Precalculated arrays-based algorithms for the calculation of the Riemann zeta-function

Lukas Kuzma, Igoris Belovas, Martynas Sabaliauskas

Vilnius University, Institute of Data Science and Digital Technologies
Akademijos st. 4, Vilnius LT-08412, Lithuania
lukas.kuzma@mif.vu.lt, igoris.belovas@mif.vu.lt, martynas.sabaliauskas@mif.vu.lt

Abstract. In this paper, we continue the study of efficient algorithms for the computation of the Riemann zeta function on the complex plane. We introduce two precalculated arrays-based modifications of MB-method. We perform numerical experiments with these algorithms using Zetafast as a benchmark and apply the algorithms for the visualizations of fractal structures associated with the Riemann zeta function.

Keywords: Riemann zeta function, numerical algorithms, fractal structures.

1 Introduction

The paper continues the studies of efficient algorithms for the computation of the Riemann zeta function on the complex plane (see [1, 2] and Lukas Kuzma's Bachelor thesis [5]). We introduce modifications of *MB*-algorithm, aiming to accelerate computations by precalculating arrays of coefficients of a series for the Riemann zeta function and combining this approach with normal (Gaussian) approximations of the coefficients. The accuracy and the processing time of the algorithms are evaluated using *Zetafast* [4] as a benchmark.

DOI: https://doi.org/10.15388/LMITT.2022.5

Throughout this paper, we denote by $\Phi(x)$ the cumulative distribution function of the standard normal distribution, and by $\overline{\Phi}(x)$ we denote the corresponding tail distribution $\overline{\Phi}(x) = 1 - \Phi(x)$. $A \times B$ stands for the Cartesian product of sets A and B. $\lfloor x \rfloor$ and $\lceil x \rceil$ stand for the floor function and the ceiling functions respectively. All limits in the paper, unless specified, are taken as $n \to \infty$.

2 *MB*-algorithm

In [3] Borwein proposed an algorithm, applied to complex numbers $s = \sigma + it$ with $\sigma \geqslant 1/2$ and arbitrary t. Suppose

$$d_{n,k} = n \sum_{j=0}^{k} \frac{(n+j-1)!4^j}{(n-j)!(2j)!}. \tag{1}$$

then the Riemann zeta function is defined by the alternating series

$$\zeta(s) = \frac{1}{d_{n,n}(1-2^{1-s})} \sum_{k=0}^{n-1} \frac{(-1)^k (d_{n,n} - d_{n,k})}{(k+1)^s} + \gamma_n(s). \tag{2}$$

The ineligibility of the numbers $d_{n,k}$ (note that with large n factorials in the definition (1) become computationally expensive) has lead Belovas et al. [2] to introduction of a modified algorithm (*MB*-algorithm), avoiding the calculation of high factorials. Let us denote, along with Proposition 1 from [2],

$$\begin{aligned} l_{max} &= \text{arg} \max_{0 \leqslant k \leqslant n} \frac{(n+k-1)!4^k}{(n-k)!(2k)!}, \\ c_{n,k} &= 1 - \frac{H_k}{H_n}, \quad n \in \mathbb{N}, \quad 0 \leqslant k \leqslant n, \end{aligned} \tag{3}$$

here

$$\begin{aligned} H_l &= H_{l-1} + e^{T_l - T_{l_{max}} + (l - l_{max}) \log 4}, \ H_0 = e^{T_0 - T_{l_{max}} - l_{max} \log 4}, \\ T_l &= T_{l-1} + \log \frac{(n-l+1)(n+l-1)}{(2l-1)(2l)}, \ T_0 = -\log n, \ 1 \leqslant l \leqslant n. \end{aligned} \tag{4}$$

Under these notations the Riemann zeta function is

$$\zeta(s) = \frac{1}{1-2^{1-s}} \sum_{k=0}^{n-1} \frac{(-1)^k c_{n,k}}{(k+1)^s} + \gamma_n(s). \tag{5}$$

The algorithm is nearly optimal in the sense that there is no sequence of n-term exponential polynomials that converge to the Riemann zeta function much faster than of the algorithm (see Theorem 3.1 in [3]).

3 Error term

The error term $\gamma_n(s)$ in (5) is characterized by the following proposition [1].

Theorem 3.1. *Let $\sigma \geqslant 1/2$, $t \geqslant 0$, $\varepsilon > 0$, $A = (2\pi)^{-1} \log 2$,*

$$s_k = 1 + ik/A, \quad k \in \mathbb{N}_0. \tag{6}$$

and $|s - s_k| \geqslant \varepsilon$, then

(i) *the error term of the series* (5) *is*

$$|\gamma_n(s)| \leqslant G_n \frac{(\cosh \pi t)^{1/2}}{|1 - 2^{1-s}|}, \tag{7}$$

(ii) *the series* (5) *to compute the Riemann zeta-function with d decimal digits of accuracy, require a number of terms*

$$n = \left\lceil B_1 t + B_2 d + C_\varepsilon \right\rceil, \tag{8}$$

where

$$\begin{aligned} G_n &= \frac{2}{(3+\sqrt{8})^n}, \quad B_1 = \frac{\pi/2}{\log(3+\sqrt{8})}, \\ B_2 &= \frac{\log 10}{\log(3+\sqrt{8})}, \; C_\varepsilon = \frac{\log 2 - \log(1-2^{-\varepsilon})}{\log(3+\sqrt{8})}. \end{aligned} \tag{9}$$

The error terms are closely linked to the problem of selection of minimal number of terms in the series (5). Notably, under the conditions of Theorem 3.1, for $\varepsilon = 10^{-m}$, $m \in \mathbb{N}$, the series (5) to compute the Riemann zeta-function with d decimal digits of accuracy, require the number of terms (see Corollary 1 in [1])

$$n = \left\lceil B_1 t + B_2 (d + m) \right\rceil + 1. \tag{10}$$

4 *NA*-modification

We have shown that the coefficients of *MB*-series satisfy (see Proposition 2 in [1])

$$c_{n,k} = \overline{\Phi}\left(\frac{k-\mu_n}{\sigma_n}\right) + O\left(\frac{1}{\sqrt{n}}\right), \quad \mu_n = \frac{n}{\sqrt{2}}, \quad \sigma_n = \frac{\sqrt{n}}{2\sqrt[4]{2}}. \tag{11}$$

This result allows us to introduce the corresponding normal approximation (*NA*) of the coefficients and to refine the number of terms n in the series (5),

$$\hat{n} = \lceil \mu_n + z_d \sigma_n \rceil, \tag{12}$$

for n large enough. Here $z_d = \Phi^{-1}(1 - 10^{-d})$. This modification is employed in Algorithm 2 (see the next section).

5 *PA*-based and *NA*-based algorithms

MB-method is well suited for calculating multiple values of the Riemann zeta functions while t is fixed. However, specific values of the Riemann zeta function for arguments with distinct t require the recalculation of the coefficients $c_{n,k}$, since the number of terms in series (5) depends on t. The first way to solve the problem is to establish limit theorems for the coefficients and to replace the coefficients with the corresponding approximations (*NA*-based approach). Alternatively, we can use a set of precomputed arrays of the coefficients $c_{n,k}$.

Let us precompute 16 arrays $\{c_{n_p,k}\}$, for $n_p = 2^{p+2}$, $1 \leqslant p \leqslant 16$ and select the number of terms in the series (5) by the rule

$$\hat{n} = \min_{n_p \geqslant n} n_p. \tag{13}$$

We can use the rule (13) for $0 < t < t_{max}$, with $t_{max} = 294000 < 2^{18}/B_1$. Note that C++ allows 16 decimal digits precision, hence we assume $c_{n,k} = 0$ if $c_{n,k} < 5 \cdot 10^{-17}$. We denote by k_p the corresponding index of the last non-zero element of $\{c_{n_p,k}\}$ array (see Table 1). Note that $k_p/n_p \to 1/\sqrt{2}$ in accordance with (11).

Algorithm 1 outlines the precalculated arrays-based modification of *MB*-method. This approach is more suitable for the calculation of specific values of the Riemann zeta function. Numerical experiments with the algorithm are presented in the next section.

Table 1: The indices of the last non-zero element k_p of $\{c_{n_p,k}\}$ array, $n_p = 2^{p+2}$.

p	n_p	k_p	p	n_p	k_p
1	8	8	9	2048	1601
2	16	16	10	4096	3111
3	32	32	11	8192	6097
4	64	64	12	16384	12013
5	128	125	13	32768	23776
6	256	233	14	65536	47189
7	512	437	15	131072	93882
8	1024	831	16	262144	187043

Algorithm 1 This algorithm will return values of the Riemann zeta function obtained by the precalculated arrays-based modification of *MB*-method. $L_k = \log k$ stand for the precalculated logarithms, $n_p = 2^{p+2}$, for $1 \leqslant p \leqslant 16$ and $t \in (0, t_{max})$.

function Zeta.PA(σ, t : real numbers; *d*, *m* : natural numbers)
    $n \leftarrow ((\pi/2)t + (d+m)L_{10})/\log(3+\sqrt{8})\rceil + 1$
    if $n \leqslant n_1$ **then**
        $n \leftarrow n_1$
    else
        $p_{min} \leftarrow 1, p_{max} \leftarrow 16$
        while $p_{max} - p_{min} > 1$ **do**
            $p \leftarrow \left\lfloor (p_{min} + p_{max} + 1)/2 \right\rfloor$
            if $n \leqslant n_p$ **then**
                $p_{max} \leftarrow p$
            else
                $p_{min} \leftarrow p$
            end if
        end while
        $n \leftarrow n_{p_{max}}$
    end if
    $S \leftarrow 0, \quad p \leftarrow -1$
    for $k \in \{0..k_p\}$ **do**
        $p \leftarrow -p$
        $S \leftarrow S + pc_{n,k}\exp(-\sigma L_{k+1})(\cos(tL_{k+1}) - i\sin(tL_{k+1}))$
    end for
    $Zeta.PA \leftarrow S/(1 - 2\exp(-\sigma L_2)(\cos(tL_2) - i\sin(tL_2)))$
end function

Algorithm 2 This algorithm will return values of the Riemann zeta function obtained by combining *PA*- and *NA*-methods.

function Zeta.NA(σ, t : real numbers; *d, m* : natural numbers)
    if $t \leqslant 32768$ **then**
        $n \leftarrow ((\pi/2)t + (d+m)L_{10})/\log(3+\sqrt{8})\Big\rceil+1$
        if $n \leqslant n_1$ **then**
            $n \leftarrow n_1$
        else
            $p_{min} \leftarrow 1, p_{max} \leftarrow 13$
            while $p_{max} - p_{min} > 1$ **do**
                $p \leftarrow \Big\lfloor(p_{min} + p_{max} + 1)/2\Big\rfloor$
                if $n \leqslant n_p$ **then**
                    $p_{max} \leftarrow p$
                else
                    $p_{min} \leftarrow p$
                end if
            end while
            $n \leftarrow n_{p_{max}}$
        end if
        $S \leftarrow 0, \quad p \leftarrow -1$
        for $k \in \{0..k_p\}$ **do**
            $p \leftarrow -p$
            $S \leftarrow S + pc_{n,k}\exp(-\sigma L_{k+1})(\cos(tL_{k+1}) - i\sin(tL_{k+1}))$
        end for
        $Zeta.NA \leftarrow S/(1 - 2\exp(-\sigma L_2)(\cos(tL_2) - i\sin(tL_2)))$
    else
        $n \leftarrow ((\pi/2)t + (d+m)L_{10} + L_2 - \log L_2)/\log(3+\sqrt{8})$
        $\mu_n \leftarrow n/\sqrt{2}, \quad \sigma_n \leftarrow \sqrt{n}/\sqrt[4]{32}, \quad z \leftarrow \Phi^{-1}(1 - 10^{-d})$
        $k_0 \leftarrow \lceil \mu_n + z\sigma_n \rceil, \quad k_1 \leftarrow \mu_n - z\sigma_n$
        function c(*n,k* : nonnegative integers)
            if $k < k_1$ **then**
                $C \leftarrow 1$
            else
                $C \leftarrow 1 - \Phi((k - \mu_n)/\sigma_n)$
            end if
        end function
        $S \leftarrow 0, \quad p \leftarrow -1$
        for $k \in \{0..k_0\}$ **do**
            $p \leftarrow -p$
            $S \leftarrow S + pC(n,k)\exp(-\sigma L_{k+1})(\cos(tL_{k+1}) - i\sin(tL_{k+1}))$
        end for
        $Zeta.NA \leftarrow S/(1 - 2\exp(-\sigma L_2)(\cos(tL_2) - i\sin(tL_2)))$
    end if
end function

Algorithm 2 combines precalculated arrays-based modification of *MB*-method (for $t \leqslant n_{\hat{p}}$) with *NA*-method (for $t > n_{\hat{p}}$). Numerical experiments have shown that $\hat{p} = 13$ is optimal.

6 Numerical experiments

Using Algorithm 1, Algorithm 2 and *Zetafast* method [4], we generate sequences of values of the Riemann zeta function $\{\zeta_{l,r}^{(PA)}\}$, $\{\zeta_{l,r}^{(NA)}\}$ and $\{\zeta_{l,r}^{(ZF)}\}$, $1 \leqslant l \leqslant N$, $1 \leqslant r \leqslant M$, $N = 10^5$, $M = 10^1$, taking as arguments uniformly distributed $z_{l,r} \in S_r \setminus \Theta$. Here (cf. Theorem 3.1)

$$\begin{gathered} S_r = \underbrace{(0.5, 2)}_{\Re z} \times \underbrace{(t_{r-1}, t_r)}_{\Im z}, \quad \Theta = \bigcup_{k=0}^{k_{max}} \{z : \ |z - s_k| \leqslant \rho\}, \\ t_r = t_0 + r\Delta_r, \quad t_0 = 0, \quad \Delta_r = t_{max}/M, \quad 1 \leqslant r \leqslant M, \end{gathered} \tag{14}$$

with $k_{max} = \lfloor A \ t_{max} \rfloor = 32433$ and $\rho = 10^{-1}$. We take decimal digits of accuracy $d = 6$ and $m = 1$ (see (10)).

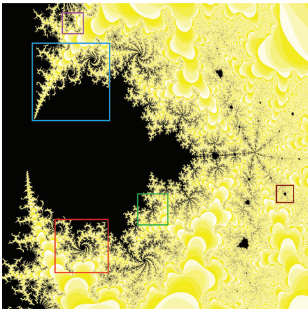
Remark 6.1. Let $t = \Im z_l$ and $k = \left\lfloor A(t + \rho) \right\rfloor$. Then $z_l \in \Theta$ if

$$k > \left\lfloor A(t - \rho) \right\rfloor \text{ and } |z_l - s_k| \leqslant \rho. \tag{15}$$

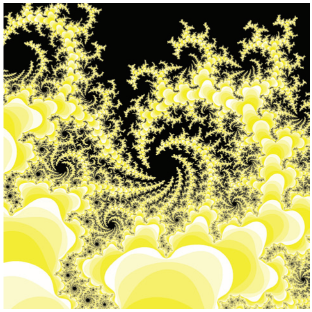
Proof. If $z_l \in \Theta$, then (cf. (6)) $\exists k \in \mathbb{N}_0$: $|t - k/A| \leqslant \rho$, or $k \in [A(t-\rho), A(t+\rho)]$. Note that the length of the interval is $2\rho A < 0.0221$. Such k exists if $\left\lfloor A(t - \rho) \right\rfloor < \left\lfloor A(t + \rho) \right\rfloor$. The proposition follows. □

7 Visualization

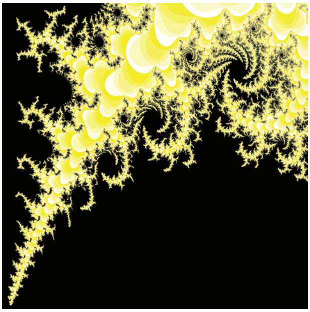
In this study we follow the guidelines of *SFH*-approach [1] visualizing fractal structures associated with the near-pole region of the Riemann zeta function (the function is calculated with $d = 6$ decimal digits of accuracy), see Figure 1. The ranges of the sets $(\sigma_1, \sigma_2) \times (t_1, t_2)$ and computation times of the figures are given in Table 2. All the frames are of 2000×2000 pixels size and have been generated using Python 3.10.2 version with AMD Ryzen 9 5950X processor (32 GB RAM, 4266 MHz).



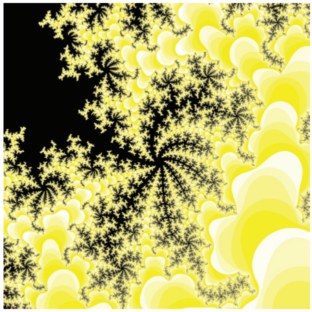
(a) main area



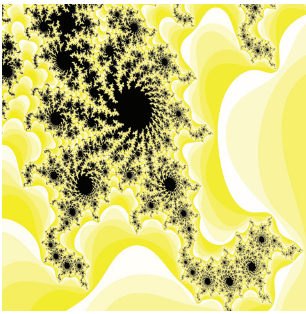
(b) area enclosed by the red square



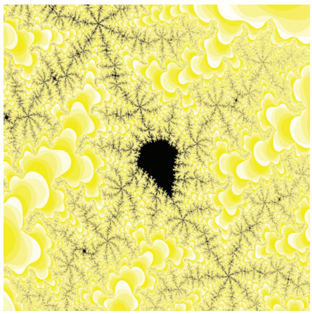
(c) area enclosed by the blue square



(d) area enclosed by the green square



(e) area enclosed by the violet square



(f) area enclosed by the brown square

Figure 1: Fractal structures associated with the near-pole region of the Riemann zeta function. Frames (b)-(f) are zoomed-in rectangles of (a). The ranges are given in Table 2.

Table 2: Ranges of the frames and corresponding computation times of Figure 1

Figure	σ_1	σ_2	t_1	t_2	Time (s)
1a	1.03000	1.04000	-0.03400	-0.02400	216
1b	1.03174	1.03347	-0.03279	-0.03106	230
1c	1.03100	1.03352	-0.02782	-0.02530	214
1d	1.03443	1.03543	-0.03123	-0.03023	222
1e	1.03197	1.03266	-0.02499	-0.02430	233
1f	1.03893	1.03953	-0.03051	-0.02995	228

8 Results

The numerical experiment has been performed on Intel® Core™ i7-8750H 2.2GHz (boosted to 4.0 GHz) processor, 16GB DDR4 RAM. The code was compiled using g++ 11.2.0 compiler using O3 optimization. Using *Zetafast* algorithm as a benchmark we calculate the accuracies

$$\begin{aligned} \delta_r^{(PA)} &= \max_{1\leqslant l\leqslant N} \left|\zeta_{l,r}^{(PA)} - \zeta_{l,r}^{(ZF)}\right|, \\ \delta_r^{(NA)} &= \max_{1\leqslant l\leqslant N} \left|\zeta_{l,r}^{(NA)} - \zeta_{l,r}^{(ZF)}\right|. \end{aligned} \tag{16}$$

and the processing times $\tau_r^{(\cdot)}$ of the sequences $\{\zeta_{l,r}^{(\cdot)}\}$ (see Table 3).

Table 3: Results of numerical experiments: accuracies $\delta_r^{(\cdot)}$ and processing times $\tau_r^{(\cdot)}$.

	Accuracies		Processing times (s)		
r	$\delta_r^{(PA)}$	$\delta_r^{(NA)}$	$\tau_r^{(PA)}$	$\tau_r^{(NA)}$	$\tau_r^{(ZF)}$
1	$5.09 \cdot 10^{-10}$	$5.09 \cdot 10^{-10}$	100.46	100.50	281.40
2	$2.99 \cdot 10^{-10}$	$2.80 \cdot 10^{-10}$	296.32	245.08	517.86
3	$8.15 \cdot 10^{-10}$	$7.58 \cdot 10^{-10}$	509.00	397.09	671.42
4	$8.29 \cdot 10^{-10}$	$7.42 \cdot 10^{-10}$	676.47	546.61	795.79
5	$1.46 \cdot 10^{-9}$	$1.43 \cdot 10^{-9}$	676.41	695.87	902.85
6	$1.89 \cdot 10^{-9}$	$1.72 \cdot 10^{-9}$	1335.08	866.20	998.96
7	$1.90 \cdot 10^{-9}$	$2.11 \cdot 10^{-9}$	1348.22	1017.30	1086.06
8	$2.09 \cdot 10^{-9}$	$2.10 \cdot 10^{-9}$	1348.04	1167.30	1171.45
9	$4.62 \cdot 10^{-9}$	$4.60 \cdot 10^{-9}$	1347.84	1317.71	1242.84
10	$3.93 \cdot 10^{-9}$	$3.85 \cdot 10^{-9}$	1348.16	1467.91	1314.32

9 Conclusions

The results show that both Algorithm 1 and Algorithm 2 return adequate, precise values of the Riemann zeta function. We see that the processing times of *PA*-algorithm are distributed in the pattern of a step function (in full compliance with the definition of Algorithm 1), while the processing times of *NA*-algorithm demonstrate a linear growth (in complete accordance with the proposition $n = O(t)$). Despite the fact that *PA*-algorithm is faster when t nears n_p, practical considerations favor the choice of *NA*-approach, especially if the range of the argument t is unrestricted. Indeed, the processing time of 10^5 Riemann zeta function values with arguments uniformly distributed in $\left(\bigcup_{r=1}^{M} S_r\right) \setminus \Theta$ is almost identical for Algorithm 1 and *Zetafast*, while Algorithm 2 is 15% faster, namely

$$\tau^{(PA)} = 904.32s, \quad \tau^{(NA)} = 785.81s, \quad \tau^{(ZF)} = 902.97s, \tag{17}$$

thus corroborating the conclusion.

References

[1] Belovas, I.; Sabaliauskas, M; Kuzma, L. Series with binomial-like coefficients for the investigation of fractal structures associated with the Riemann zeta function. *Fractal Fract.* **2022** (in review). doi.org/10.20944/ preprints202205.0122.v1

[2] Belovas, I.; Sakalauskas, L.; Starikovičius, V. A method for accelerated computation of the Riemann zeta function on the complex plane. *Publ. Math. Debrecen* **2022**, 100(1-2), 167-184. doi.org/10.5486/PMD.2022.9120

[3] Borwein, P. An efficient algorithm for the Riemann zeta function. In *Constructive, Experimental, and Nonlinear Analysis (Limoges, 1999)* Canadian Mathematical Society Conference Proceedings **2000**, 27, 29–34. cecm.sfu.ca/~pborwein/PAPERS/P155.pdf

[4] Fischer, K. *The Zetafast algorithm for computing zeta functions* **2017**. arxiv.org/abs/1703.01414

[5] Kuzma, L. Visualization of surfaces and 3D curves, associated with the Riemann zeta function and its non-trivial zeros arrangement (Bachelor thesis) **2021**. lvb.lt/permalink/f/16nmo04/ELABAETD107115273 (in Lithuanian)