

Highlights

Replaying the certificates behind a registry's unverified bounds: entropy laws, a truth table, step functions and a record package, decided from their bytes*

Carlos Toledo

- Every finite certificate behind the registry's asterisked bounds holds, decided by two programs sharing no code.
- A bound printed on the wrong side of its own certificate: refuted as printed, certified one unit up.
- A producer's checker printed OK on false claims: it compared in double precision what it enclosed at a hundred digits.
- A record package was refused because the analytic inputs its bound rests on are not in it.
- The registry removed four unverified marks on the replays and cites them beside the certificates.

Replaying the certificates behind a registry’s unverified bounds: entropy laws, a truth table, step functions and a record package, decided from their bytes

Carlos Toledo^{a,*}

^a*Independent researcher, the cert-machine project, Brazil*

Abstract

A curated registry of the best known bounds on 115 constants of analysis and combinatorics marks with an asterisk every bound whose only verification is the producer’s own, and states what removes the mark: refereeing, formalisation, or an independent replay. We replayed the certificates behind the asterisked bounds that rest on finite objects, from the bytes the registry cites and with no code of any producer’s run: entropy laws for the sums–differences constants C_{3b} and C_{3c} , a truth table on 18 variables for the Fourier entropy–influence constant C_{71} , step functions for the autocorrelation constants C_{1a} – C_{1d} , two constructions for the Zhang–Zagier height C_{82} , a step measure for the Schur–Siegel–Smyth trace constant C_{86} , and a “certified record package” for the de Bruijn–Newman constant. Each certificate is read as a finite exact object and decided by two programs that share no code with each other or with the claimants, in exact rationals where the object is rational and in outward-rounded intervals where an entropy or a logarithm enters; the deciders’ arithmetic is compared with Arb, and claims of known truth planted on the certificates are decided as constructed. Every finite certificate holds. One bound, C_{1d} , is printed below its own certificate’s exact value: refuted as printed, certified one unit up. The package is refused, because the analytic inputs its bound rests on are not in it. One certificate’s own checker printed OK on false claims, since it compares in double precision what it encloses at a hundred digits. After the replays the registry removed four asterisks and cites the replays on its pages; we report that as what happened and draw no causal conclusion. A replay decides that

*Draft v0.1 of 2026-10-21, a companion to the cert-machine method paper, built from repository commit 16849460e608; it will be archived with the next release of the repository, under the concept doi:10.5281/zenodo.22225860 (the previous version is in release v2026.10.18, doi:10.5281/zenodo.23446020). Every measured number and count is a macro read at build time from the records named in the Data availability statement; a record that no longer supports the sentence quoting it stops the build.

*Corresponding author.

Email address: `carlos@carlostoledo.co` (Carlos Toledo)

a certificate establishes a printed bound, and nothing about the constant beyond it.

Keywords: optimization constants, entropy certificates, sums–differences problem, autocorrelation inequalities, independent replay, interval arithmetic

1. Introduction

Curated records of what is known are how a field keeps score, and machines now write to them faster than referees read. The optimization-constants registry (Tao et al., 2026) keeps the best known bounds on 115 constants of analysis and combinatorics, each with its source, and many of its recent rows come from program search and AI-assisted contributors (Georgiev et al., 2025; Romera-Paredes et al., 2024; Novikov et al., 2025): two of the rows decided here carry the registry’s own note that they were prepared with an AI system, and a third producer signs its archive “AI Autoresearch” (Numaro, 2026). Such a bound arrives as a self-archived artifact—a certificate file, a script that checks it—checked so far only by the program its producer wrote. The registry’s answer is a mark: “Bounds for which the level of available verification is currently at minimal levels will be marked with an asterisk” (Tao et al., 2026). Its contributing guide states how the mark comes off: it “is removed when that changes—when the paper is refereed and published, or when the argument is formalized or independently replayed”. On 2026-09-26 the maintainer, writing that rule into the guide, added of one such bound: “If the 13-point certificate does get independently replayed, or formalized, the asterisk comes off”, and that he would take a pull request that removes it on those grounds (Tao, 2026).

This paper is the replay the rule asked for, made with cert-machine (Toledo, 2026, the method paper), a system that decides machine-generated claims independently of the claimant and records one of three verdicts—CERTIFIED, REFUTED or REFUSED—with the object that justifies it. A claim is read from its published statement and bytes; the claimant’s program is never in the trust path; a verdict is a computation with a witness, and a reading—a file inventoried, a five-sentence argument checked line by line—is recorded as a reading and never counted as a verdict. The claims decided are the registry’s asterisked bounds that rest on finite objects, as they stood on three dates: three lower bounds before the replays, the 10 bounds of its open pull-request queue on 2026-10-15, and the rows its merges of 2026-10-18 added (Section 2).

Contributions. Every finite certificate behind those bounds decided twice, by programs sharing no code with each other or with any producer, with the verdict, enclosure and scope of each recorded; two findings beside the

verdicts, a bound printed on the wrong side of its own certificate and a producer’s checker that accepts false claims; a refusal with its reason; and what the registry did afterwards, from its pinned bytes. The constants are not the contribution: a replay decides that a certificate establishes a printed bound and nothing more. Sections 2–5 give the materials, the method, the verdicts and the registry’s response; Section 6 what the exercise shows and its limits; Appendix A the command that re-derives each result.

2. Materials

The bounds. On 2c1968cd (2026-09-26) three asterisked lower bounds rested on finite certificates: two constants of the sums–differences problem (Katz and Tao, 1999), $C_{3b} \geq 1.77898884$ and $C_{3c} \geq 1.6747338950414058$, each certified by an entropy law on 13 and 147 points of $\mathbb{Z}^2$ (Mosaic Intelligence, 2026a; Lin, 2026); and the Fourier entropy–influence constant (Friedgut and Kalai, 1996), $C_{71} > 6.521845710923046575$, certified by a Boolean function on 18 variables (Numaro, 2026). On 2026-10-15 the open pull requests (#210, #217, #221, #225 and #226) offered 10 new bounds: two step functions, five one-line constructions, a nine-dimensional counterexample, a two-law entropy certificate (Jaswal, 2026), and a “certified record package” for the de Bruijn–Newman constant, $\Lambda \leq 0.1595$ (Mosaic Intelligence, 2026b), against the recorded $0 \leq \Lambda \leq 0.2$ (Rodgers and Tao, 2020; Platt and Trudgian, 2021) and an asterisked 0.1875 from the same claimant. The merges of 2026-10-18 added asterisked rows that rest on finite objects: four step functions for C_{1a} and C_{1b} , two for C_{1c} and C_{1d} , two constructions for C_{82} and a step measure for C_{86} .

The bytes. Every object decided is a byte sequence named by path and SHA-256 digest in the record that decided it, fetched from the source the registry cites and refused if its digest does not match: deposit and gist files for the entropy laws and the two-law certificate, the artifact that binds the truth table by digest, the pull requests’ bytes at the registry commit 2c1968cd, the package at its pinned commit, and the files the new pages cite. The registry’s own bytes—README, contributing guide and constant pages—are pinned at 2c1968cd before the replays and at ec37689f and 46dbf036 after each round of merges on 2026-10-18; what the registry prints beside each bound is read from those bytes. Each decision writes a record naming its inputs by digest, the verdict, the witness, the scope, the time and the second implementation’s answer; every number in this paper is a macro written from one of the records named in the Data availability statement by a generator that refuses to write when the record no longer supports the sentence quoting it. Everything ran on an 8-core laptop with 16 GB of memory, the JavaScript deciders in Node v24.14.1 and the Python ones in the standard library alone; nothing was rented.

3. Method

3.1. What each certificate claims, and how it is read

Every certificate here is a universal claim over one finite object—a law, a truth table, a step function, a polynomial, a step measure—and the claim decided is that the object establishes the printed bound, given the published theorem that turns such an object into a bound; the theorem is read and named, never re-proved, and each record’s scope sentence says so.

An entropy law. The sums–differences problem of Katz and Tao (1999) asks for the best exponents in bounds on $|A - B|$ by $|A|$, $|B|$ and $|A + B|$; its constants have entropy forms, as Ruzsa’s entropy analogues of sumset inequalities make precise (Lau and Nair, 2023), and the registry states C_{3b} and C_{3c} in that form. A certificate is a finite law μ on $\mathbb{Z}^2$ with rational weights, claiming for $(X, Y) \sim \mu$

$$C \geq \frac{H(X - Y)}{\max\{H(X), H(Y), H(X + Y)\}},$$

with $H(X + 2Y)$ added to the maximum for C_{3c} . The pushforward of μ under each linear form is computed exactly in rationals; each $-p \ln p$ is enclosed in an interval, the logarithm by its series with an outward-rounded tail; the ratio is a quotient of intervals, and $C \geq c$ is CERTIFIED when the enclosure’s lower end is at or above c , REFUTED when its upper end is below c , REFUSED when it straddles c . Earlier lower bounds were found by hand and by search (Lemm, 2015); these were found by AI-assisted search (Georgiev et al., 2025), which is why they come as artifacts. The two-law certificate draws 10^{20} independent base-55 digits from laws on 51 and 47 points; its entropies add digit by digit only if the map from digits to the linear form is injective, which holds when every form is at most 27 in absolute value on both supports—a condition of the claim, checked and not assumed.

A truth table. The Fourier entropy–influence conjecture of Friedgut and Kalai (1996) asks for a constant C with $H(\hat{f}^2) \leq C \cdot I(f)$ for every Boolean function f ; the best general results are of the form $H \leq O(I \log I)$ (Kelman et al., 2020), and lower bounds on C come from explicit functions (O’Donnell and Tan, 2013; Hod, 2017). The C_{71} bound is read from a truth table of 2^{18} entries: the integer Walsh–Hadamard transform gives the spectrum exactly, the influence I is a rational, the entropy H of the spectral weights is an enclosure, and the bound $H/(I - 1)$ follows from an amplification rule that is cited, not re-proved (O’Donnell and Tan, 2013; Hod, 2017). The decision is 10 named checks, from the digest the artifact binds and balance to $I = 261/128$ exactly, the bound, the claimant’s printed floor inside the enclosure and monotonicity.

A step function. The autocorrelation constants C_{1a} – C_{1d} are suprema and infima of autoconvolutions and correlations of functions on an interval (Mátolcsi and Vinuesa, 2010; Cloninger and Steinerberger, 2017; Barnard and Steinerberger, 2020), and a bound is exhibited by one function. A step function with n heights written as integers over one denominator has an autoconvolution that is one exact integer polynomial, so the bound it exhibits is one exact rational; the printed bound is `CERTIFIED` when it lies on the safe side of that rational and `REFUTED` as printed when it does not, in which case the rational the certificate does establish is recorded beside the refutation, as a repair.

A Zhang–Zagier construction. The essential minimum of the Zhang–Zagier height (Zhang, 1992; Zagier, 1993) is bounded above by limit-point constructions; Doche’s (Doche, 2000) gives the bound as a ratio of a Mahler measure and an integral of $\max(0, \log|A| - \log|Q|)$ on the unit circle. Every root of every composed factor is certified by Gershgorin’s theorem on exactly computed Weierstrass corrections, the integral is bounded on dyadic panels, the theorem’s hypotheses are checked exactly, and the enclosure of the construction’s value is compared with the printed bound; Doche’s theorem is read.

A step measure. The Schur–Siegel–Smyth trace problem asks for the smallest limit of $\text{tr}(\alpha)/\deg(\alpha)$ over totally positive algebraic integers (Smyth, 1984); Smith’s criterion (Smith, 2024) turns an explicit measure and finitely many integer polynomials into an upper bound, the route recent bounds take (Orloski and Talebizadeh Sardari, 2023). The C_{86} certificate is a step measure on 1,200 cells, 629 nonzero, and 11 integer polynomials with weights. Decided are the six finite conditions the claimant states—the hypotheses of the criterion and the arithmetic of the bound: densities nonnegative; each polynomial primitive, irreducible over $\mathbb{Z}$ and totally real; the weighted degree below the mass; each $\int \log|Q| d\mu$ positive, by enclosures; the potential minus the weighted logarithms nonnegative on every cell of positive density, by adaptive bisection with second-order Taylor forms; and the first moment over the mass at or below the printed bound. The chain from the conditions to the bound is the published one, read.

A one-line construction, a counterexample, a package. Two of the queue’s one-line bounds are decided from the moments of an explicit law, exactly; two more needed derivations the pull request did not contain, which their author supplied when asked, and were decided from those with no code of his run. A five-sentence argument was checked by reading alone and is counted as a reading, not a verdict. The nine-dimensional counterexample is a Lean development this project had already built and checked under two kernels; the request’s statement was read against that record. The package was inventoried without running anything.

3.2. Deciding twice, and checking the deciders

Two implementations that share no code. Each certificate was decided by two programs written from the specification in different languages and arithmetics: the entropy certificates in JavaScript over arbitrary-precision integers with dyadic intervals at 128 bits, and again by the Python standard library alone, its `decimal` arithmetic at 120 digits rounded outward; the truth table in Python `decimal` intervals at 60 digits and again in JavaScript over `BigInt` with the entropy in closed form; the step functions, autocorrelation rows, Zhang–Zagier rows and step measure each in JavaScript over `BigInt` rationals with the interval kernel’s series logarithm and again in Python with `fractions` and its own logarithm. Each record states the agreement: the same word, and the same exact fraction or overlapping enclosures. These are N-version checks written by one author with AI agents from one specification, the correlated-failure case Knight and Leveson (1986) measured: they catch a slip in one arithmetic, not a shared misreading.

The deciders’ arithmetic against Arb. The real functions of the entropy deciders—outward conversions, logarithms, entropies and ratios, directed-rounding contexts—were run on 270,746 cases from one seed, among them the registry’s own certificates, and every enclosure was compared with the exact value computed in Arb (Johansson, 2017) at 256 bits and more, or exactly in rationals when the value is rational. Of the 264,031 arithmetic cases, 264,015 returned an enclosure and every one contained the value; the other 16 were refused, each outside the amplification rule’s domain; all 71 malformed certificates were refused and all 7 planted wrong functions caught. The first run, on 2026-10-19, found 7 kinds of printed enclosure cut toward zero and 10 malformed inputs a Python decider let through; no published verdict was affected, since every verdict compared the full-precision ends, each finding was fixed the same day, and the record keeps all 17 with their witnesses. The audit took 112 s and runs in reduced form in every build.

Planted claims and red controls. A checker that refuses everything never accepts a false claim, so claims of known truth are planted beside the real ones. On the three entropy certificates: the bound claimed just above the ratio, false; the law translated by an integer vector, true; a weight moved off a probability law, and a base too small for the digit map to be injective, malformed—3 false, 3 true and 3 malformed claims, every one decided as constructed. On the truth table: 4 false, 3 true and 5 malformed, all decided as expected by both implementations after a first run whose Python word differed from the reading on 6 of them and refuted 1 true claim, kept in the record and fixed on 2026-10-19. In the method paper’s pre-registered self-audit of 2026-10-21, the 4 entropy rows of the public register were mutated in 12 ways—4 false, 4 true, 4 malformed, each witnessed by a standard-library script sharing no code with the deciders—and every one came out as

required. Every battery also carries deliberately wrong variants that must be rejected (5 on the step functions, 4 on the autocorrelation rows, 6 on the Zhang–Zagier rows, 6 on the step measure), and fails if any does not fire. The planted sets are targeted, not sampled, and give no rate.

4. Results

4.1. The three asterisked bounds

All three are CERTIFIED (Table 1), their ratios enclosed as

$$\begin{aligned} C_{3b} : & \quad 1.77898884142069354980\dots \\ C_{3c} : & \quad 1.67473389504140587006\dots \\ C_{71} : & \quad 6.52184571092304657565\dots \end{aligned}$$

The entropy certificates took 1.2 s in JavaScript and 0.2 s in Python, in the rerun kit’s timing of 2026-10-21; the truth table 1.5 s by its 10 checks, and 21 s a second time with its planted claims. The sums–differences ledger decides five side claims about the printed digits as well: the 13-point certificate’s printed “true value” is a truncation of the ratio and not a rounding, since its 37-digit prefix is CERTIFIED and the prefix’s next value is REFUTED; the 147-point certificate’s own 60-digit bound is CERTIFIED, while the same bound raised by one unit in the seventeenth significant digit—the same double as the true bound—is REFUTED. The improvement the registry prints for that row, 2.06e-11, rounds from the enclosure.

4.2. The queue of 2026-10-15

The queue was decided wherever a finite exact object exists, in 161 s for all of it (Table 2). Of its 10 rows, 7 are CERTIFIED by a computation made here; one is CERTIFIED on this project’s earlier two-kernel record of the same Lean development; one is a reading; and one is REFUSED. For the two-law certificate both implementations enclose the ratio as 1.674894211307116228764343737060..., the enclosures sharing 31 digits. The refused row is the package: of its 104 files, 63 are frozen tables and 5 are scripts, and none of the scripts evaluates the zeta function, the heat-flow function H_t or the Dirichlet polynomial whose zeros the bound is about; the analytic inputs enter as frozen outputs whose proofs cite 9 numbered banks the bundle does not contain. The package certifies that its parts fit together as declared; whether the parts are true is not in it. That is a verdict on the evidence, not on Λ , whose known bounds stand as published (Rodgers and Tao, 2020; Polymath, 2019; Platt and Trudgian, 2021); the scope is “the evidence is not in the submission”, and the refusal names what would convert it: the bank that holds the floor on the torus, the separator at the barrier point and the constants of the heat-flow approximation, each a check that would run once published.

bound	producer	object decided	verdict here	the registry afterwards
$C_{3b} \geq 1.77898884$	Mosaic Intelligence	13-point law	CERTIFIED, two programs	asterisk removed; [T2026] cited
$C_{3c} \geq 1.6747338950414058$	Y. Lin	147-point law	CERTIFIED, two programs	asterisk removed; [T2026] cited
$C_{71} > 6.521845710923046575$	Numaro	2 ¹⁸ -entry truth table	CERTIFIED, two programs	asterisk removed; [T2026] cited
$C_{3c} \geq 1.674894211307116$	M. Jaswal	51 and 47-point laws	CERTIFIED, two programs	merged with asterisk, then removed on the replay (#230); [T2026b] cited
$\Lambda \leq 0.1595$ (the registry's C_{21a})	Mosaic Intelligence	104-file package	REFUSED: the evidence is not in it	merged with asterisk, as are all 5 rows after 2021

Table 1: Five bounds of the registry, from the certificate to the registry's record (`certs/sumdiff-ledger.json`, `certs/fei-ledger.json`, `certs/optq-ledger.json`; the registry's bytes after each round of merges in `corpus/optconst-queue/registry-2026-10-18/` and `corpus/optconst-queue/registry-2026-10-18b/`). [T2026] and [T2026b] are the references by which the pages cite the replays.

4.3. The rows decided after the merges

The 9 rows the merges of 2026-10-18 added were decided the same way (Table 3). The four step functions behind the new C_{1a} and C_{1b} rows are CERTIFIED, their bounds exact rationals in both programs, in 2.3 h of Python. The C_{1c} certificate is CERTIFIED; its page prints its fraction under the decimal 0.41030287175... while the fraction is 0.4103287175491..., a printing slip that does not move the bound. The C_{1d} certificate is the one row refuted as printed: $C_{1d} \leq 1.463071582607559$ rests on a 300-cell step function whose exact ratio, the claimant's own fraction, is 1.46307158260755932746..., above the printed bound. Rounded to nearest, as a floating-point print rounds, an upper bound fell below the value it bounds; the certificate shows $C_{1d} \leq 1.4630715826075594$, and the record holds both the refutation as printed and the repair CERTIFIED. The two C_{82} rows are both Doche's construction, over $\deg Q = 999,999,992$ for the larger, whose 2,058 roots were certified and integral bounded on 13,318 dyadic panels; both are CERTIFIED in both implementations, the larger certificate's value enclosed in $[0.253444687248, 0.253444687515]$ against the printed 0.2534449, the other's construction at 0.253568930... against the printed 0.2535744749068227, a weaker bound than the construction gives

constant	request	object	verdict	how
C_{1a}	#225	65,536-step function	CERTIFIED	the autoconvolution as one exact integer
C_{1b}	#225	114,688-step function	CERTIFIED	the correlation as one exact integer
C_{3c}	#221	two laws, 51 and 47 points	CERTIFIED	exact pushforwards; entropies at 128 bits
C_{20a}	#226	one-line construction	CERTIFIED	exact moments of the exponential law
C_{20c}	#226	one-line construction	CERTIFIED	exact in $\mathbb{Q}(\sqrt{3})$
C_{23b}	#226	five-sentence argument	a reading	read, not computed: counted as a reading
C_{16a}	#226	truncated lattice sums	CERTIFIED	fixed-point enclosures at 2^{200} , from the author’s derivation
C_{70a}	#226	a Steiner sum	CERTIFIED	outward-rounded decimal recurrences, from the author’s derivation
C_{28a}	#217	a Lean development	CERTIFIED	this project’s earlier two-kernel record, the statement read
Λ (the registry’s C_{21a})	#210	104-file package	REFUSED	read file by file, nothing run

Table 2: The registry’s open pull requests of 2026-10-15, decided (`certs/optq-ledger.json`). A reading is not a verdict and is not counted as one.

and so on the safe side. The trace-constant certificate $C_{86} \leq 1.8117251$ is CERTIFIED in both on its six conditions, the density proved positive on 3,658 boxes in each, its first moment over its mass $1.811725095153609\dots$, at or below the printed bound. Each CERTIFIED here is the certificate’s: the chain from the conditions to the bound is the claimant’s published one, read.

4.4. A producer’s checker, observed

One finding is recorded beside the verdicts and is not one of them. The 147-point certificate’s own checker, at the pinned gist revision and run on 2026-09-29, encloses its ratio at 100 digits and then makes its final comparison in the default context of 53 bits, so any claim within a double of the true bound prints OK (Table 4). It printed OK on all 4 claims it was given, of which 2 are REFUTED here and 1 is not CERTIFIED here. The bound the registry prints is true, and it was decided without that checker.

5. What the registry did next

The replay of the three bounds was offered as a pull request that quoted the rule back (pull request 216 to Tao et al., 2026), and on 2026-10-18

constant	producer	object	printed bound	verdict here
C_{1a}	Atris	65,536-step function	≤ 1.5027434868	CERTIFIED, two programs
C_{1a}	AgentDiscover	65,536-step function	≤ 1.5027435320	CERTIFIED, two programs
C_{1b}	Atris	114,688-step function	≤ 0.3808585504	CERTIFIED, two programs
C_{1b}	CodexProLong	3,584-step function	≤ 0.3808585749	CERTIFIED, two programs
C_{1c}	rozety	6,148-step function	≥ 0.4103287	CERTIFIED, two programs
C_{1d}	pull request 224	300-cell step function	≤ 1.463071582607559	REFUTED as printed; the repair CERTIFIED, two programs
C_{82}	R. Pinho	Doche's construction, $\deg Q \approx 10^9$	≤ 0.2534449	CERTIFIED, two programs
C_{82}	the page's second construction	Doche's construction	≤ 0.2535744749068227	CERTIFIED, two programs
C_{86}	R. Pinho	step measure, 11 polynomials	≤ 1.8117251	CERTIFIED, two programs

Table 3: The rows decided after the merges of 2026-10-18, each by two implementations that share no code (`certs/stepfun-replay.json`, `certs/autocorr-replay.json`, `certs/c82-replay.json`, `certs/c86-replay.json`). “Two programs”: the same word and the same exact value or overlapping enclosures in both.

claim given to the checker	printed	decided here
1.6747338950414...1828	OK	CERTIFIED here (its own bound)
1.6747338950414...1900	OK	not CERTIFIED here (above the certified end at the 58th decimal)
1.6747338950414058800	OK	REFUTED here (above the ratio at the 17th significant digit)
1.67473389504140590	OK	REFUTED here

Table 4: The 147-point certificate’s own checker on four claims (`certs/sumdiff-ledger.json`): its own 60-digit bound; that bound raised at the fifty-eighth decimal; and two bounds raised in the seventeenth significant digit, within one double of the true bound.

the maintainer merged it (commit ad40c534): the three asterisks are gone, and each of the 3 constant pages cites the replay beside the certificate it replays. The same day the registry merged the two-law certificate and the package with their asterisks, marking all 5 of the latter constant’s rows after 2021 as not independently replayed. The two-law replay was then offered in the form the first was accepted (pull request 230) and merged 3.5 hours later (commit 9938edd4), so the best lower bound the registry lists for C_{3c} without the mark moved from 1.6747338950414058 to 1.674894211307116. In the same minute the maintainer merged the queue’s two step functions for C_{1a} and C_{1b} , decided CERTIFIED here and reported only in a comment, and marked them and the two rows beside them with the asterisk, as “replayed only by the submitting team’s own checker” (commit 3010f7ed): the pull requests that cited the rule removed every mark they asked about and the comment removed none, and three cases support no causal conclusion. The four step functions were then decided by two implementations and offered in the same form on 2026-10-18 (pull request 232), the C_{82} and C_{86} certificates on 2026-10-21 (pull request 242), and the C_{1c} and C_{1d} rows, with the C_{1d} bound printed one unit up and the C_{1c} decimal corrected (pull request 243). None of this is peer review: the mark records the verification available, the replay is the verification the rule asked for, and the maintainer applied the rule without re-deriving it.

6. Discussion

What the exercise shows. No bound was found false. Where there is a gap it lies between what was claimed and what was checked: a checker that compares in double precision what it encloses at a hundred digits, so that a bound one unit above the truth in its seventeenth significant digit prints OK; a bound printed below its own certificate’s exact value, because a print rounds to nearest and an upper bound rounded down is no longer a bound; a package that certifies its own assembly and not the analysis it assembles. Each was closed, or named, by a finite exact computation whose only hard part was deciding to make it; an independent decision is not a better checker but one that was not written by the claimant and so does not know where to stop. The refusal is a result with a size: neither a claim that the bound is wrong nor that it is right, it measures what an outside party can decide from the submission and names what would convert it.

Relation to the literature. The certificates are instances of certifying algorithms (McConnell et al., 2011): an output with a witness a small independent checker can confirm. The checkers the producers ship are the usual form; the replay is the independent check the form presumes and rarely gets. For the constants nothing is claimed beyond the certificates: the sums–differences and entropy literature (Katz and Tao, 1999; Lemm, 2015; Lau

and Nair, 2023), the entropy–influence results (Friedgut and Kalai, 1996; O’Donnell and Tan, 2013; Hod, 2017; Kelman et al., 2020), the autocorrelation inequalities (Matolcsi and Vinuesa, 2010; Cloninger and Steinerberger, 2017; Barnard and Steinerberger, 2020), the Zhang–Zagier height (Zhang, 1992; Zagier, 1993; Doche, 2000) and the trace problem (Smyth, 1984; Smith, 2024; Orloski and Talebizadeh Sardari, 2023) are cited, not advanced. Curated records with an explicit rule—the registry’s asterisk, the status of a problem in the Erdős problems database (Bloom, 2026)—are the setting in which a replay can change a record, and the one in which an unreplayed artifact otherwise stands indefinitely.

Threats to validity. Selection. The rows were chosen because they rest on finite objects; rows resting on proofs, or on packages whose evidence is elsewhere, were not decided, and the results describe these rows, not the registry. *The author in the outcomes.* The merges and the removed marks followed pull requests the author made; they are reported as what happened and no causal claim rests on them. *Correlated implementations.* The two deciders of each row were written by AI agents under the author’s direction from shared briefs; N-version programs written that way can fail together (Knight and Leveson, 1986), and the comparison with Arb covers the entropy deciders only, so for the step-function, Zhang–Zagier and trace deciders the second implementation is the only outside check. *Read theorems.* Every chain from a finite object to a bound—the entropy inequality, the amplification rule, Doche’s theorem, Smith’s criterion—is the published one, read and not re-derived; a slip in a published theorem would not be caught here.

What is not claimed. Nothing about any constant beyond what its certificate establishes: an exhibited bound holds and says nothing about the true value or about optimality, and the registry’s other rows are not decided here. “Independent” means independent of the claimant, not that the two implementations are independent of each other. Nothing is certified in the regulatory sense.

7. Conclusions and recommendations

Every finite certificate behind the registry’s asterisked bounds holds, each decided twice from its bytes with no code of its producer’s run; one bound was printed on the wrong side of its own certificate and is now printed on the right side; one producer’s checker accepts false claims and was not needed; one package was refused with the inputs that would decide it named. The registry removed four marks on the replays and cites them beside the certificates. The cost was seconds to hours on a laptop. To those who maintain curated records: distinguish a bound decided from outside from one only claimed, write down what removes the distinction, and keep the

mark until it is met; the registry’s rule is the model. To producers: publish the object with a digest beside each file, print bounds rounded toward the safe side, and make the checker compare at the precision it encloses. A replay that shares no code with the producer is cheap once the bytes exist and impossible until they do.

Declaration of competing interest

The author declares no competing financial or personal interests. The maintainer and contributors of the optimization-constants registry have not been consulted on this draft; the exchanges with the registry took place on the public threads cited, and the registry’s pages cite the replays. Prospective co-authors are reviewing this manuscript.

CRedit authorship contribution statement

Carlos Toledo: conceptualization, methodology, software, validation, investigation, data curation, writing—original draft, writing—review and editing. The author list and the roles are to be agreed with the prospective co-authors.

Data availability

Every record, decider, table source and pinned input is public in the cert-machine repository (<https://github.com/carlostoledo1891/cert-machine>, commit 16849460e608; archived as Toledo, 2026): the replays `certs/sumdiff-ledger.json`, `certs/fei-ledger.json`, `certs/fei-second.json`, `certs/optq-ledger.json`, `certs/stepfun-replay.json`, `certs/autocorr-replay.json`, `certs/c82-replay.json` and `certs/c86-replay.json`; the arithmetic audit `certs/entropy-arb-audit.json` and the planted claims `certs/method-eval.json`; the registry’s bytes after each round of merges under `corpus/optconst-queue/`, and the certificates under `corpus/optimization-constants/` and `corpus/optconst-queue/`, each with its digest. The certificates remain under their producers’ licences.

Declaration of generative AI and AI-assisted technologies

During the preparation of this work the author used large language models (Claude, Anthropic) to draft code, exposition and this manuscript in a certified-verification setting: constructions and prose were proposed with the help of the models, and every quantitative claim is decided by outward-rounded interval arithmetic or exact rational arithmetic, with adversarial controls that must fire for a record to be accepted. After using the tool the

author reviewed and edited the content and takes full responsibility for the content of the publication. The records, not the models, are the evidence offered; they are public and re-runnable. For this paper specifically: the deciders and their second implementations were written by AI agents (Claude) under written briefs, and the inventory of the record package is a reading produced the same way and consolidated into the public record; no verdict here rests on a reading alone.

Funding

This work received no external funding. Nothing was rented.

Appendix A. Reproduction

Each result re-derives with the command beside it in Table A.1, on the laptop of Section 2; each command that fetches a producer’s file refuses it unless it matches the pinned digest, and each battery fails unless its red controls fire. The numbers of this paper are written by `node tools/paper-numbers/optconst-replays.js` and `node tools/paper-numbers/cert-machine-method.js`, which refuse when a record no longer supports a sentence.

References

- Barnard, R.C., Steinerberger, S., 2020. Three convolution inequalities on the real line with connections to additive combinatorics. *Journal of Number Theory* 207, 42–55. doi:10.1016/j.jnt.2019.07.001.
- Bloom, T.F., 2026. Erdős problems. URL: <https://www.erdosproblems.com/>. The database of problems, with their status and the discussion threads that move it.
- Cloninger, A., Steinerberger, S., 2017. On suprema of autoconvolutions with an application to Sidon sets. *Proceedings of the American Mathematical Society* 145, 3191–3200. doi:10.1090/proc/13690.
- Doche, C., 2000. On the spectrum of the Zhang-Zagier height. *Mathematics of Computation* 70, 419–430. doi:10.1090/s0025-5718-00-01183-2.
- Friedgut, E., Kalai, G., 1996. Every monotone graph property has a sharp threshold. *Proceedings of the American Mathematical Society* 124, 2993–3002. doi:10.1090/S0002-9939-96-03732-X.
- Georgiev, B., Gómez-Serrano, J., Tao, T., Wagner, A.Z., 2025. Mathematical exploration and discovery at scale. [arXiv:2511.02864](https://arxiv.org/abs/2511.02864).

result	command	expected	time
the entropy certificates	<code>python3 tools/verify_sumdiff.py; python3 tools/verify_twolaw.py</code>	CERTIFIED; their red controls refuted or refused	0.2 s;—
the records	<code>node tools/run-sumdiff-ledger.js --check; python3 tools/run-fei-ledger.py --check; python3 -I tools/run-optq-ledger.py</code>	re-derived identical; the queue 7 CERTIFIED by computation, one on an earlier record, one read, 1 REFUSED	1.2 s; 1.5 s; 161 s
the truth table, a second time	<code>node tools/verify_fei.js</code>	CERTIFIED in both implementations; the 12 planted claims as expected	21 s
the registry's response	<code>node tools/read-optconst-registry.js --check</code>	re-read identical from its pinned bytes	—
the step functions	<code>node tools/replay-stepfun.js</code>	the four C_{1a} , C_{1b} rows CERTIFIED in both; 5 red controls as expected	2.3 h
the autocorrelation rows	<code>node tools/replay-autocorr.js</code>	C_{1c} CERTIFIED; the printed C_{1d} REFUTED, its repair CERTIFIED; 4 red controls as expected	2 min
the Zhang–Zagier rows	<code>node tools/replay-c82.js</code>	both rows CERTIFIED in both; the controls and red controls as expected	216 s
the trace-constant certificate	<code>node tools/replay-c86.js</code>	CERTIFIED in both; 6 red controls REFUTED in both	245 s
the deciders against Arb	<code>node tools/audit-entropy-arb.js</code>	no violation; every malformed input refused; every planted function caught	112 s
the planted claims	<code>node tools/run-method-eval.js</code>	every false claim refuted, every true one certified, every malformed one refused	under a second

Table A.1: Each result, the command that re-derives it, what it must print, and what it took (the times are those the records hold;—where a record holds none).

Hod, R., 2017. Improved lower bounds for the Fourier entropy/influence conjecture via lexicographic functions. URL: <https://arxiv.org/abs/1711.00762>, arXiv:1711.00762.

Jaswal, M., 2026. Entropy certificate for the sums-differences constant C_{3c} : lower bound 1.674894211307116. doi:10.5281/zenodo.23289305. Zenodo deposit, version of 2026-10-10.

Johansson, F., 2017. Arb: efficient arbitrary-precision midpoint-radius

- interval arithmetic. *IEEE Transactions on Computers* 66, 1281–1292. doi:10.1109/TC.2017.2690633.
- Katz, N.H., Tao, T., 1999. Bounds on arithmetic projections, and applications to the Kakeya conjecture. *Mathematical Research Letters* 6, 625–630. doi:10.4310/MRL.1999.v6.n6.a3.
- Kelman, E., Kindler, G., Lifshitz, N., Minzer, D., Safra, M., 2020. Towards a proof of the Fourier-entropy conjecture? *Geometric and Functional Analysis* 30, 1097–1138. doi:10.1007/s00039-020-00544-2.
- Knight, J.C., Leveson, N.G., 1986. An experimental evaluation of the assumption of independence in multiversion programming. *IEEE Transactions on Software Engineering* SE-12, 96–109. doi:10.1109/TSE.1986.6312924.
- Lau, C.W., Nair, C., 2023. Information inequalities via ideas from additive combinatorics. *arXiv:2312.11017*.
- Lemm, M., 2015. New counterexamples for sums-differences. *Proceedings of the American Mathematical Society* 143, 3863–3868. doi:10.1090/s0002-9939-2015-12603-2.
- Lin, Y., 2026. 147-point entropy certificate for the 4-slope sum-difference constant C_{3c} . URL: <https://gist.github.com/CoolRmal/5368357cd781d7e5c676c9d68ad24d22>. Gist, revision 62621d2d.
- Matolcsi, M., Vinuesa, C., 2010. Improved bounds on the supremum of autoconvolutions. *Journal of Mathematical Analysis and Applications* 372, 439–447. doi:10.1016/j.jmaa.2010.07.030.
- McConnell, R.M., Mehlhorn, K., Näher, S., Schweitzer, P., 2011. Certifying algorithms. *Computer Science Review* 5, 119–161. doi:10.1016/j.cosrev.2010.09.009.
- Mosaic Intelligence, 2026a. Certified sum-difference lower-bound certificates for C_{3a} , C_{3b} , and C_{3c} . doi:10.5281/zenodo.20794135. Zenodo deposit.
- Mosaic Intelligence, 2026b. Improve upper bound for the de Bruijn–Newman constant to 0.1595 (certified record package). URL: <https://github.com/teorth/optimizationproblems/pull/210>. Pull request 210 to the optimization-constants registry; the bundle pinned at commit cc81078.
- Novikov, A., Vĩ, N., Eisenberger, M., Dupont, E., Huang, P.S., Wagner, A.Z., Shirobokov, S., Kozlovskii, B., Ruiz, F.J.R., Mehrabian, A., Kumar, M.P., See, A., Chaudhuri, S., Holland, G., Davies, A., Nowozin, S., Kohli, P., Balog, M., 2025. AlphaEvolve: a coding agent for scientific and algorithmic discovery. *arXiv:2506.13131*.

- Numaro, 2026. Certificate archive: improved lower bound $C_{71} > 6.521845710923046575$ for the Fourier entropy-influence constant ($n = 18$ explicit construction). doi:10.5281/zenodo.21497769. Zenodo deposit.
- O'Donnell, R., Tan, L.Y., 2013. A composition theorem for the Fourier entropy-influence conjecture. arXiv:1304.1347. ICALP 2013.
- Orloski, B.J., Talebizadeh Sardari, N., 2023. Limiting distributions of conjugate algebraic integers. arXiv:2302.02872.
- Platt, D., Trudgian, T., 2021. The Riemann hypothesis is true up to $3 \cdot 10^{12}$. Bulletin of the London Mathematical Society 53, 792–797. doi:10.1112/blms.12460.
- Polymath, D.H.J., 2019. Effective approximation of heat flow evolution of the Riemann ξ function, and a new upper bound for the de Bruijn–Newman constant. Research in the Mathematical Sciences 6, 31. doi:10.1007/s40687-019-0193-1.
- Rodgers, B., Tao, T., 2020. The de Bruijn–Newman constant is non-negative. Forum of Mathematics, Pi 8, e6. doi:10.1017/fmp.2020.6.
- Romera-Paredes, B., Barekatalin, M., Novikov, A., Balog, M., Kumar, M.P., Dupont, E., Ruiz, F.J.R., Ellenberg, J.S., Wang, P., Fawzi, O., Kohli, P., Fawzi, A., 2024. Mathematical discoveries from program search with large language models. Nature 625, 468–475. doi:10.1038/s41586-023-06924-6.
- Smith, A., 2024. Algebraic integers with conjugates in a prescribed distribution. Annals of Mathematics 200, 71–122. doi:10.4007/annals.2024.200.1.2.
- Smyth, C.J., 1984. The mean values of totally real algebraic integers. Mathematics of Computation 42, 663–681. doi:10.1090/s0025-5718-1984-0736460-5.
- Tao, T., 2026. Drop the unverified asterisk on the C_{3b} record. URL: <https://github.com/teorth/optimizationproblems/pull/176#issuecomment-5848109776>. The maintainer's comment of 2026-09-26 on pull request 176 to the optimization-constants registry, which another contributor opened under this title; the comment states the rule that the registry's contributing guide then carried.
- Tao, T., et al., 2026. A collection of optimization problems in mathematics. URL: <https://github.com/teorth/optimizationproblems>. The optimization-constants registry and its contributing guide; read at commits 2c1968cd (2026-09-26), ec37689f and 46dbf036 (2026-10-18), pinned in the repository.

- Toledo, C., 2026. cert-machine: independent exact certification of machine-generated mathematics. doi:10.5281/zenodo.22225860. The repository archive: code, records and reports. The concept DOI resolves to the latest version; the version this paper was built from is named on its title page.
- Zagier, D., 1993. Algebraic numbers close to both 0 and 1. *Mathematics of Computation* 61, 485–491. doi:10.1090/s0025-5718-1993-1197513-9.
- Zhang, S., 1992. Positive line bundles on arithmetic surfaces. *Annals of Mathematics* 136, 569–587. doi:10.2307/2946601.