The de Bruijn–Newman constant: three conditional upper bounds

READING LEVEL EVERYONE · EXPERTS

Three conditional upper bounds, computer-assisted, checked within this project; not refereed.

proved in writing 10 · conditional 3 · computed 3 · open 2 · cited 1

CHECK IT YOURSELF

The check guide for this book · How to check a result yourself

---

What a match shows: a match tells you that the published files really produce the published numbers. It doesn't tell you that the question was the right one to ask, or that a written argument around the numbers is complete. Each recipe says plainly what its steps show and what they don't. For a result whose evidence is logs (logs-only), a match checks the bookkeeping of the logs, not the search that wrote them.

Cite: Sonny (2026). The de Bruijn–Newman constant: three conditional upper bounds (preliminary). Go check yourself. https://gocheckyourself.org/read/riemann-debruijn-newman/everyone/

1. Start here
2. The simple version
3. The question
4. Conditions
5. Statement
6. Theorems in this book
7. Verify: conditional upper bounds on the de Bruijn–Newman constant
8. Files and fingerprints
9. Trust and verification
10. People and works
11. Files and rights
12. Articles

1 Start here

Sonny

What this book shows

- My result. Three computer-assisted certificates (computer-checked files that show a bound holds). Here Λ is the de Bruijn–Newman constant (how much a "warming" of the zeta function is enough to make all its zeros line up): $\Lambda \le 17/100 = 0.170$, $\Lambda \le 343/2000 = 0.1715$ and $\Lambda \le 43/250 = 0.172$. Each follows from Polymath15's criterion (Polymath15 was a large online collaboration; its criterion turns one positivity check at a given height into a bound on Λ), Platt and Trudgian's verified height (how far up the zeros of zeta have been checked by computer), the inherited pipeline (the chain of computer steps I inherited from Jude Gomila) and my changed proof steps (nine, five and four). CONDITIONAL on the listed conditions. Eight of the steps are PROVED as scoped, with written arguments; the ninth, a program, is COMPUTED.

- Every passed-on number checked. Every number one stage passes to the next is rounded in its safe direction from its exact binary value, and all 76 final gates (final checks) recompose positive in exact fractions. Six written recipes let any reader recheck this in seconds.
- Why it matters. Rodgers and Tao proved $\Lambda \ge 0$, and $\Lambda = 0$ is the Riemann hypothesis (the long-standing claim that the special numbers called the zeros of the zeta function all sit on one line); the smallest refereed upper bound (one checked by independent experts before publication) is 0.2. These certificates rest on the same published inputs as that 0.2, Polymath15's criterion and Platt and Trudgian's height, with each extra assumption named.
- In Lean: nothing in this book is Lean-proved (Lean is a proof-checking program that tests every step of a proof against fixed rules).
- Every theorem in this book is listed in the theorem library.

The question

How far can the heat flow of the Riemann xi function (a way to "warm up" the zeta function a little at a time, like heat spreading through a metal bar) be run before its zeros are guaranteed to be real? That distance is a single number, the de Bruijn–Newman constant, written Λ.

Here is the setting in plain words. The Riemann hypothesis is about the zeros of the zeta function. In 1950 N. G. de Bruijn built a family of functions H_t, one for each real number t, from a Fourier transform (a way of breaking a function into waves) of the zeta function's completed form. At $t = 0$ the zeros of H_0 correspond to the non-trivial zeros of zeta, and the Riemann hypothesis says they are all real. De Bruijn showed that if all the zeros of H_t are real, then so are those of H_s for every $s \geq t$. In 1976 Charles Newman showed that there is a single threshold: all the zeros of H_t are real exactly when t is at least that threshold. He conjectured that it is not negative. The threshold Λ is the smallest value such that all zeros of H_t are real for every $t \geq \Lambda$.

The question this book works on is a bound: how small can we show Λ to be? Polymath15 (2019) proved $\Lambda \leq 0.22$, and with the verified height of Platt and Trudgian (2021) this became $\Lambda \leq 0.2$, the smallest refereed bound. Our computer-assisted certificates give $\Lambda \leq 0.172$, $\Lambda \leq 0.1715$ and $\Lambda \leq 0.170$, each conditional on a list of conditions in the statement files.

In short: Λ is at least 0 (Rodgers and Tao) and at most 0.2 by refereed work. Others have posted smaller figures that are not refereed, several of them lower than ours, some using a recent unrefereed result that zeta has no zeros with real part above $7/8$ (all credited on the credits page). Whether Λ equals 0 is the Riemann hypothesis, and it is open.

Where to go next. How is Λ defined, and why is it tied to the zeros of zeta? See the chapter on the question. What did earlier work show about it? See the same chapter.

Why it matters

Rodgers and Tao (2020) proved that $\Lambda \geq 0$. The Riemann hypothesis is the statement $\Lambda \leq 0$, so it is equivalent to $\Lambda = 0$. Every upper bound on Λ therefore says how close the Riemann hypothesis is to being forced by this route: a bound $\Lambda \leq 0.170$ would mean that the zeros of H_t are real for all $t \geq 0.170$, and the hypothesis asks for the same at $t = 0$. De Bruijn gave $\Lambda \leq 1/2$ in 1950; the bound has come down in steps since.

Where to go next. Why does a positive bound not decide the Riemann hypothesis? See the chapter on the question.

What answering it would help with

The bounds and certificate format give the following ways to examine the work.

- A smaller upper bound narrows the range of t in which a zero off the real line could still exist. This measures how much room remains between known results and the Riemann hypothesis.
- The method (a criterion that turns a positivity statement at one height into a bound, checked by large exact computations) carries over to others who run similar certificates for related questions.
- The stated bounds constrain positive values of Λ. Still open: the case $\Lambda = 0$, which is the Riemann hypothesis.

Where to go next. What does a certificate consist of? See the statement of the three bounds.

What our research shows

These are the results with their exact labels.

What this is

COMPUTER-ASSISTED CERTIFICATES THAT Λ IS AT MOST 0.170 (17/100), WITH TWO FURTHER CERTIFICATES AT 0.1715 (343/2000) AND 0.172 (43/250) (SONNY, 2026). STATUS CONDITIONAL FOR EACH: EVERY ONE RESTS ON

THE PUBLISHED PREMISES, ON THE INHERITED PIPELINE (WHOSE OWN REFEREE QUESTIONS ARE OPEN) AND ON THE CHANGED PROOF STEPS (SONNY, 2026) LISTED IN THE CONDITIONS FILE. EIGHT OF THOSE STEPS ARE PROVED AS SCOPED, WITH THEIR WRITTEN ARGUMENTS IN THE PROOFS FOLDER, READ AND RE-DERIVED WITHIN THIS PROJECT; THE NINTH, A PROGRAM, HAD ALL 13,808,993 OF ITS ROWS RECOMPUTED BY A SECOND PROGRAM. NOBODY OUTSIDE THIS PROJECT HAS READ THEM.

Still open

AN OUTSIDE READING OF THE CHANGED STEPS; THE INHERITED PIPELINE'S OWN REFEREE QUESTIONS; AND THE RIEMANN HYPOTHESIS ITSELF (THE CASE $\Lambda = 0$). SMALLER BOUNDS HAVE BEEN POSTED BY OTHERS; THE CREDITS PAGE LISTS THEM.

Checked by

SIX WRITTEN RECIPES (IN THE VERIFICATION GUIDE) THAT RE-VERIFY, IN SECONDS AND IN EXACT FRACTIONS, HOW THE STORED RESULTS JOIN, INCLUDING THE SAFE-DIRECTION ROUNDING OF EVERY NUMBER PASSED BETWEEN STAGES; THE FULL RUNS TOOK ONE TO TWO HOURS ON A SERVER AND ARE NOT REDONE HERE; THE NINE CHANGED STEPS, WHICH ARE MY OWN CHANGES TO THE INHERITED ARGUMENT, WERE READ WITHIN THIS PROJECT (DETAILS IN THE CONDITIONS FILE); SUPPORTING FILES AVAILABLE ON REQUEST.

Where to go next. Why is every bound conditional, and which steps need an outside reader? See the list of conditions.

<!--BOX-->

What's in this book

Counts are computed from the book's statement file (its list of results: 17 results, each with one status).

Status	Count
Proved and checked in Lean	0
Proved, with the written argument shipped in the proofs folder (eight changed steps: correction cap, no common-phase credit, wall time guard, total height rule, block charge, mollifier tables, handover, raised input caps; read and re-derived within this project, three of them also in separate blind sessions; and the directed export of numbers; not refereed)	9
Conjectured	0
Computed or certified (stored server runs and the arithmetic that joins them, 2; and the finite-row program, whose 13,808,993 rows at 0.170 were all recomputed by a second program written within this project, not shipped, 1)	3
Conditional (the three bounds)	3
Open	2

Item	Count
Chapters	1
Proof files	14, with an index and a fingerprint list
Statement files	3
Written check recipes	6, plus the fingerprint check
Certificate files	42 in 4 folders (0.170: 13; 0.1715: 13; 0.172: 13; directed: 3)
Lean files	0

<!--/BOX-->

Which bound to read

- 0.170 is the lowest value. It rests on nine changed proof steps, eight with written arguments in the proofs folder and the ninth, the finite-row program, recomputed row by row by a second program; its margins are thin.
- 0.1715 rests on five steps, each read and re-derived within this project.
- 0.172 rests on four steps, each read and re-derived within this project. It has the shortest list of conditions, so start there if you want to check the conditions.

All three lists are in the conditions file. Smaller figures posted by others exist, and are credited on the credits page.

Yesterday, today, tomorrow

Yesterday. N. G. de Bruijn (1950) and Charles Newman (1976) defined the constant that bears their names, and Rodgers and Tao (2020) proved it is at least 0, so that the Riemann hypothesis is the case where it equals 0. Polymath15 (2019) proved it is at most 0.22 by a criterion and an effective approximation, Platt and Trudgian (2021) verified the Riemann hypothesis to a large height, which gave 0.2, and Jude Gomila (2026) published a pipeline stating 0.1787854, which he labels not yet peer reviewed.

Today. I ran computer-assisted certificates, with Polymath15's criterion and the inherited pipeline, for the upper bounds 43/250 (0.172), 343/2000 (0.1715) and 17/100 (0.170). Every stage of each run passed, with interval arithmetic (every number carries a guaranteed lower and upper bound) used where the stages demand it, and every number passed from one stage to the next is rounded in its safe direction from its exact binary value, with the final gates recomposed in exact fractions. Each bound is conditional on the published premises, on the inherited pipeline's open referee questions, and on the changed proof steps, which are my own changes to the inherited argument: four for 0.172, five for 0.1715 and nine for 0.170, with one more condition for the last two, the frozen-radius condition (Sonny's argument shows it holds for every $y > 0$). All nine steps were read within this project against their write-ups and the Polymath15 text and found correct as scoped. The margins at 0.170 are thin (the room to spare is about two parts in ten thousand; plain rows 2.2×10^{-4}, tail ratio 0.99558). No step has been read by anyone outside this project. Others have posted smaller figures, not refereed: Mosaic Intelligence (0.1595), Stefan Gordon (0.157657), Dale James Raymond (0.1494, and 0.108 using a recent unrefereed result that zeta has no zeros with real part above 7/8) and Cetin Hakimoglu-Brown (0.13963, conditional on that same result); ours do not use that input. The Riemann hypothesis ($\Lambda = 0$) is open.

Tomorrow. A reader who examines the changed steps could turn a conditional bound into a firmer one or point to what needs repair; further research is recommended. The bound 0.172 has the shortest list of conditions,

so it is the natural place to start. The stored results may be reused by others who run similar certificates.

Where to go next. Why does 0.172 have the shortest list of conditions? See the list of conditions. What are the stored numbers? See the statement of the three bounds.

Check it yourself

What ships: the stage result files for each bound, the safe-direction list of every exported number, the written arguments for the changed steps (the proofs folder), a SHA-256 fingerprint (a 64-character code computed from a file; change one bit and the code changes) for each file, and a written recipe for each check. With these a reader can recheck how the stored results join and read the argument behind each changed step. What is not shipped: the stage programs (the full runs took one to two hours on a server and are not redone here) and the second program that recomputed the 0.170 finite rows. The recipes check the arithmetic of the stored results, not the stages. A recipe gives the exact file format, the exact condition to test, the expected answer and a standard tool or a few lines of code that any reader can write. The recipes are in the verification guide, which starts with the fingerprint check. To compare file fingerprints, any SHA-256 tool will do:

```
sha256sum -c SHA256SUMS.txt  # Linux
shasum -a 256 -c SHA256SUMS.txt   # macOS
CertUtil -hashfile <file> SHA256
```

The upper command is run inside the certificates folder on Linux (macOS: the shasum line); the lower one does the same for one file on Windows. No Lean proof is shipped in this book.

The methods are not part of this book. The files listed above ship; the files named there as not shipped are available on request. Questions about methods or tools: write to hello@gocheckyourself.org. Request access

How this research was done: https://gocheckyourself.org/about/#how

Where to go next. What does a pass of each check show, and what does it leave open? See the verification guide.

The chapters

- The question

The simple version tells the same story without equations.

The statement files are the statement of the three bounds and the list of conditions. The verification guide has the recipes, and the credits page lists the work this stands on.

We stand on the shoulders of giants: Polymath15 (2019) for the method, Rodgers and Tao for $\Lambda \geq 0$, Platt and Trudgian for the verified height, and Jude Gomila for the pipeline we inherited.

Where to go next

- Which changed steps does a bound rest on, and what did the reading find? See the list of conditions.
- How do the stage results fit together into one bound? See the statement of the three bounds.
- What can you recheck in a few seconds? See the verification guide.

Go check yourself.

Have a question about this? Ask it

2 The simple version

The simple version

Ten minutes, no equations. Each part has a short in plain words section and a short for experts note you can skip.

The question

In plain words. The Riemann hypothesis says that certain special numbers, the zeros of the zeta function, all sit on one line. In 1950 N. G. de Bruijn found a way to "warm up" the zeta function a little at a time, like heat spreading through a metal bar. The more you warm it, the more its zeros line up. Charles Newman showed that there is a point past which they are all certainly lined up, and called that point Λ. If Λ were 0, no warming would be needed: that is exactly the Riemann hypothesis. So the question is: how little warming is enough?

For experts. Λ is the de Bruijn–Newman constant: the least real number such that all zeros of H_t are real for every $t \geq \Lambda$. Rodgers and Tao (2020) proved $\Lambda \geq 0$, so the Riemann hypothesis is equivalent to $\Lambda = 0$.

What was already known

In plain words. De Bruijn showed that a warming of $1/2$ is enough. A large online collaboration, Polymath15, brought that down to 0.22 in 2019, and with a huge check of zeta's zeros by Dave Platt and Tim Trudgian it came down to 0.2. Brad Rodgers and Terry Tao showed that it is at least 0. Recently several people have posted smaller numbers that referees have not checked, some of them using a recent result, also unrefereed, about where the zeros of zeta cannot be.

For experts. Refereed: $\Lambda \leq 0.22$ (Polymath15, 2019) and $\Lambda \leq 0.2$ (Platt and Trudgian, 2021). Posted and not refereed: figures down to 0.1595 on the same route, and 0.13963 and 0.108 using a zero-free region $\mathrm{Re}(s) > 7/8$. All are credited on the credits page.

What I did

In plain words. I ran three very large computer checks that show a warming of 0.172, 0.1715 or 0.170 is enough, provided some listed conditions hold. Each check is a long chain of computer steps. Every step passed. Every number that one step hands to the next is rounded the safe way, so rounding can never make the answer look better than it is. To make the checks reach

this far, I changed nine steps of the argument I inherited from Jude Gomila's pipeline, and wrote down why each change is sound.

For experts. Three conditional certificates, $\Lambda \le 43/250$, $343/2000$ and $17/100$, by the Polymath15 criterion $\Lambda \le \mathrm{T} + \mathrm{y0}^2/2$ with the Platt–Trudgian height, the inherited pipeline and nine changed steps (four for 0.172, five for 0.1715, nine for 0.170). Eight are PROVED as scoped with written arguments; the ninth, the finite-row program, is COMPUTED (all 13,808,993 rows at 0.170 recomputed by a second program). Every exported binary64 endpoint is passed on rounded in its safe direction, and every final gate is recomposed in exact fractions. CONDITIONAL on the conditions file.

How sure it is

In plain words. It is checked within this project but not by outside experts. It also leans on the inherited pipeline, which has open questions of its own that its author lists. At 0.170 the room to spare is thin: about two parts in ten thousand. So read these as "true if the listed conditions hold", and further research is recommended.

For experts. The inherited chain's referee questions (I1 to I7 in the conditions file) are OPEN and stay so after any reading of the changed steps. Smallest margins at 0.170: plain 2.17×10^{-4}, tail ratio 0.99558.

What you can check yourself

In plain words. Every result file has a fingerprint, and six short recipes show how to recheck, with exact fractions, that the stored results fit together. Any reader with a laptop can do it in seconds. The recipes do not redo the long computer runs.

For experts. Recipes 1 to 6 in the verification guide: tuple and height, finite-row tiling, the correction cap at 80 digits, final margins, stage signs, and the directed export of every number passed on.

What is open

In plain words. Whether Λ is exactly 0, which is the Riemann hypothesis itself. A smaller upper bound does not decide it.

For experts. $\Lambda = 0$ is OPEN. These bounds do not use, and say nothing about, the $\mathrm{Re}(s) > 7/8$ input.

Experts: the files and checks are here: verification guide.

3 The question

The question

The question

What is the de Bruijn–Newman constant Λ, what was known about it, and what does this book add?

Why it matters

The Riemann hypothesis is the statement $\Lambda = 0$, and $\Lambda \geq 0$ is proved, so every proved upper bound on Λ says how close this route comes to the hypothesis.

What answering it would help with

A smaller upper bound narrows the range of t in which a zero off the real line could still exist. It does not decide the Riemann hypothesis, and nothing here claims to.

What our research shows

The constant

The Riemann hypothesis is about the zeros of the zeta function. De Bruijn (1950) built a family of functions H_t, one for each real number t, from a special Fourier transform of the zeta function's completed form. At $t = 0$ the

zeros of H_0 are, up to a change of variable, the non-trivial zeros of zeta, and the Riemann hypothesis says they are all real. De Bruijn showed that if all zeros of H_t are real, so are those of H_s for every s $\geq$ t. Newman (1976) showed that a threshold exists and conjectured that it is not negative: Λ, the de Bruijn–Newman constant, is the smallest value such that all zeros of H_t are real for every t $\geq \Lambda$.

So the Riemann hypothesis is the statement $\Lambda \leq 0$. Rodgers and Tao (2020) proved $\Lambda \geq 0$, so the hypothesis is equivalent to $\Lambda = 0$. A positive upper bound such as $\Lambda \leq 0.170$ does not decide the hypothesis.

What was known

De Bruijn gave $\Lambda \leq 1/2$. Ki, Kim and Lee (2009) gave $\Lambda < 1/2$. The Polymath15 project (2019) proved $\Lambda \leq 0.22$ by two ideas: a criterion that reduces the bound to checking that a function does not vanish in a region (Theorem 1.2), and an effective approximation of H_t by a short sum (Theorem 1.3). Platt and Trudgian (2021) verified the Riemann hypothesis up to height 3,000,175,332,800, which let the same method give $\Lambda \leq 0.2$. Others have posted lower figures, some of them using a newer input on the zeros of zeta; they are credited on the credits page.

What this family does

Using the Polymath15 criterion and a pipeline inherited from Jude Gomila's package ($\Lambda \leq 0.1787854$), we ran computer-assisted certificates for $\Lambda \leq 0.172$, $\Lambda \leq 0.1715$ and $\Lambda \leq 0.170$. Each run used interval arithmetic where the stages demand it. For each run I also changed some proof steps of the inherited argument. Every result here is conditional: it rests on the published premises and on the inherited pipeline, whose own referee questions are open, and it stays conditional on these even after the changed steps are read. The written arguments for the changed steps are in the proofs folder; we re-derived eight of the nine within this project and recomputed the ninth with a second program, and nobody outside this project has read them (see the conditions file).

We stand on the shoulders of giants. Nothing here should be read as more than a request to check.

Yesterday, today, tomorrow

Yesterday. De Bruijn built a family of functions, one for each real number t, from the zeta function, and showed that if all zeros are real at t they are real at every later time. Newman defined the threshold Λ, and the Riemann hypothesis is the statement Λ is at most 0. De Bruijn gave Λ at most 1/2, Ki, Kim and Lee gave a strict bound below it, Rodgers and Tao proved Λ is at least 0, Polymath15 proved at most 0.22, and Platt and Trudgian's verified height gave at most 0.2.

Today. I used the Polymath15 criterion and a pipeline inherited from Jude Gomila's package for 0.1787854 to run certificates for Λ at most 0.172, 0.1715 and 0.170. For each run I changed some proof steps of the inherited argument; those steps are proved as scoped within this project and have not been read by anyone outside it. Every result here is conditional on the published premises and on the inherited pipeline's open referee questions, which an outside reading of the changed steps would not remove. The published upper bounds are 0.22 (Polymath15) and 0.2 (Polymath15 with the verification of Platt and Trudgian); Jude Gomila's package, which he labels not yet peer reviewed, states 0.1787854; our values 0.172, 0.1715 and 0.170 are conditional, as stated above. Figures posted by others are credited on the credits page; several unrefereed preprints state further values, which we have not checked.

Tomorrow. The chapter asks readers to check, not to accept. Examining the changed steps and the inherited pipeline's open questions is the route to a result that is less conditional. A positive upper bound does not decide the Riemann hypothesis.

Check it yourself

The verification guide gives the written recipes that recheck the arithmetic joining the stored results, starting with the fingerprint check.

Where to go next

- How does a computer-assisted certificate produce an upper bound? See how a certificate is built.

4 Conditions

Conditions

The question these conditions answer. Under what assumptions does "every stage of the computation passed" turn into "Λ is at most the bound"?

Each bound below is conditional. Every stage of the computation passed. The conclusion "Λ is at most the bound" follows if everything on this page holds. The changed steps were read and re-derived within this project (below); nothing on this page has been read by anyone outside this project.

Terms, once: a stage is one of the five parts of the criterion (finite rows, wall, height, error, tail; see the statement of the three bounds). A changed proof step is a place where the argument we run differs from the argument in the sources we inherited, so that someone has to read the changed argument. The nine changed steps are Sonny's own; the inherited chain is Jude Gomila's. The inherited chain is the earlier pipeline these runs reuse.

Published results used as premises (all bounds)

- The Polymath15 approximation bounds. Theorem 1.2 (the criterion that turns a positivity statement at height y0 into the bound t0 + $y0^2/2$), Theorem 1.3, Proposition 6.6 and the error equations (20)–(23) of D. H. J. Polymath (2019). The finite, wall, error and tail stages rest on them as printed.
- The verified height. Platt and Trudgian (2021): the Riemann hypothesis holds up to height 3,000,175,332,800. Part (i) of the criterion, as printed by Polymath15, needs it up to $X/2 = 3{,}000{,}175{,}332{,}766$ with $X = 6{,}000{,}350{,}665{,}532$, a margin of 34. The recipe in the verification guide tests the slightly stronger height (X +

$\sqrt{(1 - y0^2)})/2$, under 3,000,175,332,767, which also passes (margin at least 33.5).

The inherited chain (all bounds)

The pipeline is built from Jude Gomila's package for $\Lambda \leq 0.1787854$ and the runs that reuse it. Its author labels it "computer-assisted unconditional proof, not yet peer reviewed" and lists the questions on which a referee is asked to judge. They are all open, in his order of priority:

- I1. That his note matches Polymath's Theorem 1.2 exactly (the "weld").
- I2. The closed barrier certificate (883 prisms).
- I3. Where the barrier coefficients and error constants come from.
- I4. The step that turns a positive floor into a bound on |f_t|, including that the correction K is counted in the floor and not in the error bound.
- I5. The transfer from one height to all heights, and from one window to all x.
- I6. The per-term Dini bound, from his sharp-bound note, together with the completeness of his enumeration of sectors and the endpoint-count treatment. It belongs to the transfer question. Because we remove the common-phase credit (the no-credit step), it also carries the nine head primes, which the credit used to bypass.
- I7. For the wall stage, the upper wall at times up to 129/800 is taken from the earlier chain and is not recomputed.

Even after the changed steps are read, each bound stays conditional on this list.

Changed proof steps

Reading status, in one place:

- Read within this project: all nine steps. Each step was read against its written argument, or, for the finite-row program, against the code change and the inherited text; and against the Polymath15 text. Result: every step is correct as scoped (the scope of each is the inherited

premises above, which the reading did not try to remove). No step was found wrong as a mathematical argument. The finite-row step is a program, not an argument: its changed lines and the block charge code were read in full, and all 873 finite-row source lines were read against the inherited sources.

- Re-derived within this project: the eight steps with written arguments. Each was re-derived from its statement and agrees with its written argument. Three of them (no common-phase credit, wall time guard, handover) were re-derived again in separate blind sessions that saw their statements and the code changes, not the written arguments; the core statements agree. The re-derivations of the correction cap, the block charge and the mollifier tables were written after reading this project's own notes on those steps (not the write-ups), so they are not independent of that reading. The correction cap was recomputed at all three tuples with outward-rounded interval arithmetic at 90 digits, and again enclosed at 95 digits.

- Recomputed: the finite-row program. All 13,808,993 rows of 0.170 were recomputed by a second program written within this project, with a different method (not shipped), in outward-rounded ball arithmetic, from the shard ranges in the finite-row stage file. Every stored row value lies inside the recomputed enclosure (to the 12 digits stored), every shard minimum sits where the stage file says, and every row is at least its stored shard floor less 1.1×10^{-12}, far under the smallest margin. It is labelled computed.

- All 13,808,993 stored 0.170 rows were also re-read (header and sequence coverage, exact floors all positive, table lines, absence of the common-phase credit). Recomputed from the definitions as well: the tail ratio D $= 0.99557360332\ldots$, and the row functional at the thinnest plain rows and at two block rows (agreeing with the stored floors to the digits shown). A search for a phase that would break the block inequality found none. None of this re-ran the interval-arithmetic stages, the stored upper wall or the full height-cell computation.

- For 0.170, the correction cap through block charge steps are used at a different tuple and with a different height build (a table branch and larger blocks) from the 0.172 and 0.1715 runs; the reading covered the 0.170 build.

How exported numbers reach the final gates (all bounds)

The height stage logs each endpoint as a binary64 number with its exact hexadecimal value, and the error and final-gate stages read decimals. The stage files carry every one of the 44 exported endpoints as a decimal rounded in its safe direction from its exact binary value: lower bounds down, upper bounds up. The final gates are recomposed from those values in exact fractions. All 76 reported gates are positive; all 49,152 high-height block rows at 0.170 (worst margin 0.005256921498) and all 12,288 at 0.1715 (worst margin 0.00513489) pass the rule and its proviso; every margin keeps its six leading significant digits. The argument is 10-directed-export.md, and Recipe 6 of the verification guide checks it from the shipped files.

Margins at 0.170 are thin

Smallest plain-row floor 0.000217735 at N = 1,104,576 and plain margin (floor − E − K_cap) 0.00021738; block margin 0.00077030; tail ratio D = 0.99558 (must be under 1) with post-error 0.002856; error E at most 3.4745×10^{-7}. The comparison margins are 0.00073740 (plain) and 0.95080 (tail ratio) at 0.1715, and 0.0038103 and 0.87283 at 0.172. Lower bounds are rounded down and upper bounds up. The numbers are exact and positive, but a slip of about one part in a few hundred in an inherited bound would show at 0.170. The bound at 0.170 is rigorous if every inherited bound is exactly as claimed.

Step	What changes, in plain words	0.172	0.1715	0.170	Status
Correction cap	The correction term K is bounded by one uniform number K_cap over all N and all heights, instead of being shown to shrink as height grows. K_cap is between 2e-10 and 3.5e-10 against smallest floors between 2e-4 and 4e-3. The finite-floor relation it uses is an inequality, floor_N(y0) $\leq$ Num_N(y0)/M_N(y0) $-$ K_N(y0).	used	used	used (J = 14.5 million)	proved as scoped; write-up: 01-correction-cap.md
No common-phase credit	The common-phase credit, an extra gain used in other runs, is switched off everywhere. Removing a nonnegative gain keeps every lower bound valid.	used	used	used	proved as scoped; write-up: 02-no-common-phase-credit.md
Wall time guard	The wall is checked for times up to T, with a time ball allowed to overhang T by 2^{-30}, and every checked box must also pass the exact test that its declared end time is at most T.	used	used	used	proved as scoped; write-up: 03-wall-time-guard.md
Total height rule	To show a positive floor at height y0 persists for all heights up to Y, we require head bound + sector mass + g′ ≤ 0 on every cell. This replaces the inherited sufficient test (head bound ≤ 0 and every sector bound ≤ 0), which fails at these y0. The rule shows that Q = 1 $-$ g $-$ Lsum is nondecreasing in height (not that Lsum itself falls, since g′ is negative and Lsum may rise at up to the rate $\lvert g' \rvert$). A floating-point test at the 0.172 rows agreed.	used	used	used (total and band form; band mass 0)	proved as scoped; write-ups: 04a, 04b, 04c
Block charge inequality, with two height rules	On a short range of rows after each start, a block of the smallest primes above 13 is treated as one exact charge (frozen form) instead of term by term. One height rule carries it from y0 to 1/8; a second rule carries the no-block floor above 1/8. The frozen charge dominates the non-frozen one, since 1 $-$ g_e $\geq$ 1 $-$ g(N) and L* is nonincreasing in g.	not used	used (k = 1000)	used (k = 3000 and k = 300)	proved as scoped; write-up: 05-block-charge.md
Optimised mollifier tables	Finite rows use stored coefficients λ_d (kept as two tables of mollifier weights) instead of the Euler shape. The lower-bound argument is unchanged (06-mollifier-tables.md), and the height rule needs a version for tables.	not used	not used	used	proved as scoped; write-up: 06-mollifier-tables.md
Handover at 1,080,000	The change from the 64-divisor mollifier to the 4-divisor one happens at N = 1,080,000 instead of 1,028,000.	not used	not used	used	proved as scoped; write-up: 07b-

Step	What changes, in plain words	0.172	0.1715	0.170	Status
					handover-1080000.md
Finite-row program on every row	The finite-row program is the certified one plus a block branch and a table branch, used on all 13.8 million rows. With no table and no block it reproduces the certified program's output line for line.	not used	not used	used	computed (recomputed by a second program, not shipped; see the reading-status list above); 07
Raised input caps	The tail program takes the join point as a bounded input (14,500,000 in the 0.170 run, as its tuple file shows) and its hard-coded lower height is 11/100 (12/100 in the inherited runs; the stored tail file shows the lower height squared as 121/10000); the error-screening program's prefilter is 0.68 (0.67 in the inherited runs). Sonny's step notes call these policy caps, not hypotheses; the gates that matter are still evaluated.	not used	not used	used	proved as scoped; write-ups: 09a, 09c

Words used in the total height rule: Lsum is the sum of per-term bounds inside the row's lower bound; the head bound bounds how fast that sum can change with height from the integers under 4096; the sector mass does the same for all larger integers, grouped into sectors; g′ is the rate of change of the factor g.

Count: 0.172 rests on 4 changed steps (the correction cap through total height rule rows of the table), 0.1715 on 5 (those four and the block charge step), 0.170 on 9.

The band transfer was built for 0.170 but is not used: no cell needs it, so the band mass subtracted in the final inequality is exactly 0 on both legs.

One more condition (0.1715 and 0.170)

- The frozen-radius condition. The frozen block step needs $g_a^2 \cdot N\hat{}y \le 1$ with $g_a = g(N_start)$. The producer refuses any row unless it certifies this, and no row was refused. It holds with room: $g_a^2\ N\hat{}y0$ is 0.2297 and 0.2184 at y0 (maxima 0.229156 and 0.217980 per shard), and for

every supplied block subshard a ≤ N ≤ b one has q_a = a² − t/16 > (25/24) b, which gives g_a(y)² N^y < 1 for every y > 0, so it is an analytic fact and not a per-row coincidence. The argument is Sonny's and ships as 05b-frozen-radius-condition.md; it was read within this project and again in the re-derivation of the block charge. It has not been read by anyone outside this project.

What the verification recipes do with these conditions

Nothing. They check arithmetic on the stored results (see the verification guide). They do not check any condition on this page.

Where to go next

- What are the stored numbers of each stage? See the statement of the three bounds.
- What can a reader recheck in the stage files? See the verification guide.

5 Statement

Statement

The question. How small can we prove the de Bruijn–Newman constant Λ to be? Λ is the smallest value such that all zeros of the heat-flow functions H_t are real for every t ≥ Λ (see the chapter on the question). Write y0 for the height at which a positivity statement is proved, T for the time up to which the heat flow is followed, and X for the real-axis position from which the criterion is applied. The criterion of D. H. J. Polymath (2019, Theorem 1.2) gives Λ ≤ T + y0²/2 when its premises hold.

The three results

Each line says what was run and what passed. Every statement is conditional on the full list in the conditions file: published results, the inherited chain, the changed proof steps and, where marked, the frozen-radius condition (Sonny's

argument shows it holds for every $y > 0$). Status: CONDITIONAL for each bound.

Bound	T	y0	Y	Rows N	Changed steps	Extra condition
$\Lambda \le 17/100 = 0.170$	3279/20000	11/100	409909/500000	691,008 to 14,500,000 (13,808,993 rows)	9	frozen radius (shown)
$\Lambda \le 343/2000 = 0.1715$	1643/10000	12/100	819391/1000000	691,008 to 12,000,000 (11,308,993 rows)	5	frozen radius (shown)
$\Lambda \le 43/250 = 0.172$	3271/20000	13/100	820305/1000000	691,008 to 12,000,000 (11,308,993 rows)	4	none

For all three, X = 6,000,350,665,532 and the common-phase credit is off. Y is a rational number at least $\sqrt{(1 - 2T)}$: the height range checked runs from y0 to Y.

Exactly: $T + y0^2/2$ equals 17/100, 343/2000 and 43/250 respectively (the recipe on the tuple in the verification guide checks this).

Stored results of each stage

Numbers are copied from the stage files in the certificates folder, with lower bounds (floors, margins, post-errors) rounded down and upper bounds (E, K_cap, D, the worst height bound) rounded up. Exact fractions are in those files. Status: COMPUTED (stored results of server runs; the recipes check how they join, not the stages themselves).

	0.170	**0.1715**	**0.172**
Finite rows, all positive	13,808,993	11,308,993	11,308,993
Smallest plain-row floor	0.000217735 at N = 1,104,576	0.000737721 at N = 695,104	0.00381064 at N = 1,028,000
Smallest block-row floor	0.000770655 at N = 1,080,000	0.00572024 at N = 1,028,000	no block
Wall: boxes, all settled without splitting	6912	6496	6272
Wall: smallest floor	0.05727	0.20105	0.30283
Height: cells, worst upper bound (must be negative)	106, −0.0419	116, −0.0812	95, −0.0864
Error E (largest piece)	3.4745e-7	3.1976e-7	3.0928e-7
Correction cap K_cap	3.4887e-10	2.4923e-10	2.2400e-10
Plain margin (floor − E − K_cap)	0.00021738	0.00073740	0.0038103
Block margin on [y0, 1/8]	0.00077030	0.0057199	none
Worst margin above height 1/8 (high-height rule)	0.0052569	0.0051348	none
Tail at J: ratio D (must be below 1)	0.99558	0.95080	0.87283
Tail at J: post-error (must be positive)	0.002856	0.031663	0.081802

Notes:

- For 0.170 the margins are small: 2.2e-4 for plain rows, and the tail ratio is 0.9956. They are exact and positive, but thin. What each changed step rests on is in the conditions file.
- For 0.172 the finite rows and the wall were produced in two earlier runs and are reused unchanged; the stage files are those earlier runs' own.
- The wall, finite-row and tail files are the producing runs' outputs, with a few self-test fields and code labels left out of their text. The height, error and correction-cap files carry every exported endpoint rounded in the safe direction from its exact binary value, with the final gates recomposed in exact fractions from those values (see the directed export and Recipe 6 of the verification guide); every margin in the table above keeps its six leading digits.

What is not stated

- No claim about the Riemann hypothesis. Rodgers and Tao (2020) proved $\Lambda \geq 0$, and the hypothesis is the statement $\Lambda = 0$, so an upper bound such as these does not decide it.
- No claim that these are the lowest public bounds. Lower figures have been posted by others (see the credits page); we have not assessed them and these certificates do not use their route.
- No claim that any condition holds. They are listed so that a reader can check them.

Where to go next

- Which changed steps does each bound rest on? See the list of conditions.
- How can you recheck these numbers from the stage files? See the verification guide.

6 Theorems in this book

Theorems in this book

Every theorem and lemma in this book, with its exact statement, status and credit. Maths is written in LaTeX notation, between dollar signs.

The entries follow the order of the results list in statement/bounds.json: the cited criterion, the three bounds, the stage results, the nine changed proof steps, the frozen-radius condition, then the open items. No Lean proof is shipped in this book. The written arguments for the changed steps are in the proofs folder.

Terms: Λ is the de Bruijn–Newman constant, the smallest value such that all zeros of the heat-flow functions H_t are real for every $t \geq \Lambda$. T is the time up to which the heat flow is followed, y_0 the height at which a positivity statement is proved, X the real-axis position from which the criterion is applied.

Theorem 1: the Polymath15 criterion,
$\Lambda \le T + y_0^2/2$ (D. H. J. Polymath, 2019, cited)

Credit: D. H. J. Polymath (2019), Theorem 1.2 of "Effective approximation of heat flow evolution of the Riemann xi function, and a new upper bound for the de Bruijn–Newman constant", Res. Math. Sci. 6 (2019), Paper 31. Cited, not proved in this book.

When its premises hold, the criterion gives $\Lambda \le T + y_0^2/2$. In the book's words: if the approximating function f_T has no zeros at positions beyond X and heights from y_0 up to 1 at the final time T, a barrier region for the earlier times is also free of zeros, and the Riemann hypothesis is verified up to a large height, then $\Lambda \le T + y_0^2/2$.

Status: cited (published result used as a premise). The verified height used is that of Platt and Trudgian (2021): the Riemann hypothesis holds up to height 3,000,175,332,800. Part (i) of the criterion, as printed by Polymath15, needs it up to $X/2 = 3{,}000{,}175{,}332{,}766$ with $X = 6{,}000{,}350{,}665{,}532$, a margin of 34.

Proof: statement/CONDITIONS.md, section "Published results used as premises".

Conditional upper bound 2: $\Lambda \le 43/250 = 0.172$

Credit: Polymath15's criterion (2019) and the pipeline of Jude Gomila (2026), used here by Sonny (2026).

$\Lambda \le \dfrac{43}{250} = 0.172$, with $T = \dfrac{3271}{20000}$, $y_0 = \dfrac{13}{100}$, $Y = \dfrac{820305}{1000000}$ (a rational number at least $\sqrt{1 - 2T}$), $X = 6{,}000{,}350{,}665{,}532$, rows N from 691,008 to 12,000,000 (11,308,993 rows), common-phase credit off, and $T + y_0^2/2 = 43/250$ exactly. Four changed steps (Lemmas 7, 8, 9 and 10 of this file: the correction cap, no common-phase credit, the wall time guard, the total height rule). No extra condition.

Status: CONDITIONAL. The conclusion follows only if everything on the conditions page holds: the Polymath15 approximation bounds as printed (Theorem 1.2, Theorem 1.3, Proposition 6.6 and equations (20)–(23)); the

verified height; the open referee questions of the inherited chain (the weld to Polymath's Theorem 1.2, the closed barrier certificate of 883 prisms, the origin of the barrier coefficients and error constants, the step from a positive floor to a bound on $|f_t|$ with the correction K counted in the floor, the transfer from one height to all heights and from one window to all x, the per-term Dini bound with the completeness of the sector enumeration (which also carries the nine head primes), and, for the wall stage, the upper wall at times up to 129/800 taken from the earlier chain); and the four changed steps. The changed steps have not been read by anyone outside this project.

Proof: statement/STATEMENT.md (https://gocheckyourself.org/read/riemann-debruijn-newman/experts/#ch-statement-statement); statement/CONDITIONS.md; certificates in certificates/0p172/; recipes in VERIFY.md (https://gocheckyourself.org/read/riemann-debruijn-newman/experts/#ch-verify).

Conditional upper bound 3: $\Lambda \le 343/2000 = 0.1715$

Credit: Polymath15's criterion (2019) and the pipeline of Jude Gomila (2026), used here by Sonny (2026).

$\Lambda \le \frac{343}{2000} = 0.1715$, with $T = \frac{1643}{10000}$, $y_0 = \frac{12}{100}$, $Y = \frac{819391}{1000000}$ (a rational number at least $\sqrt{1-2T}$), $X = 6{,}000{,}350{,}665{,}532$, rows N from 691,008 to 12,000,000 (11,308,993 rows), common-phase credit off, and $T + y_0^2/2 = 343/2000$ exactly. Five changed steps (the four of Theorem 2 and the block charge, Lemma 11 of this file). Extra condition: the frozen-radius condition (Proposition 16).

Status: CONDITIONAL, on the same conditions as Theorem 2, with the block charge step and the frozen-radius condition added. The frozen-radius condition is shown to hold for every $y > 0$ by Sonny's argument.

Proof: statement/STATEMENT.md (https://gocheckyourself.org/read/riemann-debruijn-newman/experts/#ch-statement-statement); statement/CONDITIONS.md; certificates in

certificates/0p1715/; recipes in VERIFY.md (https://gocheckyourself.org/read/riemann-debruijn-newman/experts/#ch-verify).

Conditional upper bound 4: $\Lambda \le 17/100 = 0.170$

Credit: Polymath15's criterion (2019) and the pipeline of Jude Gomila (2026), used here by Sonny (2026).

$\Lambda \le \frac{17}{100} = 0.170$, with $T = \frac{3279}{20000}$, $y_0 = \frac{11}{100}$, $Y = \frac{409909}{500000}$ (a rational number at least $\sqrt{1-2T}$), $X = 6{,}000{,}350{,}665{,}532$, rows N from 691,008 to 14,500,000 (13,808,993 rows), common-phase credit off, and $T + y_0^2/2 = 17/100$ exactly. Nine changed steps (Lemmas 7 to 15 of this file). Extra condition: the frozen-radius condition (Proposition 16).

Status: CONDITIONAL, on the same conditions as Theorem 3, with the mollifier tables, the handover, the finite-row program and the raised input caps added. The margins are thin: smallest plain-row floor 0.000217735 at $N = 1{,}104{,}576$, plain margin 0.00021738, block margin 0.00077030, tail ratio $D = 0.99558$ (must be below 1) with post-error 0.002856, error E at most 3.4745×10^{-7}. The conclusion of the book's review: the result is rigorous only if every inherited bound is exactly as claimed.

Proof: statement/STATEMENT.md (https://gocheckyourself.org/read/riemann-debruijn-newman/experts/#ch-statement-statement); statement/CONDITIONS.md; certificates in certificates/0p170/; recipes in VERIFY.md (https://gocheckyourself.org/read/riemann-debruijn-newman/experts/#ch-verify).

Computation 5: all five stages pass for the three tuples

Credit: This result (Sonny): computed.

For each of the three tuples the finite rows, wall, height, error and tail stages all pass. Stored results, with lower bounds (floors, margins, post-errors)

rounded down and upper bounds (E, K_{cap}, D, the worst height bound) rounded up, for 0.170, 0.1715 and 0.172 respectively:

- Finite rows, all positive: 13,808,993; 11,308,993; 11,308,993.
- Smallest plain-row floor: 0.000217735 at $N = 1{,}104{,}576$; 0.000737721 at $N = 695{,}104$; 0.00381064 at $N = 1{,}028{,}000$.
- Smallest block-row floor: 0.000770655 at $N = 1{,}080{,}000$; 0.00572024 at $N = 1{,}028{,}000$; no block.
- Wall, boxes all settled without splitting: 6912; 6496; 6272. Wall, smallest floor: 0.05727; 0.20105; 0.30283.
- Height, cells and worst upper bound (must be negative): 106 cells, -0.0419; 116 cells, -0.0812; 95 cells, -0.0864.
- Error E (largest piece): 3.4745×10^{-7}; 3.1976×10^{-7}; 3.0928×10^{-7}.
- Correction cap K_{cap}: 3.4887×10^{-10}; 2.4923×10^{-10}; 2.2400×10^{-10}.
- Plain margin (floor $-E - K_{\text{cap}}$): 0.00021738; 0.00073740; 0.0038103.
- Block margin on $[y_0, 1/8]$: 0.00077030; 0.0057199; none.
- Worst margin above height 1/8 (high-height rule): 0.0052569; 0.0051348; none.
- Tail at J, ratio D (must be below 1): 0.99558; 0.95080; 0.87283.
- Tail at J, post-error (must be positive): 0.002856; 0.031663; 0.081802.

Status: COMPUTED (stored results of server runs; the recipes check how they join, not the stages themselves). For 0.172 the finite rows and the wall were produced in two earlier runs and are reused unchanged.

Proof: stage files in certificates/; statement/STATEMENT.md (https://gocheckyourself.org/read/riemann-debruijn-newman/experts/#ch-statement-statement), section "Stored results of each stage".

Computation 6: the stored results agree with one another and with exact arithmetic

Credit: This result (Sonny): computed.

The stored results agree with one another and with exact arithmetic: the six recipes in the verification guide give the checks.

Status: COMPUTED.

Proof: VERIFY.md (https://gocheckyourself.org/read/riemann-debruijn-newman/experts/#ch-verify).

Lemma 7: the correction cap

Credit: Builds on: the inherited argument: Polymath15's criterion, D. H. J. Polymath (2019), and the pipeline of Jude Gomila (2026). This result (Sonny): PROVED (written argument shipped) conditional.

The correction term K is bounded by one uniform number K_{cap} over all N and all heights, instead of being shown to shrink as height grows. K_{cap} is between 2×10^{-10} and 3.5×10^{-10} against smallest floors between 2×10^{-4} and 4×10^{-3}. Used in all three bounds (for 0.170 with $J = 14.5$ million).

Status: PROVED (written argument shipped): correct as scoped, that is, under the inherited premises in the conditions file; not refereed. Read within this project: correct as scoped.

Proof: Sonny's written argument, proofs/01-correction-cap.md; read and re-derived within this project. See statement/CONDITIONS.md.

Lemma 8: no common-phase credit

Credit: Builds on: the inherited argument: Polymath15's criterion, D. H. J. Polymath (2019), and the pipeline of Jude Gomila (2026). This result (Sonny): PROVED (written argument shipped) conditional.

The common-phase credit, an extra gain used in earlier runs, is switched off everywhere. Removing a nonnegative gain keeps every lower bound valid. Used in all three bounds.

Status: PROVED (written argument shipped): correct as scoped, that is, under the inherited premises in the conditions file; read and re-derived within this project, also in a separate blind session; not refereed.

Proof: Sonny's written argument, proofs/02-no-common-phase-credit.md. See statement/CONDITIONS.md.

Lemma 9: the wall time guard

Credit: Builds on: the inherited argument: Polymath15's criterion, D. H. J. Polymath (2019), and the pipeline of Jude Gomila (2026). This result (Sonny): PROVED (written argument shipped) conditional.

The wall is checked for times up to T, with a time ball allowed to overhang T by 2^{-30}, and every checked box must also pass the exact test that its declared end time is at most T. Used in all three bounds.

Status: PROVED (written argument shipped): correct as scoped, that is, under the inherited premises in the conditions file; read and re-derived within this project, also in a separate blind session; not refereed.

Proof: Sonny's written argument, proofs/03-wall-time-guard.md. See statement/CONDITIONS.md.

Lemma 10: the total height rule

Credit: Builds on: the inherited argument: Polymath15's criterion, D. H. J. Polymath (2019), and the pipeline of Jude Gomila (2026). This result (Sonny): PROVED (written argument shipped) conditional.

To show a positive floor at height y_0 persists for all heights up to Y, we require head bound + sector mass + $g' \le 0$ on every cell. This replaces the inherited sufficient test (head bound ≤ 0 and every sector bound ≤ 0), which fails at these y_0. The rule proves that $Q = 1 - g - L_{\text{sum}}$ is nondecreasing in height (it does not prove that L_{sum} itself falls, since g' is negative and L_{sum} may rise at up to the rate $|g'|$). Used in all three bounds (for 0.170 in total and band form; band mass 0). Here L_{sum} is the sum of per-term bounds inside the row's lower bound; the head bound bounds how fast that sum can change with height from the integers below 4096; the sector mass does the same for all larger integers, grouped into sectors; g' is the rate of change of the factor g.

Status: PROVED (written argument shipped): correct as scoped, that is, under the inherited premises in the conditions file; not refereed. Read within this project: correct as scoped. A floating-point test at the 0.172 rows agreed.

Proof: Sonny's written arguments, proofs/04a-total-height-rule.md, 04b and 04c; read and re-derived within this project. See statement/CONDITIONS.md.

Lemma 11: the block charge inequality with two height rules

Credit: Builds on: the inherited argument: Polymath15's criterion, D. H. J. Polymath (2019), and the pipeline of Jude Gomila (2026). This result (Sonny): PROVED (written argument shipped) conditional.

On a short range of rows after each start, a block of the smallest primes above 13 is treated as one exact charge (frozen form) instead of term by term. One height rule carries it from y_0 to 1/8; a second rule carries the no-block floor above 1/8. Not used for 0.172; used for 0.1715 ($k = 1000$) and 0.170 ($k = 3000$ and $k = 300$). The frozen form also needs the frozen-radius condition (Proposition 16).

Status: PROVED (written argument shipped): correct as scoped, that is, under the inherited premises in the conditions file; read and re-derived within this project; not refereed. The frozen charge dominates the non-frozen charge because $1 - g_e \geq 1 - g(N)$ and L^* is nonincreasing in g; both inequalities are written in the proof. The height margin θ the rule uses reaches the final gate rounded down from its exact binary value (proofs/10-directed-export.md).

Proof: Sonny's written argument, proofs/05-block-charge.md, with proofs/10-directed-export.md for the export of θ. See statement/CONDITIONS.md.

Lemma 12: the optimised mollifier tables

Credit: Builds on: the inherited argument: Polymath15's criterion, D. H. J. Polymath (2019), and the pipeline of Jude Gomila (2026). This result (Sonny): PROVED (written argument shipped) conditional.

Finite rows use stored coefficients λ_d (two table files in the 0.170 folder) instead of the Euler shape. The lower-bound argument is unchanged, and the height rule needs a version for tables. Used for 0.170.

Status: PROVED (written argument shipped): correct as scoped, that is, under the inherited premises in the conditions file; not refereed.

Proof: Sonny's written argument, proofs/06-mollifier-tables.md; read and re-derived within this project. See statement/CONDITIONS.md.

Lemma 13: the handover at 1,080,000

Credit: Builds on: the inherited argument: Polymath15's criterion, D. H. J. Polymath (2019), and the pipeline of Jude Gomila (2026). This result (Sonny): PROVED (written argument shipped) conditional.

The change from the 64-divisor mollifier to the 4-divisor one happens at $N = 1{,}080{,}000$ instead of $N = 1{,}028{,}000$. Used for 0.170.

Status: PROVED (written argument shipped): correct as scoped, that is, under the inherited premises in the conditions file; read and re-derived within this project, also in a separate blind session; not refereed.

Proof: Sonny's written argument, proofs/07b-handover-1080000.md. See statement/CONDITIONS.md.

Computation 14: the finite-row program with block and table branches

Credit: Builds on: the inherited argument: Polymath15's criterion, D. H. J. Polymath (2019), and the pipeline of Jude Gomila (2026). This result (Sonny): computed.

The finite-row program is the certified one plus a block branch and a table branch, used on all 13.8 million rows. With no table and no block it reproduces the certified program's output line for line. Used for 0.170. All 13,808,993 rows of 0.170 were recomputed by a second program written within this project, with a different method (not shipped), in outward-rounded ball arithmetic: every stored row value lies inside the recomputed enclosure (to the 12 digits stored), every shard minimum sits where the stage file says, and every row is at least its stored shard floor less 1.1×10^{-12}, far below the smallest margin.

Status: COMPUTED: every 0.170 row recomputed by a second program written within this project; not refereed. A program step, not an argument. Read within this project: correct as scoped to the lines read.

Proof: a program step, not an argument; what it changes and what supports it are in proofs/07-steps-with-no-separate-write-up.md. The second program that recomputed the rows is not shipped. See statement/CONDITIONS.md.

Lemma 15: the raised input caps

Credit: Builds on: the inherited argument: Polymath15's criterion, D. H. J. Polymath (2019), and the pipeline of Jude Gomila (2026). This result (Sonny): PROVED (written argument shipped) conditional.

The tail program takes the join point as a bounded input (14,500,000 in the 0.170 run, as its tuple file shows); the error-screening program's prefilter changes from 0.67 to 0.68. Sonny's step notes call these policy caps, not hypotheses; the gates that matter are still evaluated. Used for 0.170. The hard-coded lower height of the tail program also changes, from 12/100 to 11/100.

Status: PROVED (written argument shipped): correct as scoped, that is, under the inherited premises in the conditions file; not refereed. Read within this project: correct as scoped.

Proof: Sonny's written arguments, proofs/09a-error-screen-guards.md and 09c; read and re-derived within this project. See statement/CONDITIONS.md.

Proposition 16: the frozen-radius condition holds for every $y > 0$

Credit: This result (Sonny): proved (written).

The frozen block step needs $g_a^2 N^y \le 1$ with $g_a = g(N_{\text{start}})$. It was shown to hold, below 1, for every $y > 0$; at y_0 the quantity $g_a^2 N^{y_0}$ is 0.2297 and 0.2184 (maxima 0.229156 and 0.217980 per shard in a second pass). The producer refuses any row unless it certifies this, and no row was refused. Needed for 0.1715 and 0.170 only.

Status: shown (an analytic fact and not a per-row coincidence); a reading within this project, not read by anyone outside it.

Proof: Sonny's argument, proofs/05b-frozen-radius-condition.md; read within this project and again in the re-derivation of the block charge. See statement/CONDITIONS.md.

Lemma 17: the directed export of numbers

Credit: This result (Sonny): proved (written).

If a finite binary64 endpoint has exact dyadic value q, exporting a lower bound as $h\lfloor q/h \rfloor$ and an upper bound as $-h\lfloor -q/h \rfloor$ on a decimal grid $h = 10^e$ gives lower $\leq q \leq$ upper. The final high-height gate $f + \theta d/M_u - E - K - I$ is nondecreasing in θ, so a θ rounded down keeps it valid, and the proviso (M_u if $f < 0$ else M_l)$f + \theta d \geq 0$ suffices. Used in all three bounds.

Status: PROVED (written argument shipped) for the export implication; COMPUTED for the shipped data (44 fields classified, every directed decimal on its safe side, all 76 gates positive); not refereed.

Proof: Sonny's written argument, proofs/10-directed-export.md; the check is Recipe 6 of VERIFY.md.

Open questions

The inherited pipeline's open referee questions and the published premises as used

Credit: Builds on: the pipeline of Jude Gomila (2026), labelled by its author "computer-assisted unconditional proof, not yet peer reviewed".

Do the inherited pipeline's open referee questions and the published premises hold as used? The questions are the ones listed under Theorem 2.

Status: OPEN.

Proof: none; see statement/CONDITIONS.md, section "The inherited chain".

The Riemann hypothesis, $\Lambda = 0$

Credit: Builds on: Rodgers and Tao (2020), who proved $\Lambda \geq 0$, so the Riemann hypothesis is the statement $\Lambda = 0$.

Is $\Lambda = 0$? An upper bound such as these does not decide it.

Status: OPEN.

Proof: none; see chapters/01-the-question.md (https://gocheckyourself.org/read/riemann-debruijn-newman/experts/#ch-chapters-01-the-question).

7 Verify: conditional upper bounds on the de Bruijn–Newman constant

Verify: conditional upper bounds on the de Bruijn–Newman constant

This guide gives a written recipe for each check, so that you can write your own checker in any language. Each recipe states the files, their format, the exact condition to test, the expected answer, and what a pass shows and does not show. No program of ours is needed.

What is claimed

The question. How small can we prove the de Bruijn–Newman constant Λ to be?

Three bounds on Λ, each the outcome of a computer run whose stages all passed, each conditional on the list in the conditions file:

Bound	Result	Changed proof steps	Outside reader
$\Lambda \leq 17/100$	stages passed	9	none
$\Lambda \leq 343/2000$	stages passed	5	none
$\Lambda \leq 43/250$	stages passed	4	none

Exact tuples and stored numbers are in the statement file.

Status

Three conditional upper bounds, computer-assisted, checked within this project; not refereed. The conditions are the published approximation bounds of Polymath15 and the verified height of Platt and Trudgian, the inherited pipeline (whose own referee questions are open), the changed proof steps, and, for 0.170 and 0.1715, one more condition on the frozen block step (the frozen-radius condition; Sonny's argument, in the proofs folder, shows it holds for every $y > 0$). All nine changed steps were read within this project and found correct as scoped. Eight have written arguments in the proofs folder and were re-derived within this project, three of them also in separate blind sessions (not refereed). The ninth, the finite-row program, had all 13,808,993 stored 0.170 rows re-read and recomputed by a second program written within this project (not shipped; it agrees with every stored row to the digits stored), and is labelled computed. Every number one stage passes to the next is rounded in its safe direction from its exact binary value, and the final gates are recomposed from those values in exact fractions (Recipe 6). No step has been read by anyone outside this project. The bounds are conditional results, not theorems: they rest on the published premises and on the inherited pipeline, whose own referee questions are open, and they stay conditional on these even after an outside reading of the changed steps.

Commands

The recipes need only a text reader and exact fractions. Python's fractions and decimal modules are enough; so is any language with big rationals. Do not use floating-point numbers where a recipe says exact.

Fingerprints. Compare the SHA-256 of each shipped file with the list shipped with the certificates. Run this inside the certificates folder; every line must report OK. On Windows, compute each file's hash with the second command and compare it with the list.

```
sha256sum -c SHA256SUMS.txt  # Linux
shasum -a 256 -c SHA256SUMS.txt   # macOS
CertUtil -hashfile <file> SHA256
```

Each of the three bounds has its own folder in the certificates folder, with six stage files, the exact final gates and height cover, and the height endpoints in hexadecimal and as directed decimals. The directed folder lists every exported endpoint and every final gate. The files carry the names below.

```
STEP3.json            wall
STEP4.json            finite rows
STEP5.json            height (0.1715 and 0.172)
STEP5B.json           height (0.170)
STEP6.json            error
STEP7.json            tail
CORRECTION-CAP.json   correction cap
GATES.json            every final gate as an exact fraction
HEIGHT.json           the height cover with directed endpoints
height-k2-hex.csv, height-k6-hex.csv            height endpoints as logged
height-k2-directed.csv, height-k6-directed.csv  the same endpoints as direc
directed/DIRECTED-EXPORTS.json   every exported endpoint: binary64 hex, exa
directed/ALL-GATE-MARGINS.csv    all 76 final gates of the three bounds, ex
directed/WRONG-DIRECTION-CONTROL.json  a wrongly rounded theta, rejected
```

Recipe 1: the tuple and the verified height

Numbers. T, y0 and Y of each bound from the table in the statement file, $X = 6{,}000{,}350{,}665{,}532$, N_start $= 691{,}008$ and J ($12{,}000{,}000$, or $14{,}500{,}000$ for 0.170); the stage files declare the same T, y0, Y and J (field "declared_tuple" of the error-stage file).

Conditions, in exact arithmetic.

- $T + y0^2/2$ equals the stated bound.
- $Y^2 \geq 1 - 2T$ (Y is at least $\sqrt{(1 - 2T)}$).
- $X + 1 \leq 2 \times 3{,}000{,}175{,}332{,}800$ (the height the criterion needs, $(X + \sqrt{(1 - y0^2)})/2$, is below the verified height of Platt and Trudgian).

Expected answer. All three hold for all three bounds. For example, for 0.170: $3279/20000 + (11/100)^2/2 = 17/100$, and $(X + 1)/2 = 3000175332766.5 \leq 3000175332800$, a margin of at least 33.5.

Recipe 2: the finite rows

File. The finite-row stage file of the bound (JSON), with the fields below.

```
status, rows, minimum_floor, minimum_N
finite: list of shards, each with
    lo, hi, rows, mtype, floor, minimum_N, status, uncert_rows
plain_rows, block_rows, minimum_block_floor, minimum_block_N      (0.170 ar
block_subshards: list of shards, same fields as above              (0.170 a
block_shards: for each table type, the N range [a, b] they cover   (0.170 a
```

Conditions.

- The overall status and every shard status are PASS, and no shard reports an uncertain row. The 0.172 shards have no "uncert_rows" field; for them, use the shard status PASS alone.
- Each shard's rows equals hi − lo + 1.
- All shards together (plain and block shards, sorted by lo) tile the range N_start to J with no gap and no overlap, and the row total equals the stored total and equals J − N_start + 1.
- Where block rows exist: the plain and block row counts add up to the total.
- The table type of each shard (field mtype) matches its range: the value 7, the 64-divisor mollifier, entirely below the handover point, and the value 3, the 4-divisor mollifier, entirely at or above it (0.170 only; no shard straddles).
- Every stored shard floor is a positive exact fraction.

- The stored minimum floor is the minimum over the plain shards, attained at the stored minimum_N; likewise the block minimum over the block shards. For each table type, the block shards of that type cover its stated N range exactly.

Expected answer. All hold. 0.170: 85 shards tile N = 691,008 to 14,500,000; 13,808,993 rows (13,759,841 plain and 49,152 block); smallest plain floor 0.000217735742 at N = 1,104,576; smallest block floor 0.000770655152 at N = 1,080,000. 0.1715 and 0.172: 11,308,993 rows to N = 12,000,000; smallest plain floors 0.000737721 and 0.00381064, rounded down (statement file).

Recipe 3: the correction cap

Files. The correction-cap file, and the numbers of Recipe 1. In the correction-cap file, the fields below.

```
K_cap_upper
domain: N [lo, hi], t, y [y0, Y]
pieces:
    head_sum (m=2..2000 termwise): [lower, upper]
    tail_upper ((J-2000) * max(f(log 2000), f(log J)); ...): a fraction
    e0_lower (...), gamma0_upper (...), k0_upper (...): fractions
```

Definition. With t, Y from Recipe 1 and e0, γ0, k0 the three stored fractions, put $f(l) = l \cdot \exp(t \cdot l^2/4 - (e0 - k0 \cdot Y) \cdot l)$.

Conditions.

- The domain agrees with the numbers of Recipe 1 (N from N_start to J, the same t, y from y0 to Y).
- With 80-digit decimal arithmetic: the sum of f(ln m) for m = 2 to 2000 lies inside the stored "head_sum" enclosure, to a tolerance of 10^{-55}.
- $(J - 2000) \cdot \max(f(\ln 2000), f(\ln J))$ is at most the stored "tail_upper" (to the same tolerance).
- $\gamma 0 \cdot k0 \cdot Y \cdot$ ("head_sum" upper + "tail_upper") $\leq$ K_cap_upper, in exact fractions.
- The error-stage file (Recipe 4) stores the same K_cap_upper.

Expected answer. All hold. 0.170: the sum recomputes to 37.913425737879336452 35 and the tail bound to 136364.4808; K_cap = 3.48865×10^{-10}.

Recipe 4: the final margins

Files. The error-stage file, the finite-row file, the correction-cap file and the height-stage file.

```
error stage:  error_upper, K_cap_upper, pieces: list with total_error [lowe
              band_mass_upper (0.170),
              0.172:  finite_floor, post_error_margin
              others: plain {finite_floor, rows, margin},
                      block_rows_y0_to_y1 {block_floor, rows, margin},
                      block_rows_y1_to_Y {subshards: list with
                          shard, rows, theta_lower, floor_noblock_at_worst,
                          M_upper_at_worst, worst_R2_margin, proviso_at_wor
                          minimum_proviso}
height stage: shards[shard].theta_lower  (0.170 and 0.1715)
```

Conditions, in exact fractions.

- E = the largest upper end of "total_error" over the pieces equals "error_upper". The band mass I is 0 (take the larger of the stored band masses and 0).
- Plain margin: smallest plain floor − E − K_cap − I equals the stored margin and is positive (for 0.172, the stored post-error margin, computed without I).
- Where block rows exist: block margin = smallest block floor − E − K_cap − I equals the stored block margin and is positive; the sub-shards of the high-height rule cover all block rows.

- For each sub-shard, with y1 = 1/8: margin = "floor_noblock_at_worst" + θ · (y1 − y0) / M_upper_at_worst − K_cap − E − I and P = M · "floor_noblock" + θ · (y1 − y0). The margin must equal the stored one and be positive, M must be positive, and θ must equal the θ stored for the same shard in the height-stage file. The stored proviso must be at least 0; it equals P where the no-block floor is negative, and is at most P where it is positive (there the lower bound of the normaliser is used, which the file does not store; see the directed export).

Expected answer. All hold. 0.170: plain margin 0.000217387, block margin 0.000770307, and all 9 sub-shards pass with worst margin 0.00525692. 0.1715: plain margin 0.00073740, block margin 0.0057199, worst 0.0051348. 0.172: post-error margin 0.0038103.

Recipe 5: the signs of the other stages

Files. The wall, height and tail stage files.

Conditions.

- Wall: status PASS; boxes equals leaves; "split_boxes" is 0; the smallest wall floor is a positive fraction; the file refers to the same T (and y0).
- Height: status PASS; the covered range contains [y0, Y] (its lower end is y0 rounded down, its upper end Y rounded up); the worst stored upper bound over all legs is negative; where block cells exist, the worst block rate is negative and θ positive.
- Tail: status PASS; T, J and $y0^2$ equal those of Recipe 1; the stored ratio D is below 1 and the stored post-error is positive.

Expected answer. All hold: 0.170 has 6,912 wall boxes (smallest floor 0.057277), 106 height cells (worst bound −0.041971) and tail D = 0.995574 with post-error 0.002856; 0.1715 has 6,496, 116, D = 0.95080; 0.172 has 6,272, 95, D = 0.87283. Lower bounds are rounded down and upper bounds up.

Recipe 6: the export of numbers

Files. directed/DIRECTED-EXPORTS.json, directed/ALL-GATE-MARGINS.csv and directed/WRONG-DIRECTION-CONTROL.json. In DIRECTED-EXPORTS.json, a list "exports"; each entry with a "direction" field has binary64_hex, binary64_exact, stored, directed_decimal and direction (lower or upper).

Conditions, in exact fractions.

- The value of binary64_hex (read as a hexadecimal float, for example with float.fromhex in Python) equals binary64_exact.
- For a lower bound, directed_decimal ≤ binary64_exact; for an upper bound, directed_decimal ≥ binary64_exact.
- Each directed decimal appears in the stage files of its bound.
- Every margin_exact in ALL-GATE-MARGINS.csv is positive.
- In the control file, deliberately_wrong_decimal exceeds binary64_exact and is marked rejected.

Expected answer. All hold: 42 directed endpoints (and 2 unused sentinels), every directed decimal on its safe side; 76 positive gates. The smallest gate is the height cover at 0.172 ($\mathrm{Y}^2 - (1 - 2\mathrm{T}) = 2.93 \times 10^{-7}$); the smallest final margin is the 0.170 plain margin, 0.000217387948.

What a pass shows, and what it does not. The six recipes show that the stored files agree with one another and with the stated arithmetic. They do not recompute any finite row, wall box, height cell, error term or tail inequality, and they do not show that the producers computed those correctly. They use the directed decimal of every exported number, which Recipe 6 checks against its exact binary value. They do not check any condition in the conditions file: those are mathematical steps that need a reader.

Expected output

Recipe	Expected answer
Fingerprints	Every line OK (42 files)
1. Tuple and height	$\mathrm{T} + \mathrm{y0}^2/2$ equals the bound; $\mathrm{Y}^2 \geq 1 - 2\mathrm{T}$; the needed height is below 3,000,175,332,800
2. Finite rows	Shards tile the whole range; row totals 13,808,993 / 11,308,993 / 11,308,993; every floor positive; stored minima are the minima
3. Correction cap	Sum and tail bound inside the stored enclosures; K_cap = 3.48865e-10 / 2.4923e-10 / 2.2400e-10 (rounded up)
4. Final margins	All margins recomputed equal the stored ones and are positive
5. Other stages	Wall floors positive; height bounds negative; tail ratio below 1 and post-error positive
6. Directed export	Every directed decimal on its safe side; all 76 gates positive

Changing any stored number makes at least one recipe fail (for example, a changed stage file fails its fingerprint, and a dropped shard breaks the tiling).

Proofs. The proofs folder holds the written argument of each changed step and its own fingerprint list. Run sha256sum -c SHA256SUMS.txt (macOS: shasum -a 256 -c SHA256SUMS.txt) inside the proofs folder; every line must

report OK (15 files). A pass shows that you hold the files as written; it does not show that the arguments are correct.

Time and memory

About two seconds and well under 100 MB for recipes 1, 2, 4, 5 and 6 (recipe 3, with 80-digit sums, takes a few seconds more) on an ordinary laptop, measured with a Python program of ours that is not shipped; a reader's own exact-fraction code following the recipes should need seconds to a few minutes. The runs that produced the stage files took one to two hours on a server and cannot be repeated on a laptop.

What is not claimed

- No claim about the Riemann hypothesis. Rodgers and Tao proved $\Lambda \geq 0$, so an upper bound does not decide it.
- No claim that any condition holds, and no claim that the bounds are true: they are true if the conditions are.
- No claim that these are the lowest public bounds. Lower figures posted by others are credited on the credits page.
- The finite rows, wall, height, error and tail results are those of the producing runs; this folder does not reproduce them, and the stage programs and the second program that recomputed the 0.170 finite rows are not shipped.
- Even if every changed step is read and accepted, the result stays conditional on the inherited pipeline's open questions.
- The review within this project is not an outside reading, and it did not re-run the interval-arithmetic stages, the stored upper wall or the full height-cell computation.

- The inputs e0, gamma0 and k0 of the correction cap are taken as stored, and the convexity of the term used for the endpoint bound is not rechecked.

Supporting records are available on request.

Our own programs and methods

The methods are not part of this book. The files listed above ship; the files named there as not shipped are available on request. Questions about methods or tools: write to hello@gocheckyourself.org. Request access

Where to go next

- Which conditions does no recipe check, and why? See the list of conditions.
- What are the stored numbers the recipes test? See the statement of the three bounds.

We hope it helps others find more results and keeps their curiosity going. We'd love to collaborate with anyone.

8 Files and fingerprints

This chapter is the book's file SHA256SUMS.txt.

9 Trust and verification

This chapter is the book's file VERIFY.md.

10 People and works

This chapter is the book's file CREDITS.md.

11 Files and rights

Licence

The files Sonny wrote for this book are under the Creative Commons Attribution 4.0 International licence (CC BY 4.0). You may copy, share and adapt them for any purpose, as long as you give credit.

How to give credit: "Sonny (Sonny), gocheckyourself.org", with a link to the page or file you used, and a note saying what you changed, if anything.

What it covers: the chapter, the statement pages, the written arguments in the proofs folder, the certificate files (including the mollifier tables in the 0.170 folder) and the fingerprint lists.

What it does not cover:

- Jude Gomila's package, from which the pipeline that produced the stage files derives (his code is under the MIT licence, his documents under CC BY 4.0). None of his code or text ships in this folder; if any does, his notice travels with it.
- Programs that are not part of this book, and work that is not published on this site.
- Trade marks, and the other rights that section 2(b) of the licence leaves out.

The licence file is LICENSE-CC-BY-4.0.txt; the full legal code is at https://creativecommons.org/licenses/by/4.0/legalcode.

Where to go next

- Whose work does this book build on, and under what terms? See Credits.
- What can a reader recheck with the files in this folder? See the verification guide.

12 Articles

- Three conditional upper bounds on the de Bruijn–Newman constant

13 Each result on its own page

1. Theorem 1: the Polymath15 criterion, $\Lambda \le T + y_0^2/2$ STATUS: CITED
2. Conditional upper bound 2: $\Lambda \le 43/250 = 0.172$ STATUS: CONDITIONAL

3. Conditional upper bound 3: $\Lambda \leq 343/2000 = 0.1715$ STATUS: CONDITIONAL
4. Conditional upper bound 4: $\Lambda \leq 17/100 = 0.170$ STATUS: CONDITIONAL
5. Computation 5: all five stages pass for the three tuples STATUS: COMPUTED
6. Computation 6: the stored results agree with one another and with exact arithmetic STATUS: COMPUTED
7. Lemma 7: the correction cap STATUS: PROVED IN WRITING
8. Lemma 8: no common-phase credit STATUS: PROVED IN WRITING
9. Lemma 9: the wall time guard STATUS: PROVED IN WRITING
10. Lemma 10: the total height rule STATUS: PROVED IN WRITING
11. Lemma 11: the block charge inequality with two height rules STATUS: PROVED IN WRITING
12. Lemma 12: the optimised mollifier tables STATUS: PROVED IN WRITING
13. Lemma 13: the handover at 1,080,000 STATUS: PROVED IN WRITING
14. Computation 14: the finite-row program with block and table branches STATUS: COMPUTED
15. Lemma 15: the raised input caps STATUS: PROVED IN WRITING
16. Proposition 16: the frozen-radius condition holds for every $y > 0$ STATUS: PROVED IN WRITING
17. Lemma 17: the directed export of numbers STATUS: PROVED IN WRITING
18. Open question: The inherited pipeline's open referee questions and the published premises as used STATUS: OPEN
19. Open question: The Riemann hypothesis, $\Lambda = 0$ STATUS: OPEN

© 2026 Sonny. The books are under CC BY 4.0 (Creative Commons Attribution 4.0 International): reuse them with credit and a link. Each book’s licence file lists the files that keep their own licences.