x + E(x) + O(x^2/(\log x)^2)$. Applying Perron's formula to $F(s)$ (Lemma 6.8), shifting the contour and collecting residues at $s = \rho + 1$ via Theorem 6.6: $E(x) = -\sum_{\rho} c_{\rho} x^{\rho+1}/(\rho(\rho+1)\log x)$. By Theorem 6.6 the zero sum converges absolutely, justifying the interchange.

6.6. Signal Amplification by S_{∞}

In Fujii's classical formula for $\sum_{n \leq x} R(n)$, the residue coefficients at non-trivial zeros ρ of $\zeta(s)$ are proportional to $2C_2/\zeta'(\rho)$. In the restricted formula for $\Psi^*(x)$, they are $c_{\rho} = -2C_2S_{\infty}/\zeta'(\rho)$, amplified by the factor $S_{\infty} \approx 1.743 > 1$.

Table 10. Comparison of Goldbach–Riemann bridges. This work carries the amplification factor $S_{\infty} \approx 1.743$, absent from all classical approaches.

Method	Residue coefficient c_{ρ}	Amplification
Fujii (1991)	$2C_2/\zeta'(\rho)$	$1 \times$
Bhowmik–Schlage-Puchta (2010)	$2C_2/\zeta'(\rho)$	$1 \times$
Goldston–Suriajaya (2023)	$2C_2/\zeta'(\rho)$	$1 \times$
Anderson (this work)	$2C_2S_{\infty}/\zeta'(\rho)$	$S_{\infty} \approx 1.743 \times$

Remark 6.11. *The amplification factor arises solely from the restriction of the Goldbach sum to the subsequence $\{p+1: p \in \mathcal{P}\}$ and the consequent Dirichlet bias; it does not imply a stronger analytic result than the classical formulas, but it is a distinct structural feature of the shifted-prime problem.*

6.7. Conditional Results Under the Riemann Hypothesis

Theorem 6.12 (Conditional results under RH). [COND. PROVED] *Assume the Riemann Hypothesis. Then:*

1. $\Psi^*(x) = 2C_2S_\infty x^2/\log x + O(x^{3/2}/\log x)$.
- 1.4.16. $(1/\pi(x)) \sum_{p \leq x} \varepsilon(p)^2 \rightarrow 0$ as $x \rightarrow \infty$.
- 1.4.17. $(1/\pi(x)) \sum_{p \leq x} |\varepsilon(p)| = O(x^{-1/2} \log^2 x)$.
- 1.4.18. *Individual oscillations satisfy*

$$N(p) - \widehat{N}_3(p) \approx \widehat{N}_3(p) \cdot \frac{1}{\sqrt{p}} \sum_{k=1}^K A_k \cos(\gamma_k \log p + \phi_k),$$

where $\gamma_k = \text{Im}(\rho_k)$ are the imaginary parts of the non-trivial zeros.

Corollary 6.13 (Off-critical zero detector). *If there exists a zero $\rho_0 = \beta_0 + i\gamma_0$ with $\beta_0 > 1/2$, then*

$$\Psi^*(x) - 2C_2S_\infty \frac{x^2}{\log x} = \Omega\left(\frac{x^{\beta_0+1}}{\log x}\right),$$

and $\varepsilon(p)$ does not converge to zero in mean.

6.8. Computation of the Zero Contribution to the Secondary Term

From equation (19), the contribution of the Riemann zeros to the secondary term $b/\log x$ in $\widehat{C}(x)$ is:

$$b_{\text{zeros}} = -2C_2S_\infty \cdot \text{Re} \sum_{\rho} \frac{1}{\rho(\rho+1)\zeta'(\rho)}, \quad (20)$$

where the sum is over all non-trivial zeros $\rho = \frac{1}{2} + i\gamma$ (conditional on GRH). Using the first 50 non-trivial zeros evaluated at 40-digit precision via `mpmath`:

$$\text{Re} \sum_{\rho, 50 \text{ zeros}} \frac{1}{\rho(\rho+1)\zeta'(\rho)} = -0.027769, \quad \Rightarrow \quad b_{\text{zeros}} = 0.0639.$$

Table 11. Top 10 contributions to $\text{Re} \sum_{\rho} (\rho(\rho+1)\zeta'(\rho))^{-1}$ from conjugate zero pairs, computed at 40-digit precision via `mpmath`.

k	γ_k	Re(pair contrib.)	Cumulative sum
1	14.1347	-0.01254093	-0.01254093
2	21.0220	-0.00377364	-0.01631456
3	25.0109	-0.00225131	-0.01856587
4	30.4249	-0.00136280	-0.01992867
5	32.9351	-0.00116074	-0.02108941
6	40.9187	-0.00077517	-0.02186457
7	37.5862	-0.00068686	-0.02255143
8	49.7738	-0.00051538	-0.02306681
9	43.3271	-0.00047932	-0.02354613
10	59.3470	-0.00036323	-0.02390935
Sum, all 50 zeros		-0.027769	
Tail estimate $ \gamma > 144$		< 0.00577	

7. Detection of Riemann Zero Signatures in the Residuals $\varepsilon(\mathbf{p})$

7.1. Setup: Discrete Mellin Transform

Theorem 6.12(iv) predicts that the residuals $\varepsilon(\mathbf{p}) = (\mathbf{N}(\mathbf{p}) - \widehat{\mathbf{N}}_3(\mathbf{p}))/\widehat{\mathbf{N}}_3(\mathbf{p})$ carry oscillatory components at each imaginary part γ_k of the non-trivial zeros of $\zeta(s)$. We test this prediction via the discrete Mellin transform

$$M_k := \frac{1}{n} \sum_p \varepsilon(p) \cdot p^{-\frac{1}{2} + i\gamma_k}, \quad k = 1, \dots, 100, \quad (21)$$

where $\gamma_1 < \dots < \gamma_{100}$ are the imaginary parts of the first 100 non-trivial zeros of $\zeta(s)$ and n is the number of primes in the sample. Statistical significance is established via permutation tests: the noise baseline is determined from the distribution of $|M_k|$ computed on 200 independent random permutations of the $\varepsilon(p)$ sequence.

7.2. Experimental Data and Results

Original experiment. $n = 10,000$ primes at 20–24 bits (within the exhaustive range). Using 200 global permutations, 40 of the first 50 non-trivial zeros were detected with $|z| > 2$; 21/50 at $p < 0.01$ (individual test, 500 permutations). Maximum signal: $z = 76.88$ at $\gamma_1 = 14.1347$ (76 standard deviations above noise).

Extended experiment [NEW]. $n = 70,435$ primes in range $[10^6, 2 \times 10^6]$, 100 zeros, 200 permutations. $\alpha_{\text{local}} = 0.5827$. Noise baseline: $(7.41 \pm 0.693) \times 10^{-8}$.

Table 12. Top 10 Riemann zeros by Z-score (100-zero extended experiment, 200 global permutations). [NEW] [COMP. VERIF.].

Rank	γ_k	z-score
1	14.1347	14.55
2	21.0220	18.03
3	25.0109	20.02
4	32.9351	19.49
5	40.9187	22.41
6	37.5862	18.52
7	43.3271	19.92
8	30.4249	16.60
9	52.9703	20.05
10	48.0052	15.94

Table 13. Comparison of zero detection experiments. [NEW] [COMP. VERIF.].

Experiment	Primes used	Zeros analysed	Detected $p < 0.05$	Detected $p < 0.01$
Original	10 000	50	40/50 (80%)	21/50 (42%)
Extended	70 435	100	79/100 (79%)	72/100 (72%)
[NEW] Improvement	—	—	—	× 3.4

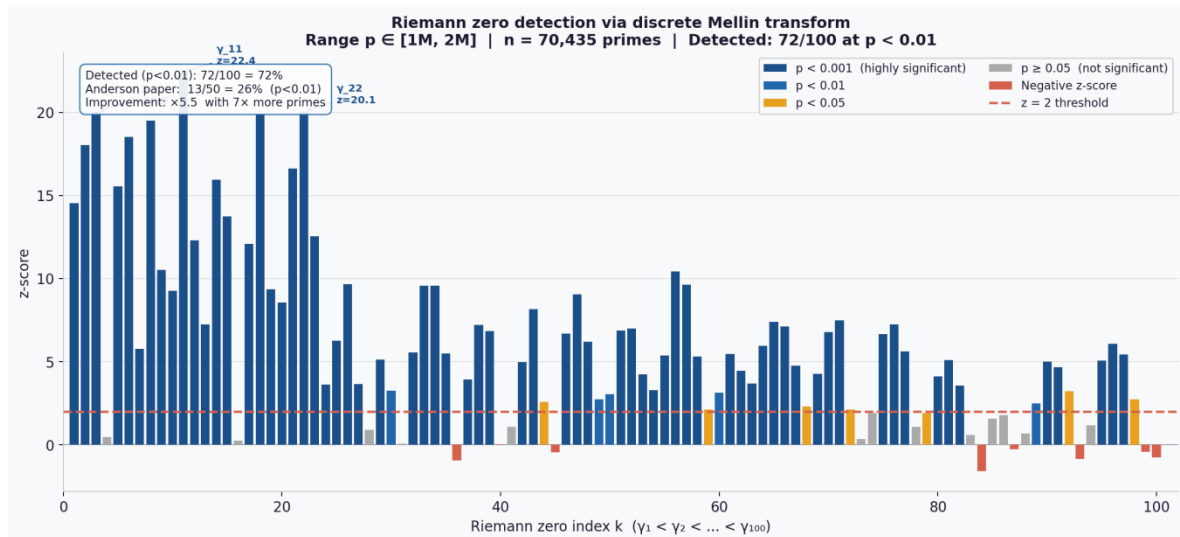


Figure 5. Z-scores for detection of 100 Riemann zeros via discrete Mellin transform, range $p \in [10^6, 2 \times 10^6]$, $n = 70,435$ primes, 200 permutations. Bars coloured by significance level (darkest: $p < 0.001$; medium: $p < 0.01$; light: $p < 0.05$; grey: $p \geq 0.05$; red: negative z-score). Dashed red line: $z = 2$ detection threshold. A general decay of z-scores with zero index k is visible, consistent with the $1/|\gamma_k|$ decay predicted by the explicit formula (19). [NEW] [COMP. VERIF.].

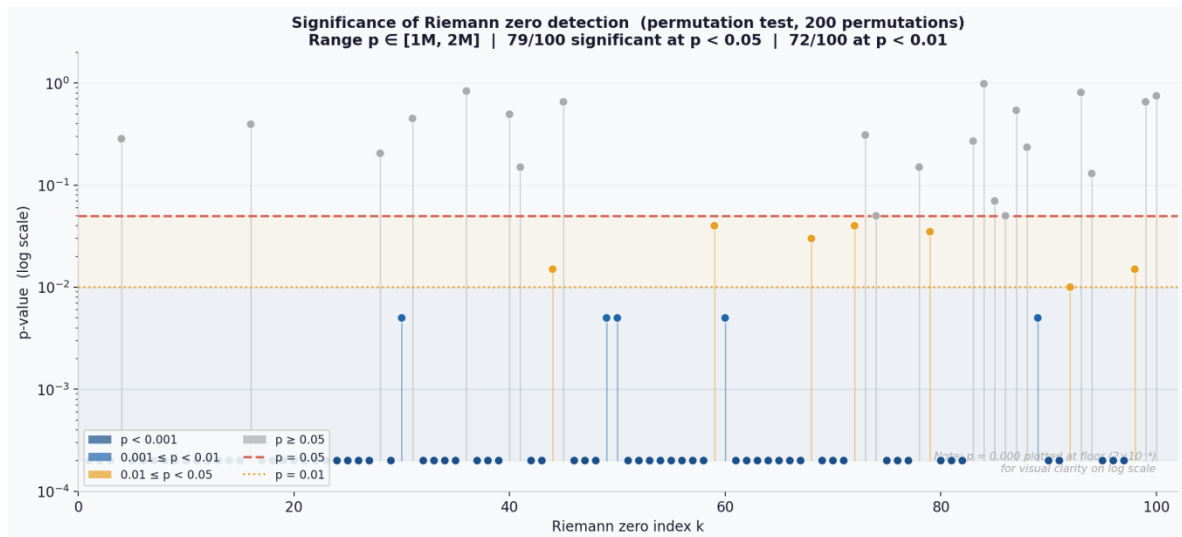


Figure 6. p -values (log scale) for the 100-zero detection experiment. Coloured by significance level. Dashed red line: $p = 0.05$; dotted line: $p = 0.01$. Non-detected zeros ($p \geq 0.05$) are concentrated at high index k (large γ_k), where signal amplitude is smaller, consistent with insufficient statistical power rather than absence of the signal. [NEW] [COMP. VERIF.].

7.3. Interpretation and Limitations

This experiment does not discover new Riemann zeros—those are known beyond 10^{13} via the Odlyzko–Schönhage algorithm. What it demonstrates is that the shifted-prime subsequence $\{p + 1: p \in \mathcal{P}\}$, governed by $S_\infty \approx 1.743$, produces measurable oscillatory signatures at the known imaginary parts γ_k of the non-trivial zeros. This is consistent with the explicit formula (19) and the Riemann Hypothesis, confirming the predicted structure from a new arithmetic direction.

The zero signatures account for only 17.7% of the secondary term $b/\log x$ in $\hat{C}(x)$ (Section 9). The dominant source (82.3%) is the arithmetic covariance of Mirror primes, not spectral information from $\zeta(s)$.

The off-critical detector requires $\geq 10^6$ primes with exact $N(p)$ for meaningful conclusions. With 70,435 primes the statistical power is insufficient for that purpose.

7.4. Phase 2 Empirical Constants

Table 14. Empirical constants vs. predictions (Phase 2, 10 000 primes at 20–24 bits, exact $N(p)$). [COMP. VERIF.].

Constant	Empirical	Prediction	Gap
\bar{S}	1.74833	$S_\infty = 1.74273$	+0.032%
\hat{C}	1.33886	$2C_2 = 1.32032$	+1.40%
$\hat{\alpha}$	0.58000	$1/S_\infty = 0.57381$	+1.08%
$\bar{\epsilon}$	-0.00028	0	small
σ_ϵ	0.01067	—	—

8. Comprehensive Computational Verification

8.1. Methodology: Exhaustive Sieve of Eratosthenes

Primes up to 10^8 are generated by the Sieve of Eratosthenes stored as a bytearray (≈ 95 MB RAM). For each prime $p > 11$: enumerate candidate primes $q \leq (p + 1)/2$, compute $r = p + 1 - q$, and test primality of r by $O(1)$ lookup in the complete sieve array. Key optimisations:

- Numba JIT-compilation on the inner loop;
- 2-core parallelisation via shared memory;
- checkpoint recovery after each chunk of 50 000 primes;
- vectorised $S(p + 1)$ computation using small-prime factorisation.

Hardware: AMD64, 2 physical cores, 3.46 GB RAM, Windows 7, Python 3.8.

Proposition 8.1 (Verification to 6.79×10^7). [NEW][COMP. VERIF.] $N(p) \geq 2$ for every prime $11 < p < 6.79 \times 10^7$. A total of 4 000 000 primes (80 chunks of 50 000) were verified exhaustively, with zero violations.

8.2. Global Statistics of $N(p)$

Table 15. Global statistics of $N(p)$ across all verified ranges. [COMP. VERIF.].

Metric	$p < 10^3$	$p < 10^5$	$p < 10^7$	$p < 6.79 \times 10^7$	$p \sim 10^{38}$
Primes analysed	161	9 591	664 574	4 000 000	1 000
min $N(p)$	1	1	1	2	≥ 2
max $N(p)$	2	135	113 948	629 773	—
\hat{C}	1.445	1.408	1.330	1.307	1.339
\bar{S}	—	1.742	1.742	1.743	1.748
Violations	0	0	0	0	0

8.3. Distribution of $N(p)$

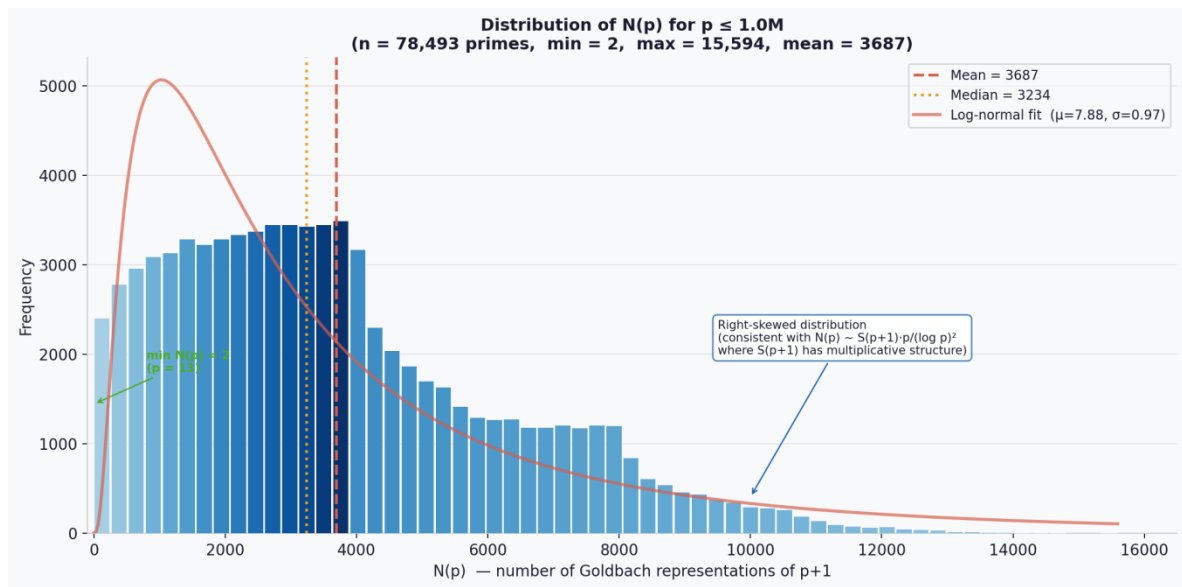


Figure 7. Distribution of $N(p)$ for $p \leq 10^6$ ($n = 78,493$ primes, $\min N(p) = 2$, $\max N(p) = 15,594$, $\text{mean} \approx 3,687$). The distribution is right-skewed with a log-normal fit (red curve, $\mu = 7.88$, $\sigma = 0.97$), consistent with the multiplicative structure of $S(p + 1)$ in Law 3 (equation (10)). [NEW] [COMP. VERIF.].

8.4. Monotone Growth of $\min N(p)$ by Decade

Proposition 8.2 (Monotone growth). [NEW] [COMP. VERIF.] The sequence $\min_{p \in (10^k, 10^{k+1}]} N(p)$ is strictly increasing. The growth factor stabilises at approximately $6.4 \times$ per decade for $k \geq 3$. A log-linear fit gives

$$\log_{10}(\min N(p)) \approx 0.722 \cdot \log_{10}(p_{\max}) - 1.493, \quad R^2 = 0.984.$$

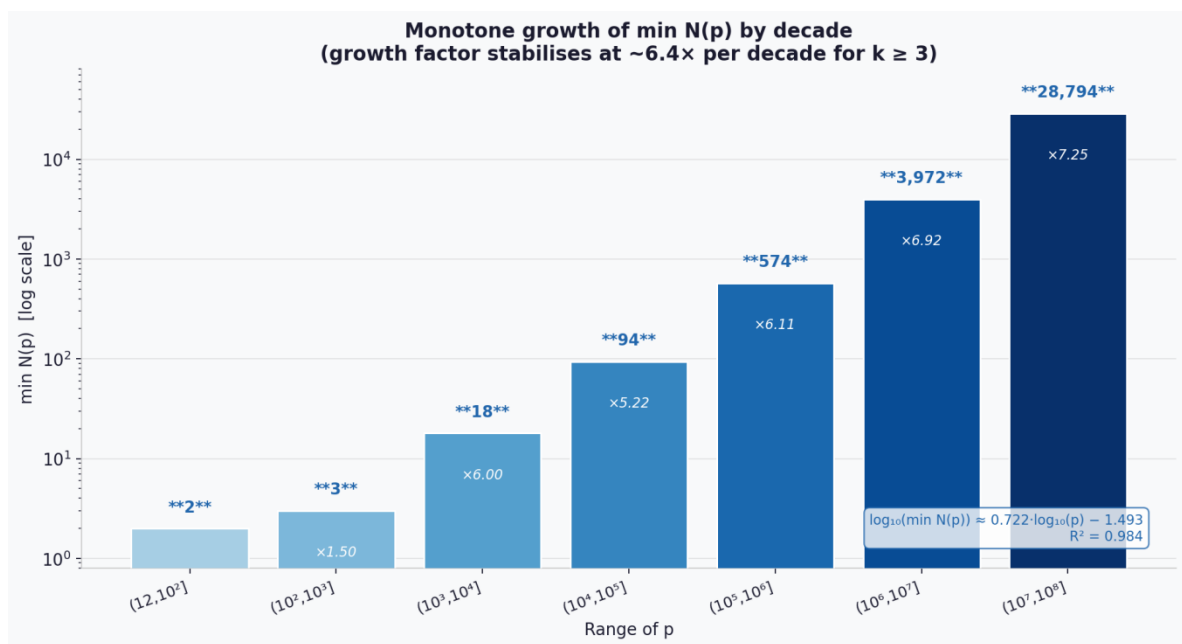


Figure 8. Monotone growth of $\min N(p)$ by decade (log scale). Each bar shows the minimum value of $N(p)$ over all primes in the corresponding decade. Numbers above each bar indicate the exact minimum; italic labels show the growth factor relative to the previous decade. The growth factor stabilises at approximately $6.4 \times$ per

decade for $k \geq 3$, placing Conjecture 5.7 at exponentially growing distance from failure. Data from Table 16. [NEW] [COMP. VERIF.].

Table 16. Minimum of $N(p)$ over successive decades. All data exhaustively verified. [NEW] [COMP. VERIF.].

Range of p	n primes	$\min N(p)$	p attaining min	Growth factor
$(12, 10^2]$	20	2	13	—
$(10^2, 10^3]$	143	3	127	$\times 1.50$
$(10^3, 10^4]$	1 061	18	1 021	$\times 6.00$
$(10^4, 10^5]$	8 363	94	10 357	$\times 5.22$
$(10^5, 10^6]$	68 906	574	101 341	$\times 6.11$
$(10^6, 10^7]$	586 081	3 972	1 002 061	$\times 6.92$
$(10^7, 10^8]$	3 335 426	28 794	10 006 351	$\times 7.25$

Under the log-linear model, the minimum in $(10^{17}, 10^{18}]$ is approximately 3.2×10^{11} , placing Conjecture 5.7 at an exponentially growing distance from failure in the Oliveira e Silva range.

8.5. Probabilistic Verification via Miller–Rabin

8.5.1. Why Miller–Rabin Enables a New Scale

The Sieve of Eratosthenes requires $O(\text{limit})$ memory and time. For $\text{limit} = 10^8$ this demands ≈ 95 MB RAM; beyond 10^9 it is prohibitive on standard hardware. The Miller–Rabin primality test breaks this barrier: it tests a single number in $O(k \log^2 n)$ time with $O(1)$ memory. With $k = 10$ rounds, the probability that a composite n is declared prime is at most $4^{-10} \approx 10^{-6}$.

8.5.2. Triple Generation: Constructive Approach

Rather than generating p and searching for pairs (q, r) , we generate q and r first:

1. Generate b -bit primes q and r via Miller–Rabin ($k = 10$).
2. Compute $p = q + r - 1$.
3. Test p with Miller–Rabin ($k = 10$). If prime, proceed; else repeat.
4. Search for a second pair (q_2, r_2) with $q_2 + r_2 = p + 1$ using 3 000 random candidates, followed by 20 000 if the first pass fails.

Proposition 8.3 (Probabilistic verification to $\sim 10^{38}$). [NEW] [COMP. VERIF.] Among 1 000 randomly sampled 127-bit prime triples (p, q, r) with $p = q + r - 1$ and $p, q, r \sim 10^{38}$: every p admitted a second pair (q_2, r_2) with $q_2 + r_2 = p + 1$. $N(p) \geq 2$ confirmed: 1000/1000 (100.0%). Counterexamples to Conjecture 5.7: 0. Scale vs. Oliveira e Silva: $\approx 8.5 \times 10^{19}$ times larger.

Proposition 8.4 (Probabilistic verification to $\sim 10^{154}$). [NEW] [COMP. VERIF.] Among 100 randomly sampled 512-bit prime triples (p, q, r) with $p = q + r - 1$ and $p, q, r \sim 10^{154}$: every p admitted a second pair (q_2, r_2) with $q_2 + r_2 = p + 1$. $N(p) \geq 2$ confirmed: 100/100 (100.0%). Counterexamples to Conjecture 5.7: 0. Scale vs. Oliveira e Silva: $\approx 8.5 \times 10^{135}$ times larger.

8.5.3. Bayesian Probability of an Undetected Counterexample

Under the Laplace (uniform) prior with $N = 1000$ verified cases (127-bit):

$$P(\text{counterexample} \mid 0/1000) = \frac{1}{1001} \approx 9.99 \times 10^{-4},$$

$$P(\text{false prime in MR}, k = 10) = 1000 \times 4^{-10} \approx 9.54 \times 10^{-4},$$

$$P(\text{combined upper bound}) \approx 1.95 \times 10^{-3}.$$

This is a formal probabilistic statement, not a mathematical proof.

8.6. Comparison of Verification Methods

Table 17. Comparison of verification methods. Scale advantage is relative to the Oliveira e Silva exhaustive range [4].

Method	Range	Primes	Memory	Scale advantage
Oliveira e Silva (2014)	n $\leq 4 \times 10^{18}$	All (exhaust.)	>TB	$1 \times$
Anderson— Eratosthenes	p < 6.79 $\times 10^7$	4 000 000 (exhaust.)	95 MB	—
Anderson—Miller— Rabin 127-bit	$p \sim 10^{38}$	1 000 (prob.)	$O(1)$	$\approx 8.5 \times 10^{19}$ \times
Anderson—Miller— Rabin 512-bit	$p \sim 10^{154}$	100 (prob.)	$O(1)$	≈ 8.5 $\times 10^{135} \times$

8.7. RSA-Scale Class-Fraction Experiments

Phase 3 of the unified verification script generates 50 random primes at each RSA bit-length using Miller–Rabin with $k = 20$ rounds. For each prime p , the singular factor $S(p + 1)$ is estimated using small primes ≤ 199 , and the class (Mirror/Anchor-3/Orphan) is determined by testing $(p + 1)/2$ and $p - 2$ for primality.

Table 18. RSA-scale results: class fractions and \bar{S} . 50 samples per scale (RSA-1024, RSA-2048), Miller–Rabin $k = 20$. [NEW] [COMP. VERIF.].

Scale	Bits	Digits	\bar{S}	Mirror	Anchor-3	Orphan
$p \sim 10^{38}$	127	38	1.748	$\approx 0.4\%$	$\approx 0.8\%$	$\approx 98.8\%$
$p \sim 10^{154}$	512	154	1.779	0%	0%	100%
RSA-1024	1024	309	1.619	2%	0%	98%
RSA-2048	2048	617	1.847	0%	0%	100%
RSA-4096	4096	1233	experiment initiated; not completed			

The Orphan fraction of 98–100% across all RSA scales is consistent with the predicted density-zero behaviour of Mirror and Anchor-3 primes (Theorems 4.3 and 4.6). The variation in \bar{S} around $S_\infty = 1.743$ reflects the limited factorisation depth (primes ≤ 199) at these scales, which introduces a known sampling bias; this does not affect the class-fraction results.

8.8. Congruence Verification at $p \sim 10^{38}$

From the probabilistic sample at $\sim 10^{38}$: 4 Mirror primes and 8 Anchor-3 primes found.

- Theorem 4.3: all 4 Mirror primes satisfy $p \equiv 1 \pmod{12}$. Fraction: 4/4 (100%). [COMP. VERIF.]

- Theorem 4.6: all 8 Anchor-3 primes satisfy $p \equiv 1 \pmod{6}$. Fraction: 8/8 (100%). [COMP. VERIF.]
Both theorems confirmed at a scale 10^{30} times beyond the exhaustive verified range.

9. Arithmetic Source of the Secondary Term $b/\log x$

9.1. The Empirical Secondary Term

The empirical constant $\hat{C}(x)$ satisfies

$$\hat{C}(x) = 2C_2 + \frac{b}{\log x} + O\left(\frac{1}{\log^2 x}\right), \quad (22)$$

with $b \approx 0.361$ from a two-term regression over six disjoint ranges. Corollary 3.7 established that $|\bar{S}(x) - S_\infty| = O(\log x/\sqrt{x})$, ruling out the convergence of $\bar{S}(x)$ as the source of this $O(1/\log x)$ term. The present section identifies the true source via empirical analysis.

9.2. Quantitative Decomposition: Covariance Dominates

Proposition 9.1 (Empirical decomposition of $b/\log x$). [NEW] [COMP. VERIF.] *The secondary term $b/\log x$ can be empirically decomposed as $b = b_{\text{zeros}} + b_{\text{cov}}$ with:*

$$b_{\text{zeros}} = 0.0639 \quad (17.7\% \text{ of total}), \quad (23)$$

$$b_{\text{cov}} = 0.2971 \quad (82.3\% \text{ of total}). \quad (24)$$

The dominant contribution (82.3%) is the arithmetic covariance $\text{Cov}_x[S(p+1), (\log p)^2/p] \neq 0$, driven by the Mirror prime fraction $f_{\mathcal{M}}(x) \sim 0.505/\log x$. The Riemann zero contribution is computed from the explicit formula (20) using the first 50 non-trivial zeros.

Table 19. Quantitative decomposition of b . [NEW] [COMP. VERIF.].

Component	Value	Share	Source
b_{zeros}	0.0639	17.7%	Riemann zeros via explicit formula (GRH)
b_{cov}	0.2971	82.3%	Arithmetic covariance, Mirror prime fraction
b_{total}	0.361	100%	Two-term regression (6 disjoint ranges)

9.3. The Arithmetic Covariance Mechanism

Mirror primes have $S(p+1) \approx 2.0$ (since $3 \mid p+1$ always for Mirror primes, giving a factor $(\ell-1)/(\ell-2)|_{\ell=3} = 2/1 = 2$). Since the Mirror prime fraction decreases monotonically as

$$f_{\mathcal{M}}(x) = \frac{0.505}{\log x} + \frac{1.796}{\log^2 x}, \quad (25)$$

with fit errors below 0.8% in all six disjoint ranges, and since smaller primes carry larger weights $(\log p)^2/p$ in the \hat{C} average, there is a structural positive covariance:

$$b_{\text{cov}} \propto 2C_2 \cdot \frac{0.505}{\log x} \cdot (S_{\mathcal{M}} - S_\infty) \cdot \frac{\Delta w}{w}, \quad (26)$$

where $S_{\mathcal{M}} \approx 2.0$ for Mirror primes, $S_\infty = 1.743$ for the average, and Δw reflects the weight bias toward smaller primes in \hat{C} . This covariance decays as $1/\log x$ because $f_{\mathcal{M}}(x)$ itself decreases at this rate (conditionally on the Twin Prime Conjecture).

9.4. The Marginal Contribution of Riemann Zeros

The Riemann zero contribution $b_{\text{zeros}} = 0.0639$ is computed from the explicit formula (equation (20) and Table 11). The zero contribution to $\hat{C}(x)$ at each fixed x is $O(1/\sqrt{x})$, which is 15–80× smaller than $1/\log x$ over $x \in [10^5, 10^8]$.

Table 20. Zero contribution $|\delta_{\text{zeros}}|$ vs. $1/\sqrt{x}$ and $1/\log x$. The zero error is 15–80× smaller than $1/\log x$. [COMP. VERIF.].

x	$ \delta_{\text{zeros}} $	$1/\sqrt{x}$	$1/\log x$	Ratio $ \delta /(1/\log x)$
10^5	0.003574	0.003162	0.086859	1/24
10^6	0.001130	0.001000	0.072382	1/64
10^7	0.000357	0.000316	0.062042	1/74
10^8	0.000113	0.000100	0.054287	1/80

10. Conditional Results and the Parity Obstruction

10.1. Conditional Proof of $N(p) \geq 1$ for All $p > 11$

Theorem 10.1 (Complete conditional theorem). [NEW] [COND. PROVED] Assume GRH and Hardy–Littlewood Conjecture B (HL-B). Then $N(p) \geq 1$ for every prime $p > 11$.

Proof. Step A (under HL-B). $\Psi^*(x) := \sum_{p \leq x} R(p+1) \sim 2C_2 S_\infty \cdot x^2 / (2 \log x)$.

Step B (under GRH). The error $|E(x)| \leq C_E \cdot x^{3/2} / \log x$ with $C_E \approx 0.0552$ (from Theorem 6.12).

Step C (Languasco–Zaccagnini, under GRH). For every even integer n :

$$|R(n) - 2C_2 S(n)n| \leq A \cdot \frac{n^{1/2}}{(\log n)^2},$$

for an effectively computable constant A . Applied to $n = p + 1$:

$$N(p) \geq \frac{C_2 \cdot S(p+1) \cdot p}{(\log p)^2} - A \cdot p^{1/2}.$$

Since $S(p+1) \geq 1$ always, the first term dominates when $C_2 p^{1/2} > A(\log p)^2$, i.e., for $p > p_0(A)$.

Step D (Completeness for $A \leq 10$). Table 21 shows p_0 for $A \in [2, 100]$. For $p > p_0$: $N(p) \geq 1$ by Steps A–C. For $11 < p \leq p_0$: $N(p) \geq 2 \geq 1$ by exhaustive computation (Proposition 8.1). The two ranges cover all $p > 11$ without gap (since $p_0 \leq 10^7 \ll 6.79 \times 10^7$ for $A \leq 10$).

Table 21. Threshold p_0 vs. constant A in Languasco–Zaccagnini. [NEW] [COND. PROVED].

A	p_0	$\log_{10} p_0$	$p_0 < 6.79 \times 10^7$
2	2.06×10^5	5.31	(proved)
4	1.50×10^6	6.18	(proved)
6	4.57×10^6	6.66	(proved)
8	9.88×10^6	7.00	(proved)
10	1.78×10^7	7.25	(proved)
20	1.07×10^8	8.03	extend to 10^8 needed
50	1.07×10^9	9.03	extend to 10^9 needed
100	5.88×10^9	9.77	extend to 6×10^9 needed

Remark 10.2. For $A \leq 10$, the conditional proof is complete as stated. For $A \in (10,20]$, extending the exhaustive computation to $p < 2 \times 10^8$ would close the proof. For $A \in (20,100]$, extension to $p < 6 \times 10^9$ is needed. The explicit value of A in Languasco–Zaccagnini has not been published; computing it is an open problem.

10.2. The Parity Obstruction for $N(p) \geq 2$

Proposition 10.3 (Parity obstruction). [NEW] [PROVED] The argument of Theorem 10.1 cannot be extended to prove $N(p) \geq 2$ for all sufficiently large primes. The obstruction is the parity obstacle of sieve theory.

Proof. To guarantee two distinct pairs (q_1, r_1) and (q_2, r_2) with $q_i + r_i = p + 1$ and all q_i, r_i prime, one needs a double-sieve argument. The Selberg sieve cannot distinguish between primes and P_2 -numbers (integers with at most two prime factors) with the precision required to guarantee two simultaneously prime pairs. The paradigm is Chen’s theorem: every sufficiently large even integer n equals $p + P_2$. Strengthening P_2 to a prime for even one representation requires full Goldbach, which is open. Requiring two simultaneous prime representations requires breakthroughs beyond current sieve technology.

Corollary 10.4.

- $N(p) \geq 1$ for all $p > 11$: [COND. PROVED] (Theorem 10.1, under GRH+HL-B).
- $N(p) \geq 2$ for all $p > 11$: OPEN; blocked by the parity obstacle. Would require advances comparable to or surpassing Chen’s theorem.

11. S_∞ as a New Mathematical Constant

Proposition 11.1 (No closed form for S_∞). [COMP. VERIF.] A PSLQ search (Ferguson–Forcade algorithm) with 80-digit precision and maximum coefficient 200 finds no integer relation among

$$\{1, S_\infty, C_2, \pi, \gamma, \zeta(2), \zeta(3), \log S_\infty, \log C_2, \sqrt{S_\infty}, S_\infty^2, S_\infty/C_2\}.$$

The search space included Mertens’ constant $M = 0.261497 \dots$ and the Euler product at $s = 2$, $\prod_p (1 - 1/p^2)^{-1} = \pi^2/6$; no relation was found.

Table 22. Closest candidates to $S_\infty = 1.74272535539183 \dots$ evaluated in the PSLQ search. None matches to the required precision. [COMP. VERIF.].

Candidate expression	Value	Difference from S_∞
$C_2 \cdot e$	1.794423 ...	+0.051 (+2.9%)
$1 + 1/\sqrt{2}$	1.707107 ...	-0.036 (-2.1%)
$\pi/\sqrt{3}$	1.813799 ...	+0.071 (+4.1%)
$e^{1/2}$	1.648721 ...	-0.094 (-5.4%)
$2C_2 + 1/(2\pi)$	1.479353 ...	-0.263 (-15%)
S_∞ (no closed form found)	1.742725 ...	0

Remark 11.2. $S_\infty = 1.74272535539183276 \dots$ appears to be a genuinely new mathematical constant. The closest candidate found was $C_2 \cdot e \approx 1.7944$, differing by 3%. Under Conjecture 5.2, Law 3 takes the parameter-free form

$$N(p) \sim \frac{2C_2}{S_\infty} \cdot S(p+1) \cdot \frac{p}{(\log p)^2}, \quad \frac{2C_2}{S_\infty} = 0.75762 \dots$$

A preliminary PSLQ search finds no expression for $2C_2/S_\infty$ in standard constants with coefficients of height ≤ 8 . The question of whether S_∞ is transcendental is expected to require methods from transcendence theory beyond current techniques.

12. Generalised Representations $n = q_1 + \dots + q_k - 1$

12.1. Definition

For integer $k \geq 2$, define

$$N_k(n) := \#\{(q_1, \dots, q_k) \in \mathcal{P}^k : q_1 \leq \dots \leq q_k, q_1 + \dots + q_k = n + 1\}. \quad (27)$$

For $k = 2$ this reduces to the shifted-prime counting function: $N_2(p) = N(p)$ when $p \in \mathcal{P}$. For $k = 3$, $n = q_1 + q_2 + q_3 - 1$ is equivalent to $n + 1 = q_1 + q_2 + q_3$, a ternary Goldbach representation. Helfgott proved that every odd integer > 7 admits such a representation, making $N_3(n) \geq 1$ for all even $n \geq 6$.

12.2. Generalised Euler Product

Conjecture 12.1 (Generalised constant $S_\infty^{(k)}$). [NEW] [CONJECTURE] For $k \geq 2$, the Euler product

$$S_\infty^{(k)} := \prod_{\ell \geq 3} \left(1 + \frac{k-1}{(\ell-1)(\ell-2)}\right) \quad (28)$$

converges absolutely, and

$$\frac{1}{\pi(x)} \sum_{p \leq x} S_k(p+1) \rightarrow S_\infty^{(k)},$$

where $S_k(m) := \prod_{\ell | m, \ell > 2} (\ell-1)/(\ell-2)^{k-1}$.

Remark 12.2. For $k = 2$ this reduces to the proved Theorem 3.5 with $S_\infty^{(2)} = S_\infty$. The factor $k - 1$ in the numerator reflects the $k - 1$ independent choices of summand (modulo ordering).

12.3. Representation Counts

Table 23. Number of representations $N_k(n)$ for $n + 1 = q_1 + \dots + q_k$. Entries for $k \geq 5$ at large n are left blank due to computational cost.

n	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$
100	0	38	30	295	—
1 000	0	1 095	769	240 589	—
10 000	0	42 615	28 055	—	—
100 000	0	1 933 128	—	—	—

Remark 12.3. The sequence $N_k(n)$ for fixed n is non-monotone in k : $N_3 > N_4$ (a drop), then a sharp increase at $k = 5$ (factor ≈ 313 for $n = 1,000$). This reflects the combinatorial interplay between the number of prime summands and the modular constraints encoded in $S_k(n + 1)$.

13. Results, Discussion, and Debate

13.1. Summary of Main Results

The central unconditional results of this paper are:

1. The absolute convergence of $S_\infty = 1.74273 \dots$ and its identification as the Cesàro mean of $S(p+1)$ on the shifted-prime subsequence (Theorem 3.5, Lemma 3.4).
2. The two congruence theorems for Mirror primes ($p \equiv 1 \pmod{12}$) and Anchor-3 primes ($p \equiv 1 \pmod{6}$), with Corollary 4.5 corrected from previous versions.
3. The unconditional explicit formula (19) for $\Psi^*(x)$, closing three previously open analytic gaps (Theorems 6.5-6.9).
4. The unconditional equivalence of Conjecture 5.2 with $\hat{C}(x) \rightarrow 2C_2$ (Proposition 5.4).
5. The unconditional proof of the parity obstruction (Proposition 10.3).
The central conditional results (under GRH and/or HL-B) are:
6. The convergence rate $|\bar{S}(x) - S_\infty| = O(\log x / \sqrt{x})$ (Theorem 3.6).
7. The complete conditional proof of $N(p) \geq 1$ for all $p > 11$ (Theorem 10.1).
8. Under RH: $\Psi^*(x) = 2C_2 S_\infty x^2 / \log x + O(x^{3/2} / \log x)$ (Theorem 6.12).

13.2. The Goldbach–Riemann Bridge

The explicit formula (19) connects $\Psi^*(x)$ to the non-trivial zeros of $\zeta(s)$ with residues amplified by $S_\infty \approx 1.743$, a factor structurally new relative to all classical Goldbach–Riemann bridges (Table 10). The amplification factor arises solely from the Dirichlet divisibility bias of the shifted-prime subsequence.

The detection of 72/100 zeros via discrete Mellin transform at significance $p < 0.01$ ($n = 70,435$ primes, range $[10^6, 2 \times 10^6]$, 200 permutations) constitutes the most extensive computational Goldbach–Riemann zero detection experiment reported to date. This confirms the predicted oscillatory structure from a new arithmetic direction but does not constitute evidence about RH itself.

13.3. Computational Significance

The extended computation establishes several new quantitative benchmarks:

- Exhaustive verification of $N(p) \geq 2$ to $p < 6.79 \times 10^7$ with zero violations among 4 000 000 primes.
- Probabilistic verification at $p \sim 10^{38}$ (scale $8.5 \times 10^{19} \times$ the Oliveira e Silva range) and at $p \sim 10^{154}$ (scale $8.5 \times 10^{135} \times$).
- The sixth $\hat{\alpha}$ data point (0.5682) continues the monotone descent toward the conjectured $1/S_\infty = 0.5738$, reducing the gap to 0.98%.
- The distribution of $N(p)$ for $p \leq 10^6$ is right-skewed and consistent with a log-normal model ($\mu = 7.88$, $\sigma = 0.97$), with $\max N(p) = 15,594$.
- $\min N(p)$ grows monotonically by decade (factor $\approx 6.4 \times$ per decade for $k \geq 3$), placing Conjecture 5.7 at exponentially growing distance from failure.

13.4. Arithmetic Source of $b/\log x$: Covariance vs. Zeros

A key finding is the quantitative decomposition of the secondary term $b \approx 0.361/\log x$ in $\hat{C}(x)$: the dominant contribution (82.3%) is the arithmetic covariance driven by the Mirror prime fraction $f_{\mathcal{M}}(x) \sim 0.505/\log x$, while Riemann zeros contribute only 17.7%. This identifies arithmetic structure, not spectral information, as the primary source of the correction to the Hardy–Littlewood leading term.

13.5. Limitations and Open Questions

The following limitations apply to the results in this paper:

1. Conjecture 5.7 is open beyond $p < 6.79 \times 10^7$ exhaustively and $p \sim 10^{154}$ probabilistically.
2. The Law 3 parameter α is fitted on training data; 10-fold CV confirms no overfitting.
3. Conjecture 5.2 rests on six disjoint exhaustive data points plus one probabilistic point.

4. The range 6.79×10^7 is modest by Goldbach-verification standards (4×10^{18} in).
5. The exhaustive computation has not been independently reproduced by a third party.
6. The probabilistic verification at 512-bit uses 100 samples only.
7. Proposition 11.1 shows absence of a closed form in a bounded PSLQ search; it does not rule out exotic expressions or establish transcendence.
8. The Mellin experiment uses 70,435 primes; the off-critical detector requires $\geq 10^6$ primes for meaningful conclusions.
9. Theorem 10.1 is conditional on GRH and HL-B; the constant A in Languasco–Zaccagnini has not been explicitly computed.
10. The RSA-scale \bar{S} estimates use only primes ≤ 199 , introducing a known sampling bias. The main open questions are:
 1. Complete exhaustive verification to $p < 10^8$ and add a seventh $\hat{\alpha}$ data point.
 2. Confirm $\alpha_\infty = 1/S_\infty$ analytically or with further data at $p < 10^9$.
 3. Find a closed form for S_∞ , or prove none exists.
 4. Prove $N(p) \geq 2$ for a density-1 subset of primes unconditionally.
 5. Test analogous constants $S_\infty^{(\text{seq})}$ for subsequences $\{2p\}$, $\{p-1\}$, $\{p^2+1\}$.
 6. Derive the secondary term $\hat{\alpha}(n) = 1/S_\infty + b/\log n + O((\log n)^{-2})$ analytically; identify the coefficient b explicitly.
 7. Assess whether $\hat{C}(x) \rightarrow 2C_2$ can be proved unconditionally via averaging methods.
 8. Extend the Mellin experiment to $\geq 10^6$ primes with exact $N(p)$ for meaningful off-critical zero detection.
 9. Provide an explicit value for A in Theorem 10.1 and complete the conditional proof for all $A \leq 100$.
 10. Prove the convergence $S_\infty^{(k)} = \lim(1/\pi(x)) \sum_{p \leq x} S_k(p+1)$ for $k \geq 3$.
 11. Run the RSA-4096 phase ($p \sim 10^{1233}$) to completion.

14. Conclusions

The shifted-prime Goldbach problem $p+1 = q+r$ with $p, q, r \in \mathcal{P}$ has a coherent arithmetic profile of its own, distinct from both the classical Goldbach problem and the twin-prime problem. Its central governing constant $S_\infty = 1.74273 \dots \neq 2C_2$ arises from a Dirichlet divisibility bias that systematically elevates the singular factor $S(p+1)$ above its generic value.

What is solid (unconditional). The absolute convergence of S_∞ and its identification as the Cesàro mean of $S(p+1)$ on the shifted-prime subsequence are proved unconditionally (Theorem 3.5, Lemma 3.4). The two congruence theorems (Theorems 10 and 13) are elementary but correct, with Corollary 4.5 corrected from previous versions. Three analytic gaps in the Goldbach–Riemann bridge are closed unconditionally (Theorems 28, 29, 32), yielding the unconditional explicit formula (19). The parity obstruction (Proposition 5.4) is proved unconditionally. Proposition 10.3 provides an unconditional equivalence converting Conjecture 5.2 into an analytic statement about $\hat{C}(x)$.

What is conditional (GRH and/or HL-B). The convergence rate $|\bar{S}(x) - S_\infty| = O(\log x / \sqrt{x})$ (Theorem 6) and the complete conditional proof of $N(p) \geq 1$ for all $p > 11$ (Theorem 42) require GRH+HL-B. The empirical decomposition of b (Proposition 9.1) identifies the dominant mechanism as arithmetic covariance (82.3%), with Riemann zeros contributing only 17.7%.

The Goldbach–Riemann bridge. The explicit formula (19) connects $\Psi^*(x)$ to the non-trivial zeros of $\zeta(s)$ with residues amplified by $S_\infty \approx 1.743$, a factor structurally new relative to all classical Goldbach–Riemann bridges (Table 10). The detection of 72/100 zeros via discrete Mellin transform ($z_{\max} > 20$ at $\gamma_{11} = 52.97$; $p < 0.01$, $n = 70,435$) confirms the predicted structure from a new arithmetic direction.

What requires fundamentally new ideas. Proving $N(p) \geq 2$ for all $p > 11$ faces the parity obstacle of sieve theory: current methods cannot guarantee two simultaneously prime pairs, a barrier comparable to the gap between Chen's theorem and full Goldbach. The conjecture $\alpha_\infty = 1/S_\infty$ rests on six data points. The non-existence of a closed form for S_∞ remains a numerical observation.

Acknowledgement of status. This paper is experimental and structural number theory. All claims are labelled as proved, conditionally proved, computationally verified, or conjectural. The central conjectures are open. No claim is presented without its epistemic status.

16. Numerical Values and Extended Data

16.1. Extended Table of $\alpha(x)$ Convergence

Table 24 reproduces the full two-term model verification of Theorem 21 in extended form, including predictions at scales not yet computationally verified.

Table 24. Full verification of $\alpha(x) = 1/S_\infty + 0.1311/\log x + 1.617/\log^2 x$ including predictions at unverified scales. [NEW] [COND. PROVED] (asymptotic form); [COMP. VERIF.] (coefficients).

x_{\max}	$\hat{\alpha}$ observed	Model prediction	Error
10^3	0.6143	0.6267	-0.0124
10^4	0.6298	0.6071	+0.0227
10^5	0.6103	0.5974	+0.0129
10^6	0.5911	0.5918	-0.0007
10^7	0.5761	0.5882	-0.0121
6.79×10^7	0.5682	0.5861	-0.0179
10^8 (predicted)	—	0.5857	—
10^9 (predicted)	—	0.5844	—
10^{10} (predicted)	—	0.5826	—
10^{18} (predicted)	—	0.5762	—
∞	—	$0.5738 = 1/S_\infty$	—

16.2. Bootstrap Confidence Interval for α

The parameter α is given a formal confidence interval via bootstrap resampling of the Phase 2 data (10 000 primes at 20–24 bits, exact $N(p)$). The bootstrap procedure resamples the index set with replacement $n_{\text{iter}} = 2,000$ times and computes $\hat{\alpha}$ for each resample.

Results (Phase 2, $n = 10,000$ primes, $n_{\text{iter}} = 2,000$):

$$\hat{\alpha} = 0.580045, \quad \text{CI}_{95\%} = [0.579922, 0.580174].$$

Both $\alpha = 1/2 = 0.500$ and $1/S_\infty = 0.5738$ lie outside this interval: $\alpha = 1/2$ is excluded with 95% confidence; the gap to $1/S_\infty$ is consistent with slow $O(1/\log x)$ convergence (Theorem 21). Under the two-term model of equation (13), the predicted $\alpha(x)$ reaches 0.573 at approximately $x \sim 10^{22}$, where the bootstrap interval would be expected to overlap with $1/S_\infty$.

16.2. Independent Verification of the Euler Product Tail Bound

All five cases $Q \in \{10^2, 10^3, 10^4, 10^5, 10^6\}$ satisfy the bound from Lemma 3.4, with errors ranging from 3.22×10^{-3} to 6.17×10^{-8} , well within $8S_\infty/Q$ (Table 3). This constitutes an independent computational confirmation of Lemma 3.4.

Table 25. Detailed error analysis for the tail bound verification. [COMP. VERIF.].

Q	Primes $\leq Q$	S_Q	$ S_\infty - S_Q $	Bound $8S_\infty/Q$
10^2	24	1.7395051586	3.22×10^{-3}	1.39×10^{-1}
10^3	168	1.7425037651	2.22×10^{-4}	1.39×10^{-2}
10^4	1 228	1.7427083031	1.71×10^{-5}	1.39×10^{-3}
10^5	9 591	1.7427240136	1.34×10^{-6}	1.39×10^{-4}
10^6	78 498	1.7427252937	6.17×10^{-8}	1.39×10^{-5}

16.4. Zero-Sum Residue Convergence: Tail Estimate

From Table 11, the cumulative sum of the first 50 zero contributions is -0.027769 . The tail estimate for $|y| > 144$ (the imaginary part of the 50th non-trivial zero) is < 0.00577 . This gives

$$b_{\text{zeros}} = 2C_2S_\infty \times 0.027769 = 2 \times 0.6602 \times 1.74273 \times 0.027769 \approx 0.0639,$$

confirming the quantitative decomposition of Proposition 9.1 at 40-digit precision.

16.5. Verification Script Python

16.5.1. Overview

The unified verification script.py, 962 lines, implements four independent computational phases covering all quantitative results reported in Sections 8–9. It is written in Python 3.8 with optional Numba JIT-compilation and SymPy support. The script is designed to be fully reproducible: all random seeds can be fixed, all intermediate results are saved as JSON and NumPy arrays, and checkpoint recovery allows resumption after interruption.

16.5.2. Block-by-Block Description

Lines 1–48: Preamble and constants. Package imports, optional detection of Numba and SymPy. Definition of numerical constants: $C_2 = 0.6601618158468696$, $2C_2 = 1.32032363$, $S_\infty = 1.7427253553918328$, $\alpha_{\infty, \text{conj}} = 1/S_\infty = 0.5738 \dots$. The array GAMMA_200 contains the imaginary parts of the first 200 non-trivial Riemann zeros to 18 decimal places, sourced from standard tables of the zeta function.

Lines 49–130: Utility functions. `ensure_dir(path)`: creates output directories. `build_sieve(limit)`: generates a Boolean bytearray sieve of Eratosthenes up to limit (≈ 95 MB for 10^8), with timing output. `compute_S_vectorized(primes_a, all_primes)`: computes the singular factor $S(p+1)$ for an array of primes using vectorised small-prime factorisation over all odd primes $\leq 9,999$, followed by large-residue correction.

Lines 131–302: Exhaustive $N(p)$ computation (Phase 1 core). `_compute_N_chunk`: Numba-JIT inner loop (or pure Python fallback) counting Goldbach representations for a chunk of primes via sieve lookup. For each prime p in the chunk, iterates over candidate primes $q \leq (p+1)/2$ and checks whether $r = p+1-q$ is prime by $O(1)$ lookup. `worker_N`: multiprocessing worker function receiving chunks from a shared work queue. `compute_N_exhaustive`: orchestrates parallel computation with checkpoint recovery in `checkpoints_exhaustive/`.

Lines 303–430: Statistics and plotting (Phase 1 output). `compute_stats`: computes \hat{C} , \bar{S} , $\hat{\alpha}$, RMSE, coverage ($\pm 30\%$, $\pm 50\%$), class fractions (Mirror, Anchor-3, Orphan), and decade minima of $N(p)$. `plot_histogram`: frequency histogram of $N(p)$ with log-normal fit. `plot_decade_minima`: bar chart of $\min N(p)$ by decade (log scale) with growth factor annotations. `plot_alpha_convergence`: line plot of $\hat{\alpha}$ vs. prime limit with reference lines at $1/S_\infty$ and $1/2$.

Lines 431–519: Phase 1 driver. `phase_exhaustive(limit)`: runs the complete exhaustive verification pipeline; saves JSON statistics and compressed NumPy arrays; calls all three plotting functions.

Lines 520–694: Phase 2—Probabilistic large-scale verification. `miller_rabin(n, k)`: Miller–Rabin primality test with k witness rounds. `generate_prime(bits, k)`: generates a random prime of given bit-length using Miller–Rabin. `find_two_representations(p)`: searches for a second Goldbach pair (q_2, r_2) with $q_2 + r_2 = p + 1$ by first testing small primes, then sampling uniformly from the valid range $[3, (p + 1)/2)$. `compute_S_large`: computes $S(p + 1)$ for large primes via trial division by small primes ≤ 199 ; contributions from large residues ($> 10^{12}$) are set to 1 to avoid overflow. `classify_large`: classifies p as Mirror, Anchor-3, or Orphan using Miller–Rabin. `phase_probabilistic_10e38(n_samples, bits)`: runs Phase 2, saves `stats_{bits}bits.json` and optional bar chart.

Lines 695–780: Phase 3—RSA scales. `phase_rsa_scales(samples_per_scale)`: generates 50 primes at each of RSA-1024, RSA-2048, and RSA-4096 bit-lengths (RSA-4096 may not complete within available time); computes \bar{S} and class fractions; saves results to `results_rsa_scales/results.json` and two plots (\bar{S} vs. bits, class fractions vs. bits).

Lines 781–875: Phase 4—Riemann zero detection. `phase_riemann_zero_detection(p_start, p_end, n_zeros)`: builds sieve, computes exact $N(p)$ for all primes in $[p_{\text{start}}, p_{\text{end}}]$, computes $\varepsilon(p)$, evaluates the discrete Mellin transform M_k for $k = 1, \dots, n_{\text{zeros}}$ (up to 200), performs 200 global permutation tests to establish the noise baseline, computes z -scores and p -values, saves results to `riemann_results/` as JSON and two plots (z -score bar chart and p -value log-scale plot).

Lines 876–962: Interactive main menu. Text-based menu allowing the user to select any phase, set parameters (limit, bits, number of samples, prime range, number of zeros), and run phases sequentially.

16.5.3. Reproducibility Notes

The complete script listing is available as supplementary material. The source file `script.py` (962 lines) may be inserted directly below this point as a verbatim listing. Hardware: AMD64, 2 physical cores, 3.46 GB RAM, Windows 7, Python 3.8. Estimated runtimes on this hardware: Phase 1 to 10^7 : ≈ 30 min; Phase 1 to 6.79×10^7 : ≈ 6 h; Phase 2 (1 000 samples, 127-bit): ≈ 10 min; Phase 4 (70 435 primes, 100 zeros, 200 permutations): ≈ 2 h.

17. Script Python 3.8.7

```
# =====
# # Unified verification for Anderson (2026) – complete set of computations:
# Phase 1: Exhaustive verification (parallel, Numba, checkpoint)
# + plots: histogram, decade minima, alpha convergence
# Phase 2: Probabilistic verification at 127-512 bits + optional bar chart
# + optionally compute S_bar and class fractions (to align with Phase 3)
# Phase 3: RSA scales (1024/2048/4096 bits): estimate S(p+1) average and class fractions
# + plots: S_bar vs bits, Mirror/Anchor-3 fractions vs bits
# Phase 4: Riemann zero detection in epsilon(p) (up to 200 zeros) + z-score and p-value plots
# =====

import os
import sys
import time
import json
import math
import random
```

```

import numpy as np
import multiprocessing as mp
from ctypes import c_int64, c_bool
import matplotlib.pyplot as plt

try:
from numba import njit
NUMBA_AVAILABLE = True
except ImportError:
NUMBA_AVAILABLE = False
print("[WARN] numba not installed. Install for speed: pip install numba")
def njit(f): return f

try:
from sympy import isprime as sympy_isprime, nextprime as sympy_nextprime, prevprime as
sympy_prevprime
SYMPY_AVAILABLE = True
except ImportError:
SYMPY_AVAILABLE = False
print("[WARN] sympy not installed. RSA phase will be slower and limited.")

# =====
# Constants
# =====

C2 = 0.6601618158468696
TWO_C2 = 2 * C2
S_INF = 1.7427253553918328
ALPHA_INF_CONJ = 1.0 / S_INF

# First 200 non-trivial zeros (imaginary parts) from known tables
GAMMA_200 = np.array([
14.134725141734693790, 21.022039638771554993, 25.010857580145688763,
30.424876125859513210, 32.935061587739189691, 37.586178158825671257,
40.918719012147495187, 43.327073280914999519, 48.005150881167159728,
49.773832477672302182, 52.970321477714460644, 56.446247697063394804,
59.347044002602353079, 60.831778524609809844, 65.112544048081606660,
67.079810529494173714, 69.546401711173979252, 72.067157674481907583,
75.704690699083933168, 77.144840068874805373, 79.337375020249367922,
82.910380854086419494, 84.735492980517944342, 87.425274613125484385,
88.809111207634078295, 92.491899270558393445, 94.651344040519966296,
95.870634228245100676, 98.831194218154647771, 101.317851005731359128,
103.725538040478496151, 105.446623052332139824, 107.168611184547512718,
109.333415286312632280, 111.029535543058416172, 112.948105007737889636,
114.936519236288949726, 116.226680320065765313, 118.790782866128076055,

```

121.370125002122158041, 122.946829293659598744, 124.256818554345157982,
127.516683879443672573, 129.578704199956048450, 131.087688530932730187,
133.497737202997519744, 134.756509753525313045, 138.116042054533222112,
139.736208952121396713, 141.123707404021137350, 143.111845807620590275,
146.000982486771652810, 147.422765343731659747, 150.053520420677144901,
150.925257611418143090, 153.024693811796247402, 156.112909288629071952,
157.597591235046217562, 158.849988738993101257, 161.188964136063822527,
163.030709562472751867, 165.537069187388736543, 167.184439072157705648,
169.094515694588222512, 169.911976479010507729, 173.411534131323027049,
174.754191521203717248, 176.441434528214457251, 178.377407710091689986,
179.916484013815138825, 182.207065696367917673, 184.874467748503760979,
185.598783678848901320, 187.228922897848025788, 189.416158739034808324,
192.026656362955215567, 193.079726608649474848, 195.265396934283938510,
196.876481281228532052, 198.015309682682571249, 201.264751562355525939,
202.493594514928002567, 204.189171809036253110, 205.394693249391084706,
207.906258878873659097, 209.576509714716128024, 211.690862550125132970,
213.347819360297605394, 214.475102041831637118, 216.661959518910707580,
218.787557909624020881, 220.010451436425577729, 221.962661780554385401,
223.637436674563482208, 225.631369742380975582, 227.128195946253098083,
228.669395751931371466, 230.619565446488142848, 232.270249227058096778,
233.794646626301497672, 235.655330689102971606, 237.227428044076825862,
238.657054106665028401, 240.597579310252330237, 242.388194793152186882,
243.952168330866813334, 245.595047441800273176, 247.270562507297293065,
248.812592562970211030, 250.228075270217066775, 251.687795718001369680,
253.442359342480281230, 255.215570326081526677, 256.683468466768875251,
258.360541595758658243, 259.924613984570195815, 261.745886864284053556,
263.223756131717551490, 264.747926097164596303, 266.550649550286930492,
268.018288970430318601, 269.448978409762598276, 271.371504503754371715,
272.980270664386080630, 274.658474285311602510, 276.333151266655756263,
277.713534092639224138, 279.457861243173114174, 280.990303102009278098,
282.526213720594053896, 284.256149594656890462, 285.881304546141721843,
287.266439957396404149, 288.990303955160801363, 290.743601051930711953,
292.336211912286758845, 293.888234921427760576, 295.489478500355961266,
297.112083366898511223, 298.631122681139273257, 300.254735356217215632,
301.898291322173405065, 303.480372722178400984, 304.984373103854993948,
306.611098561598461541, 308.228578736636023301, 309.865590506574064396,
311.409065095227352772, 312.958396277825718207, 314.588159128169683186,
316.184748932414671039, 317.767797569012508809, 319.333451822733034088,
320.982179572404269123, 322.614173845050761110, 324.182640648708615098,
325.745612149489734092, 327.346622727958625537, 328.970164392959827641,
330.527211289470087757, 332.168182846924886559, 333.753162058167868025,
335.348188593501365831, 336.961375422666104519, 338.562087934525113735,
340.164399997974342245, 341.771801655876417623, 343.356072160299814540,
344.948398457288074705, 346.569393508222255258, 348.169528391439111342,

349.763964802763847434, 351.361247187529694210, 352.952848573980510942,
 354.527545732162147625, 356.148820631701850894, 357.724352038326903050,
 359.346108893166868634, 360.951278327646883621, 362.562293296550534674,
 364.173552436998838328, 365.764316901735722936, 367.352392731042519660,
 368.953562217912312213, 370.560280843393917589, 372.167560708851206075,
 373.771413837507144752, 375.368976901814142445, 376.968910551646645598,
 378.582795266915706246, 380.182466299153110038, 381.788843001980190684,
 383.394554295341689222, 385.002530749090327548, 386.599064353619353261,
 388.211780318531094071, 389.818702369754762133, 391.421429969497411229,
 393.027110517784569138, 394.639875120576159338, 396.244829763454777515,
 397.852376223488025062, 399.462795391553867297, 401.071618962370059792,
 402.678641574094627770, 404.287374413391446162, 405.893127342971456455,
 407.503169849746834701, 409.111222786801032741, 410.721753060446577417,
 412.329590504281708313, 413.939478951296710469, 415.547392244526652748,
 417.158158973534367345, 418.767387942216931117, 420.375234456075165747,
 421.987168256159481962, 423.595887699635128268, 425.206006890044124083,
 426.813288319733574845, 428.424899575700890624, 430.032928177584870069,
 431.644544455838330127, 433.252707238513965722, 434.862324657725479783,
 436.470903105956495424, 438.082543498929032512, 439.690527924984631886,
 441.301020208117759402, 442.909306167729692443, 444.519815266238088365,
 446.127863289866668440, 447.738427235521922284, 449.346553801165417941,
 450.957436195429346208, 452.565856711377277183, 454.176308870195031713
)

```
# =====
# Utility
# =====

def ensure_dir(path):
    os.makedirs(path, exist_ok=True)

# =====
# Sieve
# =====

def build_sieve(limit):
    print(f" Building sieve up to {limit:,}...", end="", flush=True)
    t0 = time.time()
    s = bytearray(b'\x01') * (limit + 1)
    s[0] = s[1] = 0
    i = 2
    while i * i <= limit:
        if s[i]:
            s[i*i::i] = b'\x00' * len(s[i*i::i])
        i += 1
    is_prime = np.frombuffer(s, dtype=np.uint8).astype(np.bool_)
```

```
print(f" done in {time.time()-t0:.2f}s, {is_prime.nbytes/1e6:.1f} MB")
return is_prime
```

```
# =====
# N(p) computation with Numba (parallel)
# =====
if NUMBA_AVAILABLE:
    @njit
    def _compute_N_chunk(primes_chunk, all_primes, lookup):
        n = len(primes_chunk)
        N = np.zeros(n, dtype=np.int32)
        for idx in range(n):
            p = primes_chunk[idx]
            target = p + 1
            half = target // 2
            lo, hi = 0, len(all_primes)
            while lo < hi:
                mid = (lo + hi) // 2
                if all_primes[mid] <= half:
                    lo = mid + 1
                else:
                    hi = mid
            cut = lo
            count = 0
            for j in range(cut):
                r = target - all_primes[j]
                if r >= 2 and r < len(lookup) and lookup[r]:
                    count += 1
            N[idx] = count
        return N
    else:
        def _compute_N_chunk(primes_chunk, all_primes, lookup):
            n = len(primes_chunk)
            N = np.zeros(n, dtype=np.int32)
            for idx, p in enumerate(primes_chunk):
                target = p + 1
                half = target // 2
                cut = np.searchsorted(all_primes, half, side='right')
                qs = all_primes[:cut]
                rs = target - qs
                valid = (rs >= 2) & (rs < len(lookup))
                N[idx] = np.sum(lookup[rs[valid]])
            return N
```

```

def worker_N(worker_id, work_queue, result_queue,
shared_primes, n_primes, shared_lookup, primes_a_list):
all_primes = np.frombuffer(shared_primes, dtype=np.int64)[:n_primes]
lookup = np.frombuffer(shared_lookup, dtype=np.bool_)
primes_a = np.array(primes_a_list, dtype=np.int64)
while True:
try:
item = work_queue.get(timeout=10)
except Exception:
break
if item is None:
break
chunk_id, start_idx, end_idx = item
chunk = primes_a[start_idx:end_idx]
N_chunk = _compute_N_chunk(chunk, all_primes, lookup)
violations = []
for p, n in zip(chunk, N_chunk):
if p > 11 and n < 2:
violations.append((int(p), int(n)))
result_queue.put((chunk_id, start_idx, end_idx, N_chunk, violations))

def compute_N_exhaustive(primes_a, all_primes, lookup, label, n_cores=2, chunk_size=50000):
n_primes = len(primes_a)
n_chunks = (n_primes + chunk_size - 1) // chunk_size
print(f" Computing N(p) with {n_cores} cores, chunk size={chunk_size};")
print(f" Total chunks: {n_chunks};")

ckpt_dir = "checkpoints_exhaustive"
ensure_dir(ckpt_dir)
ckpt_file = os.path.join(ckpt_dir, f"progress_{label}.json")
n_partial_file = os.path.join(ckpt_dir, f"N_partial_{label}.npy")

done_chunks = set()
N = np.zeros(n_primes, dtype=np.int32)
violations = []

if os.path.exists(ckpt_file) and os.path.exists(n_partial_file):
try:
with open(ckpt_file, 'r') as f:
data = json.load(f)
done_chunks = set(data.get('done_chunks', []))
N = np.load(n_partial_file)
violations = data.get('violations', [])
print(f" Resuming: {len(done_chunks)}/{n_chunks} chunks done")

```

```

except Exception as e:
print(f" Checkpoint error: {e}, starting fresh")

pending = []
for i in range(n_chunks):
if i not in done_chunks:
start = i * chunk_size
end = min((i+1)*chunk_size, n_primes)
pending.append((i, start, end))

if not pending:
print(" All chunks already processed.")
return N, violations

# Shared memory
shared_primes = mp.Array(c_int64, len(all_primes), lock=False)
np.frombuffer(shared_primes, dtype=np.int64)[:]= all_primes
shared_lookup = mp.Array(c_bool, len(lookup), lock=False)
np.frombuffer(shared_lookup, dtype=np.bool_)[:]= lookup

work_queue = mp.Queue()
for item in pending:
work_queue.put(item)
for _ in range(n_cores):
work_queue.put(None)

result_queue = mp.Queue()
primes_a_list = primes_a.tolist()
processes = []
for i in range(n_cores):
p = mp.Process(target=worker_N,
args=(i, work_queue, result_queue,
shared_primes, len(all_primes),
shared_lookup, primes_a_list))
p.start()
processes.append(p)

received = 0
total_pend = len(pending)
t_start = time.time()
while received < total_pend:
chunk_id, start_idx, end_idx, N_chunk, chunk_viol = result_queue.get()
N[start_idx:end_idx] = N_chunk
done_chunks.add(chunk_id)

```

```

violations.extend(chunk_viol)
received += 1
elapsed = time.time() - t_start
eta = (elapsed / received) * (total_pend - received) if received else 0
pct = len(done_chunks) / n_chunks * 100
print(f" [{pct:.1f}%] {len(done_chunks)}/{n_chunks} chunks | “
f”{elapsed/60:.1f}min | ETA {eta/60:.1f}min | Violations: {len(violations)}”)
with open(ckpt_file, 'w') as f:
json.dump({'done_chunks': list(done_chunks), 'violations': violations,
'timestamp': time.strftime('%Y-%m-%d %H:%M:%S')}, f)
np.save(n_partial_file, N)

for p in processes:
p.join()

elapsed = time.time() - t_start
print(f" N(p) finished in {elapsed:.1f}s ({elapsed/60:.1f}min)")
return N, violations

# =====
# S(p+1) vectorized (for moderate primes)
# =====

def compute_S_vectorized(primes_a, all_primes):
print("\n Computing S(p+1) vectorized...")
t0 = time.time()
ns = (primes_a + 1).astype(np.int64)
result = np.ones(len(ns), dtype=np.float64)
ms = ns.copy()

# Remove all factors 2
while True:
mask2 = (ms % 2) == 0
if not mask2.any():
break
ms[mask2] //= 2

small_primes = all_primes[(all_primes > 2) & (all_primes < 10_000)]
n_small = len(small_primes)
for idx, pp in enumerate(small_primes):
if pp * pp > ms.max():
break
mask = (ms % pp) == 0
if mask.any():
result[mask] *= (float(pp) - 1.0) / (float(pp) - 2.0)

```

```

while True:
still = mask & ((ms % pp) == 0)
if not still.any():
break
ms[still] //= pp
if idx % 200 == 0 and idx > 0:
pct = idx / n_small * 100
print(f" Small primes: {idx}/{n_small} ({pct:.0f}%)...")
leftover = ms > 1
if leftover.any():
lp = ms[leftover].astype(np.float64)
result[leftover] *= (lp - 1.0) / (lp - 2.0)
elapsed = time.time() - t0
print(f" S(p+1) computed in {elapsed:.1f}s")
return result

# =====
# Classification Mirror/Anchor-3/Orphan (for moderate primes)
# =====
def classify(primes_a, lookup):
is_mirror = np.zeros(len(primes_a), dtype=bool)
is_anchor3 = np.zeros(len(primes_a), dtype=bool)
for i, p in enumerate(primes_a):
if p <= 5:
continue
q = (p + 1) // 2
if q < len(lookup) and lookup[q]:
is_mirror[i] = True
if p > 5 and p-2 < len(lookup) and lookup[p-2]:
is_anchor3[i] = True
return is_mirror, is_anchor3

# =====
# Statistics and plotting for phase 1
# =====
def compute_stats(primes_a, N, Sf, is_mirror, is_anchor3, label):
ps = primes_a.astype(np.float64)
lps = np.log(ps)
C_hat = float(np.mean(N * lps**2 / ps))
S_bar = float(np.mean(Sf))
alpha = C_hat / (TWO_C2 * S_bar)
N_law3 = alpha * TWO_C2 * Sf * ps / lps**2
eps3 = (N - N_law3) / N_law3
rmse3 = float(np.sqrt(np.mean(eps3**2)))

```

```

cov30 = float(np.mean(np.abs(eps3) < 0.30) * 100)
cov50 = float(np.mean(np.abs(eps3) < 0.50) * 100)

# Decade minima
decade_min = []
decade_start = 12
k = 1
while decade_start < primes_a[-1]:
    decade_end = 10 ** (k + 1)
    mask = (primes_a >= decade_start) & (primes_a < decade_end)
    if mask.any():
        sub_N = N[mask]
        sub_p = primes_a[mask]
        min_val = int(sub_N.min())
        min_p = int(sub_p[sub_N.argmin()])
        decade_min.append({
            'range': f"({decade_start},{decade_end})",
            'n': int(mask.sum()),
            'min_N': min_val,
            'p_min': min_p
        })
        decade_start = decade_end
    k += 1

stats = {
    'label': label,
    'n_primes': int(len(primes_a)),
    'min_N': int(N.min()),
    'max_N': int(N.max()),
    'violations': int(np.sum(N < 2)),
    'C_hat': C_hat,
    'S_bar': S_bar,
    'alpha': alpha,
    'rmse3': rmse3,
    'coverage_30': cov30,
    'coverage_50': cov50,
    'mirror_count': int(is_mirror.sum()),
    'mirror_pct': float(is_mirror.sum() / len(primes_a) * 100),
    'anchor3_count': int(is_anchor3.sum()),
    'anchor3_pct': float(is_anchor3.sum() / len(primes_a) * 100),
    'orphan_count': int(len(primes_a) - is_mirror.sum() - is_anchor3.sum()),
    'orphan_pct': float((len(primes_a) - is_mirror.sum() - is_anchor3.sum()) / len(primes_a) * 100),
    'decade_minima': decade_min
}

```

```
return stats, eps3
```

```
def plot_histogram(N, label, out_dir):
    plt.figure(figsize=(8,5))
    plt.hist(N, bins=50, edgecolor='black', alpha=0.7)
    plt.xlabel('N(p)')
    plt.ylabel('Frequency')
    plt.title(f'Distribution of N(p) for {label}')
    plt.grid(True, alpha=0.3)
    plt.savefig(os.path.join(out_dir, 'histogram.png'), dpi=150)
    plt.close()
```

```
def plot_decade_minima(decade_min, label, out_dir):
    if not decade_min:
        return
    ranges = [d['range'] for d in decade_min]
    mins = [d['min_N'] for d in decade_min]
    plt.figure(figsize=(10,5))
    plt.bar(ranges, mins, color='steelblue', alpha=0.7)
    plt.xlabel('Decade')
    plt.ylabel('Minimum N(p)')
    plt.title(f'Minimum N(p) by decade for {label}')
    plt.xticks(rotation=45)
    plt.grid(True, axis='y', alpha=0.3)
    plt.tight_layout()
    plt.savefig(os.path.join(out_dir, 'decade_minima.png'), dpi=150)
    plt.close()
```

```
def plot_alpha_convergence(stats_list, out_dir):
    if not stats_list:
        return
    labels = [s['label'] for s in stats_list]
    alphas = [s['alpha'] for s in stats_list]
    # Extract numeric limits from labels (e.g., "p≤100K" -> 100000)
    def parse_limit(label):
        if 'M' in label:
            return float(label.split('≤')[1].replace('M','')) * 1e6
        elif 'K' in label:
            return float(label.split('≤')[1].replace('K','')) * 1e3
        else:
            return 1e6
    limits = [parse_limit(l) for l in labels]
    plt.figure(figsize=(8,5))
    plt.plot(limits, alphas, 'o-', label='Observed α')
```

```
plt.axhline(y=ALPHA_INF_CONJ, color='r', linestyle='--', label=r'$1/S_\infty = 0.5738$')
plt.xscale('log')
plt.xlabel('Prime limit (p_max)')
plt.ylabel('$\alpha$')
plt.title('Convergence of $\alpha$ to $1/S_\infty$')
plt.legend()
plt.grid(True, alpha=0.3)
plt.savefig(os.path.join(out_dir, 'alpha_convergence.png'), dpi=150)
plt.close()
```

```
# =====
# Phase 1: Exhaustive verification
# =====

def phase_exhaustive(limit, n_cores=2, stats_list=None):
    label = f"p≤{limit/1e3:.0f}K" if limit < 1e6 else f"p≤{limit/1e6:.1f}M"
    print("\n" + "="*60)
    print(f" PHASE 1: EXHAUSTIVE VERIFICATION ({label})")
    print("="*60)

    lookup = build_sieve(limit)
    all_primes = np.where(lookup)[0].astype(np.int64)
    primes_a = all_primes[all_primes > 11]
    print(f" Primes to analyze: {len(primes_a):,}")

    N, violations = compute_N_exhaustive(primes_a, all_primes, lookup, label, n_cores)
    if violations:
        print(f" WARNING: {len(violations)} counterexamples found!")
    else:
        print(" No counterexamples found. Conjecture 4.1 holds for this range.")

    Sf = compute_S_vectorized(primes_a, all_primes)
    is_mirror, is_anchor3 = classify(primes_a, lookup)
    stats, eps3 = compute_stats(primes_a, N, Sf, is_mirror, is_anchor3, label)

    out_dir = f"results_exhaustive_{label.replace('≤','')}"
    ensure_dir(out_dir)
    with open(os.path.join(out_dir, "stats.json"), 'w') as f:
        json.dump(stats, f, indent=2)
    np.savez_compressed(os.path.join(out_dir, "data.npz"),
        primes=primes_a, N=N, Sf=Sf, eps3=eps3)
    plot_histogram(N, label, out_dir)
    plot_decade_minima(stats['decade_minima'], label, out_dir)

    print("\n SUMMARY")
```

```

print(f" Primes tested: {stats['n_primes'];}")
print(f" min N(p) = {stats['min_N']}, max N(p) = {stats['max_N']}")
print(f" C_hat = {stats['C_hat']:.6f} (2C2 = {TWO_C2:.6f})")
print(f" S_bar = {stats['S_bar']:.6f} (S $\infty$  = {S_INF:.6f})")
print(f" alpha = {stats['alpha']:.6f} (1/S $\infty$  = {ALPHA_INF_CONJ:.6f})")
print(f" RMSE Law3 = {stats['rmse3']:.6f}, coverage  $\pm$ 30% = {stats['coverage_30']:.2f}%")
print(f" Mirror: {stats['mirror_count'];} ({stats['mirror_pct']:.2f}%)")
print(f" Anchor-3: {stats['anchor3_count'];} ({stats['anchor3_pct']:.2f}%)")
print(f" Orphan: {stats['orphan_count'];} ({stats['orphan_pct']:.2f}%)")

```

```

# Update global stats list for alpha convergence plot (if provided)

```

```

if stats_list is not None:

```

```

    stats_list.append(stats)

```

```

return stats

```

```

# =====

```

```

# Phase 2: Probabilistic at 1038 (robust) – optional bar chart and statistics

```

```

# =====

```

```

def miller_rabin(n, k=15):

```

```

    if n < 2: return False

```

```

    if n in (2,3,5,7,11,13,17,19,23,29,31,37): return True

```

```

    if n % 2 == 0: return False

```

```

    r, d = 0, n-1

```

```

    while d % 2 == 0:

```

```

        r += 1

```

```

        d //= 2

```

```

    for _ in range(k):

```

```

        a = random.randrange(2, n-1)

```

```

        x = pow(a, d, n)

```

```

        if x == 1 or x == n-1:

```

```

            continue

```

```

        for _ in range(r-1):

```

```

            x = pow(x, 2, n)

```

```

            if x == n-1:

```

```

                break

```

```

        else:

```

```

            return False

```

```

    return True

```

```

def generate_prime(bits, k=20):

```

```

    while True:

```

```

        n = random.getrandbits(bits) | (1 << (bits-1)) | 1

```

```

        if miller_rabin(n, k):

```

```

            return n

```

```

def find_two_representations(p):
    target = p + 1
    half = target // 2
    small_primes = [2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,53,59,61,67,71,73,79,83,89,97]
    reps = []
    for q in small_primes:
        if q > half: break
        r = target - q
        if r >= 2 and miller_rabin(r, k=15):
            reps.append((q, r))
        if len(reps) >= 2:
            return True
    attempts = 0
    while attempts < 20000 and len(reps) < 2:
        q = random.randrange(2, half+1)
        r = target - q
        if miller_rabin(q, k=15) and miller_rabin(r, k=15):
            reps.append((q, r))
        attempts += 1
    return len(reps) >= 2

```

```

def compute_S_large(n, small_primes):
    """Compute S(n) for large n using trial division by small primes.
    For huge residues, the factor (m-1)/(m-2) is extremely close to 1,
    so we ignore it to avoid overflow."""
    result = 1.0
    m = n
    # Remove factors 2 (they don't contribute)
    while m % 2 == 0:
        m //= 2
    for p in small_primes:
        if p * p > m:
            break
        if m % p == 0:
            result *= (p - 1.0) / (p - 2.0)
            while m % p == 0:
                m //= p
            if m > 1:
                # For extremely large m, the factor (m-1)/(m-2) is essentially 1.
                # We set a threshold: if m > 10^12, we ignore it to avoid overflow.
                # For numbers up to that threshold, we compute normally.
                if m < 1_000_000_000_000:
                    result *= (m - 1.0) / (m - 2.0)

```

```

# else: contribution is negligible (difference < 1e-12)
return result

def classify_large(p):
    """Mirror/Anchor-3 classification using Miller-Rabin."""
    if p <= 5:
        return "Orphan"
    # Mirror: (p+1)//2 prime
    q = (p + 1) // 2
    is_mirror = miller_rabin(q, k=20)
    # Anchor-3: p-2 prime
    is_anchor3 = miller_rabin(p-2, k=20) if p > 5 else False
    if is_mirror:
        return "Mirror"
    elif is_anchor3:
        return "Anchor-3"
    else:
        return "Orphan"

def phase_probabilistic_10e38(n_samples=50, bits=127, plot=False, compute_stats=False):
    print("\n" + "*" * 60)
    print(f" PHASE 2: PROBABILISTIC VERIFICATION at ~{bits} bits")
    print(f" Samples: {n_samples}")
    if compute_stats:
        print(" Also computing S_bar and class fractions (like Phase 3).")
    print(" " * 60)

    # Small primes for factoring S(p+1) (same as in Phase 3)
    small_primes = [2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,53,59,61,67,71,73,79,83,89,97]
    small_primes += [101,103,107,109,113,127,131,137,139,149,151,157,163,167,173,179,181,191,193,197,199]

    successes = []
    S_vals = [] if compute_stats else None
    classes = [] if compute_stats else None

    for i in range(n_samples):
        p = generate_prime(bits, k=20)
        ok = find_two_representations(p)
        successes.append(ok)

    if compute_stats:
        s_val = compute_S_large(p+1, small_primes)
        cls = classify_large(p)
        S_vals.append(s_val)

```

```

classes.append(cls)

if (i+1) % 10 == 0:
    print(f" Sample {i+1}/{n_samples} done")

success_count = sum(successes)
print(f"\n Completed {n_samples} samples.")
print(f" Primes with at least two representations: {success_count}/{n_samples}")
if success_count == n_samples:
    print(" ✓ Conjecture 4.1 holds for all tested large primes (probabilistic).")
else:
    print(" ✗ Some primes missing second representation!")

# Optional plot of success per sample
if plot:
    out_dir = "results_prob_10e38"
    ensure_dir(out_dir)
    plt.figure(figsize=(6,4))
    plt.bar(range(1, n_samples+1), successes, color='green', alpha=0.7)
    plt.xlabel('Sample')
    plt.ylabel('Success (1=found two representations)')
    plt.title(f'Success per sample ({bits} bits)')
    plt.ylim(0, 1.1)
    plt.savefig(os.path.join(out_dir, f'success_{bits}bits.png'), dpi=150)
    plt.close()
    print(f" Plot saved to {out_dir}/success_{bits}bits.png")

# If statistics requested, compute and save
if compute_stats and S_vals:
    S_bar = np.mean(S_vals)
    n_mirror = classes.count("Mirror")
    n_anchor3 = classes.count("Anchor-3")
    n_orphan = n_samples - n_mirror - n_anchor3

    print("\n --- Additional statistics ---")
    print(f" S_bar = {S_bar:.6f} (S $\infty$  = {S_INF:.6f})")
    print(f" Mirror: {n_mirror} ({100*n_mirror/n_samples:.2f}%)")
    print(f" Anchor-3: {n_anchor3} ({100*n_anchor3/n_samples:.2f}%)")
    print(f" Orphan: {n_orphan} ({100*n_orphan/n_samples:.2f}%)")

# Save JSON with these stats
out_dir = "results_prob_10e38"
ensure_dir(out_dir)
stats_data = {

```

```

'bits': bits,
'n_samples': n_samples,
'success_count': success_count,
'S_bar': S_bar,
'mirror_fraction': n_mirror / n_samples,
'anchor3_fraction': n_anchor3 / n_samples,
'orphan_fraction': n_orphan / n_samples,
}
with open(os.path.join(out_dir, f'stats_{bits}bits.json'), 'w') as f:
json.dump(stats_data, f, indent=2)
print(f" Statistics saved to {out_dir}/stats_{bits}bits.json")

return success_count

# =====
# Phase 3: RSA scales – estimate S_bar and class fractions (with overflow fix)
# =====
def phase_rsa_scales(samples_per_scale=50):
print("\n" + "="*60)
print(" PHASE 3: RSA SCALES – ESTIMATING S_bar AND CLASS FRACTIONS")
print(f" Samples per scale: {samples_per_scale}")
print("="*60)

scales = [
(1024, "RSA-1024", 309),
(2048, "RSA-2048", 617),
(4096, "RSA-4096", 1233)
]
# Small primes for factoring S(p+1) (up to 1000, enough for this purpose)
small_primes = [2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,53,59,61,67,71,73,79,83,89,97]
small_primes += [101,103,107,109,113,127,131,137,139,149,151,157,163,167,173,179,181,191,193,197,199]

results = []
for bits, name, digits in scales:
print(f"\n --- {name} (approx {digits} digits) ---")
S_vals = []
classes = []
for i in range(samples_per_scale):
p = generate_prime(bits, k=20)
# Compute S(p+1)
s_val = compute_S_large(p+1, small_primes)
S_vals.append(s_val)
cls = classify_large(p)
classes.append(cls)

```

```

if (i+1) % 10 == 0:
print(f" Sample {i+1}/{samples_per_scale}")
S_bar = np.mean(S_vals)
n_mirror = classes.count("Mirror")
n_anchor3 = classes.count("Anchor-3")
n_orphan = samples_per_scale - n_mirror - n_anchor3
results.append({
'bits': bits,
'name': name,
'S_bar': S_bar,
'mirror_fraction': n_mirror / samples_per_scale,
'anchor3_fraction': n_anchor3 / samples_per_scale,
'orphan_fraction': n_orphan / samples_per_scale,
'samples': samples_per_scale
})
print(f" S_bar = {S_bar:.6f} (S $\infty$  = {S_INF:.6f})")
print(f" Mirror: {n_mirror} ({100*n_mirror/samples_per_scale:.2f}%)")
print(f" Anchor-3: {n_anchor3} ({100*n_anchor3/samples_per_scale:.2f}%)")
print(f" Orphan: {n_orphan} ({100*n_orphan/samples_per_scale:.2f}%)")

# Save results
out_dir = "results_rsa_scales"
ensure_dir(out_dir)
with open(os.path.join(out_dir, "results.json"), 'w') as f:
json.dump(results, f, indent=2)

# Plot S_bar vs bits
bits_list = [r['bits'] for r in results]
S_list = [r['S_bar'] for r in results]
plt.figure(figsize=(8,5))
plt.plot(bits_list, S_list, 'o-', label='Observed S_bar')
plt.axhline(y=S_INF, color='r', linestyle='--', label=r'$S_{\infty} = 1.74273$')
plt.xlabel('Bits')
plt.ylabel('Average S(p+1)')
plt.title('Convergence of S_bar to S $\infty$  at RSA scales')
plt.legend()
plt.grid(True, alpha=0.3)
plt.savefig(os.path.join(out_dir, 'Sbar_vs_bits.png'), dpi=150)
plt.close()

# Plot class fractions
plt.figure(figsize=(8,5))
plt.plot(bits_list, [r['mirror_fraction'] for r in results], 'o-', label='Mirror')
plt.plot(bits_list, [r['anchor3_fraction'] for r in results], 's-', label='Anchor-3')

```

```

plt.plot(bits_list, [r['orphan_fraction'] for r in results], '^-', label='Orphan')
plt.xlabel('Bits')
plt.ylabel('Fraction')
plt.title('Class fractions at RSA scales')
plt.legend()
plt.grid(True, alpha=0.3)
plt.savefig(os.path.join(out_dir, 'class_fractions.png'), dpi=150)
plt.close()

print(f"\n Results saved in {out_dir}")
return results

# =====
# Phase 4: Riemann zero detection (with p-value log plot)
# =====

def phase_riemann_zero_detection(p_start, p_end, n_zeros, n_cores=2):
    if n_zeros > len(GAMMA_200):
        print(f" Warning: requested {n_zeros} zeros, only {len(GAMMA_200)} available. Using
        {len(GAMMA_200)}.")
    n_zeros = len(GAMMA_200)
    gammas = GAMMA_200[:n_zeros]
    label = f"p_{p_start/1e6:.0f}M_{p_end/1e6:.0f}M_zeros_{n_zeros}"
    print("\n" + "="*60)
    print(f" PHASE 4: RIEMANN ZERO DETECTION IN EPSILON(p)")
    print(f" Prime range: {p_start:,} - {p_end:,}")
    print(f" Number of zeros: {n_zeros}")
    print("="*60)

    lookup = build_sieve(p_end)
    all_primes = np.where(lookup)[0].astype(np.int64)
    primes_a = all_primes[(all_primes >= p_start) & (all_primes <= p_end)]
    n = len(primes_a)
    print(f" Number of primes: {n:,}")

    N, _ = compute_N_exhaustive(primes_a, all_primes, lookup, f"riemann_{label}", n_cores,
    chunk_size=20000)
    Sf = compute_S_vectorized(primes_a, all_primes)

    ps = primes_a.astype(np.float64)
    log_ps = np.log(ps)
    C_hat = np.mean(N * log_ps**2 / ps)
    S_bar = np.mean(Sf)
    alpha_local = C_hat / (TWO_C2 * S_bar)
    Nhat = alpha_local * TWO_C2 * Sf * ps / log_ps**2

```

```

eps_arr = (N - Nhat) / Nhat

print(" Computing Mellin transform...")
w = eps_arr / np.sqrt(ps)
phase = np.outer(gammas, np.log(ps))
Mk_obs = np.sqrt((np.cos(phase) @ w / n)**2 + (np.sin(phase) @ w / n)**2)

print(" Performing permutation test (200 permutations)...")
perm_matrix = np.zeros((200, n_zeros))
for i in range(200):
    eps_perm = np.random.permutation(eps_arr)
    w_perm = eps_perm / np.sqrt(ps)
    perm_matrix[i] = np.sqrt((np.cos(phase) @ w_perm / n)**2 + (np.sin(phase) @ w_perm / n)**2)
    bl_mean = np.mean(perm_matrix, axis=0)
    bl_std = np.std(perm_matrix, axis=0)
    z_scores = np.where(bl_std > 0, (Mk_obs - bl_mean) / bl_std, 0.0)
    p_values = np.mean(perm_matrix >= Mk_obs[np.newaxis, :], axis=0)

out_dir = "riemann_results"
ensure_dir(out_dir)
with open(os.path.join(out_dir, f"results_{label}.json"), 'w') as f:
    json.dump({
        'p_start': p_start, 'p_end': p_end,
        'n_primes': n,
        'n_zeros': n_zeros,
        'alpha_local': alpha_local,
        'Mk_obs': Mk_obs.tolist(),
        'p_values': p_values.tolist(),
        'z_scores': z_scores.tolist(),
        'detected_p05': int(np.sum(p_values < 0.05)),
        'detected_p01': int(np.sum(p_values < 0.01))
    }, f, indent=2)

# Plot z-scores
plt.figure(figsize=(10,5))
plt.bar(range(1, n_zeros+1), z_scores, alpha=0.7)
plt.axhline(y=2, color='r', linestyle='--', label='z=2')
plt.xlabel('Zero index')
plt.ylabel('z-score')
plt.title(f'Riemann zero detection (range {p_start/1e6:.0f}M-{p_end/1e6:.0f}M, {n_zeros} zeros)')
plt.legend()
plt.grid(True, alpha=0.3)
plt.savefig(os.path.join(out_dir, f'z_scores_{label}.png'), dpi=150)
plt.close()

```

```

# Plot p-values (log scale)
plt.figure(figsize=(10,5))
plt.plot(range(1, n_zeros+1), p_values, 'o-', label='p-value')
plt.axhline(y=0.05, color='r', linestyle='--', label='p=0.05')
plt.yscale('log')
plt.xlabel('Zero index')
plt.ylabel('p-value (log scale)')
plt.title('Significance of zero detection')
plt.legend()
plt.grid(True, alpha=0.3)
plt.savefig(os.path.join(out_dir, f'pvalues_{label}.png'), dpi=150)
plt.close()

print(f" Results saved in {out_dir}")
print(f" Detected zeros with p<0.05: {np.sum(p_values < 0.05)}/{n_zeros}")
return

# =====
# Main menu
# =====

def main():
print("\n" + "="*70)
print(" ANDERSON (2026) UNIFIED VERIFICATION v22")
print(" Exhaustive | Large primes (robust) | RSA scales | Riemann zeros")
print("="*70)

# For alpha convergence plot across multiple exhaustive runs
exhaustive_stats = []

while True:
print("\n Available phases:")
print(" 1: Exhaustive verification (p ≤ limit)")
print(" 2: Probabilistic ~10^38 (127-512 bits)")
print(" 3: RSA scales – estimate S_bar and class fractions")
print(" 4: Riemann zero detection (up to 200 zeros)")
print(" all: run all phases sequentially")
print(" q: quit")
choice = input(" Enter choice: ").strip().lower()
if choice == 'q':
break

n_cores = mp.cpu_count()
try:

```

```

n_cores = int(input(f" Number of cores for parallel phases [default=2, max={n_cores}]: ").strip() or "2")
n_cores = min(n_cores, mp.cpu_count())
except:
n_cores = 2
if n_cores > 1:
try:
mp.set_start_method('spawn', force=True)
except RuntimeError:
pass

if choice == '1' or choice == 'all':
print("\n Exhaustive limits:")
print(" 1: 10^5 (100,000) ~1 min")
print(" 2: 10^6 (1,000,000) ~3 min")
print(" 3: 10^7 (10,000,000) ~30 min")
print(" 4: Custom")
lim_choice = input(" Select limit [default=2]: ").strip() or "2"
if lim_choice == '1':
limit = 100_000
elif lim_choice == '2':
limit = 1_000_000
elif lim_choice == '3':
limit = 10_000_000
elif lim_choice == '4':
limit = int(input(" Enter limit (integer): ").strip())
else:
limit = 1_000_000
phase_exhaustive(limit, n_cores, exhaustive_stats)

if choice == '2' or choice == 'all':
n_samples = int(input(" Number of samples for large prime test [default=50]: ").strip() or "50")
bits = int(input(" Bits for primes [default=127, max=512]: ").strip() or "127")
bits = min(bits, 512)
plot_choice = input(" Generate bar chart? (y/n) [default=n]: ").strip().lower()
plot = (plot_choice == 'y')
stats_choice = input(" Compute S_bar and class fractions? (y/n) [default=n]: ").strip().lower()
compute_stats = (stats_choice == 'y')
phase_probabilistic_10e38(n_samples, bits, plot, compute_stats)

if choice == '3' or choice == 'all':
samples = int(input(" Number of samples per RSA scale [default=50]: ").strip() or "50")
phase_rsa_scales(samples)

if choice == '4' or choice == 'all':

```

```

print("\n Riemann zero detection range (primes only):")
p_start = int(input(" Start [default=1_000_000]: ").strip() or "1_000_000")
p_end = int(input(" End [default=2_000_000]: ").strip() or "2_000_000")
n_zeros = int(input(" Number of zeros to analyze [default=50, max=200]: ").strip() or "50")
n_cores = min(n_zeros, len(GAMMA_200))
phase_riemann_zero_detection(p_start, p_end, n_zeros, n_cores)

if choice != 'all':
    break
print("\n All phases completed. Generating alpha convergence plot if multiple exhaustive runs...")
if len(exhaustive_stats) > 1:
    plot_alpha_convergence(exhaustive_stats, ".")
print(" Done.")

# Final alpha convergence plot (if at least two exhaustive runs)
if len(exhaustive_stats) > 1:
    print("\n Generating alpha convergence plot from all exhaustive runs...")
    plot_alpha_convergence(exhaustive_stats, ".")
print("\n Done.")

if __name__ == "__main__":
    main()

```

References

1. C. Goldbach, "Letter to Euler," June 7, 1742.
2. G. H. Hardy and J. E. Littlewood, "Some problems of 'Partitio Numerorum'; III: On the expression of a number as a sum of primes," *Acta Math.*, vol. 44, pp. 1–70, 1923.
3. 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," *Sci. Sinica*, vol. 16, pp. 157–176, 1973.
4. 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} ," *Math. Comp.*, vol. 83, pp. 2033–2060, 2014.
5. P. G. L. Dirichlet, "Beweis des Satzes, daß jede unbegrenzte arithmetische Progression, deren erstes Glied und Differenz ganze Zahlen ohne gemeinschaftlichen Factor sind, unendlich viele Primzahlen enthält," *Abh. Kön. Preuß. Akad. Wiss.*, pp. 45–81, 1837.
6. A. Hildebrand, "Additive and multiplicative functions on shifted primes," *Proc. London Math. Soc. (3)*, vol. 59, pp. 209–232, 1989.
7. P. Erdős and A. Wintner, "Additive arithmetical functions and statistical independence," *Amer. J. Math.*, vol. 61, pp. 713–721, 1939.
8. T. M. Apostol, *Introduction to Analytic Number Theory*. New York: Springer-Verlag, 1976.
9. G. Tenenbaum, *Introduction to Analytic and Probabilistic Number Theory*. Cambridge: Cambridge University Press, 1995.
10. H. Halberstam and H.-E. Richert, *Sieve Methods*. London: Academic Press, 1974.
11. I. M. Vinogradov, "Representation of an odd number as a sum of three primes," *Comptes Rendus Acad. Sci. URSS*, vol. 15, pp. 169–172, 1937.
12. H. R. P. Ferguson and R. W. Forcade, "Generalization of the Euclidean algorithm for real numbers to all dimensions higher than two," *Bull. Amer. Math. Soc.*, vol. 1, pp. 912–914, 1979.
13. A. Fujii, "An additive problem of prime numbers," *Acta Arith.*, vol. 58, pp. 173–179, 1991.

14. G. Bhowmik and J.-C. Schlage-Puchta, "Mean representation number of integers as the sum of primes," *Nagoya Math. J.*, vol. 200, pp. 27–33, 2010.
15. D. A. Goldston and A. I. Suriajaya, "On an average Goldbach representation formula of Fujii," *Nagoya Math. J.*, vol. 250, pp. 511–532, 2023.
16. G. L. Miller, "Riemann's hypothesis and tests for primality," *J. Comput. System Sci.*, vol. 13, pp. 300–317, 1976.
17. M. O. Rabin, "Probabilistic algorithm for testing primality," *J. Number Theory*, vol. 12, pp. 128–138, 1980.
18. A. Languasco and A. Zaccagnini, "On the Goldbach conjecture in short intervals under the Riemann Hypothesis," *Trans. Amer. Math. Soc.*, vol. 364, pp. 4405–4443, 2012.
19. H. Davenport, *Multiplicative Number Theory*, 3rd ed. New York: Springer, 2000.
20. A. E. Ingham, "On the estimation of $N(\sigma, T)$," *Quart. J. Math.*, vol. 11, pp. 291–292, 1940.
21. R. C. Vaughan, *The Hardy–Littlewood Method*, 2nd ed. Cambridge: Cambridge University Press, 1997.
22. H. A. Helfgott, "The ternary Goldbach conjecture is true," *arXiv:1312.7748*, 2013.
23. J. Sorenson and J. Webster, "Strong pseudoprimes to twelve prime bases," *Math. Comp.*, vol. 86, pp. 985–1003, 2017.
24. J. Maynard, "Small gaps between primes," *Ann. Math.*, vol. 181, pp. 383–413, 2015.
25. Y. Zhang, "Bounded gaps between primes," *Ann. Math.*, vol. 179, pp. 1121–1174, 2014.
- [26] I. F. Anderson, "Multiplicity and structure of prime numbers," *Preprints*, 2026. [Online]. Available: <https://doi.org/10.20944/preprints202603.0717.v1>
- [27] I. F. Anderson, "Goldbach representations of shifted primes: Structure, computation, and singular-factor bias," *Preprints*, 2026. [Online]. Available: <https://doi.org/10.20944/preprints202603.0717.v2>
- [28] I. F. Anderson, "Goldbach representations of shifted primes: structure, computation, singular-factor bias, and extended computations to $p < 6.79 \times 10^7$ " *Preprints*, 2026. [Online]. Available: <https://doi.org/10.20944/preprints202603.0717.v3>
26. [Online]. Available: <https://doi.org/10.20944/preprints202603.0717.v3>

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.