Annales Univ. Sci. Budapest., Sect. Comp. 48 (2018) 53–64

IMPROVEMENTS ON FINDING LARGE VALUES OF THE RIEMANN ZETA FUNCTION ON THE CRITICAL LINE

Attila Kovács and Norbert Tihanyi

(Budapest, Hungary)

Communicated by Imre Kátai

(Received March 30, 2018; accepted June 14, 2018)

Abstract. In this paper we present some improvements of finding large values of the Riemann–Siegel function $Z(t)$. In order to analyse $Z(t)$ the authors developed a function $F(t)$ which shows in some aspect similar characteristics to $Z(t)$ but easier to compute.

1. Introduction

Many important functions exist in analytical number theory, including the various zeta functions, such as Epstein zeta, Dirichlet-L, Dedekind zeta, or the Hurwitz zeta function. One of the most important and most studied zeta function is the *Riemann zeta function*. It is defined by the sum of the infinite series

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}$$

Key words and phrases: Riemann zeta function, prime number theory.
2010 Mathematics Subject Classification: 11M26.
The project has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00001).

with the complex variable $s = \sigma + it$. $\zeta(s)$ is convergent if $Re(s) > 1$. By analytic continuation the function can be extended to the whole complex plane, except for a simple pole at $s = 1$. It satisfies the functional equation

$$\zeta(1-s) = 2(2\pi)^{-s} \cos\left(\tfrac{1}{2}s\pi\right) \Gamma(s)\zeta(s) .$$

Riemann conjectured that all non-trivial zeros of $\zeta(s)$ have real part $\sigma = 1/2$ (critical line). This is the famous Riemann hypothesis, one of the most important and unsolved problems in number theory. The Riemann zeta function describes the behaviour of the distribution of the prime numbers.

Many authors computed the zeroes of $\zeta(s)$ proving that the Riemann hypothesis holds up to some bound. In 1979 Brent showed [1] that the first 75 million non-trivial zeros lie on the critical line. In 1986 van de Lune et al. showed [2] that the first 1.5 billion non-trivial zeros also lie on the critical line. In 1985 the Mertens conjecture [3] was disproved by Odlyzko and te Riele based on extensive computations of the zeros of the zeta function. Their method is a striking example of a mathematical proof including a large amount of computational evidence.

1.1. The Riemann–Siegel Formula

For investigating $\zeta(s)$ on the critical line the usual way is to consider the formula

$$\zeta\left(\tfrac{1}{2} + it\right) = Z(t)e^{-i\theta(t)} \tag{1.1}$$

where $Z(t)$ is the *Riemann–Siegel function*. Clearly, we have $|\zeta(1/2 + it)| = |Z(t)|$. Equation (1.1) implies that $Z(t)$ and $\theta(t)$ are real for real t and can be used to investigate the behaviour of $\zeta(s)$. Analysing $Z(t)$ reduces the problem of finding zeros of the zeta function on the critical line. The Riemann–Siegel function has also a very deep connection to the Riemann hypothesis:

Theorem 1.1. *Suppose that there exist a real number t_0 with $Z(t_0) \neq 0$ such that $Z(t)$ had either a positive local minimum or a negative local maximum at $t = t_0$. Then RH is false* [4].

$Z(t)$ can be calculated in time complexity of $O(t^{1/2})$ by the Riemann–Siegel Formula ([1], [5])

$$Z(t) = 2 \sum_{n=1}^{\lfloor\sqrt{t/2\pi}\rfloor} \frac{1}{\sqrt{n}} \cos(\theta(t) - t \cdot \ln n) + O(t^{-1/4}) , \tag{1.2}$$

where $\theta(t)$ is the the *Riemann–Siegel theta* function and is defined in terms of the Gamma function for real values of t by

$$\theta(t) = \arg\left(\Gamma\left(\frac{2it+1}{4}\right)\right) - \frac{\ln\pi}{2}t \approx \frac{t}{2}\ln\frac{t}{2\pi} - \frac{t}{2} - \frac{\pi}{8} + \frac{1}{48t} + \frac{7}{5760t^3} + \cdots$$

Equation (1.2) was the most used tool for verification of the Riemann hypothesis till 1988, when Odlyzko and Schönhage introduced a more efficient algorithm for evaluating the Riemann zeta function at multiple points simultaneously. In 2004 the Odlyzko-Schönhage algorithm [6] was used by Gourdon in order to verify the Riemann hypothesis for the first 10^{13} zeros of the zeta function [7].

1.2. The importance of large $Z(t)$ values

Based on numerical verification and analysis of the behaviour of $\zeta(s)$ it is conjectured that if a counterexample of RH exist then it should be in the neighbourhood of unusually large peaks of $\zeta(1/2 + it)$ [1].

For the integers $m \geq 0$ the m^{th} Gram point g_m is defined by the unique solution of the equation

$$\theta(g_m) = m\pi\,.$$

Gram's law states that $Z(t)$ "usually" changes sign in the Gram intervals $G_j = [g_j, g_{j+1})$ for $j \geq 0$. A Gram point g_j is said to be "good" if $(-1)^j Z(g_j) > > 0$ and "bad" otherwise. A Gram block with length k is an interval $M_j = = [g_j, g_{j+k})$ such that g_j and g_{j+k} are good Gram points and $g_{j+1}, \ldots g_{j+k-1}$ are bad Gram points for $k \geq 1$. The interval M_j satisfies *Rosser's rule* if $Z(t)$ has at least k zeroes in M_j. Rosser's rule is violated infinitely often, but only for a small fraction of the Gram blocks [13].

In 1979 Brent computed the first 75 000 000 zeros of $\zeta(s)$ and observed an unusually large $Z(t)$ (> 79.6) near the 70354406^{th} Gram point [1]. Experiences showed that in all the cases, where an exception to Rosser's rule was observed, there was a large local peak of $Z(t)$ nearby.

Another reason why calculating peak values of $Z(t)$ is interesting: analysing them may help to discover and explain new interesting behaviour of the distribution of prime numbers.

At present (2018) one can calculate $Z(t)$ within $O(t^{1/3})$ time complexity applying the algorithm of Hiary [8] (published in 2011). Since $\zeta(1/2 + it)$ is unbounded $Z(t)$ can take arbitrarily large values as t goes to infinity, however, finding peak values of $Z(t)$ is computationally expensive even with modern computer technology.

In 2013 the authors published an algorithm for solving n-dimensional simultaneous Diophantine approximation problems efficiently [9]. Using this method and based on previous work of Kotnik [10] and Odlyzko [11] the `RS-Peak` algorithm [12] was presented in 2014. The `RS-Peak` algorithm can be used to find those t values efficiently where large $Z(t)$ are likely.

2. The RS-Peak algorithm

In this section we summarize the `RS-Peak` algorithm in a nutshell. For further details see the full specification and description of the algorithm in [12]. During a 4-year period many new records achieved applying the `RS-Peak` algorithm on the SZTAKI Desktop Grid [15] operated by the Laboratory of Parallel and Distributed Systems in the Institute for Computer Science and Control of the Hungarian Academy of Sciences. The largest known $Z(t)$ and other new achievements were published in 2017 [14].

The `RS-Peak` algorithm has three main parts:

- Part I – Fast Diophantine approximation;
- Part II – Filtering with a special function;
- Part III – Summand filtering.

The second and third part eliminate weak candidates by sieving.

2.1. Part I – Fast Diophantine approximation

The first part is about generating the candidates t where large $Z(t)$ is likely by solving simultaneous Diophantine approximations. Consider a set of irrationals $\Upsilon = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$, let $\varepsilon > 0$ and let us define the set

$$\Lambda(\Upsilon, \varepsilon) = \{k \in \mathbb{N} : \|k\alpha_i\| < varepsilon \text{ for all } \alpha_i \in \Upsilon\} \tag{2.1}$$

where $\|\cdot\|$ denotes the nearest integer distance function. Finding *many* appropriate $k \in \Lambda(\Upsilon, \varepsilon)$ is a *simultaneous Diophantine approximation problem.* One of the most efficient algorithms for solving such simultaneous Diophantine approximation problems is the Lenstra-Lenstra-Lovász (LLL) basis reduction algorithm [16]. One can use the LLL algorithm in order to find *one* appropriate k satisfying (2.1).

In 2013 the authors presented a method for solving n-dimensional simultaneous Diophantine approximation problems efficiently [12]. The method is based on the following theorem:

Theorem 2.1. *Let* $\Upsilon = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ *be a set of irrationals and* $\varepsilon > 0$. *Then there is a set* Γ_n *with* 2^n *elements with the following property: if* $k \in \Lambda(\Upsilon, \varepsilon)$ *then* $(k + \gamma) \in \Lambda(\Upsilon, \varepsilon)$ *for some* $\gamma \in \Gamma_n$.

The generation of Γ_n can be done efficiently and can be used to produce *many* $k \in \mathbb{N}$ much faster than with LLL in small dimensions ($n \leq 20$), see [12, 17]. Based on this result the authors introduced the *Multithreaded Advanced Fast Rational Approximation* algorithm (MAFRA) [17] for solving simultaneous Diophantine approximation problems. Then, the algorithm `RS-Peak` [12] was introduced, which is based on the following idea: one has to find an integer k such that the real numbers $k\frac{\ln p_i}{\ln 2}$ are all close to some integers for as many primes p_i as possible. In this case at the points $t = \frac{2k\pi}{\ln 2}$ the approximation

$$\cos\big(\theta(t) - t \ln n\big) \approx \cos\big(\theta(t)\big)$$

holds. For this particular Diophantine approximation problem LLL is substituted by MAFRA achieving significant improvement in locating appropriate integers k.

2.2. Part II – Filtering with a special function

Investigating many large values of $Z(t)$ the authors introduced the function

$$F(t) = \sum_{n=1}^{\lfloor \ln (t/2\pi) \rfloor} \frac{1}{\sqrt{n}} \cos(\theta(t) - t \cdot \ln n)\,. \tag{2.2}$$

We expect large $Z(t)$ where $F(t)$ is large due to the fact that $\frac{1}{\sqrt{n}}$ is dominant in the beginning of the summand and less relevant for larger n. Until now the function $F(t)$ was not thoroughly analysed. In the last few years it has turned out that it is much more powerful.

2.3. Part III – Summand filtering

Part III is the last filtering part of the `RS-Peak` algorithm. Here the

$$A(t, B_1, B_2) = \sum_{n=B_1}^{B_2} \frac{1}{\sqrt{n}} \cos(\theta(t) - t \cdot \ln n) \tag{2.3}$$

function was used with appropriate B_1 and B_2 parameters.

The main focus of this paper is to strengthen the filtering part II of the `RS-Peak` algorithm and to analyse the behaviour of $F(t)$ function, which is, in some sense, very similar to the original Riemann–Siegel function.

3. Numerical investigation of $F(t)$

The importance and usefulness of studying $F(t)$ has been already observed in [12]: "Where peak values of $F(t)$ occur we can expect in a high proportion of the cases that there are peak values of $Z(t)$ as well. $F(t)$ can be used for a given t to presume the order of $Z(t)$. Of course, there are cases where the behaviour of $F(t)$ and $Z(t)$ is different, but in general, the methods can be used very effectively to eliminate unlikely candidates."

3.1. Computing $F(t)$

Calculating $Z(t)$ is a very expensive task with the original Riemann–Siegel formula, even with the $O(t^{1/3})$ time complexity algorithm of Hiary. Note that the complexity of $F(t)$ is only $O(\ln t)$, which is significantly better. On a modern computer architecture $F(t)$ can be calculated in less than a second even for $t \approx 10^{1000}$ and can be used to calculate $F(t)$ for large t values. Calculating $F(t)$ gives us a lot of useful information about the behaviour of $Z(t)$, such as the expected growth of $Z(t)$ or the magnitude of $Z(t)$.

For testing purposes we used a Supermicro server equipped with 2 Intel Xeon Processor E5-2650 v4 CPU. Each CPU has 12 physical cores. With hyperthreading we have 48 threads for testing purposes. For calculating $Z(t)$ the $O(t^{1/3})$ time complexity algorithm of Hiary was used[1]. $F(t)$ is very simple, so we implemented it in the PARI/GP computer algebra system. Table 1 shows the difference between the calculation speed of $Z(t)$ and $F(t)$, denoted by $\Omega_{Z(t)}$ and $\Omega_{F(t)}$, respectively.

#	t	$Z(t)$	$F(t)$	$\Omega_{Z(t)}$	$\Omega_{F(t)}$
1	69903941711014013853520029.49	3794.501	13.382	$5434s$	$< 1ms$
2	132209240212483009855439237 3.32	−5012.013	−13.841	$17478s$	$< 1ms$
3	5964500070917012502334744833.72	−4619.42	−14.007	$31753s$	$< 1ms$
4	7214695626747977979984985146.68	6089.99	14.007	$34194s$	$< 1ms$
5	31616488911549318255796390329.65	−7135.605	−13.467	$60561s$	$< 1ms$
6	10^{100}	N/A	0.059	N/A	$3ms$
7	10^{340}	N/A	1.720	N/A	$31ms$
8	10^{1000}	N/A	0.07	N/A	$824ms$

Table 1: Computation time of $Z(t)$ and $F(t)$ for some values

One can see that the calculation speed of $F(t)$ is fast enough even for large values of t. One can compute $F(t)$ for $t = 10^{340}$ easily, which may show

[1]The algorithm can be downloaded from Github https://github.com/jwbober/zetacalc

interesting behaviour of the Riemann–Siegel function. Calculating $Z(t)$ for $t > 10^{40}$ is beyond the current computational capacity.

Let us consider the following simple maximum searching algorithm:

Algorithm 1 `FindLargest`(t, N, Δ)

$t_{max}, F_{max} \leftarrow 0$
while $t < t + N$ **do**
  $f \leftarrow$ `abs`$(F(t))$
  if $f > F_{max}$ **then**
    $t_{max} \leftarrow t$
    $F_{max} \leftarrow f$
  end if
  $t \leftarrow t + \Delta$
end while
return (t_{max}, F_{max})

One can observe that the most expensive computations in the main summand of $F(t)$ are calculating the square root and the natural logarithm function many times. E.g., for the value $t = 699039417110140138535200000$ the `FindLargest`$(t, 1000, 0.1)$ algorithm calculates $\frac{1}{\sqrt{n}}$ and $\log n$ every time when $F(t)$ is called. In our particular case, $F(t)$ is invoked 10 000 times. However, in our experiments the values of t are $0 \leq t \leq 10^{40}$, and since $\lfloor \ln(10^{40}/2\pi) \rfloor \approx 90$ one can use a precomputed lookup table for storing the values $\sqrt{n}$ and $\log n$, respectively. Running `FindLargest`$(t, 1000, 0.1)$ without a precomputed table took approximately 3.5 seconds in our test environment. Using a precomputed table the same task took approximately 1.5 seconds. The difference is significant. Let us analyse the `FindLargest` algorithm applying different Δ stepsizes. Let the starting point $t = 699039417110140138535200000$, where we have

#	Δ	N	t_{max}	F_{max}
1	1	50	699039417110140138535200029	11.355
2	0.1	50	699039417110140138535200029.6	13.08
3	0.01	50	699039417110140138535200029.49	13.382

Table 2: The result of the `FindLargest` algorithm with different stepsizes

The algorithm finds the appropriate t_{max} values in the interval $[t, t + N]$, where the largest $Z(t)$ occur at $t = 699039417110140138535200029.49$ using $\Delta = 0.01$ stepsize. Figure 1 and 2 show $F(t)$ in the interval $[t, t + 50]$ with $\Delta = 1$ and $\Delta = 0.1$ stepsizes. The difference is striking. One can observe that $F(t)$ is more dense with $\Delta = 0.1$ than with $\Delta = 1$.

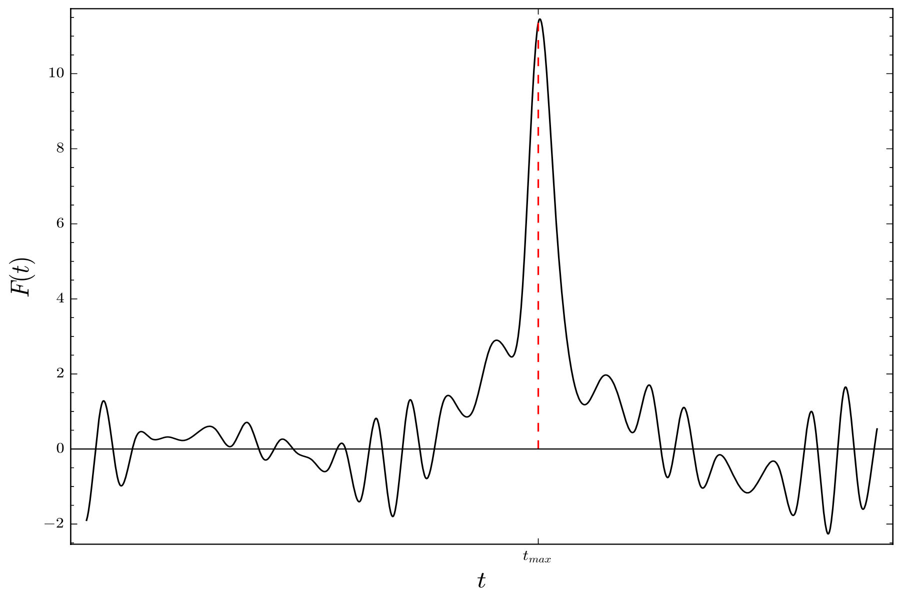


Figure 1: Plotting $F(t)$ with stepsize $\Delta = 1$

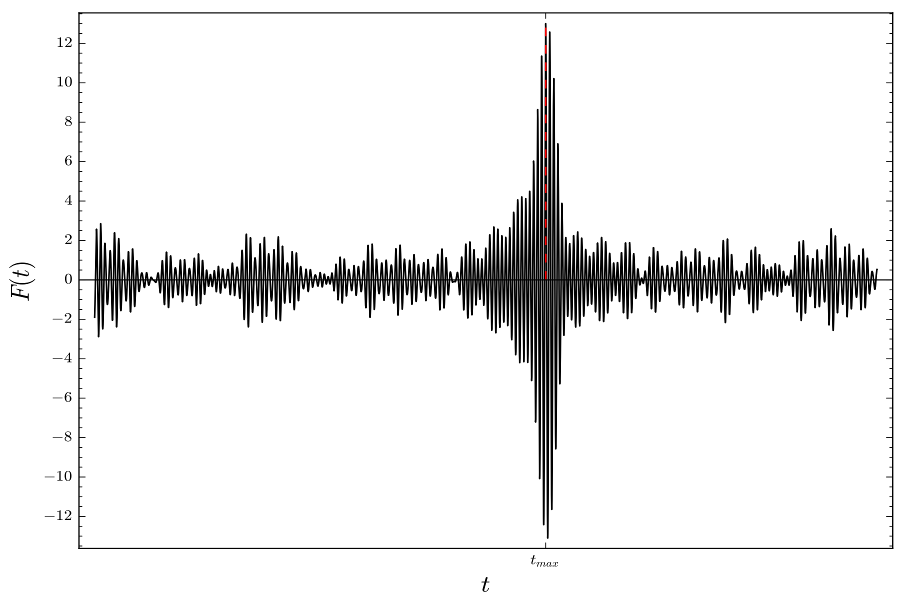


Figure 2: Plotting $F(t)$ with stepsize $\Delta = 0.1$

One can observe that the fine behaviour of $F(t)$ can only be seen with the smaller stepsize $\Delta = 0.1$. Computational experiments show that in the range of interest, the stepsize $\Delta = 1$ is sufficient for finding t_{max} with small deviation.

3.2. Deviation of F(t) and Z(t)

The previous sections suggest that in many cases $F(t)$ can be used effectively to indicate where large values of $|Z(t)|$ are likely. We analysed the `FindLargest` algorithm for many different t values with different N and Δ parameters. Let $Z_{max} = Z_{max}(t_0, N)$ denote such a value t where $|Z(t)|$ is the largest in the interval $[t_0, t_0 + N]$. Similarly, let $F_{max} = F_{max}(t_0, N)$ denote such a value t where $|F(t)|$ is the largest in the interval $[t_0, t_0 + N]$. In order to measure the strength of the $F(t)$ function we are interested in the deviation of $Z_{max}(t_0, N)$ and $F_{max}(t_0, N)$. Let us define the function σ as

$$\sigma = \sigma(t_0, N) = 100 * \frac{|Z(Z_{max}(t_0, N)) - Z(F_{max}(t_0, N))|}{Z(Z_{max}(t_0, N))}$$

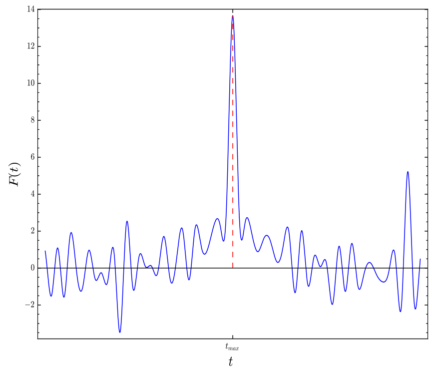
Table 3 displays the output of the `FindLargest`(t, N, Δ) algorithm (t values found by `RS-Peak`) for different parameters together with σ.

<table>
<tr><th colspan="6">$t_0 = 356071078353654500$</th></tr>
<tr><td>$Z(Z_{max})$</td><td>N</td><td>Δ</td><td>Z_{max}</td><td>F_{max}</td><td>σ</td></tr>
<tr><td>1287.14</td><td>100</td><td>0.01</td><td>$t_0 + 62.22$</td><td>$t_0 + 62.22$</td><td>0%</td></tr>
<tr><th colspan="6">$t_0 = 6578787583549202400$</th></tr>
<tr><td>$Z(Z_{max})$</td><td>N</td><td>Δ</td><td>Z_{max}</td><td>F_{max}</td><td>σ</td></tr>
<tr><td>−1368.459</td><td>100</td><td>0.01</td><td>$t_0 + 0.03$</td><td>$t_0 + 0.03$</td><td>0%</td></tr>
<tr><th colspan="6">$t_0 = 1322092402124830098554392000$</th></tr>
<tr><td>$Z(Z_{max})$</td><td>N</td><td>Δ</td><td>Z_{max}</td><td>F_{max}</td><td>σ</td></tr>
<tr><td>−5012.013</td><td>1000</td><td>0.01</td><td>$t_0 + 373.32$</td><td>$t_0 + 373.32$</td><td>0%</td></tr>
<tr><th colspan="6">$t_0 = 31616488911549318255796390000$</th></tr>
<tr><td>$Z(Z_{max})$</td><td>N</td><td>Δ</td><td>Z_{max}</td><td>F_{max}</td><td>σ</td></tr>
<tr><td>−7135.606</td><td>1000</td><td>0.01</td><td>$t_0 + 329.65$</td><td>$t_0 + 329.66$</td><td>0.46%</td></tr>
</table>

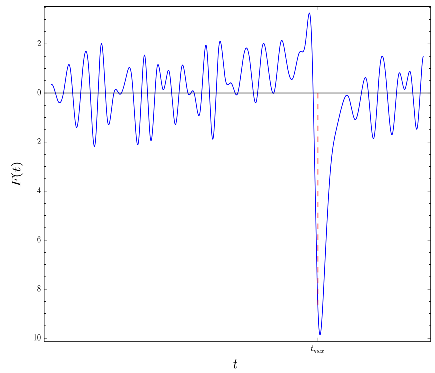
Table 3: Output of the `FindLargest`(t, N, Δ) algorithm for different t values

We also tested our $F(t)$ function on different values published by other authors. Figure 3 shows the neighbourhood of F_{max} for large values of t published by Hiary[2]. In each case the appropriate F_{max} values differ only with $\sigma < 0.5\%$.

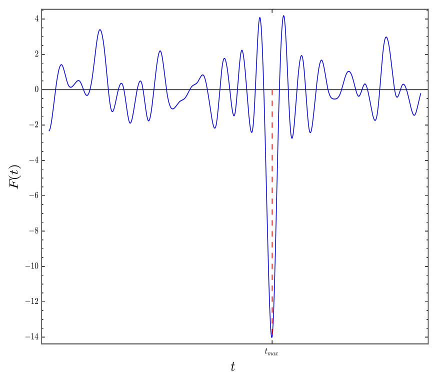
[2]https://people.math.osu.edu/hiary.1/



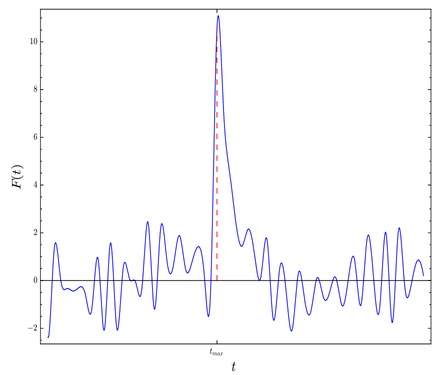
(a) $Z(t) \approx 16244.86, F(t) \approx 13.67$



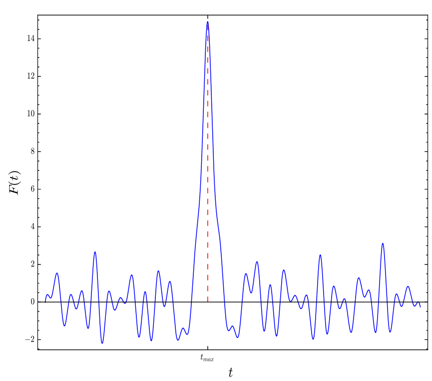
(b) $Z(t) \approx -14055.89, F(t) \approx -14.218$



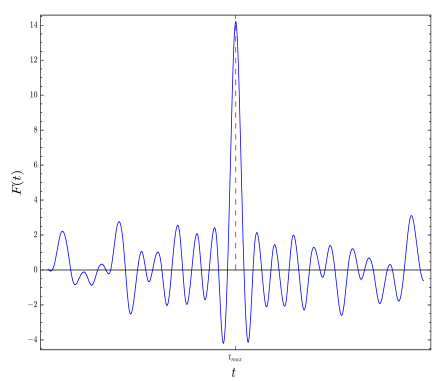
(c) $Z(t) \approx -13558.833, F(t) \approx -14.242$



(d) $Z(t) \approx 13338, F(t) \approx 11.430$



(e) $Z(t) \approx 12021.094, F(t) \approx 14.807$



(f) $Z(t) \approx 11196.79, F(t) \approx 14.074$

Figure 3: Output of the `FindLargest`$(t, 100, \Delta = 1)$ algorithm for t values published by Ghaith A. Hiary and Jonathan W. Bober

4. Further research

The aim of the `Riemann Zeta Search Project` is locating peak values of the $\zeta(s)$ function on the critical line in order to have a better understanding of the distribution of prime numbers. Applying `RS-Peak` algorithm many large values of $Z(t)$ have been found in the last 4 years. Thousands of new large $Z(t)$ values calculated and published on the `Riemann Zeta Search Project` website `https://www.Riemann-Siegel.com`.

The Lindelöf hypothesis is a conjecture about the rate of growth of the Riemann zeta function on the critical line and says that for any $\epsilon > 0$

$$\zeta\left(\tfrac{1}{2} + it\right) = O(t^{\epsilon}).$$

The calculation of millions of large $Z(t)$ values for $10^{10} \le t \le 10^{37}$ shows that as t goes to infinity the $\varphi(t) = \frac{\log|Z(t)|}{\log(t)}$ gets smaller and smaller. It seems that these values can be used to numerically support the Lindelöf hypothesis. We plan to continue our research in this direction.

Acknowledgment. The authors gratefully acknowledge the constructive comments of the referee. The authors would like to thank the opportunity for accessing to the ATLAS Super Cluster operating at Eötvös Loránd University and for using the capacity of SZTAKI Desktop Grid project operated by the Laboratory of Parallel and Distributed Systems in the Institute for Computer Science and Control of the Hungarian Academy of Sciences.

References

[1] **Brent, R.P.,** On the zeros of the Riemann zeta function in the critical strip, *Math. Comp.*, **33**, Number 148, (1979), 1361–1372.

[2] **van de Lune, J., H.J.J. te Riele and D.T. Winter,** On the zeros of the Riemann zeta function in the critical strip. IV, *Math. Comp.*, **46**, Number 174, (1986), 667–681.

[3] **Odlyzko, A.M. and H.J.J. te Riele**, Disproof of the Mertens conjecture, *Journal für die reine und angewandte Mathematik*, **357** (1985), 138–160.

[4] **Broughan, K.,** *Equivalents of the Riemann Hypothesis, Volume Two: Analytic Equivalents*, Cambridge University Press., (2017) ISBN:9781108178228, pp. 79, Theorem 5.13.

[5] **Lehman, R.S.,** Separation of zeros of the Riemann zeta-function, *Math. Comp.*, **20** (1966), 523–541.

[6] **Odlyzko, A.M. and A. Schönhage,** Fast algorithms for multiple evaluations of the Riemann zeta function, *Trans. Am. Math. Soc.*, **309**, (1988), 797–809.

[7] **Gourdon, X.,** The 10^{13}-rst zeros of the Riemann Zeta function, and zeros computation at very large height, `http://numbers.computation.free.fr/Constants/Miscellaneous/zetazeros1e13-1e24.pdf`, 2004.

[8] **Hiary, G.A.,** Fast methods to compute the Riemann zeta function, *Ann. Math.*, **174(2)**, (2011), 891–946.

[9] **Kovács, A. and N. Tihanyi,** Efficient computing of n-dimensional simultaneous Diophantie approximation problems, *Acta Univ. Sapientiae, Informatica*, **5(1)** (2013), 6–34.

[10] **Kotnik, T.,** Computational estimation of the order of $\zeta(1/2 + it)$, *Math. Comp.*, **73**, Number 246, (2004), 949–956.

[11] **Odlyzko, A.M.**, The 10^{20}-th zero of the Riemann zeta function and 175 million of its neighbors, `http://www.dtc.umn.edu/~odlyzko/unpublished/`, Accessed 15 April 2018,. (1992)

[12] **Tihanyi, N.,** Fast method for locating peak values of the Riemann zeta function on the critical line, *16th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing* (*SYNASC*), 2014, IEEE Explorer 10.1109/SYNASC.2014.20

[13] **Trudgian, T.,** On the success and failure of Gram's Law and the Rosser Rule, *Acta Arithmetica*, **148** (2011), 225–256.

[14] **Tihanyi, N., A. Kovács and J. Kovács,** Computing extremely large values of the Riemann zeta function, *J Grid Computing*, **15** (2017), 527. `https://doi.org/10.1007/s10723-017-9416-0`

[15] **Kacsuk, P. et al**, SZTAKI Desktop Grid (SZDG): A Flexible and Scalable Desktop Grid System, *J. Grid Computing*, **7** (2009), 439. `https://doi.org/10.1007/s10723-009-9139-y`

[16] **Lenstra, A.K., H.W. Lenstra Jr. and L. Lovász,** Factoring polynomials with rational coefficients, *Math. Ann.*, **261(4)** (1982), 515–534.

[17] **Tihanyi, N., A. Kovács and Á. Szűcs**, Distributed computing of simultaneous Diophantine approximation problems, *Stud. Univ. Babes-Bolyai Math.*, **59(4)** (2014), 557–566.

A. Kovács and N. Tihanyi
Eötvös Loránd University
Budapest
Hungary
`attila.kovacs@inf.elte.hu`
`norbert.tihanyi@inf.elte.hu`