NEW COMPUTATIONS OF THE RIEMANN ZETA FUNCTION ON THE CRITICAL LINE

JONATHAN W. BOBER AND GHAITH A. HIARY

Abstract. We present highlights of computations of the Riemann zeta function around large values and high zeros. The main new ingredient in these computations is an implementation of the second author's fast algorithm for numerically evaluating quadratic exponential sums. In addition, we use a new simple multi-evaluation method to compute the zeta function in a very small range at little more than the cost of evaluation at a single point.

1. Introduction

Computations of $\zeta(1/2+it)$ have a long history and are of interest in number theory because of fundamental links to the prime numbers. (The primes and zeta zeros are Fourier transforms of each other.) With the exception of Riemann's hand calculation in the 1850s, which remained unknown to the outside world for many decades, computations of $\zeta(1/2+it)$ until 1932 had relied on the Euler–Maclaurin formula for approximating sums by integrals.

An important advance came in 1932 when Siegel rediscovered the formula used by Riemann in his hand calculation. This formula, now known as the Riemann–Siegel (RS) formula, allows computing $\zeta(1/2+it)$ by summing $\lfloor\sqrt{t/2\pi}\rfloor$ terms, far fewer than in the Euler–Maclaurin method.

For a long time after that, progress in zeta computations came from advances in computing technology rather than faster algorithms. This is nicely illustrated in Turing's effort in the 1930s to build a mechanical computer to calculate zeta, and his pioneering use of the electronic computer for that purpose two decades later. To check the Riemann Hypothesis (RH), Turing introduced a novel method to prove that a given list of zeta zeros in an interval is complete. This method is still the state of the art today. See §3.

The main algorithmic improvements on the RS formula did not arrive until the 1980s, starting with the Odlyzko–Schönhage algorithm [18] for multiple evaluations of zeta, and the algorithms of Schönhage [20] and Heath-Brown (see [9]) for evaluation at a single point. Odlyzko implemented the Odlyzko–Schönhage algorithm and computed a dataset of 20 billion zeros around zero number 10^{23}. Gourdon [8] then computed two billion zeros around the 10^{24}-th zero using another implementation of the Odlyzko–Schönhage algorithm.

2010 *Mathematics Subject Classification.* Primary: 11Y35. Secondary: 65Y20.

Key words and phrases. The Riemann zeta function, large values, exponential sums, the van der Corput iteration, Theta algorithm.

GH is partially supported by the National Science Foundation under agreements No. DMS-1406190. Both authors are pleased to thank IAS, MSRI, and ICERM, where parts of this work was conducted.

In the meantime, the theoretical complexity of computing $\zeta(1/2+it)$ at a single point became of interest in its own right. New algorithms by the second author [9, 10] have bounded this complexity by $t^{1/3+o(1)}$ time and little space, and then by $t^{4/13+o(1)}$ time and space. The purpose of this article is to report on our implementation of the $t^{1/3+o(1)}$ algorithm and subsequent computations. The $t^{4/13+o(1)}$ algorithm is significantly more complicated and it is not clear that it would offer a practical improvement in the range of our computations.

Most of our computations were in small "targeted intervals" where zeta was expected to be unusually large. Our hope was that unusual behaviors of the zeros would be discovered in such intervals. The heuristic method used to find these intervals naturally produced very large t, and so most computations at large t were done out of necessity. This resulted in the largest computed value of $Z(t) \approx 16244.8652$ and $S(t) \approx 3.3455$, and in checking the RH for more than 50000 zeros in over 200 small intervals going up to the 10^{36}-th zero. See §4.

Our computations demonstrate that the $t^{1/3+o(1)}$ algorithm is practical. Indeed, almost all computations finished in a few days on the "riemann machine," a computer cluster at the University of Waterloo. (The implementation source code is available on the authors' websites.) This cluster has 16 nodes (though we usually limited our use to about 12 nodes) and each node has 8 cores at 2.27GHz clock speed. Probably, the range of t feasible via the current implementation and using easily available computational resources is past the 10^{40}-th zero.

2. The Riemann–Siegel formula

Let $s = \sigma + it$ where σ and t are real numbers. The Riemann zeta function $\zeta(s)$ is defined by

$$\zeta(s) := \sum_{n=1}^{\infty} n^{-s} \qquad (\sigma > 1). \tag{1}$$

It has a meromorphic continuation with a simple pole at $s = 1$, and satisfies the functional equation $\xi(s) = \xi(1-s)$ where $\xi(s) := \pi^{-s/2}\Gamma(s/2)\zeta(s)$. The RH is the conjecture that all zeros (i.e. roots) of $\xi(s)$ lie on the critical line $\mathrm{Re}(s) = 1/2$.

The RS formula is an asymptotic formula to compute $\zeta(s)$. Actually, the RS formula is usually stated for $Z(t) := e^{i\theta(t)}\zeta(1/2+it)$, which is a real-valued version of zeta on the critical line often called the Hardy Z-function. In particular, one can isolate simple non- trivial zeros of zeta merely by looking for sign changes of $Z(t)$. The phase factor $\theta(t)$ satisfies

$$\left|\theta(t) - \left(\frac{1}{2}\log\frac{t}{2\pi e} - \frac{\pi}{8} + \frac{1}{48t}\right)\right| \le \frac{0.129}{t^3}, \qquad (t > 1). \tag{2}$$

So one can compute $\theta(t)$ precisely and quickly for large t. Also, since $Z(t) = Z(-t)$ by the functional equation, we may restrict to $t \ge 0$.

The version of the RS formula that we used in our computations is the same as in [19]. For $t > 2\pi$, let $a := \sqrt{t/(2\pi)}$, $N := \lfloor a \rfloor$ the integer part of a, and $z = a - \lfloor a \rfloor$ the fractional part of a. Then the RS formula consists of a main sum

$$\mathcal{M}(t) = \sum_{n=1}^{N} \frac{e^{it\log n}}{\sqrt{n}}, \tag{3}$$

a correction $\mathcal{C}_m(t)$, and a remainder $R_m(t)$. Specifically, for each $m \in \mathbb{Z}_{\geq 0}$,

$$Z(t) = 2\,\Re e^{-i\theta(t)}\mathcal{M}(t) + \mathcal{C}_m(t) + R_m(t), \tag{4}$$

where $\mathcal{C}_m(t)$ is a sum of $m+1$ terms,

$$\mathcal{C}_m(t) = \frac{(-1)^{N+1}}{\sqrt{a}} \sum_{r=0}^{m} \frac{C_r(z)}{a^r}, \tag{5}$$

and $R_m(t)$ satisfies the bound $R_m(t) \ll t^{-(2m+3)/4}$. The functions $C_r(u)$ on the r.h.s. in (5) are linear combinations of derivatives of

$$F(u) := \frac{\cos(2\pi(u^2 - u - 1/16))}{\cos(2\pi u)}, \tag{6}$$

up to the $3r$-th derivative; see [7, 1]. For example, $C_0(u) = F(u)$ and $C_1(u) = F^{(3)}(u)/96\pi^2$, where $F^{(3)}(u)$ is the third derivative of $F(u)$. To evaluate $F(u)$, and more generally $C_r(u)$, near removable singularities at $u = \pi(2n+1)/2$, we used a numerically stable Taylor expansion.

The remainder $R_m(t)$ in (4) is well-controlled when t is large. Gabcke derived in his thesis [7] explicit bounds for $R_m(t)$ when $m \leq 10$. For example, if $t \geq 200$, then $|R_1(t)| < .053t^{-5/4}$, $|R_4(t)| < 0.017t^{-11/4}$, and $|R_{10}(t)| < 25966t^{-23/4}$. Even the bound for $R_1(t)$ is sufficient in our computations since t is of size $> 10^{20}$, so already $|R_1(t)| < 10^{-26}$.

3. Turing's method

The zeros of $\xi(s)$ are called the non-trivial zeros of zeta. They are denoted by $\rho_n = 1/2 + i\gamma_n$, $n \neq 0$. For example, $\rho_1 = 1/2 + i14.134725\ldots$, $\rho_2 = 1/2 + i21.022039\ldots$, $\rho_3 = 1/2 + i25.010857\ldots$, and $\rho_{-n} = \overline{\rho_n}$. The RH is the statement that the ordinates γ_n are always real. The counting function of zeros is

$$N(t) := |\{0 < \operatorname{Im}\rho_n < t\}| + \frac{1}{2}|\{\operatorname{Im}\rho_n = t\}|. \tag{7}$$

Using the functional equation and the argument principle from complex analysis, if $t \neq \operatorname{Im}\rho_n$ for any n, then

$$N(t) = \frac{1}{\pi}\theta(t) + 1 + S(t), \tag{8}$$

where $\theta(t)$ and $S(t)$ are defined[1] by a continuous variation in the argument of $\pi^{-s/2}\Gamma(s/2)$ and $\zeta(s)$, respectively, as s moves along the line segments from 2, where the argument is defined to be zero, to $2+it$ to $1/2+it$; see [21, 5, 6]. If $t = \operatorname{Im}\rho_n$ for some n, then we define $S(t) := \lim_{\epsilon \searrow 0} \frac{1}{2}(S(t+\epsilon) + S(t-\epsilon))$.

As mentioned earlier, $\theta(t)$ can be computed easily; it increases roughly linearly with no unpredictable oscillations. Therefore, in view of formula (8), the main difficulty is to compute $S(t)$, for which we employ Turing's method [22]. This is a particularly attractive method as it uses the already computed list of zeros together with a few evaluations of $Z(t)$. Basically, the value of $S(t)$ is determined at two points $t_2 > t_1$, which in turn determines $N(t_2) - N(t_1)$.[2] If the number of zeros found in $[t_1, t_2]$ matches this difference, then the completeness of the zeros list is

[1] This is the same $\theta(t)$ appearing in the definition of $Z(t)$.

[2] We assume that $Z(t_1)$ and $Z(t_2)$ are nonzero.

verified, and we are in a position to compute $S(t)$ throughout $[t_1, t_2]$. It is thus clear that the main issue is to find such t_1 and t_2.

To this end, Turing first observed that if the sign of $Z(t)$ is known, then the value of $S(t)$ is known modulo 2. It is therefore not necessary to compute $S(t)$ to any great accuracy. In fact, as observed in [6], it is advantageous to specialize to t a good gram point[3], for then $S(t)$ must be an even integer, and it suffices to prove that $|S(t)| < 2$ in order to conclude that $S(t) = 0$. The basic idea here is that $S(t)$ is small on average, satisfying the bound $|\int_t^{t+\Delta} S(y)dy| \le 0.128 \log(t+\Delta) + 2.30$, provided that $t > 168\pi$ and $\Delta > 0$. So if one incorrectly assumes that $S(t) \ge 2$, then, provided that sign changes of the Z-function are sufficiently regularly spaced around t, the average of $S(y)$ over $[t, t+\Delta]$ will contradict the required bound once Δ is large enough (roughly of size $\gg \log(t+\Delta)$). And therefore one can conclude that $S(t) < 2$. An analogous argument can be used to prove that $S(t) > -2$; see [22, 6] for details.

4. Examples of computations

The new methods described in this paper are suitable for evaluation of $Z(t)$ in short intervals. Accordingly, most of our computations have focused on evaluating the zeta function high on the critical line at spots where we might expect interesting behavior. Additionally, we have done some evaluation at spots where t or $N(t)$ is a nice "round" number; in these spots we expect to see "typical" behavior.

To find points where we expect to see interesting behavior, we used the LLL algorithm [15], as done in [17], to search for values of t where $p^{it} \approx 1$ for many initial primes p. By multiplicativity, then, there should be unusually many values of n^{it} close to 1, making the initial segment of the main sum large. Though we have not attempted to make this argument rigorous, it works well in practice, and we have observed many values of $Z(t)$ which are much larger than average by using LLL to line up the values of just one hundred or so initial p^{it}.

As a byproduct of our search for large values, we also find large values of $S(t)$. It is always the case in our computations that when $\zeta(1/2+it)$ is very large there is a large gap between the zeros around the large value. And it seems that to compensate for this large gap the zeros nearby get "pushed" to the left and right. A typical trend in the large values that we have found is that $S(t)$ is particularly large and positive before the large value and large and negative afterwards. This behavior can be seen in the plots in Figures 1 and 2.

As a consequence on the Riemann Hypothesis, $\zeta(1/2+it)$ is known to grow slower than any fixed power of t. Currently, the best conditional upper bound is

$$|\zeta(1/2+it)| \ll \exp\left(\frac{\log 2}{2}\frac{\log t}{\log\log t} + O\left(\frac{\log t \log\log\log t}{(\log\log t)^2}\right)\right), \tag{9}$$

[3]The m-th gram point g_m is the unique solution the equation $\theta(t) = \pi m \in \mathbb{Z}_{\ge -1}$ for $t \ge 7$. It is called good if $(-1)^m Z(g_m) > 0$. One usually finds a good gram point on testing $Z(t)$ at few consecutive gram points.

t	$Z(t)$
3924676458989430915525116928410.0506	16244.8652
7039106631049132430879196955445 3.2490	−14055.8928
5521664100099312888686808632346.5052	−13558.8331
3557586000421470624922724880597 7.2412	13338.6875
6632378187823588974002457910706.5963	12021.0940
6981562889715199161359429404 60.3337	11196.7919
2892860767193253077183805490 50.2563	10916.1145
5005475723107396211588045467 1617.4008	−10622.1763
8036257285923443631238142187 78.1993	10282.6496
6904226398239362545403022694 42.4854	10268.7134
1907915287180786223131860607197.5463	10251.5994
9832284408046499500622869540131.7445	−10138.5908

TABLE 1. All local maxima of $|Z(t)| > 10000$ found by our computations.

a result of Chandee and Soundararajan [3] improving the leading constant in earlier results. On the other hand, Bondarenko and Seip [2] have recently shown unconditionally that there exist values of t for which

$$|\zeta(1/2+it)| > \exp\left(\left(\frac{1}{\sqrt{2}}+o(1)\right)\sqrt{\frac{\log t \log\log\log t}{\log\log t}}\right),$$

improving on previous results by a factor of $\sqrt{\log\log\log t}$. Where exactly in this range the largest values lie is still an open question.

Table 1 has a list of the 12 local maxima of $|Z(t)|$ larger than 10000 that we have found to date. We do not know if the values that we have found are close to as large as possible.

Bounds for the growth of $S(t)$ are similar to the bounds for the growth of $\log|\zeta(1/2+it)|$. Again assuming the Riemann Hypothesis, Montgomery established in [16] that there are values of t for which

$$|S(t)| \gg \left(\frac{\log t}{\log\log t}\right)^{1/2}. \tag{10}$$

And Goldston and Gonek proved that $|S(t)| \le (1/2+o(1))\log t/\log\log t$. Even unconditionally, $|S(t)|$ is known to be unbounded by results of Selberg. Nevertheless, previous to these computations, the largest observed value of $S(t)$ seems to have been -2.9076, as reported by Gourdon [8]. Table 2 lists 11 spots where we have found values of $|S(t)| > 3.1$, the largest of which is $S(t) \approx 3.3455$ for $t \approx 7.75\times10^{27}$.

The data is just meant to be a sample of what we have computed. Further examples of computations can be found on the authors' websites. All together, we have checked the Riemann Hypothesis for more than 50000 zeros in over 200 separate small intervals with values of t ranging from 10^{24} to 8×10^{34}.

t	$S(t)$
77573049903678614171502130536386	3.3455
54657756232105712480149851681946 09	−3.2748
35575860004214706249227248805976.9763	3.2722
31774695316763918183637654364.8066	3.2573
11580026442432493576924087062.5414	−3.2371
10758662450340950434456735185.3359	−3.2261
50054757231073962115880454671617.8419	−3.1826
50054757231073962115880454671617.8419	−3.1826
39246764589894309155251169284103.7774	3.1694
10251393160473423776137882271.3031	3.1660
77590565202125505656738011641.6876	3.1431

TABLE 2. Spots where $|S(t)| > 3.1$ found by our computations. In this table, t actually denotes the imaginary part of a zero of $\zeta(s)$, so the value of $S(t)$ is attained just before (if negative) or after this zero.

n	γ_n
10^{25}	11944793301783015851478 71.32909
10^{26}	11452628915113964213507127.18757
10^{27}	10999095561574854224192062 1.36163
98094362213058141112271181439	$10^{28} + 0.00366$
10^{29}	10191135223869807023206505980.23860
$10^{30} + 484$	98297762869274424758690514889.09764
10175904020745527981663511857 65	$10^{29} + 0.07316$
10^{31}	94929882975455496405878655 9878.40484
10^{32}	91783586564949893364312590048 05.28194
10^{33}	88837796029624663862630219091104.93992
$10^{36} + 42420637374017961984$	81029194732694548890047854481676712.98790

TABLE 3. Some examples of zeros of $\zeta(1/2+it)$. Here the notation means that $\zeta(1/2+i\gamma_n) = 0$ for some real number γ_n within about 10^{-5} of the number in the right column, and that there are exactly $n-1$ zeros in of $\zeta(s)$ in the critical strip with positive imaginary part $< \gamma_n$.

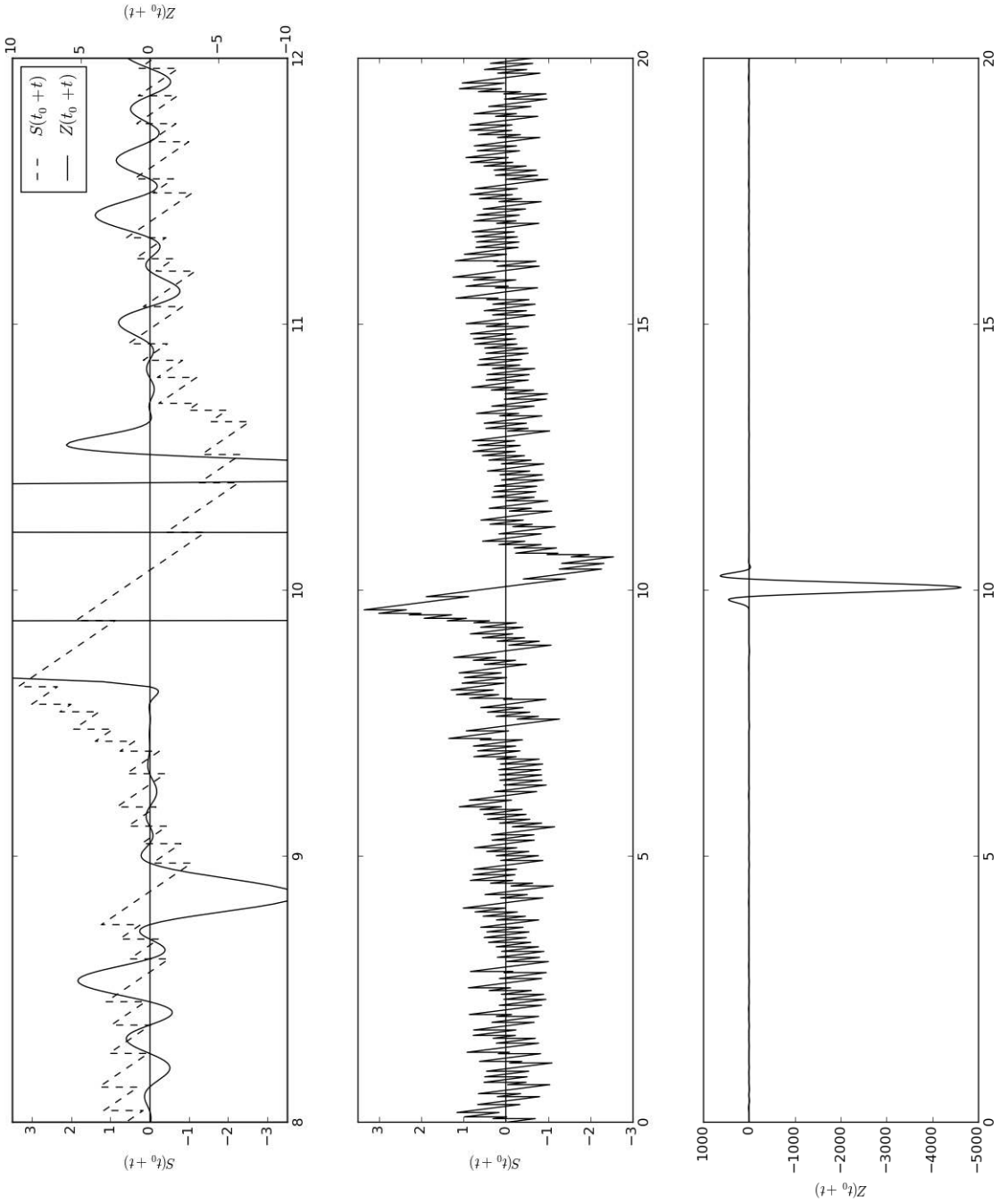

FIGURE 1. $Z(t)$ and $S(t)$ around the largest value of $S(t)$ we have found. Here $t_0 = 7757304990367861417150213044$.

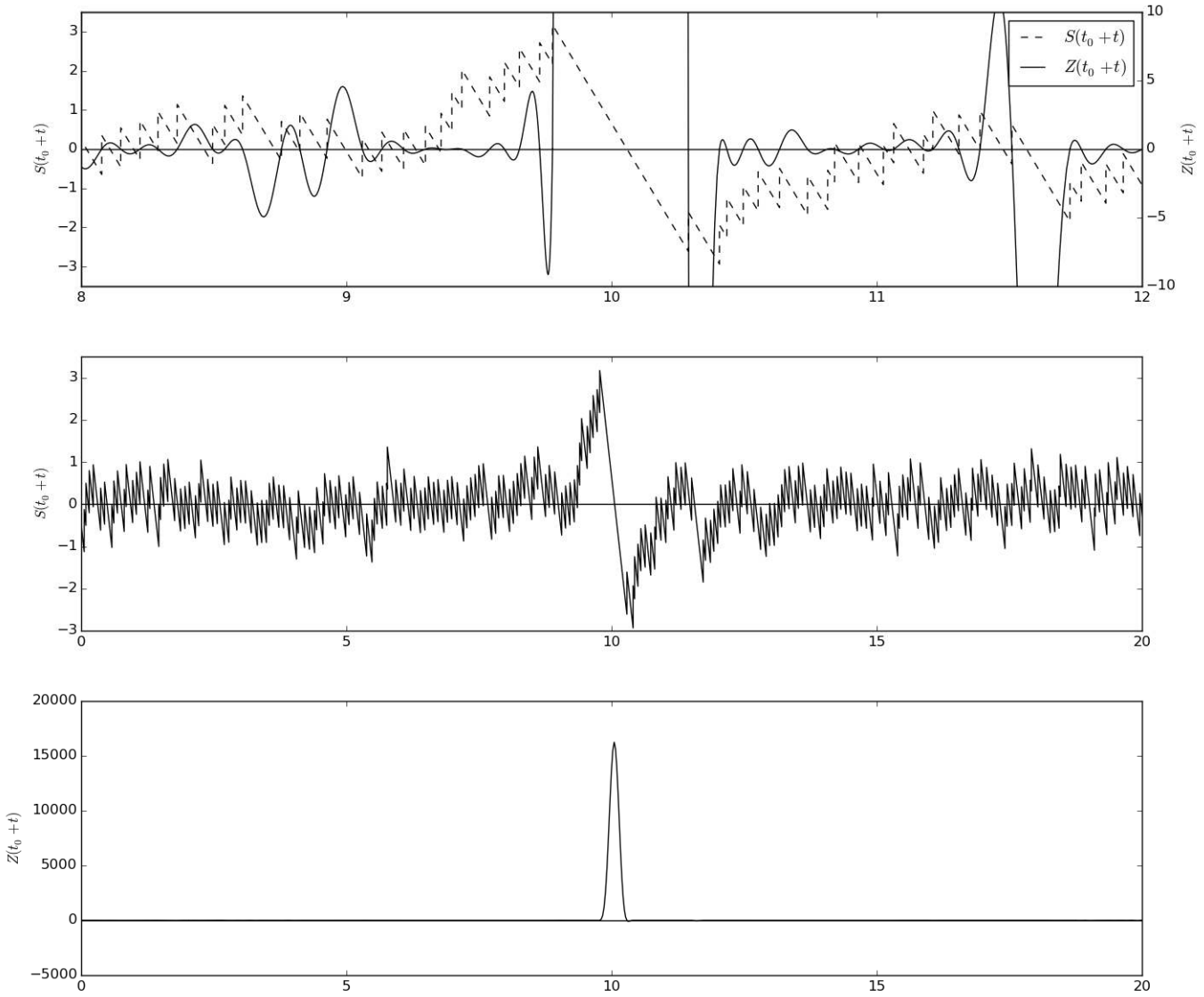

FIGURE 2. $Z(t)$ and $S(t)$ around the largest value of $Z(t)$ we have found. Here $t_0 = 392467645898943091552511692840 94$.

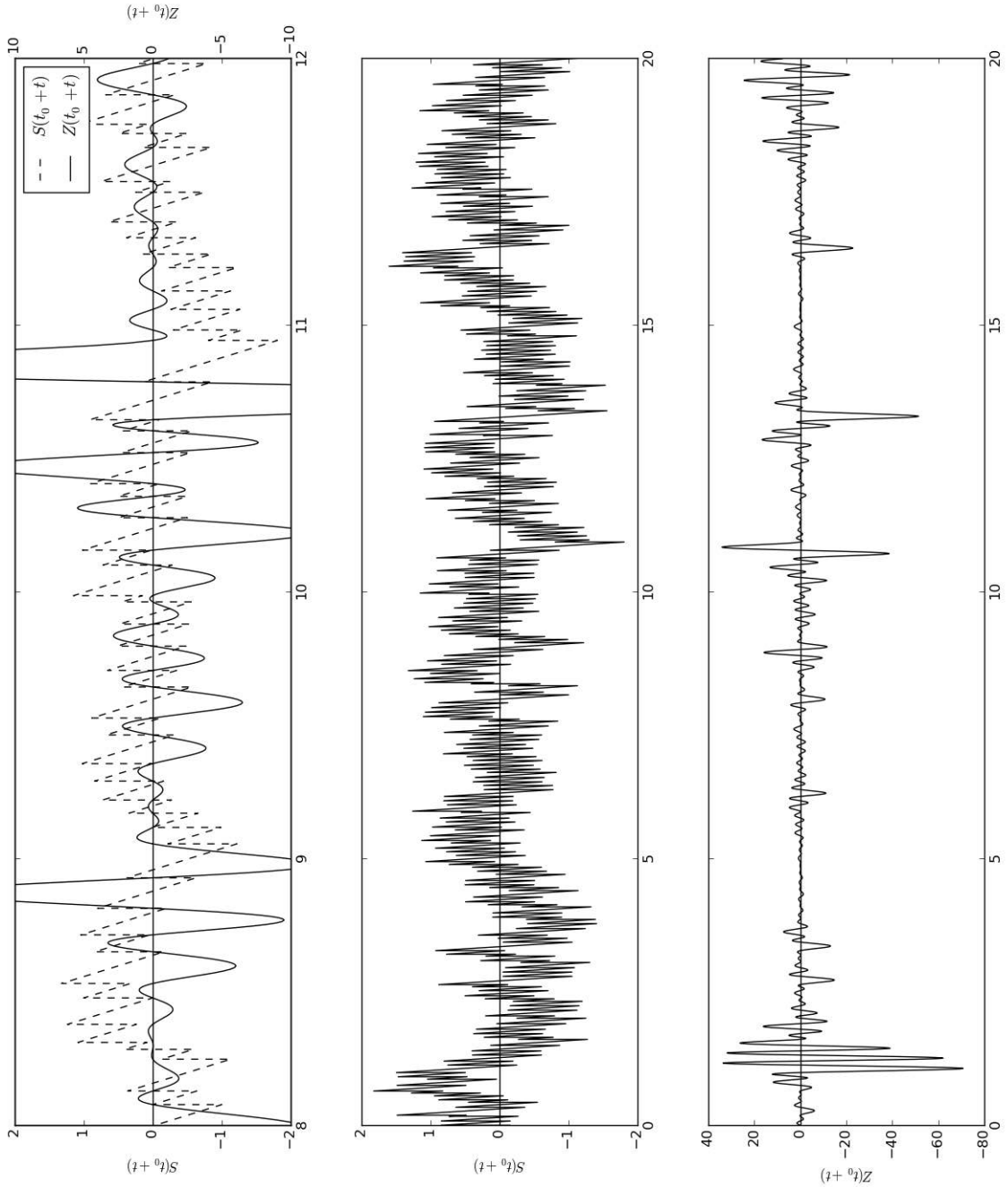

FIGURE 3. $Z(t)$ and $S(t)$ around the zero number $10000000000000004242063737401796198 4 > 10^{36}$, where $t_0 = 81029194732694548890047854481676703$. In contrast to Figures 1 and 2, there is nothing particularly special about this range of t, so we should expect this to exhibit typical behavior of $S(t)$ and $Z(t)$.

5. Reducing to quadratic exponential sums

To start, we choose a positive integer $v_0 \le N+1$ and a real number $u_0 > 1$. Then we construct the sequences $K_r = \lceil v_r/u_0 \rceil$ and $v_{r+1} = v_r + K_r$ for $0 \le r < R$, where R is the largest integer such that $v_R < N+1$. We define $K_R := \min\{\lceil v_R/u_0 \rceil, N+1-v_R\}$. So $v_{R+1} = N+1$, which is the first point outside the range of summation of the main sum $\mathcal{M}(t)$. We subdivide $\mathcal{M}(t)$ into an initial sum of $v_0 - 1$ terms, followed by $R+1$ consecutive blocks where the r-th block starts at v_r and consists of K_r consecutive terms. We use the Taylor expansion to express the r-th block as a linear combination of the quadratic exponential sums

$$F(K, j; a, b) = \frac{1}{K^j} \sum_{k=0}^{K-1} k^j e^{2\pi i a k + 2\pi i b k^2}, \qquad j \in \mathbb{Z}_{\ge 0}. \tag{11}$$

(We treat k^j as 1 when $k = j = 0$.) To this end, we define

$$f_s(z) := \frac{e^{(s-1/2)(z-z^2/2)}}{(1+z)^s}, \qquad f_s(0) = 1. \tag{12}$$

For $n \in [v_r, v_r + K_r)$, we write $n = v_r + k$ where $0 \le k < K_r$. Then, letting $s^* = 1/2 - it$ and noting that

$$\frac{e^{it\log n}}{\sqrt{n}} = \frac{e^{it\log v_r}}{\sqrt{v_r}} e^{itk/v_r - itk^2/2v_r^2} f_{s^*}(k/v_r), \tag{13}$$

we obtain, on expanding $f_{s^*}(z)$ around 0, that

$$\sum_{v_r \le n < v_r + K_r} \frac{e^{it\log n}}{\sqrt{n}} = \frac{e^{it\log v_r}}{\sqrt{v_r}} \sum_{j=0}^{\infty} c_r(j) F(K_r, j; a_r, b_r), \tag{14}$$

where the linear and quadratic arguments a_r and b_r are given by the formulas $a_r := t/2\pi v_r$ and $b_r := -t/4\pi v_r^2$. The coefficients $c_r(j)$ are given by

$$c_r(j) := \left(\frac{K_r}{v_r}\right)^j \frac{f_{s^*}^{(j)}(0)}{j!}, \tag{15}$$

where $f_s^{(j)}(z)$ is the j-th derivative in z. For instance, $f_s^{(1)}(z) = -1/2$, $f_s^{(2)}(z)/2! = 3/8$, and $f_s^{(3)}(z)/3! = -(16s+7)/48$. In order to avoid clutter, we have suppressed dependence on t, u_0, and v_0 in all of $c_r(j)$, a_r, b_r, and R.

Let J denote a positive integer, and let $\mathbf{c}_r = (c_r(0), \dots, c_r(J))$ denote a tuple of the first $J+1$ coefficients. Define

$$\mathbf{F}(K, \mathbf{c}; a, b) = \sum_{j=0}^{J} c(j) F(K, j; a, b), \tag{16}$$

which is a linear combination of quadratic sums. Proposition 5.1, which is proved in §9, bounds the truncation error $\epsilon_J(t)$ in the main sum when the series on the r.h.s. of (14) is stopped after $J+1$ terms for each block. That is, $\epsilon_J(t)$ satisfies

$$\mathcal{M}(t) = \sum_{n=1}^{v_0-1} \frac{e^{it\log n}}{\sqrt{n}} + \sum_{r=0}^{R} \frac{e^{it\log v_r}}{\sqrt{v_r}} \mathbf{F}(K_r, \mathbf{c}_r; a_r, b_r) + \epsilon_J(t), \tag{17}$$

Proposition 5.1. *Let $\alpha = 1/0.9$, $K_{\min} = 2000$. If $v_0 = \lceil K_{\min} t^{1/3} \rceil \le \frac{1}{3}\sqrt{t/(2\pi)}$ say, and $u_0 = \alpha|s|^{1/3} \ge 1$, then $\epsilon_J(t)$ is bounded as in the following table.*

t \ J	18	21	24	27	30
10^{24}	0.000831893	0.0000386099	0.0000108755	7.46138×10^{-6}	5.42799×10^{-6}
10^{26}	0.00283154	0.0000925556	7.50489×10^{-6}	3.85022×10^{-6}	2.76739×10^{-6}
10^{28}	0.00839067	0.000256977	9.10602×10^{-6}	1.82102×10^{-6}	1.21035×10^{-6}
10^{30}	0.0230403	0.000698646	0.0000196934	1.1228×10^{-6}	4.96568×10^{-7}
10^{32}	0.0603446	0.00182714	0.0000495358	1.44903×10^{-6}	2.13244×10^{-7}
10^{34}	0.15309	0.00463399	0.000124900	3.12046×10^{-6}	1.36063×10^{-7}
10^{36}	0.37952	0.011489	0.00030938	7.53404×10^{-6}	1.89985×10^{-7}

TABLE 4. Bounds on $\epsilon_J(t)$ for various J and t.

The bounds in Table 5.1 are calculated as $|\epsilon_J(t)| \le c_{\max}(J,t)\mathcal{M}_{\max}(t)$ where $c_{\max}(J,t)$ is essentially a bound on the truncation error after J terms in the Taylor expansion of $f_{s^*}(z)$ at $z=0$, and $\mathcal{M}_{\max}(t)$ is a bound on the sum of the $R+1$ blocks. The bound $\mathcal{M}_{\max}(t)$ that we proved is of the form $\ll t^{1/6}\log t$, which we know is a significant overestimate, and is in fact the main source of inefficiency in Proposition 5.1. In practice, $\epsilon_J(t)$ is bounded by something like $c_{\max}(J,t)\sqrt{\log(N/v_0)}$, though we cannot prove this. In any case, Table 5 furnishes bounds on $c_{\max}(J,t)$ alone. The numbers appearing there should be much closer to the true truncation error in our computations. Note that $c_{\max}(J,t)$ depends mostly on J, and that its dependence on t is weak in comparison.

t \ J	18	21	24	27	30
10^{24}	1.048×10^{-8}	4.864×10^{-10}	1.370×10^{-10}	9.399×10^{-11}	6.838×10^{-11}
10^{26}	1.028×10^{-8}	3.358×10^{-10}	2.723×10^{-11}	1.397×10^{-11}	1.004×10^{-11}
10^{28}	1.025×10^{-8}	3.137×10^{-10}	1.112×10^{-11}	2.223×10^{-12}	1.478×10^{-12}
10^{30}	1.024×10^{-8}	3.105×10^{-10}	8.751×10^{-12}	4.989×10^{-13}	2.207×10^{-13}
10^{32}	1.024×10^{-8}	3.100×10^{-10}	8.404×10^{-12}	2.459×10^{-13}	3.618×10^{-14}
10^{34}	1.024×10^{-8}	3.099×10^{-10}	8.353×10^{-12}	2.087×10^{-13}	9.099×10^{-15}
10^{36}	1.024×10^{-8}	3.099×10^{-10}	8.345×10^{-12}	2.033×10^{-13}	5.125×10^{-15}

TABLE 5. Bounds on $c_{\max}(J,t)$ for various J and t.

We consider the sensitivity of the main sum to perturbations in the quadratic sums in (17). Such perturbations arise from the accumulation of roundoff errors when using floating point arithmetic. If each sum $\mathbf{F}(K_r, \mathbf{c}_r; a_r, b_r)$ in (17) is computed to within ε, say, then this introduces a total error $\le \varepsilon \sum_{r=0}^{R} 1/\sqrt{v_r}$. Choosing u_0 and v_0 as in Proposition 5.1, and noting $v_{r+1} = v_r + K_r \ge v_r(1+1/u_0)$, it is easy to show that this error is $\le 0.05\varepsilon t^{1/6}$, provided that $u_0 \ge 200$ and $t \ge 10^{10}$ say. In practice though, this maximal size is never reached. Instead, due to pseudorandom nature of roundoff errors, one observes square-root cancellation in their sum. So the cumulative error is typically $\le \varepsilon(\sum_{r=0}^{R} 1/v_r)^{1/2}$. Thus, under the same assumptions on u_0 and t as before, the error is $\le 0.0011\varepsilon$.

6. Computing quadratic exponential sums

The main new ingredient in our algorithm to compute the zeta function is an implementation of Hiary's algorithm to compute quadratic exponential sums $\mathbf{F}(K, \mathbf{c}; a, b)$. The algorithm runs in $O((J+1)^A \log^A(K/\epsilon))$ bit operations, where ϵ is the desired accuracy and $A \le 3$. This algorithm was derived in [10] to compute $\zeta(1/2+it)$ in $t^{1/3+o(1)}$ time, which is the method implemented in our computations. Currently, the fastest method for computing zeta at a single point has asymptotic running time $t^{4/13+o(1)}$; see [9]. This method relies on computing cubic exponential sums instead of quadratic sums, and is practical only if t is extremely large. (The issue of practical t threshold for the $t^{4/13+o(1)}$ method is being tackled in upcoming work by the second author.)

From a high level point of view, in the typical case the algorithm for quadratic sums works by using Poisson summation to write $\mathbf{F}(K, \mathbf{c}; a, b)$ as a combination of a shorter sum and some integrals which can be calculated to whatever precision we like. The length of the new sum will be $\lfloor a + 2bK \rfloor$, and we will be able to assume that $0 \le b < 1/4$, so the length of the sum decreases quickly. There are some cases where this Poisson summation does not work well, but these correspond precisely to the case where b is very small and we can compute this sum by using Euler–Maclaurin summation.

From [10], we distill Proposition 6.1. The proof of this proposition is essentially the content of Equations (3.37), (3.38), and (3.39) [10], though we have modified the notation in some ways, and have explicitly written out the result for general $\mathbf{v}$, rather than only for $\mathbf{v} = (0, 0, \dots, 1)$, as is done in [10]. (This implicitly involves changing an order of summation, which causes the appearance of z_j and z'_j and makes computation more efficient.) The formula in Proposition 6.1 is fairly complicated since, following [10], it is completely explicit and avoids numerical differentiation. This ensures better and more robust performance in practice.

The algorithm works by applying the formula in Proposition 6.1 repeatedly, until either $b \ll 1/K$, in which case the Euler–Maclaurin summation is used, or $K \le K_{\min}$, in which case direct summation is used.

The algorithm for computing quadratic sums has been subsequently implemented by Kuznetsov [14], but using Mordell integral identities instead of the Poisson summation and relying on numerical differentiation.

Proposition 6.1. *If K is a positive integer, $a \ge 0$, and $b > 0$, then we have the transformation*

$$\mathbf{F}(K, \mathbf{v}; a, b) = \mathbf{F}(q, \mathbf{v}'; a', b') + \mathbf{R}(K, \mathbf{v}; a, b),$$

where $q = \lfloor a + 2bK \rfloor$ is the length of the new quadratic sum, $a' = a/(2b)$ is the new linear argument, and $b' = -1/(4b)$ is the new quadratic argument. The new coefficient vector $\mathbf{v}' = (v'_0, \dots, v'_J)$ is defined by

$$v'_j = e^{-\frac{i\pi a^2}{2b}} q^j b^{-\frac{j}{2}} \sum_{\ell=j}^{J} v_\ell b^{-\frac{1+\ell}{2}} K^{-\ell} A_{j,\ell} \sum_{\substack{k \equiv \ell - j \bmod 2 \\ 0 \le k \le \ell - j}} a^k b^{-\frac{k}{2}} B_{\ell-j,k},$$

$$A_{j,\ell} = \frac{\ell!}{j!} \pi^{\frac{j-\ell}{2}} 2^{\frac{j-3\ell-1}{2}} e^{\frac{\pi i}{4}(1+3(\ell-j))}, \qquad B_{j,k} = \frac{(-1)^{\frac{k+j}{2}}}{\left(\frac{j-k}{2}\right)! \, k!} (2\pi)^{\frac{k}{2}} e^{-\frac{3\pi i k}{4}}.$$

The remainder $\mathbf{R}(K,\mathbf{v};a,b)$ *is given explicitly by*

$$\mathbf{R}(K,\mathbf{v};a,b) = e^{2\pi iaK+2\pi ibK^2}\left[e^{\frac{\pi i}{2}}\sum_{j=0}^{J} S_j + \frac{1}{2}\sum_{j=0}^{J} v_j\right] + \frac{v_0}{2} - \delta_{\lceil a\rceil -1}v_0'.$$

The S_j *are defined as follows. Let* $z_j = i^j\sum_{\ell=j}^{J} v_\ell\binom{\ell}{j}$, $z_j' = \sum_{\ell=j}^{J} v_\ell\binom{\ell}{j}2^{\frac{\ell+1}{2}}e^{\frac{i\pi(\ell+1)}{4}}$, $\omega = \{a+2bK\}$, *and* $\omega_1 = \lceil a\rceil - a$. *Then*

$$\begin{aligned}
S_j = z_j\Bigg[&\mathcal{I}_{\tilde{C}_1}(K,j,\omega,b) - \mathcal{I}_{C_7}(K,j,\omega,b) - \mathcal{J}_1(\omega,b,j,q-\lceil a\rceil,K)\\
&+ (-1)^j\mathcal{I}_{C_{9H}}(K,j,1-\omega,b) + \mathcal{J}_2(2bK-\omega_1,1-\omega,b,j,K)\Bigg]\\
&+ v_j\Bigg[(-1)^{j+1}\mathcal{J}_1(\omega_1,b,j,q-\lceil a\rceil,K) + (-1)^{j+1}\mathcal{I}_{C_7}(K,j,\omega_1,b)\\
&+ \mathcal{I}_{C_{9H}}(K,j,1-\omega_1,b) + (-i)^{j+1}\mathcal{J}_2(2bK-\omega,1-\omega_1,b,j,K)\Bigg]\\
&- z_j' e^{2\pi iaK-2\pi\omega K}\mathcal{I}_{C_{9E}}(K,j,\omega,b).
\end{aligned}$$

Here, the $\mathcal{J}$ *terms are given by the integrals*

$$\mathcal{J}_1(\alpha,\beta,j,M,K) = \frac{1}{K^j}\int_0^K t^j\exp(-2\pi\alpha t - 2\pi i\beta t^2)\frac{1-\exp(-2\pi Mt)}{\exp(2\pi t)-1}\mathrm{d}t,$$
$$\mathcal{J}_2(\alpha_1,\alpha_2,\beta,j,K) = \frac{1}{K^j}\int_0^K t^j\exp(-2\pi\beta t^2)\frac{\exp(-2\pi\alpha_1 t)+(-1)^{j+1}\exp(-2\pi\alpha_2 t)}{\exp(2\pi t)-1}\mathrm{d}t.$$

The $\mathcal{I}$ *terms are all integrals of the same integrand along different paths in the complex plane, which we can write explicitly as integrals over segments of* $\mathbb{R}_{\geq 0}$ *as*

$$\begin{aligned}
\mathcal{I}_{C_7}(K,j,\alpha,\beta) &= e^{-\frac{i\pi(j+1)}{4}}\frac{1}{K^j}\int_0^{K\sqrt{2}} t^j\exp\left(-(1+i)\pi\sqrt{2}\alpha t - 2\pi\beta t^2\right)\mathrm{d}t,\\
\mathcal{I}_{C_{9H}}(K,j,\alpha,\beta) &= \frac{1}{K^j}\int_0^{\infty} t^j\exp(-2\pi\alpha t - 2\pi i\beta t^2)\mathrm{d}t,\\
\mathcal{I}_{C_{9E}}(K,j,\alpha,\beta) &= \frac{1}{K^j}\int_0^{\infty} t^j\exp(-2\pi(\alpha - i\alpha + 2\beta K + 2i\beta Kt - 4\pi\beta t^2)dt,\\
\mathcal{I}_{\tilde{C}_1}(K,j,\alpha,\beta) &= -ie^{-2\pi\alpha K-2\pi i\beta K^2}\int_0^K t^j\exp(2\pi i\alpha t - 4\pi\beta Kt + 2\pi i\beta t^2)\mathrm{d}t.
\end{aligned}$$

Some remarks:

- The formula in Proposition 6.1 takes on a simpler form if $J = 0$.
- The integral $\mathcal{J}_2$ occurs from certain "boundary" terms in the computation, while the bulk of the contribution to the sum generally comes from the $\mathcal{J}_1$.
- If ω is not too small (compared to the target precision) then the terms involving $\mathcal{I}_{C_{9E}}$ and $\mathcal{I}_{\tilde{C}_1}$ are not large enough to make any significant contribution, so we do not need to compute them.

7. Simple multi-evaluation

Thus far we have only described how to evaluate $\zeta(1/2+it)$ at a single point. To locate zeros and make plots of zeta, we of course want to evaluate at more points. We note that the sum $\mathcal{M}(t)$, which consumes almost all the computation time, is a bandlimited function with highest frequency $\tau = \log\lfloor\sqrt{t/2\pi}\rfloor$; see [17]. So we can use interpolation to recover $\mathcal{M}(t)$ for any t in a small window if we have already evaluated it on a relatively coarse grid of points covering a slightly larger window. Such a grid consists of points spaced π/β apart, with $\beta > \tau$. In light of this, we focus first on the problem of computing $\mathcal{M}(t_0 + \delta j)$ for a range of integers j, with t_0 large and $\delta j \ll 1$. For concreteness, we might imagine that $10^{24} < t_0$, $\delta = .04$, and $0 \le j \le 1000$, which corresponds to the evaluation of $\zeta(1/2 + it)$ in a window of size 40. To further simplify matters, we avoid the vicinity of t_0 where the length of $\mathcal{M}(t)$ changes; i.e. t_0 of the form $2\pi n^2$ for some integer n.

We perform the multi-evaluation on each block, which we recall have the shape

$$\sum_{n=v}^{v+K-1} \frac{e^{i(t_0+\delta j)\log n}}{\sqrt{n}} = \frac{e^{i(t_0+\delta j)\log v}}{\sqrt{v}} \sum_{k=0}^{K-1} \frac{e^{i(t_0+\delta j)\log(1+k/v)}}{\sqrt{1+k/v}}. \tag{18}$$

The entirety of our multi-evaluation is based on the simple observation that in factoring out the first term in this sum we have removed most of the oscillation from the summands. So the inner sum in (18) changes little with j, and it suffices to approximate by its value at t_0 only.

Let us denote the inner sum in (18) by $V(t, v, K)$. Then using the inequality $\log(1 + x) \le x$ for $0 \le x \le 1$, and the bound $(K - 1)/v \le 1/u_0$, we obtain

$$|V(t_0 + \delta j, K, v) - V(t_0, K, v)| \le \frac{\rho\delta j}{u_0} \max_{0\le\Delta<K} \left| \sum_{k=\Delta}^{K-1} e^{it_0\log(v+k)} \right|, \tag{19}$$

where $\rho > 1$ depends on t_0. Given the range of t_0 under consideration, we can show that $\rho = 1.1$ is admissible. So, summing over all blocks, the total error in the multi-evaluation method is at most $\rho\delta j u_0^{-1}\mathcal{M}_{\max}(t_0)$. Now, $\mathcal{M}_{\max}(t_0) \le At_0^{1/6}\log t_0$ for some constant A (see Lemma 9.1), and by the main result in [13] we should be able to take $A \le 1$. Moreover, we have $u_0 \ge \alpha t^{1/3}$ where we took $\alpha = 1/0.9$ in our computations. Thus, the error in the multi-evaluation is

$$\le \frac{\rho A\delta j\log t_0}{\alpha t_0^{1/6}} \le \frac{\delta j\log t_0}{t_0^{1/6}}. \tag{20}$$

In light of this, the multi-evaluation can be carried out safely over a large range of j (of length $t_0^{1/6-\epsilon}$). In practice, the estimate (20) is conservative because the estimate for $\mathcal{M}_{\max}(t_0)$ is wasteful. Almost surely the cumulative error will be significantly below the maximal size (20), and is much closer to

$$\le \frac{\delta j\sqrt{\log(N/v_0)}}{t_0^{1/3}}. \tag{21}$$

8. Our implementation

In this section, we document our implementation of the $t^{1/3+o(1)}$ algorithm. We remark that due to special features of the theta algorithm in [10] (e.g. strong links

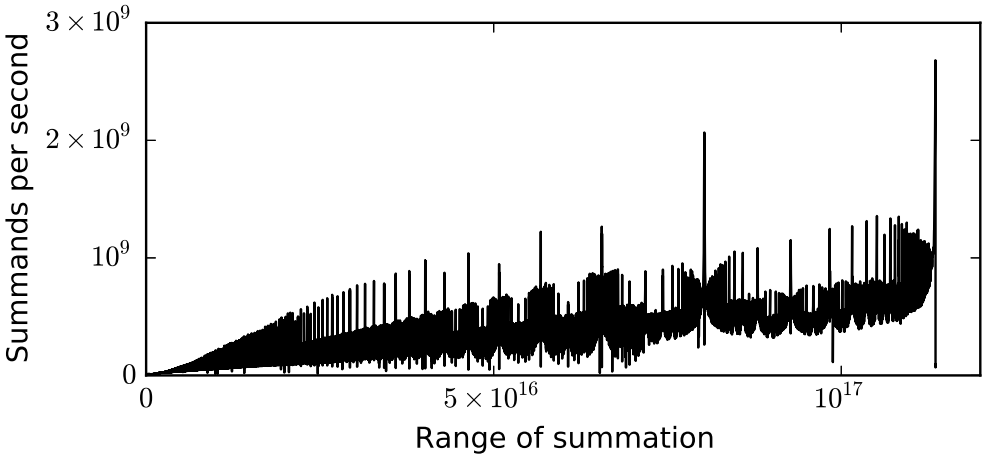

FIGURE 4. The computation at the largest height was split into over 130000 independent blocks of computation, each computing a consecutive piece of the main sum. Here we show the "speed" of our implementation in terms of the number of summands per second summed in each block. We can see here that the diophantine properties of the input affect the running time of the theta algorithm; for example, the largest spikes correspond to summands $n^{-1/2+it}$ where $t/(4\pi n^2)$ is very close to an integer.

between the running time and number-theoretic properties of the inputs), there are several spots of optimization, but we have explored only few of them.

8.1. **Quadratic exponential sums.** We have a general implementation of the algorithm described by Proposition 6.1, focused on moderate precision output for general input. The entirety of our `C++` code currently consists of a bit over 3000 lines, though some of that is for precomputed tables and constants that are produced quickly. It was quite useful during the implementation to constantly compare answers obtained from the `C++` code with answers obtained from a basic version of the algorithm that was implemented in `Mathematica`. The end result is a single function which computes the sum $F(K, \mathbf{v}; a, b)$ to a specified "attempted" absolute precision ϵ. By this we mean that if our implementation were bug-free, and if our subroutines employ sufficient working precision consistently (as they hopefully should), then we would compute the answer to within ϵ.

In practice, floating point arithmetic is not exact, and we limit our use of high precision arithmetic (using `MPFR`) to small areas of the algorithm where it is completely unavoidable. For example, if K is very large then $F(K, \mathbf{v}; a, b)$ typically oscillates rapidly as b changes (on a scale like $1/K^2$), so it is necessary to specify the value of b sufficiently precisely. For typical input (for example, with $J = 18$, $\mathbf{v} = (1, 1, \ldots, 1)$, and a and b arbitrary) we expect to be able to able to compute up to an absolute precision of around 10^{-10} with our implementation, and we usually get over 40 bits of relative precision.

There are also some other limitations in our implementation that come from our use of `C++` doubles. For example, there are numbers that occur in the computation which may become too large or too small to be represented as double precision numbers, but in practice our implementation will usually fail before we reach this point. We hope, at least, that our implementation will return "not a number" in such a case to indicate an error, rather than returning a wrong answer. These are technical limitations, however, and with some effort they could be removed at little cost to the overall running time of our implementation.

For efficient evaluation of the integrals arising in Proposition 6.1, we cannot rely on straightforward numeric integration. Instead we must deal with each integral on a case-by-case basis and evaluate as appropriate. Many of the details are given in [10], and we do not include them all here, but as a typical example we explain here how one might evaluate the integral

$$\mathcal{J} := \mathcal{J}_1(\alpha,\beta,0,M,1) = \int_0^1 \exp(-2\pi\alpha t - 2\pi i\beta t^2)\frac{1-\exp(-2\pi Mt)}{\exp(2\pi t)-1}\mathrm{d}t.$$

Here, $\alpha \in [0,1]$ and $\beta \in [0,1/4]$. We can begin by replacing $\exp(-2\pi i\beta t^2)$ by its Taylor series, and get

$$\mathcal{J} = \sum_{r=0}^{\infty}\frac{(-2\pi i\beta)^r}{r!}\int_0^1 t^{2r}\exp(-\pi at)\frac{1-\exp(-2\pi Mt)}{\exp(2\pi t)-1}\mathrm{d}t.$$

The integral inside the sum will be small, so we can truncate the infinite sum after a small number of terms, and we now focus on this inner integral, in which we can expand the geometric sum

$$\frac{1-\exp(-2\pi Mt)}{\exp(2\pi t)-1}$$

to get that the inner integral equals

$$\sum_{m=1}^{M}\int_0^1 t^{2r}\exp(-2\pi(\alpha+m)t)dt.$$

This integral is now a fairly simple function, and we have a number of choices for how to evaluate it. For example, we can evaluate it explicitly using its antiderivative, or we can again use the Taylor expansion for the exponential and integrate term-by-term, or we can write it in terms of an incomplete gamma function and use a continued fraction expansion; these methods each work well for different ranges of parameters.

This still does not completely solve the problem, as M may be very large. To deal with this, once m is past a certain size we use a simple approximation to the antiderivative of the integrand and Euler–Maclaurin summation to compute the sum over the full range.

8.2. **The main sum.** We compute the sum $\mathcal{M}(t) = \sum_{n=1}^{N} n^{-1/2+it}$ in three stages, as different methods of computation are appropriate for different sizes of n. We write

$$\mathcal{M}(t) = \mathcal{M}_1(t) + \mathcal{M}_2(t) + \mathcal{M}_3(t),$$

where

$$\mathcal{M}_1(t) = \sum_{n_0\le n<n_1}\frac{n^{it}}{n^{1/2}}, \quad \mathcal{M}_2(t) = \sum_{n_1\le n<n_2}\frac{n^{it}}{n^{1/2}}, \quad \mathcal{M}_3(t) = \sum_{n_2\le n<n_3}\frac{n^{it}}{n^{1/2}},$$

with $n_0 = 1, n_1 \asymp t^{1/4}, n_2 \asymp t^{1/3}$ and $n_3 = N$.

In stage 1, we compute the sum $\mathcal{M}_1(t)$ directly. More specifically, we compute each term to roughly 50 bits of relative precision, then add up the terms. This ensures that in the ranges of t where we computed the roundoff error $|\mathcal{M}_1(t) - \mathrm{fl}(\mathcal{M}_1(t))|$ is far subsumed by the "practical truncation error" discussed in §5.

In stage 2 we still add up each term in the sum individually, but gain efficiency by approximating the exponent $(it - 1/2)\log(1 + k/v)$ in successive terms using a rapidly decaying Taylor series instead of computing it directly using expensive multiprecision arithmetic for the logarithm function. Another advantage to this approach is that, now, double precision arithmetic suffices to to compute many of the terms in the Taylor expansion, and higher precision is only needed for the first few terms (the exact number of these terms and the needed working precision are both determined by the attempted absolute precision ϵ and using formulas coded in the implementation). The overall effect is to ensure that the roundoff error from stage 2 is again subsumed by the practical truncation error.

In stage 3 we approximate by quadratic exponential sums as described in §5 and apply the theta algorithm described in §6, passing a requested precision parameter ε to the algorithm. In turn, ε is passed to subfunctions in our code in order to determine the needed working precision for each subfunction, and so on. Significant effort was spent on sharpening formulas to calculate the working precision for each subfunctions. This enabled a numerically more robust implementation and facilitated the discovery of programming bugs. Actually, the precision of the theta algorithm in practice is much better than the requested precision ε. Here is a typical example. On a test suite of 29952 sums which might come from a computation of $\zeta(1/2 + i10^{36})$, when the theta algorithm is called with a precision of $\varepsilon = 10^{-5}$, the worst error is 5.361×10^{-6} but the typical error is around 7.45×10^{-9}, and the vast majority of the errors are $< 3 \times 10^{-8}$.

In summary, we expect the values of $Z(t)$ that we computed to be accurate to within $\pm 10^{-7}$ typically, and accurate to within $\pm 10^{-5}$ in the worst cases. The bulk of the error comes from the practical truncation error discussed in §5. In comparison, the practical roundoff and multi-evaluation errors are small.

8.3. **Example running time.** Figure 4 gives an indication of the speed of our current implementation. We graph the number of terms per second computed in large blocks of the main sum at the largest height computation we ran, in a range around $t = 8102919473269454889004785448167671 2$. For this computation, we used approximately 22.5 cpu-core-years on the BlueCrystal Phase 2 cluster at the University of Bristol, which has 2.8 Ghz Intel Xeon E5462 cpus. Using just our "stage 2" code, we estimate that the same computation would take around 300 core-years instead. Towards the end of the range the "stage 3" code often has a speedup over "stage 2" by a factor of around 200.

9. Proofs

Proof of Proposition 5.1. By definition

$$\epsilon_J(t) = \sum_{r=0}^{R} \frac{e^{it\log v_r}}{\sqrt{v_r}} \sum_{j>J} c_r(j) F(K_r, j; a_r, b_r). \tag{22}$$

Applying partial summation to $F(K_r, j; a_r, b_r)$ gives

$$|F(K_r, j; a_r, b_r)| \le \left(\frac{K_r - 1}{K_r}\right)^j F_{\max}(K_r; a_r, b_r), \tag{23}$$

where

$$F_{\max}(K; a, b) = \max_{0 \le \Delta < K} \left| \sum_{k=\Delta}^{K-1} e^{2\pi i a k + 2\pi i b k^2} \right|. \tag{24}$$

In view of this, let us define

$$\begin{aligned} c_{\max}(J, t) &= \max_{0 \le r \le R} \sum_{J < j} \left(\frac{K_r - 1}{K_r}\right)^j |c_r(j)|, \\ \mathcal{M}_{\max}(t) &= \sum_{r=0}^{R} \frac{F_{\max}(K_r; a_r, b_r)}{\sqrt{v_r}}. \end{aligned} \tag{25}$$

(The dependence of $c_{\max}(J, t)$ on t is through the coefficients $c_r(j)$.) By the triangle inequality then,

$$|\epsilon_J(t)| \le c_{\max}(J, t) \mathcal{M}_{\max}(t). \tag{26}$$

Now, recall that

$$c_r(j) = \left(\frac{K_r}{v_r}\right)^j \frac{f_{s^*}^{(j)}(0)}{j!}. \tag{27}$$

By construction $K_r = \lceil v_r/u_0 \rceil$, hence, $(K_r - 1)/v_r \le 1/u_0$. Also, if we write

$$\frac{f_{s^*}(z)}{j!} = e^{-\frac{s^* z^3}{3}} \frac{e^{\frac{s^* z^3}{3}} f_{s^*}(z)}{j!} = e^{-\frac{s^* z^3}{3}} \sum_{m=0}^{\infty} d_m z^m, \tag{28}$$

then

$$\left| \frac{f_{s^*}^{(j)}(0)}{j!} \right| \le \sum_{\substack{0 \le m, h \\ m + 3h = j}} \frac{|s|^h}{h! 3^h} |d_m|, \tag{29}$$

where, by Cauchy's theorem applied with a circle of radius $\rho > 0$ around the origin,

$$|d_m| \le \frac{1}{2\pi} \left| \int_{|z|=\rho} \frac{e^{\frac{s^* z^3}{3}} f_{s^*}(z)}{z^{m+1}} dz \right| \le \frac{e^{\frac{\rho}{2} + \frac{\rho^2}{4} + |s| \sum_{\ell=4}^{\infty} \frac{\rho^\ell}{\ell}}}{\rho^m}. \tag{30}$$

Choosing $\rho = 1/|s|^{1/4}$ therefore gives

$$|d_m| \le |s|^{m/4} e^{\lambda(s)}, \tag{31}$$

where

$$\lambda(s) \le \frac{1}{2|s|^{1/4}} + \frac{1}{4|s|^{1/2}} + \frac{1}{4(1 - |s|^{-1/4})}. \tag{32}$$

Combining (31), (29), (27), and observing that $d_0 = 1$ and $|d_1| + |d_2| + |d_3| = 49/48$ (since they come exclusively from expanding $e^{-\frac{1}{2}(z - z^2/2)}$) we obtain (on treating

the cases $m=0$, $1\le m\le 3$, and $4\le m$ separately) that

(33)
$$c_{\max}(J,t)\le \Big[e^{1/3\alpha^3}-\sum_{0\le h\le \frac{J}{3}}\frac{1}{h!}\frac{1}{(3\alpha^3)^h}\Big]+\frac{49}{48|s|^{1/3}\alpha}\Big[e^{1/3\alpha^3}-\sum_{0\le h\le \frac{J-1}{3}}\frac{1}{h!}\frac{1}{(3\alpha^3)^h}\Big]$$
$$+\frac{e^{\lambda(s)}}{|s|^{1/3}\alpha^4}\Big[e^{1/3\alpha^3}-\sum_{0\le h\le \frac{J-4}{3}}\frac{1}{h!}\frac{1}{(3\alpha^3)^h}\Big]+\frac{e^{\lambda(s)+1/3}}{|s|^{5/12}\alpha^{J-1}(\alpha-1)}.$$

The proposition now follows on substituting into (26) this estimate for $c_{\max}(J,t)$ and the bound for $\mathcal{M}_{\max}(t)$ from Lemma 9.1. □

Lemma 9.1. *Let $\alpha=1/0.9$, $K_{\min}=2000$. If $v_0=\lceil K_{\min}t^{1/3}\rceil\le \frac{1}{3}\sqrt{t/(2\pi)}$ and $u_0=\alpha|s|^{1/3}\ge 1$, then*

(34)
$$\mathcal{M}_{\max}(t)\le\sqrt{\frac{1}{u_0}+\frac{1}{v_0}}\left(\frac{\log\frac{t}{2\pi v_0^2}}{\log(1+\frac{1}{u_0})}+\frac{t^{1/4}}{3\pi^{1/4}}\frac{\sqrt{2}}{\sqrt{u_0}}+\frac{2t}{3\pi}\frac{\sqrt{2}}{\sqrt{u_0v_0}v_0}\right.$$
$$\left.+\frac{\sqrt{t}}{\sqrt{\pi}}\frac{1}{v_0}+2u_0+2u_0\log\frac{t}{\pi v_0^2}+5\right).$$

Proof. Recall that

$$\mathcal{M}_{\max}(t)=\sum_{r=0}^{R}\frac{F_{\max}(K_r;a_r,b_r)}{\sqrt{v_r}}. \tag{35}$$

To bound $F_{\max}(K_r;a_r,b_r)$, we use the Weyl-van der Corput Lemma in [4, Lemma 5]. This lemma gives rise to a certain geometric sum, which is in turn bounded using the Kusmin–Landau Lemma in [4, Lemma 2]. This gives for each positive integer M,

$$F_{\max}(K_r;a_r,b_r)^2\le(K+M)\left(\frac{K_r}{M}+\min(1/\pi\|2b_r\|+1,K_r)\right), \tag{36}$$

where $\|x\|$ is the distance to the nearest integer to x. Choosing $M=K_r$ yields

$$F_{\max}(K_r;a_r,b_r)\le\sqrt{4K_r+2K_r\min(1/\pi\|2b_r\|,K_r)}. \tag{37}$$

We partition $\{b_r\}_{r=0}^R$ into subsets $I_\ell=\{b_r:2|b_r|\in[\ell-1/2,\ell+1/2),\,0\le r\le R\}$. So $\mathcal{M}_{\max}=\sum_\ell\mathcal{M}_\ell$ where

$$\mathcal{M}_\ell=\sum_{b_r\in I_\ell}\sqrt{K_r/v_r}\sqrt{4+2\min(1/\pi\|2b_r\|,K_r)}. \tag{38}$$

Let K_ℓ^* denote the maximum block length K_r associated with a subset I_ℓ. Analogously define v_ℓ^* to be the maximum such v_r. Then appealing to the inequality

$$K_r/v_r\le\sqrt{1/u_0+1/v_0}, \tag{39}$$

and the bound $\sqrt{x+y}\le\sqrt{x}+\sqrt{y}$ (valid for $x,y\ge0$), we obtain

$$\mathcal{M}_\ell\le\sqrt{1/u_0+1/v_0}\left(2|I_\ell|+\sqrt{2/\pi}M_\ell^*\right) \tag{40}$$

where $|I_\ell|$ is the cardinality of I_ℓ and

$$M_\ell^*=\sum_{b_r\in I_\ell}\sqrt{\min(1/\|2b_r\|,\pi K_\ell^*)}. \tag{41}$$

We bound $\mathcal{M}_\ell^*$ in terms of the minimum distance between distinct points $b_r, b_{r'} \in I_\ell$,

$$\delta_\ell = \min_{\substack{b_r, b_{r'} \in I_\ell \\ b_r \neq b_{r'}}} |2b_r - 2b_{r'}|. \tag{42}$$

To this end, proceed similarly to the proof of [11, Lemma 3.1] (starting with Equation (25) there and using the monotonicity of the b_r) to obtain

$$\mathcal{M}_\ell^* \leq \sum_{0 \leq w \leq 1/2\delta_\ell} \sqrt{2\min\left(1/w\delta_\ell, K_\ell^*\right)} \leq \sqrt{2K_\ell^*} + 2/\delta_\ell. \tag{43}$$

The last inequality follows on isolating the term in the sum corresponding to $w = 0$, fixing the min to be $1/w\delta_\ell$ in the remainder of the sum, and estimating that by an integral.

At this point, we observe that if $\ell < t/2\pi v_R^2 - 1/2$ or $\ell > t/2\pi v_0^2 + 1/2$, then I_ℓ is empty. In view of this, and since $v_R \leq \sqrt{t/2\pi}$, we may restrict the range of summation in (38) to $1 \leq \ell \leq \ell^* = t/2\pi v_0^2 + 1/2$. So, substituting (43) into (40), summing over ℓ, and using the obvious formula $\sum_\ell |I_\ell| = R + 1$, we obtain

$$\mathcal{M}_{\max}(t) \leq \sqrt{1/u_0 + 1/v_0} \left(2R + 2 + \sum_{1 \leq \ell \leq \ell^*} \sqrt{2K_\ell^*} + \sum_{1 \leq \ell \leq \ell^*} 2/\delta_\ell \right). \tag{44}$$

Now, a simple calculation shows that

$$\frac{t}{\pi(2\ell+1)} \leq v_r^2 \leq \frac{t}{\pi(2\ell-1)}, \qquad (b_r \in I_\ell). \tag{45}$$

Hence,

$$v_\ell^* \leq \frac{\sqrt{t}}{\sqrt{\pi(2\ell-1)}}, \qquad \text{and so} \qquad K_\ell^* \leq \frac{\sqrt{t}}{\sqrt{\pi(2\ell-1)}} \frac{1}{u_0} + 1. \tag{46}$$

So, using the inequality $\sqrt{x+y} \leq \sqrt{x} + \sqrt{y}$ once again, we obtain

$$\sum_{1 \leq \ell \leq \ell^*} \sqrt{2K_\ell^*} \leq \frac{t^{1/4}}{\pi^{1/4}} \frac{\sqrt{2}}{\sqrt{u_0}} \left(\frac{1}{3} + \frac{2}{3} \frac{t^{3/4}}{\pi^{3/4}} \frac{1}{v_0^{3/2}} \right) + \frac{\sqrt{t}}{\sqrt{\pi} v_0} + 1. \tag{47}$$

Here, we additionally estimated $\sum_{1 \leq \ell \leq \ell^*} 1/(2\ell-1)^{1/4} \leq 1/3 + (2/3)(2\ell^* - 1)^{3/4}$, which follows on isolating the term corresponding to $\ell = 1$ and bounding the rest by an integral. Furthermore, since the sequence $\{b_r\}_{r=0}^R$ is monotonically increasing), then

$$\delta_\ell = \min_{b_r, b_{r+1} \in I_\ell} (2b_{r+1} - 2b_r) = \min_{b_r, b_{r+1} \in I_\ell} \frac{t}{2\pi} \frac{K_r(v_{r+1} + v_r)}{v_r^2 v_{r+1}^2}. \tag{48}$$

Thus, using the inequalities $K_r/v_r \geq 1/u_0$ and $v_{r+1} \leq v_\ell^*$, we arrive at the lower bound $\delta_\ell \geq (2\ell - 1)/u_0$. Consequently, as $\sum_{1 \leq \ell \leq \ell^*} 1/(2\ell-1) \leq 1 + \frac{1}{2}\log(2\ell^* - 1)$, we obtain

$$\sum_{1 \leq \ell \leq \ell^*} 2/\delta_\ell \leq 2u_0 + u_0 \log(t/\pi v_0^2). \tag{49}$$

Last, a routine application of induction (see [12, Lemma 3.1]) gives

$$R \leq \frac{\log(\sqrt{t/2\pi}/v_0)}{\log(1 + 1/u_0)} + 1. \tag{50}$$

The claim follows on substituting (50), (49), and (47) into (44). □

References

1. M. V. Berry, *The Riemann-Siegel expansion for the zeta function: high orders and remainders*, Proc. Roy. Soc. London Ser. A **450** (1995), no. 1939, 439–462. MR 1349513 (96f:11105)
2. Andriy Bondarenko and Kristian Seip, *Large GCD sums and extreme values of the Riemann zeta function*, arXiv:1507.05840 (2015).
3. Vorrapan Chandee and K. Soundararajan, *Bounding $|\zeta(\frac{1}{2}+it)|$ on the Riemann hypothesis*, Bull. Lond. Math. Soc. **43** (2011), no. 2, 243–250. MR 2781205
4. Yuanyou F. Cheng and Sidney W. Graham, *Explicit estimates for the Riemann zeta function*, Rocky Mountain J. Math. **34** (2004), no. 4, 1261–1280. MR 2095256 (2005f:11179)
5. Harold Davenport, *Multiplicative number theory*, third ed., Graduate Texts in Mathematics, vol. 74, Springer-Verlag, New York, 2000, Revised and with a preface by Hugh L. Montgomery. MR 1790423 (2001f:11001)
6. H. M. Edwards, *Riemann's zeta function*, Dover Publications, Inc., Mineola, NY, 2001, Reprint of the 1974 original [Academic Press, New York; MR0466039 (57 #5922)]. MR 1854455 (2002g:11129)
7. W. Gabcke, *Neue herleitung und explicite restabschätzung der Riemann-Siegel-formel.*, Ph.D. thesis, Göttingen, 1979.
8. X. Gourdon, *The 10^{13} first zeros of the Riemann zeta function and zero computation at very large heights*, Online manuscript (2004).
9. Ghaith A. Hiary, *Fast methods to compute the Riemann zeta function*, Ann. of Math. (2) **174** (2011), no. 2, 891–946. MR 2831110 (2012g:11154)
10. ———, *A nearly-optimal method to compute the truncated theta function, its derivatives, and integrals*, Ann. of Math. (2) **174** (2011), no. 2, 859–889. MR 2831109 (2012g:11215)
11. ———, *An explicit hybrid estimate for $L(1/2+it,\chi)$*, arXiv:1510.00950 (2015).
12. ———, *An alternative to Riemann-Siegel type formulas*, Math. Comp. **85** (2016), no. 298, 1017–1032. MR 3434892
13. ———, *An explicit van der Corput estimate for $\zeta(1/2+it)$*, Indag. Math. (N.S.) **27** (2016), no. 2, 524–533. MR 3479170
14. A. Kuznetsov, *Computing the truncated theta function via Mordell integral*, Math. Comp. **84** (2015), no. 296, 2911–2926. MR 3378853
15. A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász, *Factoring polynomials with rational coefficients*, Math. Ann. **261** (1982), no. 4, 515–534. MR 682664
16. Hugh L. Montgomery, *Extreme values of the Riemann zeta function*, Comment. Math. Helv. **52** (1977), no. 4, 511–518. MR 0460255
17. A. M. Odlyzko, *The 10^{20}-th zero of the Riemann zeta function and 175 million of its neighbors*, Unpublished manuscript, 1992.
18. A. M. Odlyzko and A. Schönhage, *Fast algorithms for multiple evaluations of the Riemann zeta function*, Trans. Amer. Math. Soc. **309** (1988), no. 2, 797–809. MR 961614 (89j:11083)
19. Michael Rubinstein, *Computational methods and experiments in analytic number theory*, Recent perspectives in random matrix theory and number theory, London Math. Soc. Lecture Note Ser., vol. 322, Cambridge Univ. Press, Cambridge, 2005, pp. 425–506. MR 2166470 (2006d:11153)
20. A. Schönhage, *Numerik analytischer Funktionen und Komplexität*, Jahresber. Deutsch. Math.-Verein. **92** (1990), no. 1, 1–20. MR 1037441
21. E. C. Titchmarsh, *The theory of the Riemann zeta-function*, second ed., The Clarendon Press Oxford University Press, New York, 1986, Edited and with a preface by D. R. Heath-Brown. MR 882550 (88c:11049)
22. A. M. Turing, *Some calculations of the Riemann zeta-function*, Proc. London Math. Soc. (3) **3** (1953), 99–117. MR 0055785

JB: Heilbronn Institute for Mathematical Research, School of Mathematics, University of Bristol, Howard House, Queens Avenue, Bristol BS8 1SN, United Kingdom

E-mail address: j.bober@bris.ac.uk

GH: Department of Mathematics, The Ohio State University, 231 West 18th Ave, Columbus, OH 43210, USA

E-mail address: hiary.1@osu.edu