A Width–2 Boundary Program for Excluding Off–Axis Quartets with an η–Absorption Tail Closure and a Finite–Height Front–End (v33)

Dylan Anthony Dupont

v33 compiled: 2026-01-21

Abstract

This document is a post-pivot build (v33) of the width–2 boundary program. Relative to v32, the main changes are:

1. *Forcing constant provenance:* the forcing allocation constant is restored to $K_{\mathrm{alloc}} = 3 + 8\sqrt{3}$ and carried consistently through the tail inequality and certificate scaffolding.

2. *Bridge 1 hypotheses hardened:* the outer factor is constructed via the Dirichlet problem on the box, and the zero-free inner collapse is proved via Dirichlet uniqueness (removing boundary-regularity ambiguity).

3. *Local term made explicit:* the local window count is bounded unconditionally and explicitly: $N_{\mathrm{loc}}(m) \leq 1.01 \log m + 17$ for $m \geq 10$, yielding an explicit majorant in the tail program.

4. *Admissibility / δ control:* the nominal scale δ_0 is separated from the chosen κ–admissible $\delta \leq \delta_0$, with explicit lemmas guaranteeing existence and monotone safety under δ–shrinking.

A refreshed reproducibility pack is included; numerical certificates are optional sanity checks and are not logically required by the η–absorption closure.

Contents

Executive Proof Status

Status (v33): This version integrates Batch 1 patch packets into the v32 post-pivot spine. Concretely: (i) K_{alloc} is restored to its forcing-lemma value (closing G–3 at the manuscript level), (ii) Bridge 1 is rewritten with a Dirichlet outer factor and a Dirichlet-uniqueness inner collapse (closing G–6), (iii) the local window term is made explicit via an unconditional short-interval bound for $N(T)$ (closing G–8), and (iv) admissibility/absorption hygiene is improved by separating the nominal scale δ_0 from an admissible $\delta \le \delta_0$ and proving monotone safety under δ–shrinking (partial progress on G–4/G–5).

Numerical certificates are retained as *optional sanity checks* in the reproducibility pack, but are not logically required for tail closure under the η–absorption posture.

Open proof-grade blockers (tracked in `v33_status_decision.md`):

1. *Constant provenance and δ–uniformity:* show $C_{\text{up}}, C_1, C_2, C_h''$ are defined and proved finite with no hidden dependence on δ or on unknown zeros (G–4, G–12).

2. *Upper-envelope scaling audit:* Lemma 10.2 must be rechecked for correct δ–scaling (no unjustified $\sqrt{\delta}$ steps) (G–2, E–5).

3. *Residual envelope proof:* Lemma 7.1 requires a complete, unconditional derivation in the exact v–frame (G–1).

4. *Front-end dependence:* the finite-height hypothesis remains an external input (Appendix B) (G–11); S2 remains the audit harness.

This v33 build includes: (i) a hardened Bridge 1 package, (ii) a restored forcing constant ledger, (iii) an explicit local-window bound feeding the tail, (iv) admissibility/absorption monotonicity lemmas, and (v) a rebuilt reproducibility pack with updated scripts.

Part I
Reader's Guide / Definitions and Reduction

1 Width–2 normalization

Define the width–2 objects

$$u := 2s, \qquad \zeta_2(u) := \zeta\left(\tfrac{u}{2}\right), \qquad \Lambda_2(u) := \pi^{-u/4}\, \Gamma\left(\tfrac{u}{4}\right)\, \zeta\left(\tfrac{u}{2}\right). \tag{1}$$

Then Λ_2 is entire and satisfies the functional equation

$$\Lambda_2(u) = \Lambda_2(2 - u). \tag{2}$$

We recenter at $u = 1$:

$$v := u - 1, \qquad E(v) := \Lambda_2(1 + v). \tag{3}$$

The functional equation becomes the evenness relation

$$E(v) = E(-v), \tag{4}$$

and complex conjugation gives $E(\overline{v}) = \overline{E(v)}$.

2 Heights and horizontal displacement (RH–free)

Let $\rho = \beta + i\gamma$ be any nontrivial zero of $\zeta(s)$ (no assumption on β). In width–2 we write

$$u_\rho := 2\rho = (1 + a) + im, \qquad a := 2\beta - 1 \in (-1, 1), \qquad m := 2\gamma > 0. \tag{5}$$

Thus RH is equivalent to $a = 0$ for every nontrivial zero.

3 Quartet symmetry in width–2

The functional equation and conjugation imply that any off–axis zero with parameters (a, m) produces a quartet

$$\{\, 1 \pm a \pm im \,\} \subset \{u \in \mathbb{C} : \Lambda_2(u) = 0\}. \tag{6}$$

In the centered v–coordinate this becomes $\{\pm a \pm im\} \subset \{v \in \mathbb{C} : E(v) = 0\}$.

4 Finite-height front-end after lowering the tail anchor

Once the tail anchor is lowered to $m_\star$, the analytic tail argument covers all $m \geq m_\star$. The remaining region corresponds to classical heights

$$0 < \text{Im}(s) < H_0 := m_\star/2. \tag{7}$$

In v31 we take $m_\star = 10$, hence $H_0 = 5$.

Definition 4.1 (Front-end statement). We say that *RH holds up to height H_0* if every nontrivial zero $\rho = \beta + i\gamma$ with $0 < \gamma \leq H_0$ satisfies $\beta = 1/2$.

Remark 4.2 (How v31 discharges the front-end). The required statement for v31 is RH up to height $H_0 = 5$. This is a tiny special case of published rigorous verifications of RH to enormous heights. For example, Platt–Trudgian prove RH for all zeros with $0 < \gamma \leq 3 \cdot 10^{12}$ using interval arithmetic, which immediately implies RH up to $H_0 = 5$. Appendix B records this discharge in a pinned JSON file.

Part II

Self-Contained Boundary Program and Tail Closure

5 Aligned boxes and the $\delta(m)$ scale

Let $m > 0$ and $\alpha \in (0,1]$. Fix a parameter $\eta \in (0,1)$ and define the *nominal* box scale

$$\delta_0 = \delta_0(m,\alpha) := \frac{\eta\alpha}{(\log m)^2}. \tag{8}$$

We will work with aligned boxes $B(\alpha, m, \delta)$ at scales $0 < \delta \leq \delta_0$. By default one may take $\delta = \delta_0$, but later (Definition 10.4) we allow shrinking δ to enforce κ–admissibility; this is non-circular and monotone-safe (Lemmas 10.5 and 11.2).

Define the (width–2) box centered at $\alpha + im$ by

$$B(\alpha, m, \delta) := \{\, v \in \mathbb{C} : |\text{Re}\, v - \alpha| \leq \delta, \ |\text{Im}\, v - m| \leq \delta \,\}. \tag{9}$$

We will also use the symmetric dial centers $v_\pm := \pm\alpha + im$.

6 Local factor and finiteness

For a fixed $m > 0$, let

$$Z(m) := \{\, \rho : E(\rho) = 0, \ |\text{Im}\, \rho - m| \leq 1 \,\} \tag{10}$$

(zeros counted with multiplicity). Define the local zero factor and residual:

$$Z_{\text{loc}}(v) := \prod_{\rho \in Z(m)} (v - \rho)^{m_\rho}. \tag{11}$$

$$F(v) := \frac{E(v)}{Z_{\text{loc}}(v)}. \tag{12}$$

Lemma 6.1 (Finiteness of Z_{loc}). *For each fixed $m > 0$ the set $Z(m)$ is finite; hence Z_{loc} is a finite product and F is meromorphic globally and analytic in any neighborhood of $\partial B(\alpha, m, \delta)$ that contains no zeros of E.*

Proof. Nontrivial zeros of ζ satisfy $0 < \beta < 1$, hence in the v–coordinate one has $\operatorname{Re} v \in (-1, 1)$ for all nontrivial zeros. Therefore the set $\{|\operatorname{Im} v - m| \leq 1\} \cap \{|\operatorname{Re} v| \leq 1\}$ is compact. Since E is entire and its zeros are discrete, only finitely many zeros can lie in this compact set. $\square$

7 Residual envelope bound and the constants ledger

Lemma 7.1 (Residual envelope inequality). *There exist absolute constants $C_1, C_2 > 0$ such that for all $m \geq 10$, all $\alpha \in (0, 1]$, and $\delta = \eta\alpha/(\log m)^2$, one has*

$$\sup_{v \in \partial B(\alpha, m, \delta)} \left| \frac{F'(v)}{F(v)} \right| \leq C_1 \log m + C_2. \tag{13}$$

Remark 7.2 (Hard gate). The tail certificates in Appendix A use explicit numerical interval enclosures for the constant ledger (e.g. $C_1, C_2, C_{\text{up}}, C_h'', \kappa$), stored in `v33_repro_pack/v33_constants_m10.json`. They verify the tail inequality *conditional on* these enclosures being correct. An unconditional claim therefore requires an independent certification of the analytic constant ledger, in particular Lemma 7.1 and the upper-envelope constant inputs (Lemma 10.2).

8 Short-side forcing

Assume an off-axis pair at height m with displacement $a > 0$ exists. On an aligned box with $\alpha = a$, the two upper zeros in the centered v–plane are at $v = \pm a + im$. The pair factor

$$Z_{\text{pair}}(v) := (v - (a + im))(v - (-a + im)) \tag{14}$$

produces a large phase rotation on the near vertical side.

Lemma 8.1 (Short-side forcing lower bound). *Let $I_+ := \{\alpha + iy : |y - m| \leq \delta\}$ with $|\alpha - a| \leq \delta$. Then*

$$\Delta_{I_+} \arg Z_{\text{pair}} = 2\arctan\left(\frac{\delta}{|\alpha - a|}\right) + 2\arctan\left(\frac{\delta}{\alpha + a}\right) \geq \frac{\pi}{2}. \tag{15}$$

9 Outer factorization and the inner quotient (Bridge 1)

We work on a fixed box $B = B(\alpha, m, \delta)$ and write B° for its interior. Assume the boundary-contact convention: E has no zeros on ∂B.

Lemma 9.1 (Dirichlet outer factor on a box). *Let $B = B(\alpha, m, \delta)$ be the closed rectangle and B° its interior. Assume E is holomorphic on a neighborhood of $\overline{B}$ and $E \neq 0$ on ∂B. Then $\log|E| \in C(\partial B)$. Let $U \in C(\overline{B}) \cap \operatorname{Harm}(B^\circ)$ be the unique solution of the Dirichlet problem with boundary data $U|_{\partial B} = \log|E|$. Since B° is simply connected, there exists a harmonic conjugate V on B° (unique up to an additive constant) such that $U + iV$ is holomorphic on B°. Define*

$$G_{\text{out}}(v) := \exp(U(v) + iV(v)), \qquad v \in B^\circ.$$

Then G_{out} is holomorphic and zero-free on B°, satisfies $|G_{\text{out}}(v)| = e^{U(v)}$ for $v \in B^\circ$, and

$$\lim_{z \to \xi, \; z \in B^\circ} |G_{\text{out}}(z)| = |E(\xi)| \qquad (\xi \in \partial B).$$

5

Proof. Continuity of $\log|E|$ on ∂B follows from $E \neq 0$ on ∂B. Existence and uniqueness of U on a rectangle are standard. Since B° is simply connected, U admits a harmonic conjugate V on B°, unique up to an additive constant. The function $U + iV$ is holomorphic, hence so is $G_{\text{out}} = \exp(U + iV)$, and it is zero-free. Finally $|G_{\text{out}}| = e^U$ on B°, and by continuity of U on $\overline{B}$ we have $e^{U(\xi)} = |E(\xi)|$ on ∂B, yielding the boundary modulus identity in interior-limit form. $\quad\square$

Define on B° the inner quotient

$$W(v) := \frac{E(v)}{G_{\text{out}}(v)}.$$

Then W is holomorphic on B° and $|W| = 1$ on ∂B in the sense of interior limits in modulus.

Proposition 9.2 (Bridge 1: zero-free inner collapse). *Assume the setup of Lemma 9.1 and define $W = E/G_{\text{out}}$ on B°. If W is zero-free on B° (equivalently, E is zero-free on B°), then W is constant on B°; in fact $W \equiv e^{i\theta_B}$ for some $\theta_B \in \mathbb{R}$.*

Proof. Since W is zero-free on B° and G_{out} is zero-free, the function E is zero-free on B°. Because B° is simply connected, E admits a holomorphic logarithm on B°, so $\log|E|$ is harmonic on B°. By construction U is harmonic on B°, continuous on $\overline{B}$, and equals $\log|E|$ on ∂B. Thus $U - \log|E|$ is harmonic on B° with zero boundary values, so by Dirichlet uniqueness $U \equiv \log|E|$ on B°. Therefore for $v \in B^\circ$,

$$|W(v)| = \frac{|E(v)|}{|G_{\text{out}}(v)|} = \frac{|E(v)|}{e^{U(v)}} = \frac{|E(v)|}{e^{\log|E(v)|}} = 1.$$

An analytic function of constant modulus on a connected open set is constant, hence $W \equiv e^{i\theta_B}$. $\quad\square$

Remark 9.3 (Boundary modulus convention). Under boundary-contact, U extends continuously to ∂B and satisfies $U|_{\partial B} = \log|E|$. Hence $|G_{\text{out}}| = |E|$ holds pointwise on ∂B as interior limits in modulus, and therefore $|W| = 1$ holds pointwise in modulus on ∂B. In boundary integral estimates this may be used in the a.e. sense without change.

10   Shape-only invariance and the envelope constants

Let $T(v) := (v - (\alpha + im))/\delta$, mapping ∂B affinely onto the fixed square boundary ∂Q with $Q = [-1, 1]^2$.

Lemma 10.1 (Shape-only invariance). *Any constant arising solely from geometric or boundary-operator estimates on ∂B that are invariant under affine rescaling depends only on ∂Q and is independent of (α, m, δ).*

Proof. Under T, arclength scales by δ and tangential derivatives by $1/\delta$. After normalization, all purely geometric quantities and operator norms reduce to fixed quantities on ∂Q. $\quad\square$

Lemma 10.2 (Upper envelope bound (outer-aligned form)). *Let $B = B(\pm a, m, \delta)$ be an aligned box and let G_{out} be the outer factor on B constructed from $\log|E|$ on ∂B (Section 9). Define the inner quotient*

$$W(v) := \frac{E(v)}{G_{\text{out}}(v)}.$$

Assume the boundary-contact convention: E has no zeros on ∂B (hence W has unimodular boundary values a.e.). For each sign $\pm$ let $v_\pm := \pm a + im$ and let $e^{i\varphi_0^\pm} \in \mathbb{T}$ be an $L^2(\partial B, ds)$-best constant phase,

$$e^{i\varphi_0^\pm} \in \arg\min_{|c|=1} \int_{\partial B} |W(v) - c|^2 \, ds(v).$$

6

Then there exists a shape-only *constant* $C_{\mathrm{up}} > 0$ *(depending only on the normalized square* $Q = [-1, 1]^2$*) such that*

$$\sum_{\pm} \left| W(v_\pm) - e^{i\varphi_0^\pm} \right| \;\leq\; 2\, C_{\mathrm{up}}\, \delta^{3/2} \sup_{v \in \partial B} \left| \frac{E'(v)}{E(v)} \right|. \tag{16}$$

One admissible explicit definition is

$$C_{\mathrm{up}} := \left(\sup_{\xi \in \partial Q} P_Q(0, \xi) \right)^{1/2} \cdot \frac{4}{\pi} \cdot \sqrt{8} \cdot \left(1 + \|H_{\partial Q}\|_{L^2 \to L^2} \right),$$

where $P_Q(0, \xi) = d\omega_0^Q/ds(\xi)$ *is the Poisson kernel of* Q *at the center* 0 *with respect to arclength on* ∂Q*, and* $H_{\partial Q}$ *is the boundary conjugation (Hilbert/Cauchy) operator on* ∂Q*.*

Remark 10.3 (No residual proxying in the upper envelope). Lemma 10.2 controls the inner quotient $W = E/G_{\mathrm{out}}$ and therefore depends on $\sup_{\partial B} |E'/E|$. Residual bounds for $F = E/Z_{\mathrm{loc}}$ control $\sup_{\partial B} |F'/F|$ and do *not* by themselves bound $\sup_{\partial B} |E'/E|$. Whenever the residual envelope is used to control dial deviation, it must be routed through the log-derivative split $E'/E = F'/F + Z'_{\mathrm{loc}}/Z_{\mathrm{loc}}$ (Lemma 10.6) together with the collar bound (Lemma 10.7), yielding Corollary 10.9.

Proof. Fix one sign and write $v_0 = v_\pm$ and $B = B(\pm a, m, \delta)$. We record the (RH-free) chain and indicate the scale factors explicitly.

1. **Evaluation from the boundary (harmonic measure; produces $\delta^{-1/2}$).** For any constant $c \in \mathbb{T}$, subharmonicity of $|W - c|^2$ implies

$$|W(v_0) - c|^2 \leq \int_{\partial B} |W(\xi) - c|^2 \, d\omega_{v_0}^B(\xi) = \int_{\partial B} |W(\xi) - c|^2 \, P_B(v_0, \xi) \, ds(\xi),$$

   so

$$|W(v_0) - c| \leq \|P_B(v_0, \cdot)\|_{L^\infty(\partial B)}^{1/2} \|W - c\|_{L^2(\partial B, ds)}.$$

   Under the similarity $T(\xi) = (\xi - v_0)/\delta$ mapping ∂B onto ∂Q, Poisson kernels scale by $\|P_B(v_0, \cdot)\|_\infty^{1/2} = \delta^{-1/2} \|P_Q(0, \cdot)\|_\infty^{1/2}$.

2. **Poincaré/Wirtinger on ∂B (produces δ).** For the L^2-best constant $c = e^{i\varphi_0^\pm}$ and $|\partial B| = 8\delta$, periodic Poincaré on a loop of length 8δ gives

$$\|W - c\|_{L^2(\partial B)} \leq \frac{|\partial B|}{2\pi} \|\partial_s W\|_{L^2(\partial B)} = \frac{4\delta}{\pi} \|\partial_s W\|_{L^2(\partial B)}.$$

3. **Outer factor control (no δ; uses bounded boundary conjugation).** Write $\log G_{\mathrm{out}} = U + i\widetilde{U}$ with $U|_{\partial B} = \log |E|$ and $\widetilde{U} = H_{\partial B} U$. Differentiating tangentially, $\partial_s \log G_{\mathrm{out}} = \partial_s U + i\, H_{\partial B}(\partial_s U)$. Since $\log W = \log E - \log G_{\mathrm{out}}$,

$$\|\partial_s \log W\|_{L^2(\partial B)} \leq \left(1 + \|H_{\partial B}\|_{L^2 \to L^2} \right) \|\partial_s \log E\|_{L^2(\partial B)} \leq \left(1 + \|H_{\partial B}\|_{L^2 \to L^2} \right) \left\| \frac{E'}{E} \right\|_{L^2(\partial B)}.$$

   On ∂B we have $|W| = 1$ a.e., hence $|\partial_s W| = |\partial_s \log W|$.

4. **L^2 to sup (produces $\delta^{1/2}$).** Using $|\partial B| = 8\delta$,

$$\left\| \frac{E'}{E} \right\|_{L^2(\partial B)} \leq \sqrt{|\partial B|} \sup_{\partial B} \left| \frac{E'}{E} \right| = \sqrt{8\delta} \sup_{\partial B} \left| \frac{E'}{E} \right|.$$

Combining the four steps yields

$$|W(v_0) - e^{i\varphi_0^{\pm}}| \leq \|P_Q(0, \cdot)\|_{\infty}^{1/2} \cdot \frac{4}{\pi} \cdot \sqrt{8} \cdot \left(1 + \|H_{\partial Q}\|_{L^2 \to L^2}\right) \cdot \delta^{3/2} \sup_{\partial B} \left|\frac{E'}{E}\right|,$$

where we used the similarity invariance $\|H_{\partial B}\|_{L^2 \to L^2} = \|H_{\partial Q}\|_{L^2 \to L^2}$. Summing over $\pm$ gives (16). $\qquad\square$

10.1 Local factor split and collar control

Definition 10.4 (Collar-admissible aligned boxes)**.** Fix once and for all a collar parameter $\kappa \in (0, 1/10)$. An aligned box $B = B(\alpha, m, \delta)$ is called κ–*admissible* if

$$\text{dist}\big(\partial B,\, \mathcal{Z}(E)\big) \geq \kappa\delta.$$

Given any nominal scale $\delta_0 > 0$ and any center, there exists some $0 < \delta \leq \delta_0$ for which κ–admissibility holds (Lemma 10.5). Whenever a chosen box is not κ–admissible, we shrink δ until κ–admissibility holds. Moreover the assembled tail inequality is monotone-safe under such δ–shrinking (Lemma 11.2).

Lemma 10.5 (Existence of a κ–admissible shrink)**.** *Fix $\kappa \in (0, 1/10)$ and a center $v_0 \in \mathbb{C}$. For every $\delta_0 > 0$ there exists $\delta' \in (0, \delta_0]$ such that the closed box*

$$B(v_0, \delta') := \{\, v \in \mathbb{C} : \ \|v - v_0\|_{\infty} \leq \delta' \,\}$$

satisfies

$$\text{dist}\big(\partial B(v_0, \delta'),\, \mathcal{Z}(E)\big) \geq \kappa\delta'.$$

In particular, given (α, m) and nominal $\delta_0 = \eta\alpha/(\log m)^2$, one may always choose a scale $0 < \delta \leq \delta_0$ for which $B(\alpha, m, \delta)$ is κ–admissible.

Proof. Zeros of the entire function E are isolated. Choose $\varepsilon > 0$ such that $\mathcal{Z}(E) \cap \{\, 0 < \|v - v_0\|_{\infty} \leq \varepsilon \,\}$ is empty (if $E(v_0) = 0$) or such that $\mathcal{Z}(E) \cap \{\, \|v - v_0\|_{\infty} \leq \varepsilon \,\}$ is empty (if $E(v_0) \neq 0$). Set $\delta' := \min\{\delta_0, \varepsilon/(1 + \kappa)\}$. Then every boundary point satisfies $\|v - v_0\|_{\infty} = \delta'$. Any zero $\rho \in \mathcal{Z}(E)$ is either $\rho = v_0$ (in which case $\text{dist}(v, \rho) = \delta' \geq \kappa\delta'$) or satisfies $\|\rho - v_0\|_{\infty} \geq \varepsilon$ (in which case $\text{dist}(v, \rho) \geq \varepsilon - \delta' \geq \kappa\delta'$). Therefore $\text{dist}(\partial B(v_0, \delta'), \mathcal{Z}(E)) \geq \kappa\delta'$. $\qquad\square$

Lemma 10.6 (Log-derivative decomposition)**.** *With Z_{loc} and F as in (11) and (12), one has on any region where E and Z_{loc} are holomorphic and nonvanishing (in particular on ∂B under the boundary-contact convention)*

$$\frac{E'}{E} = \frac{F'}{F} + \frac{Z'_{\text{loc}}}{Z_{\text{loc}}}.$$

Lemma 10.7 (Buffered local factor bound on ∂B)**.** *Let $B = B(\alpha, m, \delta)$ be κ–admissible in the sense of Definition 10.4. Then*

$$\sup_{v \in \partial B} \left|\frac{Z'_{\text{loc}}(v)}{Z_{\text{loc}}(v)}\right| \leq \frac{N_{\text{loc}}(m)}{\kappa\,\delta},$$

where $N_{\text{loc}}(m)$ counts zeros of E in the local window used to define Z_{loc}, with multiplicity.

Lemma 10.8 (Explicit local window zero count). *Let $N(T)$ denote the number of nontrivial zeros $\rho = \beta + i\gamma$ of $\zeta(s)$ with $0 < \gamma \le T$, counted with multiplicity. Then for every $T \ge 5$,*

$$N(T+1) - N(T-1) \le 1.01 \log T + 17. \tag{17}$$

Consequently, for every $m \ge 10$,

$$N_{\mathrm{loc}}(m) \le 1.01 \log m + 17. \tag{18}$$

Proof. By [7, Theorem 1.1], for every $x \ge e$,

$$\left| N(x) - \frac{x}{2\pi} \log\left(\frac{x}{2\pi e}\right) \right| \le 0.10076 \log x + 0.24460 \log\log x + 8.08344.$$

Let $M(x) := \frac{x}{2\pi} \log(\frac{x}{2\pi e})$, so $M'(x) = \frac{1}{2\pi} \log(\frac{x}{2\pi})$. For $T \ge 5$ we have $\log(T \pm 1) \le \log(2T)$ and $\log\log x \le \log x$ for $x \ge e$, hence

$$N(T+1) - N(T-1) \le (M(T+1) - M(T-1)) + 2(0.10076 + 0.24460)\log(2T) + 2 \cdot 8.08344.$$

Moreover

$$M(T+1) - M(T-1) = \int_{T-1}^{T+1} M'(x)\,dx \le \int_{T-1}^{T+1} \frac{1}{2\pi} \log x\,dx \le \frac{1}{\pi} \log(2T).$$

Combining these bounds gives $N(T+1) - N(T-1) \le 1.00903 \log T + 16.8663 \le 1.01 \log T + 17$, establishing (17). Finally, in width–2 one has $m = 2T$. The local window $|\operatorname{Im}\rho - m| \le 1$ corresponds to $|\gamma - T| \le 1/2$ in the s–plane, so $N_{\mathrm{loc}}(m) = N(T + \frac{1}{2}) - N(T - \frac{1}{2}) \le N(T+1) - N(T-1)$, yielding (18). $\qquad\square$

Corollary 10.9 (Outer-aligned upper envelope in residual+local form). *Let B be κ–admissible. Assume the residual envelope bound of Lemma 7.1, i.e. $\sup_{\partial B} |F'/F| \le L(m) := C_1 \log m + C_2$. Then*

$$\sum_{\pm} |W(v_\pm) - e^{i\varphi_0^\pm}| \le 2C_{\mathrm{up}}\left(\delta^{3/2} L(m) + \delta^{1/2} \frac{N_{\mathrm{loc}}(m)}{\kappa} \right) \le 2C_{\mathrm{up}}\left(\delta^{3/2} L(m) + \delta^{1/2} \frac{1.01 \log m + 17}{\kappa} \right).$$

10.2 Horizontal non-forcing budget in residual form

Definition 10.10 (Horizontal non-forcing phase budget). Let $B = B(\pm a, m, \delta)$ be an aligned box and let $F = E/Z_{\mathrm{loc}}$ be the residual factor. Assume F is holomorphic and zero-free on a neighborhood of ∂B. Let $H_\pm$ denote the top and bottom edges of ∂B:

$$H_+ := \{x + i(m+\delta) : \ x \in [\pm a - \delta, \pm a + \delta]\}, \qquad H_- := \{x + i(m-\delta) : \ x \in [\pm a - \delta, \pm a + \delta]\}.$$

Define

$$\Delta_{\mathrm{nonforce}}(B) := \int_{H_+} |\partial_s \arg F|\,ds + \int_{H_-} |\partial_s \arg F|\,ds.$$

Lemma 10.11 (Horizontal budget (residual form; audit-grade)). *In the setting of Definition 10.10,*

$$\Delta_{\mathrm{nonforce}}(B) \le 4\delta \sup_{v \in \partial B} \left| \frac{F'(v)}{F(v)} \right|.$$

Consequently, if $\sup_{\partial B} |F'/F| \le C_1 \log m + C_2$, then

$$\Delta_{\mathrm{nonforce}}(B) \le C_h'' \,\delta\,(\log m + 1), \qquad C_h'' := 4\max\{C_1, C_2\}.$$

Proof. On either horizontal edge, $|\partial_s \arg F| \le |F'/F|$ pointwise. Each edge has length 2δ, hence each integral is bounded by $2\delta \sup_{\partial B} |F'/F|$. Summing top and bottom gives the first inequality, and the second follows from $\sup_{\partial B} |F'/F| \le C_1 \log m + C_2 \le \max\{C_1, C_2\}(\log m + 1)$. $\qquad\square$

11 The explicit tail inequality (post-pivot)

For $m \geq 10$ we use the growth surrogate

$$L(m) := C_1 \log m + C_2,$$

with constants as in Lemma 7.1. For the local window term we use the explicit majorant from Lemma 10.8:

$$N_{\text{up}}(m) := 1.01 \log m + 17 \quad \text{so that} \quad N_{\text{loc}}(m) \leq N_{\text{up}}(m) \quad (m \geq 10).$$

For a parameter $\eta \in (0,1)$ and a dial displacement $\alpha \in (0,1]$ define the *nominal* scale

$$\delta_0 := \delta_0(m,\alpha) := \frac{\eta \alpha}{(\log m)^2}.$$

Fix a collar parameter $\kappa \in (0,1/10)$ as in Definition 10.4. For each (m,α) we choose any scale $0 < \delta \leq \delta_0$ such that the aligned boxes $B = B(\pm\alpha, m, \delta)$ are κ–admissible; existence is guaranteed by Lemma 10.5. By Lemma 11.2, shrinking δ only helps in the tail inequality, so it is safe to treat δ_0 as the worst-case scale in one-height reductions.

Theorem 11.1 (Tail inequality (audit-grade post-pivot form)). *Fix $m \geq 10$ and $\eta \in (0,1)$. Assume:*

1. *the forcing lemma producing the positive constant*

$$c_0 := \frac{3 \log 2}{8\pi}, \qquad c := \frac{3 \log 2}{16}, \qquad K_{\text{alloc}} := 3 + 8\sqrt{3};$$

2. *the residual envelope bound (Lemma 7.1) providing C_1, C_2;*

3. *the audit-grade horizontal budget bound (Lemma 10.11), giving a constant C_h'' independent of (α, m, δ);*

4. *the explicit local window bound (Lemma 10.8) providing the majorant $N_{\text{up}}(m) = 1.01 \log m + 17$.*

Then for every $\alpha \in (0,1]$ and every κ–admissible aligned box $B = B(\pm\alpha, m, \delta)$, absence of off–axis quartets at height m follows from the strict inequality

$$2C_{\text{up}}\left(\delta^{3/2}L(m) + \delta^{1/2}\frac{N_{\text{up}}(m)}{\kappa}\right) < c - \delta\left(K_{\text{alloc}} c_0 L(m) + C_h''(\log m + 1)\right). \tag{19}$$

Proof sketch / bookkeeping. The forcing side is unchanged from v31. The only post-pivot modification is on the upper-envelope side: Lemma 10.2 bounds dial deviation in terms of $\sup_{\partial B}|E'/E|$. Applying the log-derivative split (Lemma 10.6), the residual envelope for $\sup_{\partial B}|F'/F| \leq L(m)$ (Lemma 7.1), and the collar bound $\sup_{\partial B}|Z'_{\text{loc}}/Z_{\text{loc}}| \leq N_{\text{loc}}(m)/(\kappa\delta)$ (Lemma 10.7) yields

$$\sup_{\partial B}\left|\frac{E'}{E}\right| \leq L(m) + \frac{N_{\text{loc}}(m)}{\kappa\delta} \leq L(m) + \frac{N_{\text{up}}(m)}{\kappa\delta}.$$

Plugging this into Lemma 10.2 gives the left-hand side of (19). The right-hand side is the forcing lower bound, with the horizontal non-forcing term bounded by Lemma 10.11. $\square$

Lemma 11.2 (Monotonicity under δ–shrinking). *Fix $m \geq 10$, $\alpha \in (0,1]$, and constants $C_{\text{up}}, \kappa, c, c_0, K_{\text{alloc}}, C_h'', C_1, C$ Let $L(m) = C_1 \log m + C_2$ and $N_{\text{up}}(m) = 1.01 \log m + 17$. For $\delta \in (0,1]$ define*

$$\text{LHS}(\delta) := 2C_{\text{up}}\left(\delta^{3/2} L(m) + \delta^{1/2} \frac{N_{\text{up}}(m)}{\kappa}\right), \qquad \text{RHS}(\delta) := c - \delta\left(K_{\text{alloc}} c_0 L(m) + C_h'' (\log m + 1)\right).$$

Then $\text{LHS}(\delta)$ is (weakly) increasing in δ and $\text{RHS}(\delta)$ is (weakly) decreasing. Consequently, if $\text{LHS}(\delta_0) < \text{RHS}(\delta_0)$ for some $\delta_0 \in (0,1]$, then $\text{LHS}(\delta) < \text{RHS}(\delta)$ holds for every $\delta \in (0, \delta_0]$.

Proof. For $\delta > 0$, both $\delta^{3/2}$ and $\delta^{1/2}$ are increasing functions, hence so is $\text{LHS}(\delta)$. The bracketed factor in $\text{RHS}(\delta)$ is nonnegative and independent of δ, so $\text{RHS}(\delta)$ decreases linearly in δ. $\quad\square$

Lemma 11.3 (Worst case in α is $\alpha = 1$ at the nominal scale). *Fix $m \geq 10$ and $\eta \in (0,1)$. Define the nominal scale $\delta_0(m, \alpha) = \eta\alpha/(\log m)^2$. Consider the tail inequality (19) evaluated at $\delta = \delta_0(m, \alpha)$. Then the left-hand side is (weakly) increasing in $\alpha \in (0,1]$, while the right-hand side is (weakly) decreasing. Therefore it suffices to verify (19) at $\alpha = 1$ and $\delta = \delta_0(m, 1)$. If one later shrinks $\delta \leq \delta_0(m, \alpha)$ to enforce κ–admissibility, the inequality only becomes easier (Lemma 11.2).*

Proof. With $\delta = \delta_0(m, \alpha) = \eta\alpha/(\log m)^2$, both $\delta^{3/2}$ and $\delta^{1/2}$ are increasing functions of α, so the left-hand side increases. The right-hand side equals $c - \delta \cdot \Xi(m)$ for a nonnegative factor $\Xi(m)$ independent of α, hence it decreases. $\quad\square$

Lemma 11.4 (Monotonicity in m (including the local term)). *Fix $\alpha \in (0,1]$ and $\eta \in (0,1)$ and write $x := \log m$. Evaluate the tail inequality (19) at the nominal scale $\delta = \delta_0(m, \alpha) = \eta\alpha/x^2$. Using the bounds $L(m) = C_1 x + C_2$ and $N_{\text{up}}(m) = 1.01x + 17$, the left-hand side of (19) is (weakly) decreasing in $m \geq 10$, and the right-hand side is (weakly) increasing. Therefore it suffices to verify (19) at the minimal $m_\star = 10$ (and nominal scale $\delta_0(10, \alpha)$). If one later shrinks $\delta \leq \delta_0$ to enforce κ–admissibility, the inequality only becomes easier (Lemma 11.2).*

Proof. Write $\delta = \eta\alpha/x^2$. The left-hand side is bounded by

$$2C_{\text{up}}\left(\eta^{3/2}\alpha^{3/2} \frac{C_1 x + C_2}{x^3} + \frac{\sqrt{\eta\alpha}}{\kappa} \frac{1.01x + 17}{x}\right) = 2C_{\text{up}}\left(\eta^{3/2}\alpha^{3/2} \frac{C_1 x + C_2}{x^3} + \frac{\sqrt{\eta\alpha}}{\kappa}\left(1.01 + \frac{17}{x}\right)\right),$$

and both $\frac{C_1 x + C_2}{x^3}$ and $1.01 + 17/x$ decrease for $x > 0$. Hence the left-hand side decreases as m increases.

For the right-hand side, the subtracted term has the form $\delta \cdot \Psi(x)$ with

$$\Psi(x) = K_{\text{alloc}} c_0 (C_1 x + C_2) + C_h''(x + 1), \qquad \delta = \eta\alpha/x^2.$$

Since both $(C_1 x + C_2)/x^2$ and $(x + 1)/x^2$ decrease for $x > 0$, the product $\delta\,\Psi(x)$ decreases, and therefore the right-hand side increases in m. $\quad\square$

Theorem 11.5 (Tail closure from a one-height check). *Fix $\eta \in (0,1)$ and evaluate (19) at the nominal scale $\delta = \delta_0(m, \alpha) = \eta\alpha/(\log m)^2$. If (19) holds at $(m, \alpha) = (10, 1)$ (with $\delta = \delta_0(10, 1)$ and the same constants), then it holds for all $m \geq 10$ and all $\alpha \in (0,1]$ at the corresponding nominal scales. Consequently, for every κ–admissible choice $0 < \delta \leq \delta_0(m, \alpha)$ the inequality also holds (Lemma 11.2), so no off–axis quartets exist at any height $m \geq 10$.*

Proof. By Lemma 11.3 it suffices to check $\alpha = 1$ at the nominal scale. By Lemma 11.4 the inequality becomes easier as m increases. Therefore checking $(m, \alpha) = (10, 1)$ implies all $m \geq 10$ and all $\alpha \in (0,1]$ at the corresponding nominal scales. Finally, shrinking $\delta \leq \delta_0$ to enforce κ–admissibility only helps (Lemma 11.2). The quartet exclusion is exactly the forcing-vs-envelope contradiction. $\quad\square$

12 η–absorption at the low anchor

The post-pivot tail inequality at $(m, \alpha) = (10, 1)$ reads

$$A\,\eta^{3/2} + B\,\eta^{1/2} < c - D\,\eta,$$

with the explicit coefficients

$$A := \frac{2C_{\mathrm{up}}L(10)}{(\log 10)^3}, \qquad B := \frac{2C_{\mathrm{up}}N_{\mathrm{up}}(10)}{\kappa\,\log 10}, \qquad D := \frac{K_{\mathrm{alloc}}c_0 L(10) + C_h''(\log 10 + 1)}{(\log 10)^2}.$$

Since $A, B, D < \infty$ (by the lemmas cited above), the right-hand side is positive for all sufficiently small $\eta > 0$, and the inequality is forced by choosing η small enough.

Proposition 12.1 (η–absorption (explicit sufficient condition)). *Let A, B, D be as above. Define*

$$\eta_\star := \min\left\{1,\ \left(\frac{c}{4A}\right)^{2/3},\ \left(\frac{c}{4B}\right)^2,\ \frac{c}{4D}\right\}.$$

Then for every $\eta \in (0, \eta_\star]$ the tail inequality (19) holds at $(m, \alpha) = (10, 1)$, hence (by Theorem 11.5) at all $m \geq 10$ and all $\alpha \in (0, 1]$.

Proof. The definition of $\eta_\star$ ensures three inequalities simultaneously:

$$A\eta^{3/2} \leq c/4, \qquad B\eta^{1/2} \leq c/4, \qquad D\eta \leq c/4.$$

Thus the left-hand side $A\eta^{3/2} + B\eta^{1/2} \leq c/2$ and the right-hand side $c - D\eta \geq 3c/4$, yielding a strict inequality. $\square$

13   Global RH from a small front-end $+$ an η–absorbed tail

Theorem 13.1 (Global closure (post-pivot logical form)). *Assume:*

1. *(Front-end) All nontrivial zeros with $0 < \mathrm{Im}(s) \leq 5$ lie on the critical line.*

2. *(Analytic inputs) Lemmas 10.2–10.8 and Lemma 10.11 hold with finite constants.*

Then there exists $\eta_\star > 0$ (as in Proposition 12.1) such that for every choice $\eta \in (0, \eta_\star]$ the tail program excludes off–axis zeros for all $\mathrm{Im}(s) \geq 5$. Consequently, all nontrivial zeros lie on the critical line.

Proof. Proposition 12.1 and Theorem 11.5 exclude off–axis quartets at all heights $m \geq 10$. By the front-end hypothesis there are no off-axis zeros below height 5. Therefore there are no off-axis zeros at any height. $\square$

Remark 13.2 (Computations in v33). The post-pivot logical closure uses η–absorption and does not require numerical values for the constants, only finiteness and δ–uniformity. Appendix A nevertheless provides a small interval-arithmetic tail check for one concrete choice of parameters as a sanity check, and Appendix B records a pinned literature front-end discharge.

A Tail certificate bundle and reproducibility (v33)

A.1 What the tail certificates prove (and what they do not)

Each tail certificate proves the statement:

> Given a constants file that provides interval enclosures for $(C_1, C_2, C_{\text{up}}, C_h'', \kappa)$ and the chosen parameters (m, η, α) (with $N_{\text{up}}(m) = 1.01 \log m + 17$ hard-coded from Lemma 10.8), the generated interval bounds satisfy LHS < RHS with strict separation in the sense $\text{LHS}_{\text{hi}} < \text{RHS}_{\text{lo}}$.

It does *not* certify that the constants file is correct.

A.2 SHA−256 table (exact artifacts)

The file `v33_repro_pack/SHA256SUMS.txt` is the canonical hash list.

```
c1a9d56949b2dd58d617a21428c3497e18be6f8b47f7782ff5d01351f7efaf97   README.md
cbb8de4d781683c4a73d2bcb284ef476e10e90f09fbbb1d864fcc687cc305374   v33_constants_m10.json
ec3701b63e2b343981b01bc48b338b3c1da08d6b44d1f4ad21259ba82e12aa25   v33_frontend_certificate.json
0ba807ea35a9930a954dca01bd5261fbbfeb38747febef2125bb6c6ad58b9ae5   v33_tail_certificate_m10.json
8858142a403cb81b2c61b1710ffa3a0c7e0412b8c80dbe446c76cadb78f4a6cf   v33_generate_frontend_certificate
    .py
9517c1e42946becfbbaaa826733ef67779bd80643154fcb7bf6b8b3aad6e7247   v33_generate_tail_certificate.py
d2e4981ba29745925df84db9b06415defe4738ff1f749efee10dafe3f65d2209   v33_verify_frontend_certificate.
    py
67920531ecf65031488415c56cf54112b9247e1b19e963e543378f25fa76f874   v33_verify_tail_certificate.py
c1debbda3583dbaf0dc7120684ba89c457fef1227f4aa13504b21cf11e029acb   v33_frontend_verifier_output.txt
acc667dec23169d66c954328a1ea3dc5dc099749d5045c771537b0d294d5f40f   v33_verifier_output_m10.txt
```

A.3 Commands

From the directory `v33_repro_pack/`:

1. `sha256sum -c SHA256SUMS.txt`

2. `python3 v33_verify_tail_certificate.py --constants v33_constants_m10.json --certificate v33_tail_certificate_m10.json`

3. `python3 v33_verify_frontend_certificate.py --certificate v33_frontend_certificate.json`

A.4 Expected verifier output: $m = 10$ (verbatim)

```
LHS_hi =
    0.036946013258747960323533969194609059995630327458382820910011029509013094932490047158355836
RHS_lo =
    0.129965096347944215724970679716013192260392769855133829588479000426675277738893819116146351
STRICT (LHS_hi < RHS_lo) = True
PASS
```

A.5 Bundle files (verbatim)

```json
{
  "certificate_version": "v33",
  "created_utc": "2026-01-21T00:00:00Z",
  "m_band": "10",
  "eta": "1e-14",
  "alpha_worst": "1",
  "kappa": "0.05",
  "intervals": {
    "C1": {
      "lo": "15.1",
      "hi": "15.2"
    },
    "C2": {
      "lo": "37.3",
      "hi": "37.4"
    },
    "C_up": {
      "lo": "1100",
      "hi": "1100.5"
    },
    "C_hpp": {
      "lo": "1100",
      "hi": "1100.5"
    }
  },
  "notes": [
    "Demo-only intervals carried forward from v31-style scaffolding; replace with audit-proven
    enclosures when G-1/G-12 are closed.",
    "The verifier/generator implement directed-rounding interval arithmetic with Python's decimal
    module.",
    "The local-window majorant N_up(m)=1.01*log(m)+17 is hard-coded from Lemma Nloc-logm in
    manuscript_v33."
  ]
}
```

```json
{
  "certificate_version": "v33",
  "m_band": "10",
  "eta": "1e-14",
  "alpha": "1",
  "kappa": "0.05",
  "prec": 90,
  "pi_interval": {
    "lo": "3.14159265358979323846264338327950288419716939937510",
    "hi": "3.14159265358979323846264338327950288419716939937511"
  },
  "logm_interval": {
    "lo":
    "2.302585092994045684017991454684364207601101488628772976033327900967572609677352480235999721",
    "hi":
    "2.302585092994045684017991454684364207601101488628772976033327900967572609677352480235999721"
  },
  "delta_interval": {
    "lo":
    "1.886116970116139292199608299650608736659005451762204889419088795910853616229630107611974
    68E
    -15",
```

```
      "hi":
      "1.88611697011613929219960829965060873665900545176220488941908879591085361622963010761197469E
      -15"
    },
    "L_interval": {
      "lo":
      "72.06903490421008982867167096573389953477663247829447193810325130461034640612800224515635578",
      "hi":
      "72.39929341350949439707347011120233595553674262715734923570658409470710366709575769958871576"
    },
    "Nup_interval": {
      "lo":
      "19.32561094392398614085817136923120784967711250351506070579366117997724833577741260050383571",
      "hi":
      "19.32561094392398614085817136923120784967711250351506070579366117997724833577741260050383572"
    },
    "kappa_interval": {
      "lo": "0.05",
      "hi": "0.05"
    },
    "lhs_interval": {
      "lo":
      "0.036929227246363249695767376248112465003163963048376923276890330108202426654631411759145525",

      "hi":
      "0.036946013258747960323353396919460905999563032745838282091001102950901309493249004715835836"
    },
    "rhs_interval": {
      "lo":
      "0.129965096347944215724970679716013192260392769855133829588479000426675277738893819116146351",
      "hi":
      "0.129965096347948199005209691457697222838229716361348668790498959759558243588339370271250251"
    },
    "derived_constants": {
      "ln2_interval": {
        "lo":
      "0.693147180559945309417232121458176568075500134360255254120680009493393621969694715605863327",
        "hi":
      "0.693147180559945309417232121458176568075500134360255254120680009493393621969694715605863327"
      },
      "c_interval": {
        "lo":
      "0.129965096354989745515731022773408106514156275192547860147627501780011304119317759176099373",
        "hi":
      "0.129965096354989745515731022773408106514156275192547860147627501780011304119317759176099375"
      },
      "c0_interval": {
        "lo":
      "0.0827383500572443475236711620442491341185086557736206913728528561387020242248387512851407512",

        "hi":
      "0.0827383500572443475236711620442491341185086557736209547372007536994885577445868650239268751"
      },
      "Kalloc_interval": {
        "lo":
      "16.8564064605510183482195707320469789355424420304830450244464558356154641352704002966491695",
        "hi":
      "16.8564064605510183482195707320469789355424420304830450244464558356154641352704002966491696"
      }
```

```
    },
    "pass": true
}




#!/usr/bin/env python3
"""
v33_generate_tail_certificate.py

Deterministically generates v33_tail_certificate_m10.json from v33_constants_m10.json using
directed-rounding interval arithmetic implemented with Python's decimal module.

This generator is intended to be auditable: no network access, no randomness, and no external
    libraries.

Tail inequality certified (for given inputs):
  LHS(delta) < RHS(delta), where
  LHS(delta) = 2*C_up*( delta^(3/2)*L(m) + delta^(1/2)*N_up(m)/kappa )
  RHS(delta) = c - delta*( Kalloc*c0*L(m) + C_hpp*(log(m)+1) )

with
  L(m)    = C1*log(m) + C2,
  N_up(m) = 1.01*log(m) + 17,
  c   = (3 ln 2)/16,
  c0 = (3 ln 2)/(8 pi),
  Kalloc = 3 + 8*sqrt(3).

Usage:
  python3 v33_generate_tail_certificate.py v33_constants_m10.json v33_tail_certificate_m10.json
"""

import json
import sys
from dataclasses import dataclass
from decimal import Decimal, getcontext, localcontext, ROUND_FLOOR, ROUND_CEILING


# ---- Fixed enclosure for pi (50 decimal places) ----
# pi = 3.14159265358979323846264338327950288419716939937510...
PI_LO = Decimal("3.14159265358979323846264338327950288419716939937510")
PI_HI = Decimal("3.14159265358979323846264338327950288419716939937511")


@dataclass
class Interval:
    lo: Decimal
    hi: Decimal

    def __post_init__(self) -> None:
        if self.lo > self.hi:
            raise ValueError(f"Bad interval: {self.lo} > {self.hi}")


def ctx(prec: int, rounding):
    c = getcontext().copy()
    c.prec = prec
    c.rounding = rounding
```

```python
        return c


def iv(lo: str, hi: str | None = None) -> Interval:
    if hi is None:
        hi = lo
    return Interval(Decimal(lo), Decimal(hi))


def add(a: Interval, b: Interval, prec: int) -> Interval:
    with localcontext(ctx(prec, ROUND_FLOOR)):
        lo = a.lo + b.lo
    with localcontext(ctx(prec, ROUND_CEILING)):
        hi = a.hi + b.hi
    return Interval(lo, hi)


def sub(a: Interval, b: Interval, prec: int) -> Interval:
    with localcontext(ctx(prec, ROUND_FLOOR)):
        lo = a.lo - b.hi
    with localcontext(ctx(prec, ROUND_CEILING)):
        hi = a.hi - b.lo
    return Interval(lo, hi)


def mul(a: Interval, b: Interval, prec: int) -> Interval:
    with localcontext(ctx(prec, ROUND_FLOOR)):
        cands_lo = [a.lo*b.lo, a.lo*b.hi, a.hi*b.lo, a.hi*b.hi]
        lo = min(cands_lo)
    with localcontext(ctx(prec, ROUND_CEILING)):
        cands_hi = [a.lo*b.lo, a.lo*b.hi, a.hi*b.lo, a.hi*b.hi]
        hi = max(cands_hi)
    return Interval(lo, hi)


def div(a: Interval, b: Interval, prec: int) -> Interval:
    if b.lo <= 0 <= b.hi:
        raise ZeroDivisionError("Interval division by an interval containing 0.")
    with localcontext(ctx(prec, ROUND_FLOOR)):
        rlo = Decimal(1) / b.hi
    with localcontext(ctx(prec, ROUND_CEILING)):
        rhi = Decimal(1) / b.lo
    return mul(a, Interval(rlo, rhi), prec)


def sqrt(a: Interval, prec: int) -> Interval:
    if a.lo < 0:
        raise ValueError("sqrt of negative interval")
    with localcontext(ctx(prec, ROUND_FLOOR)):
        lo = a.lo.sqrt()
    with localcontext(ctx(prec, ROUND_CEILING)):
        hi = a.hi.sqrt()
    return Interval(lo, hi)


def ln(a: Interval, prec: int) -> Interval:
    if a.lo <= 0:
        raise ValueError("ln of nonpositive interval")
    with localcontext(ctx(prec, ROUND_FLOOR)):
```

```
        lo = a.lo.ln()
    with localcontext(ctx(prec, ROUND_CEILING)):
        hi = a.hi.ln()
    return Interval(lo, hi)


def pow_3_2(a: Interval, prec: int) -> Interval:
    return mul(a, sqrt(a, prec), prec)


def compute(constants: dict, prec: int = 90) -> dict:
    m = iv(constants["m_band"])
    eta = iv(constants["eta"])
    alpha = iv(constants["alpha_worst"])
    kappa = iv(constants["kappa"])

    C1 = iv(constants["intervals"]["C1"]["lo"], constants["intervals"]["C1"]["hi"])
    C2 = iv(constants["intervals"]["C2"]["lo"], constants["intervals"]["C2"]["hi"])
    Cup = iv(constants["intervals"]["C_up"]["lo"], constants["intervals"]["C_up"]["hi"])
    Chpp = iv(constants["intervals"]["C_hpp"]["lo"], constants["intervals"]["C_hpp"]["hi"])

    logm = ln(m, prec)
    delta = div(mul(eta, alpha, prec), mul(logm, logm, prec), prec)

    # L(m) = C1*logm + C2
    L = add(mul(C1, logm, prec), C2, prec)

    # N_up(m) = 1.01*logm + 17
    Nup = add(mul(iv("1.01"), logm, prec), iv("17"), prec)

    # ln 2
    ln2 = ln(iv("2"), prec)

    # c = (3 ln 2)/16
    c = div(mul(iv("3"), ln2, prec), iv("16"), prec)

    # c0 = (3 ln 2)/(8 pi), pi enclosed
    pi = Interval(PI_LO, PI_HI)
    c0 = div(mul(iv("3"), ln2, prec), mul(iv("8"), pi, prec), prec)

    # Kalloc = 3 + 8 sqrt(3)
    sqrt3 = sqrt(iv("3"), prec)
    Kalloc = add(iv("3"), mul(iv("8"), sqrt3, prec), prec)

    logm_plus1 = add(logm, iv("1"), prec)

    # local term: delta^(1/2) * Nup / kappa
    local_term = mul(sqrt(delta, prec), div(Nup, kappa, prec), prec)

    # residual term: delta^(3/2) * L
    residual_term = mul(pow_3_2(delta, prec), L, prec)

    lhs = mul(mul(iv("2"), Cup, prec), add(residual_term, local_term, prec), prec)

    # RHS = c - delta*(Kalloc*c0*L + Chpp*(logm+1))
    term1 = mul(mul(Kalloc, c0, prec), L, prec)
    term2 = mul(Chpp, logm_plus1, prec)
    rhs = sub(c, mul(delta, add(term1, term2, prec), prec), prec)
```

```python
        passed = (lhs.hi < rhs.lo)

        return {
            "prec": prec,
            "pi_interval": {"lo": str(PI_LO), "hi": str(PI_HI)},
            "logm_interval": {"lo": str(logm.lo), "hi": str(logm.hi)},
            "delta_interval": {"lo": str(delta.lo), "hi": str(delta.hi)},
            "L_interval": {"lo": str(L.lo), "hi": str(L.hi)},
            "Nup_interval": {"lo": str(Nup.lo), "hi": str(Nup.hi)},
            "kappa_interval": {"lo": str(kappa.lo), "hi": str(kappa.hi)},
            "lhs_interval": {"lo": str(lhs.lo), "hi": str(lhs.hi)},
            "rhs_interval": {"lo": str(rhs.lo), "hi": str(rhs.hi)},
            "derived_constants": {
                "ln2_interval": {"lo": str(ln2.lo), "hi": str(ln2.hi)},
                "c_interval": {"lo": str(c.lo), "hi": str(c.hi)},
                "c0_interval": {"lo": str(c0.lo), "hi": str(c0.hi)},
                "Kalloc_interval": {"lo": str(Kalloc.lo), "hi": str(Kalloc.hi)},
            },
            "pass": bool(passed),
        }


def main() -> int:
    if len(sys.argv) != 3:
        print("Usage: v33_generate_tail_certificate.py constants.json tail_certificate.json", file=
    sys.stderr)
        return 2

    with open(sys.argv[1], "r", encoding="utf-8") as f:
        constants = json.load(f)

    out = {
        "certificate_version": "v33",
        "m_band": constants["m_band"],
        "eta": constants["eta"],
        "alpha": constants["alpha_worst"],
        "kappa": constants["kappa"],
    }
    out.update(compute(constants, prec=90))

    with open(sys.argv[2], "w", encoding="utf-8") as f:
        json.dump(out, f, indent=2)

    print("[generate] wrote", sys.argv[2])
    print("[generate] PASS =", out["pass"])
    print("[generate] lhs_interval.hi =", out["lhs_interval"]["hi"])
    print("[generate] rhs_interval.lo =", out["rhs_interval"]["lo"])
    return 0


if __name__ == "__main__":
    raise SystemExit(main())




#!/usr/bin/env python3
"""
v33_verify_tail_certificate.py
```

```
Verifier for v33_tail_certificate_m10.json. This script:
- loads the constants JSON and the pinned certificate JSON
- regenerates the certificate from constants
- checks exact JSON equality on the computed fields
- reports PASS/FAIL and prints the strict-separation check LHS_hi < RHS_lo.

Usage:
  python3 v33_verify_tail_certificate.py --constants v33_constants_m10.json --certificate
    v33_tail_certificate_m10.json

Exit codes:
- 0 on PASS
- nonzero on FAIL
"""

from __future__ import annotations

import argparse
import json
import sys

from v33_generate_tail_certificate import compute


def main() -> int:
    ap = argparse.ArgumentParser(description="Verify v33 tail certificate (m=10).")
    ap.add_argument("--constants", required=True, help="Path to v33_constants_m10.json")
    ap.add_argument("--certificate", required=True, help="Path to v33_tail_certificate_m10.json")
    args = ap.parse_args()

    with open(args.constants, "r", encoding="utf-8") as f:
        constants = json.load(f)

    with open(args.certificate, "r", encoding="utf-8") as f:
        cert = json.load(f)

    regen = {
        "certificate_version": "v33",
        "m_band": constants["m_band"],
        "eta": constants["eta"],
        "alpha": constants["alpha_worst"],
        "kappa": constants["kappa"],
    }
    regen.update(compute(constants, prec=90))

    # Compare all keys that regen produces (ignore any extra keys in cert)
    ok = True
    for k, v in regen.items():
        if cert.get(k) != v:
            ok = False
            print(f"MISMATCH key={k}")
            print("  cert :", cert.get(k))
            print("  regen:", v)

    lhs_hi = regen["lhs_interval"]["hi"]
    rhs_lo = regen["rhs_interval"]["lo"]
    strict = (float(lhs_hi) < float(rhs_lo))
```

```
        print("LHS_hi =", lhs_hi)
        print("RHS_lo =", rhs_lo)
        print("STRICT (LHS_hi < RHS_lo) =", strict)

        if not ok or not strict or not regen["pass"]:
            print("FAIL")
            return 1

        print("PASS")
        return 0


if __name__ == "__main__":
    raise SystemExit(main())
```

B Finite-height front-end certificate (literature-based)

The required front-end is RH up to height $H_0 = 5$. We record a discharge using Platt–Trudgian's published verification of RH up to $3 \cdot 10^{12}$.

```
{
  "certificate_version": "v33",
  "created_utc": "2026-01-21T00:00:00Z",
  "needed_frontend_statement": {
    "type": "RH_to_height",
    "H0": 5.0,
    "text": "All nontrivial zeros rho=beta+i gamma with 0<gamma<=H0 satisfy beta=1/2."
  },
  "discharged_by": {
    "type": "literature_citation",
    "verification_height": 3000000000000.0,
    "reference": {
      "authors": "D. J. Platt and T. S. Trudgian",
      "title": "The Riemann hypothesis is true up to 3*10^12",
      "venue": "Bulletin of the London Mathematical Society",
      "year": 2021,
      "doi": "10.1112/blms.12460",
      "arxiv": "2004.09765",
      "statement": "All zeros beta+i gamma with 0<gamma<=3*10^12 satisfy beta=1/2 (rigorous
    interval arithmetic)."
    },
    "logic": "If RH holds for 0<gamma<=H_cited and H0<=H_cited, then RH holds for 0<gamma<=H0."
  },
  "notes": [
    "This JSON is not itself a computation of zeros; it is a pinned statement+reference used by v31
    .",
    "For a fully self-contained proof without external computational input, one would need to
    implement and certify an argument-principle zero count in this region using ball arithmetic (
    not provided here)."
  ]
}
```

```
H0 (needed) = 5.0
H_cited     = 3000000000000.0
CHECK: H0 <= H_cited : True
PASS
```

```python
#!/usr/bin/env python3
"""v33_generate_frontend_certificate.py

Creates a pinned JSON certificate for the finite-height front-end assumption used by v33.

This script does NOT compute zeta zeros. It encodes a minimal (H0, citation) logic statement:
if RH has been verified up to H_cited and H0 <= H_cited, then RH holds up to height H0.

Usage:
  python3 v33_generate_frontend_certificate.py v33_frontend_certificate.json
"""

from __future__ import annotations

import json
from datetime import datetime, timezone
import sys


def main() -> int:
    if len(sys.argv) != 2:
        print("Usage: v33_generate_frontend_certificate.py output.json", file=sys.stderr)
        return 2

    out = {
        "certificate_version": "v33",
        "created_utc": datetime.now(timezone.utc).strftime("%Y-%m-%dT%H:%M:%SZ"),
        "needed_frontend_statement": {
            "type": "RH_to_height",
            "H0": 5.0,
            "text": "All nontrivial zeros rho=beta+i gamma with 0<gamma<=H0 satisfy beta=1/2."
        },
        "discharged_by": {
            "type": "literature_citation",
            "verification_height": 3e12,
            "reference": {
                "authors": "D. J. Platt and T. S. Trudgian",
                "title": "The Riemann hypothesis is true up to 3*10^12",
                "venue": "Bulletin of the London Mathematical Society",
                "year": 2021,
                "doi": "10.1112/blms.12460",
                "arxiv": "2004.09765",
                "statement": "All zeros beta+i gamma with 0<gamma<=3*10^12 satisfy beta=1/2 (
    rigorous interval arithmetic)."
            },
            "logic": "If RH holds for 0<gamma<=H_cited and H0<=H_cited, then RH holds for 0<gamma<=
    H0."
        },
        "notes": [
            "This JSON is not itself a computation of zeros; it is a pinned statement+reference
    used by v33.",
```

```
            "For a fully self-contained proof without external computational input, one would need
        to implement and certify an argument-principle zero count in this region using ball arithmetic
        (not provided here)."
            ]
    }

    with open(sys.argv[1], "w", encoding="utf-8") as f:
        json.dump(out, f, indent=2)

    print("[generate] wrote", sys.argv[1])
    return 0


if __name__ == "__main__":
    raise SystemExit(main())


#!/usr/bin/env python3
"""v33_verify_frontend_certificate.py

Verifier for the front-end certificate JSON produced by v33_generate_frontend_certificate.py.

This verifier checks the internal logic only:
- parses the JSON
- confirms that the required finite-height H0 is <= the cited verification height

It does NOT re-run the cited large-scale computation (Platt--Trudgian); that result is treated as
    an
external, peer-reviewed input in the manuscript.

Usage:
  python3 v33_verify_frontend_certificate.py --certificate v33_frontend_certificate.json

Exit codes:
- 0 on PASS
- nonzero on FAIL
"""

from __future__ import annotations

import argparse
import json


def main() -> int:
    ap = argparse.ArgumentParser(description="Verify v33 front-end certificate JSON (internal logic
        only).")
    ap.add_argument("--certificate", required=True, help="Path to v33_frontend_certificate.json")
    args = ap.parse_args()

    with open(args.certificate, "r", encoding="utf-8") as f:
        cert = json.load(f)

    needed = cert.get("needed_frontend_statement", {})
    discharged = cert.get("discharged_by", {})

    H0 = float(needed.get("H0"))
    Hc = float(discharged.get("verification_height"))

    ok = H0 <= Hc
```

```
    print("H0 (needed) =", H0)
    print("H_cited     =", Hc)
    print(f"CHECK: H0 <= H_cited : {ok}")

    if not ok:
        print("FAIL")
        return 1

    print("PASS")
    return 0


if __name__ == "__main__":
    raise SystemExit(main())
```

References

References

[1] R. Coifman, A. McIntosh, and Y. Meyer, *L'intégrale de Cauchy définit un opérateur borné sur L^2 pour les courbes lipschitziennes*, Annals of Mathematics (2) **116** (1982), no. 2, 361–387.

[2] T. A. Driscoll and L. N. Trefethen, *Schwarz–Christoffel Mapping*, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, 2002.

[3] P. L. Duren, *Theory of H^p Spaces*, Academic Press, 1970.

[4] J. B. Garnett, *Bounded Analytic Functions*, Graduate Texts in Mathematics, Springer, 2007.

[5] A. Ivić, *The Riemann Zeta-Function: Theory and Applications*, Wiley-Interscience, 1985.

[6] E. C. Titchmarsh, *The Theory of the Riemann Zeta-Function*, 2nd ed., revised by D. R. Heath-Brown, Oxford University Press, 1986.

[7] A. Bellotti and T. Wong, *An improved explicit bound on the argument of the Riemann zeta function on the critical line*, arXiv:2412.15470v2 (2024).

[8] D. Platt and T. Trudgian, *The Riemann hypothesis is true up to $3 \cdot 10^{12}$*, Bulletin of the London Mathematical Society **53** (2021), no. 3, 792–797.