Abstract

In this paper, we present an optimization of Odlyzko and Schönhage algorithm that computes efficiently Zeta function at large height on the critical line, together with computation of zeros of the Riemann Zeta function thanks to an implementation of this technique. The first family of computations consists in the verification of the Riemann Hypothesis on all the first 10^{13} non trivial zeros. The second family of computations consists in verifying the Riemann Hypothesis at very large height for different height, while collecting statistics in these zones. For example, we were able to compute two billion zeros from the 10^{24}-th zero of the Riemann Zeta function.

The 10^{13} first zeros of the Riemann Zeta function, and zeros computation at very large height

Xavier Gourdon

version of : October 24th 2004

1 Introduction

The Riemann Zeta function is defined by

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

for complex values of s. While converging only for complex numbers s with $\Re(s) > 1$, this function can be analytically continued to the whole complex plane (with a single pole at $s = 1$). The Riemann Zeta-function was first introduced by Euler with the computation of

$$\sum_{n=1}^{\infty} \frac{1}{n^2}$$

but it was Riemann who, in the 1850's, generalized its use and showed that the distribution of primes is related to the location of the zeros of Zeta. Riemann conjectured that the non trivial zeros of $\zeta(s)$ are located on the critical line $\Re(s) = 1/2$. This conjecture, known as the Riemann Hypothesis (RH), has never been proved or disproved, and is probably the most important unsolved problem in mathematics.

1.1 Numerical verification of the RH on the first zeros

History of numerical verifications

Numerical computations have been made threw the ages to check the RH on the first zeros. Computer age, starting with Turing computations, permitted to perform verification higher than billions of zeros. An history of the RH verification on the first n zeros is given below.

Year	n	Author
1903	15	J. P. Gram [8]
1914	79	R. J. Backlund [1]
1925	138	J. I. Hutchinson [10]
1935	1,041	E. C. Titchmarsh [30]
1953	1,104	A. M. Turing [33]
1956	15,000	D. H. Lehmer [16]
1956	25,000	D. H. Lehmer [15]
1958	35,337	N. A. Meller [18]
1966	250,000	R. S. Lehman [14]
1968	3,502,500	J. B. Rosser, J. M. Yohe, L. Schoenfeld [28]
1977	40,000,000	R. P. Brent [3]
1979	81,000,001	R. P. Brent [4]
1982	200,000,001	R. P. Brent, J. van de Lune, H. J. J. te Riele, D. T. Winter [34]
1983	300,000,001	J. van de Lune, H. J. J. te Riele [12]
1986	1,500,000,001	J. van de Lune, H. J. J. te Riele, D. T. Winter [13]
2001	10,000,000,000	J. van de Lune (unpublished)
2003	250,000,000,000	S. Wedeniwski [35]

It is important to notice here that the recent fast method by Odlyzko and Schönhage to perform multi-evaluation of the Riemann Zeta function (see later for detail) has not been used even in the recent important distributed computation directed by S. Wedeniwski (on a Pentium 4 2Ghz equivalent time, this computation required about 700 years of computation). The method he used is more classical and is more likely to control potential numerical errors and implementation bugs.

Consequences of numerical verifications of the RH

Without the assumption of the RH, numerical verifications of the RH on the first N zeros permit to derive explicit estimates for some number theoretic functions, like

$$\pi(x) = \sum_{p \leq x} 1, \qquad \psi(x) = \sum_{(p,\nu):p^\nu \leq x} \log p, \qquad \theta(x) = \sum_{p \leq x} \log p,$$

where p runs other prime numbers. This development have been initiated by Rosser [26] in 1941, later improved by Rosser and Schoenfeld [27, 29] in 1975-1976. For example, Rosser and Schoenfeld, based on the verification on the RH on the first $3,502,500$ zeros [28], proved that for $x > 1.04 \times 10^7$,

$$|\psi(x) - x| < 0.0077629 \frac{x}{\log x}, \qquad |\theta(x) - x| < 0.0077629 \frac{x}{\log x}.$$

From RH verification at larger height, more recent progress have been obtained. For example, Dusart in [6] obtained several tighter estimates of this kind based on the RH verification until the first $1,500,000,001$ zeros [13]. Very recently, Ramar and Saouter [25], from S. Wedeniwski computations that shows that all non-trivial zeros $s = \sigma + it$ of Zeta for $|t| < T_0 = 3.3 \times 10^9$ lie on the critical line, obtained an estimate of a different kind by proving that for every real number $x \geq 10,726,905,041$, there exists at least one prime number p such that

$$x \left(1 - \frac{1}{28,314,000}\right) < p \leq x.$$

Our result of the RH verification until the 10^{13}-th zero should permit to improve such quantitative estimates a little more. It is of importance here to state that numerical verification of the RH is proven by a large computation. Thus, in addition to possible errors in the validity of used results and algorithms, it is subject to several possible other errors that would not be easily controlled (human coding bug, compiler bug, system bug, processor bug, etc). Unlike other numerical computations for which the notion of certificate permits relatively easy verification, (examples include primality testing with Elliptic curve for example (ECPP), integer factorization, odd perfect number bounds) here the verification has the same cost as the total computation which was used to obtain the result (unlike computations with certificates). It is thus difficult to consider such results as "proved" in a strong sense as a pure mathematical proof. This problematic is expected to be more and more important in the future, as results "computationnaly proved" are likely to be more frequent. Discussion about validity of our RH verification until the 10^{13}-th zero is the object of section 3.3.1.

1.2 Numerical computations of the distribution of the zeros of the Zeta function

While numerical computations on zeros of the Zeta function have long been focused on RH verification only (to check the RH, isolating the zeros is sufficient so no precise computations of the zeros are needed) it was Odlyzko who the first, computed precisely large consecutive sets of zeros to observe their distribution. More precisely, Odlyzko made some empirical observations of the distribution on the spacing between zeros of $\zeta(s)$ in various zones and checked the correspondence with the *GUE hypothesis*, which conjectures that normalized spacing between zeros behaves like eigenvalues of random hermitian matrices (see section 4.2 for more details). In 1987, Odlyzko computed numerically 10^5 zeros of the Riemann Zeta function between index $10^{12} + 1$ and $10^{12} + 10^5$ to the accuracy of 10^{-8} and was the first to observe a good agreement with the GUE hypothesis (see [20]). Later, in order to reach much higher heights, Odlyzko with Schönhage [24] developed a fast algorithm for multi-evaluation

of $\zeta(s)$. After refinements to make this method efficient for practical purposes, Odlyzko was able to compute 70 million zeros at height 10^{20} in 1989 and then 175 million in 1992 at the same height (see [21]). Later he reached the height 10^{21} (see [22]), and in 2001 he computed ten billion zeros at height 10^{22} (see [23]). In a more recent unpublished work in 2002, Odlyzko computed twenty billion zeros at height 10^{23}.

1.3 Notations and definitions

All results in this section are classical and can be found in [31] or [7] for example.

It is known that all non-trivial zeros are located in the band $0 < \Re(s) < 1$. The Riemann Hypothesis is the conjecture that all these zeros are located on the *critical line* $\Re(s) = 1/2$. Restricting on zeros with positive imaginary part, the n-th zero (sorted in increasing order of its imaginary part) is denoted by ρ_n and we denote $\gamma_n = \Im(\rho_n)$. Thus if the RH is true, we have $\rho_n = 1/2 + i\gamma_n$.

We define

$$\theta(t) = \arg\left(\pi^{-it/2}\Gamma(1/4 + it/2)\right) \tag{1}$$

where the argument is defined by continuous variation of t starting with the value 0 at $t = 0$. A consequence of the Zeta functional equation is that the function

$$Z(t) = e^{i\theta(t)}\zeta(\tfrac{1}{2} + it),$$

known as the *Riemann-Siegel Z-function*, is real valued. Moreover we have $|Z(t)| = |\zeta(1/2+it)|$ thus the zeros of $Z(t)$ are the imaginary part of the zeros of $\zeta(s)$ which lie on the critical strip. We are lead to finding change of sign of a real valued function to find zeros on the critical strip, and this is a very convenient property in numerical verification of the RH. This is why our computations concentrate on $Z(t)$ evaluation.

We define

$$S(t) = \frac{1}{\pi}\arg\zeta(\tfrac{1}{2} + it), \tag{2}$$

where the argument is defined by continuous variation of s in $\zeta(s)$ starting at $s = 2$, then vertically to $s = 2 + it$, then horizontally to $s = 1/2 + it$. The number of zeros ρ of Zeta function with $0 < \Im(\rho) < t$ is denoted by $N(t)$. We have

$$N(t) = 1 + \frac{1}{\pi}\theta(t) + S(t). \tag{3}$$

It is known unconditionally that

$$S(t) = O(\log t)$$

so that with the asymptotic expansion of $\theta(t)$ we get

$$N(t) = \frac{t}{2\pi}\log\frac{t}{2\pi} - \frac{t}{2\pi} + O(\log t).$$

This entails that $\gamma_n \sim 2\pi n/\log n$. The function $S(t)$ is important and has been widely studied. It is known that unconditionally,

$$S_1(t) \equiv \int_0^t S(u)du = O(\log t).$$

Thus the mean value of the $S(t)$ function is zero. Another result states that the mean value of $S(t)^2$ is $2\pi^2 \log\log T$, so $S(t)$ is a small and very very slowly increasing function on average. As we will see later, unexpected behavior of the $\zeta(1/2 + it)$ function is likely to occur when $S(t)$ is large, and this is why some particular statistics were done on this function.

2 Computation of the Riemann-Siegel Z-function

In this section, we describe an algorithm due to Odlyzko and Schönhage that computes efficiently values of the Riemann Zeta function on the critical line, together with practical considerations and optimizations that were used in our implementation.

2.1 Notations

We first recall that on the critical line $\sigma = 1/2$, the Riemann Zeta function satisfies

$$\zeta(\tfrac{1}{2} + it) = e^{-i\theta(t)} Z(t) \tag{4}$$

where $\theta(t)$ is a real valued function defined in (1). As t goes to infinity, $\theta(t)$ satisfies the asymptotic formula

$$\theta(t) = \frac{t}{2}\log\frac{t}{2\pi} - \frac{t}{2} - \frac{\pi}{8} + \frac{1}{48t} + \frac{7}{5760t^3} + \cdots \tag{5}$$

The Riemann-Siegel Z-function is a real valued function of the real variable t and satisfies the *Riemann-Siegel expansion*

$$\begin{aligned} Z(t) &= 2\sum_{n=1}^{m} \frac{\cos(\theta(t) - t\log n)}{\sqrt{n}} + R(t) \\ R(t) &= (-1)^{m+1}\tau^{-1/2}\sum_{j=0}^{M}(-1)^j \tau^{-j}\Phi_j(z) + R_M(t), \end{aligned} \tag{6}$$

with $R_M(t) = O(t^{-(2M+3)/4})$, where we used the notations

$$\tau = \sqrt{\frac{t}{2\pi}}, \quad m = \lfloor \tau \rfloor, \quad z = 2(t-m) - 1.$$

The first functions $\Phi_j(z)$ are defined by

$$\begin{aligned} \Phi_0(z) &= \frac{\cos(\frac{1}{2}\pi z^2 + \frac{3}{8}\pi)}{\cos(\pi z)} \\ \Phi_1(z) &= \frac{1}{12\pi^2}\Phi_0^{(3)}(z) \\ \Phi_2(z) &= \frac{1}{16\pi^2}\Phi_0^{(2)}(z) + \frac{1}{288\pi^4}\Phi_0^{(6)}(z) \end{aligned}$$

The general expression of $\Phi_j(z)$ for $j > 2$ Is quite complicated and we refer to [36] or [31] for it. As exposed in [11], explicit bounds have been rigorously obtained on the error term $R_M(t)$, and for $t \geq 200$, one has

$$|R_0(t)| \leq 0.127\, t^{-3/4}, \qquad |R_1(t)| \leq 0.053\, t^{-5/4}, \qquad |R_2(t)| \leq 0.011\, t^{-7/4}.$$

In the practice for computations of zeros of $Z(t)$ above the 10^{10}-th zero for example, the choice $M = 1$ permits to obtain an absolute precision of $Z(t)$ smaller than 2×10^{-14} and this is sufficient to locate the zeros.

2.2 Presentation of the Odlyzko-Schönhage algorithm

The Odlyzko-Schönhage algorithm permits efficient evaluations of $Z(t)$ in a range of the form $T \leq t \leq T + \Delta$, with $\Delta = O(\sqrt{T})$. For t in this range, we write

$$Z(t) = \sum_{n=1}^{k_0-1} \frac{\cos(\theta(t) - t\log n)}{\sqrt{n}} + \Re(e^{-i\theta(t)}F(t)) + \sum_{n=k_1+1}^{m} \frac{\cos(\theta(t) - t\log n)}{\sqrt{n}} + R(t)$$

where $R(t)$ is the remainder term defined in (6) and where $F(t)$ is a complex function defined by

$$F(t) = F(k_0 - 1, k_1; t) := \sum_{k=k_0}^{k_1} \frac{1}{\sqrt{k}} \exp(it\log k) \tag{7}$$

where $k_1 = \lfloor \sqrt{T/(2\pi)} \rfloor$ and k_0 is a small fixed integer (we will discuss its value later). What counts is that, for a given interval $[T, T + \Delta]$ and t in this range, the values of k_0 and k_1 are fixed in our computations of $Z(t)$. Since k_0 is chosen as very small compared to $T^{1/2}$ and since $m - k_1$ is bounded (because $\Delta = O(\sqrt{T})$), we see that the most time consuming part in the computation of $Z(t)$ is the computation of $F(k_0 - 1, k_1; t)$. The rest of the technique is mainly dedicated to fast evaluation of this sum.

Main steps of the algorithm

To obtain fast evaluations of $F(t)$ in the range $[T, T+\Delta]$, the Odlyzko-Schönhage algorithm is divided in two steps :

- first, multi-evaluations of $F(t)$ are computed on a well chosen regular grid of abscissa for t.
- from these values, an interpolation formula permits to obtain efficiently any value of $F(t)$ at a certain accuracy when t stays in our range.

In the original Odlyzko and Schönhage algorithm of [24], multi-evaluations of not-only $F(t)$ were needed, but also multi-evaluations of derivatives of $F(t)$ on the regular grid. An important improvement of this have been obtained by Odlyzko (see [21]) that permits to have an interpolation formula from multi-evaluations of $F(t)$ only.

We first concentrate on the multi evaluation part of the algorithm, which requires numerous treatments to make it efficient, and which is the most time consuming for computation at very large height (say above the 10^{20}-th zero). Interpolation considerations will be presented afterward. Of course, the way the regular grid should be chosen in the previous step is dependent of the interpolation formula.

2.3 Fast multi-evaluation of $F(t)$ on a regular grid

We describe here how Odlyzko-Schönhage algorithm permits to evaluate efficiently approximations at a controlled accuracy of the values $F(t)$ defined in (7) at evenly spaced values

$$t = T_0, \quad T_0 + \delta, \quad \ldots, \quad T_0 + (R-1)\delta.$$

We will later precise what values of δ and R we should choose. Instead of computing directly the values $F(T_0)$, $F(T_0+\delta)$, ..., $F(T_0+(R-1)\delta)$, the key idea of [24] is to compute their discrete Fourier transform, defined by

$$u_k = \sum_{j=0}^{R-1} F(T_0 + j\delta)\omega^{-jk}, \qquad \omega = \exp(2i\pi/R),$$

for $0 \le k < R$. The property of the inverse Fourier transform gives

$$F(T_0 + j\delta) = \frac{1}{R}\sum_{k=0}^{R-1} u_k \omega^{jk}. \tag{8}$$

In the algorithm, the value of R is chosen to be a power of two, so the values $F(T_0+j\delta)$ are efficiently obtained from the (u_k) with an FFT transform. As Odlyzko described in [22], we found that this FFT takes a small portion of the total time (not more than a few percent of the total time in our implementation, even using disk FFT for very large values of R). So the problem is now reduced to computing efficiently the (u_k). Using the definition (7) of $F(t)$ and exchanging the order of summations, we obtain that $u_k = \omega^k f(\omega^k)$, where $f(z)$ is defined by

$$f(z) = \sum_{k=k_0}^{k_1} \frac{a_k}{z - b_k}, \qquad b_k = e^{i\delta \log k}, \quad a_k = \frac{e^{iT_0 \log k}}{k^{1/2}}(1 - e^{iR\delta \log k}). \tag{9}$$

The key point is now to compute efficiently the complex values $f(\omega^k)$ for $0 \le k < R$. The approach presented in [24] and [22] is to use Taylor series expansions for different subset of the indices in the sum. Given a complex number z_0 on the unit circle and given L, $0 < L < 1$, consider for example the subset I of indices in $\{k_0, k_0+1, \ldots, k_1\}$ such that $|b_k - z_0| > L$ for all $k \in I$. We have the Taylor series expansion

$$f_I(z) \equiv \sum_{k\in I} \frac{a_k}{z - b_k} = \sum_{n\ge 0} A_n (z_0 - z)^n, \qquad A_n = \sum_{k\in I} (z_0 - b_k)^{-n-1}$$

valid for all z on the unit circle with $|z - z_0| < L$. When restricting to z with $|z - z_0| < L/2$ for example, the convergence of the Taylor series expansion is better than $(1/2)^n$ and a fixed number of terms (say V) in this expansion is sufficient to obtain a given absolute precision.

Approximations of values $f_I(\omega^j)$ for indices j such that $|\omega^j - z_0| < L/2$ (let us call J this set of indices) are then obtained from this Taylor series expansion. When the number of elements in I is much bigger than the number of terms V used in the series expansion, this gives a much more efficient method to compute the values $(f_I(\omega^j))_{j\in J}$. This idea is used in [24] as a basic ingredient of a sophisticated method that writes $f(z)$ in the form $\sum f_{I_\ell}$, the (I_ℓ) being a partition of $\{k_0, k_0+1, \ldots, k_1\}$ chosen such that Taylor series convergence of f_{I_ℓ} is well controlled. In our implementation, we used two different approaches which are more efficient. The first approach, described in 2.3.1, is dedicated to computations of the Riemann Zeta function at large but reasonable height (say until the 10^{13}-th zero). The second approach, suited to computations at larger height, is the Greengard-Rokhlin algorithm, that Odlyzko presented in [22] as a possible improvement but he did not implement it in his computations. Implementation aspects of the Greengard-Rokhlin algorithm are described in 2.3.2.

2.3.1 Fast multi-evaluations of $f(\omega^k)$ at reasonable height

The approach of Odlyzko and Schöhnage for the computation of $(f(\omega^k))_{0\le k<R}$ is not optimal for practical concerns. We propose another technique which benefits from several optimizations. First, it permits a certain flexibility that can be used at different recursive level of the computation (tuning of the speed of convergence of approximating series for example). Then we make use of a better approximation compared to the Taylor series expansion, which permits to decrease the number of terms. Finally, instead of evaluating complex Taylor series expansion of a complex variable, our approximations are computed from a function of a real variable, and benefits additionally from symmetry in the abscissa. We did not implement Odlyzko and Schönhage algorithm in order to have a precise estimation of the saving, but we estimate that our approach is probably at least twice of three times faster (however, only a constant factor is saved).

Chebychev interpolation. The basic ingredient in our approach is, instead of using Taylor series of the complex variable z of $a_k/(z-b_k)$, to use Chebychev interpolation of the function $a_k/(e^{i\theta}-b_k)$ of the real variable θ. On the interval $[\theta_0 - L, \theta_0 + L]$, the interpolation of degree N is performed on the abscissas

$$\alpha_j = \theta_0 + L\gamma_j, \qquad \gamma_j = \cos\frac{(2j+1)\pi}{2N}$$

and the resulting interpolating polynomial of a function $G(\theta)$ on this interval is

$$P_{N,\theta_0,L}(\theta) = \sum_{j=0}^{N-1} \frac{G(\alpha_j)R_N(\theta)}{R'_N(\alpha_j)(\theta-\alpha_j)}, \qquad R_N(\theta) = \prod_{j=0}^{N-1}(\theta-\alpha_j). \tag{10}$$

When the singularities of $G(\theta)$ are far from the interval $[\theta_0 - L, \theta_0 + L]$, this interpolating polynomial is a good approximation to $G(\theta)$ in this interval. The following result gives a quantitative information of the approximation bound.

Proposition 1 *[Point interpolation error bound] Let*

$$G(\theta) = \sum_k \frac{a_k}{b_k - e^{i\theta}}, \qquad b_k = e^{i\beta_k}$$

be a function of the real variable θ. Suppose that for all k,

$$\|\beta_k - \theta_0\| \equiv \min_n |\beta_k - \theta_0 + 2n\pi| > \lambda L \qquad (\lambda > 1).$$

Then for all θ in $[\theta_0 - L, \theta_0 + L]$ we have

$$|G(\theta) - P_{N,\theta_0,L}(\theta)| < \frac{4 + 2^{1-N}}{(\lambda-1)L\,K(\lambda)^N}\sum_k |a_k|, \qquad K(\lambda) = \lambda + \sqrt{\lambda^2-1},$$

where $P_{N,\theta_0,L}(\theta)$ is the degree N Chebyshev interpolating polynomial of $G(\theta)$ other $[\theta_0 - L, \theta_0 + L]$ defined in (10).

Proof : We make use of the residue theorem, from the integral on the complex domain of t

$$\int_{|t-\theta_0|=R} \frac{G(t)}{R_N(t)} \frac{dt}{t-\theta}$$

where $R = 2n\pi$. As $n \to \infty$, the function $G(t)$ stays bounded on $|t-\theta_0| = R$ and we obtain that this integral vanishes. Thus the sum of its residues vanishes. The residues of the integrand are the points $t = \theta$, the zeros α_j of $R_N(t)$ and the poles $\beta_k + 2n\pi$ of $G(t)$, thus

$$\frac{G(\theta)}{R_N(\theta)} + \sum_{j=0}^{N-1} \frac{G(\alpha_j)}{R'_N(\alpha_j)(\alpha_j - \theta)} + \sum_k \sum_{n=-\infty}^{\infty} \frac{a_k}{R_N(\beta_k + 2n\pi)(\beta_k + 2n\pi - \theta)} = 0.$$

This identity rewrites in a form that gives an explicit error term in the approximation obtained with the interpolating polynomial

$$G(\theta) - P_{N,\theta_0,L}(\theta) = E(\theta), \qquad E(\theta) = \sum_k a_k \sum_{n=-\infty}^{\infty} \frac{R_N(\theta)}{R_N(\beta_k + 2n\pi)(\theta - \beta_k + 2n\pi)}.$$

The polynomial $R_N(t)$ is related to the $[-1,1]$ Chebychev polynomial $T_N(t)$ thanks to the formula

$$R_N(\theta_0 + Lt) = \frac{L^N}{2^{N-1}} T_N(t), \qquad T_N(t) = \cos(N \arccos(t)). \tag{11}$$

which entails

$$E(\theta) = \sum_k a_k \sum_{n=-\infty}^{\infty} \frac{T_N((\theta - \theta_0)/L)}{T_N((\beta_k - \theta_0 + 2n\pi)/L)(\theta - \beta_k + 2n\pi)}.$$

Since $|\theta - \theta_0| < L$, we have $|T_N((\theta-\theta_0)/L)| \leq 1$. Now we concentrate on $T_N((\beta_k - \theta_0 + 2n\pi)/L)$. We can choose β_k modulo 2π such that the inequality

$$\left| \frac{\beta_k - \theta_0 + 2n\pi}{L} \right| > |2n+1|\lambda$$

is always fulfilled. Since for real x and $|x| > 1$ we have

$$|T_N(x)| = \cosh(N \operatorname{arccosh}(|x|)) > \frac{K(|x|)^N}{2}, \qquad K(x) = x + \sqrt{x^2 - 1},$$

we deduce

$$\begin{aligned} \sum_{n=-\infty}^{+\infty} & \frac{T_N((\theta - \theta_0)/L)}{T_N((\beta_k - \theta_0 + 2n\pi)/L)(\theta - \beta_k + 2n\pi)} \\ & < \sum_{n=-\infty}^{+\infty} \frac{2}{K(|2n+1|\lambda)^N |2n+1|(\lambda - 1)L} \\ & < \frac{2}{K(\lambda)^N (\lambda - 1)L} \sum_{n=-\infty}^{+\infty} \frac{1}{|2n+1|^{N+1}} < \frac{4 + 2^{1-N}}{K(\lambda)^N (\lambda - 1)L}, \end{aligned}$$

where we have used the property $K(k\lambda) > kK(\lambda)$. The result follows easily. •

In fact, it is cheaper in terms of performance to compute G and G' at a given point rather than computing G at two different points. This remark motivates an interpolation at point and derivatives of G instead of points of G only. The previous proposition easily generalizes to the following result :

Proposition 2 (Point and derivatives interpolation error bound) *Under the assumptions of proposition 1, for all real values of θ such that $|\theta - \theta_0| \leq L$, we have*

$$|G(\theta) - Q_{N,\theta_0,L}(\theta)| < \frac{8 + 2^{2-N}}{(\lambda - 1)L\, K(\lambda)^{2N}} \sum_k |a_k|, \qquad K(\lambda) = \lambda + \sqrt{\lambda^2 - 1},$$

where $Q_{N,\theta_0,L}(\theta)$ is the degree $2N$ point-and-derivative Chebyshev interpolating polynomial of $G(\theta)$ other $[\theta_0-L,\theta_0+L]$, with interpolating abscissa $(\alpha_j)_{0\leq j<N}$. The interpolating polynomial is defined by

$$Q_{N,\theta_0,L}(\theta)=\sum_{j=0}^{N-1}\left(\frac{G(\alpha_j)R_N(\theta)^2}{R'_N(\alpha_j)^2(\theta-\alpha_j)^2}-\frac{G(\alpha_j)R''(\alpha_j)R_N(\theta)^2}{R'_N(\alpha_j)^2(\theta-\alpha_j)}+\frac{G'(\alpha_j)R_N(\theta)^2}{R'_N(\alpha_j)^2(\theta-\alpha_j)}\right).$$

Proof : The proof is similar to the previous one, starting from the contour integral

$$\int_{|t-\theta_0|=R}\frac{G(t)}{R_N(t)^2}\,\frac{dt}{t-\theta}$$

and taking into account residues of the integrand. •

Practical implementation of the interpolation. In our implementation, we used the point-and-derivative Chebychev interpolation defined in proposition 2. Undesired numerical errors can appear if one does not achieve the interpolation in the right way. To overcome this problem, we keep a non expanded form of the interpolation which is described below.

It is convenient to express the interpolation on $[-1,1]$ instead of $[\theta_0-L,\theta_0+L]$, which writes in the form $G(\theta_0+Lt)\simeq Q_N(t)$ with

$$Q_N(t)=\sum_{j=0}^{N-1}\left(\frac{G(\theta_0+L\gamma_j)T_N(t)^2}{T'_N(\gamma_j)^2(t-\gamma_j)^2}-\frac{G(\theta_0+L\gamma_j)T''(\gamma_j)T_N(t)^2}{T'_N(\gamma_j)^3(t-\gamma_j)}+\frac{L\,G'(\theta_0+L\gamma_j)T_N(t)^2}{T'_N(\gamma_j)^2(t-\gamma_j)}\right),$$

with $\gamma_j=\cos\frac{(2j+1)\pi}{2N}$ and $T_N(t)=2^{N-1}\prod_j(t-\gamma_j)$ the N-th Chebychev polynomial. The interpolating polynomial $Q_N(t)$ also writes as

$$Q_N(t)=\sum_{j=0}^{N-1}G(\theta_0+L\gamma_j)V_j(t)+L\,G'(\theta_0+L\gamma_j)W_j(t)$$

with $V_j(t)$ and $W_j(t)$ the interpolation base polynomials of degree $2N-1$ defined by

$$V_j(t)=\frac{T_N(t)^2}{T'_N(\gamma_j)^2(t-\gamma_j)^2}-\frac{T''_N(\gamma_j)T_N(t)^2}{T'_N(\gamma_j)^3(t-\gamma_j)}=\prod_{k\neq j}\left(\frac{t-\gamma_k}{\gamma_j-\gamma_k}\right)^2\left(1-2(t-\gamma_j)\sum_{k\neq j}\frac{1}{\gamma_j-\gamma_k}\right)$$

and

$$W_j(t)=\frac{T_N(t)^2}{T'_N(\gamma_j)^2(t-\gamma_j)}=(t-\gamma_j)\prod_{k\neq j}\left(\frac{t-\gamma_k}{\gamma_j-\gamma_k}\right)^2.$$

In fact, coefficients of the polynomials $V_j(x)$ and $W_j(x)$ are quite big compared to the evaluations of these polynomials in $[-1,1]$, and this creates important numerical imprecisions. To avoid them, it is much better to keep a non-expanded form of the polynomials. We define the linear polynomials

$$A_j(t)=\frac{1}{\prod_{k\neq j}(\gamma_j-\gamma_k)}\left(1-2(t-\gamma_j)\sum_{k\neq j}\frac{1}{\gamma_j-\gamma_k}\right)\quad\text{and}\quad B_j(t)=\frac{(t-\gamma_j)}{\prod_{k\neq j}(\gamma_j-\gamma_k)},$$

so that

$$V_j(t)=\left(\prod_{k\neq j}(t-\gamma_k)^2\right)A_j(t),\qquad W_j(t)=\left(\prod_{k\neq j}(t-\gamma_k)^2\right)B_j(t).$$

Now suppose that N is even. During the interpolation, we define the linear polynomials $L_j(t)$ by

$$L_j(t)=G(\theta_0+L\gamma_j)A_j(t)+LG'(\theta_0+L\gamma_j)B_j(t)$$

and then cubic polynomials $C_j(t)$ (for $j<N/2$) by

$$C_j(t)=(t-\gamma_j)^2L_{n-j}(t)+(t-\gamma_{n-j})^2L_j(t).$$

The symmetry of Chebyshev roots $\gamma_j = -\gamma_{n-j}$ entails

$$Q_N(t) = \sum_{0 \le j < N/2} \left(\prod_{\substack{k \neq j \\ 0 \le k < N/2}} (t^2 - \gamma_k^2)^2 \right) C_j(t).$$

Taking into account additional easy factorizations, this form is well suited to fast and precise evaluation of both $Q_N(t)$ and $Q_N(-t)$.

In our implementation, we choose the parameters $\lambda = 2$ and $N = 8$, that provide enough precision for the $Z(t)$ evaluation in the context of RH verification until the 10^{13}-th zero.

Recursive process for fast computation of $f(\omega^j)$. Recall that our objective is to obtain a fast computation of the values $f(\omega^j)$ where $\omega = \exp(2i\pi/R)$ and

$$f(z) = \sum_{k=k_0}^{k_1} \frac{a_k}{z - b_k}, \qquad b_k = e^{i\beta_k}, \quad \beta_k = \delta \log k.$$

Now suppose we want to compute approximations of the values $f(\omega^j)$ for consecutive indexes $j = j_0, j_0 + 1, \ldots, j_1$. We define $j_c = (j_0 + j_1)/2$ and $L = (j_1 - j_0)\pi/R$. We construct the subset I of indices in $K = \{k_0, k_0 + 1, \ldots, k_1\}$ such that $\|\beta_k - \omega^{j_c}\| > \lambda L$ for all $k \in I$. The contribution $f_I(\omega^j)$, where

$$f_I(z) = \sum_{k \in I} \frac{a_k}{z - b_k},$$

can be approximated efficiently for values of $j = j_0, j_0 + 1, \ldots, j_1$ by using a point-and-derivative Chebychev interpolation of $f_I(e^{i\theta})$ on the interval $[2j_0\pi/R, 2j_1\pi/R] = [\theta_0 - L, \theta_0 + L]$ where $\theta_0 = 2j_c\pi/R$; moreover, the approximation error can be estimated thanks to proposition 2. To obtain contribution on the remaining k indices in $K - I$, the process is made recursive, by considering the two half of indexes $j_0, \ldots, j_c$ and $j_c + 1, \ldots, j_1$ and considering that the remaining set of k indices is $K - I$ instead of K. When the number of j indices is small, direct evaluation of $f_I(\omega^j)$ is made instead of using the Chebychev interpolation. A particular attention to numerical errors should be payed for cases for which ω^j is very close to a β_k value. When this situation occurs, the stable expression

$$\frac{a_k}{\omega^j - b_k} = -\frac{e^{iT_0 \log k}}{k^{1/2}} \exp(-i((R+1)x + \delta \log k)) \frac{\sin Rx}{\sin x}, \qquad x = \frac{j\pi}{R} - \frac{\delta \log k}{2}. \tag{12}$$

should be used.

It is interesting to notice that, at each step of our recursive process, a different value of λ and N (number of points considered for the interpolation) can be taken, giving potential high flexibility to our algorithm. However, in our implementation, we did not make use of this potential and kept the fixed values $\lambda = 2$ and $N = 8$ that gave good results. We choose to implement the recursive process presented above by starting with four sets of j indices corresponding to quarters of the complete j indices set.

2.3.2 Implementation of the Greengard-Rokhlin algorithm for huge height

When the number of k indices is much bigger than the number of j indices in the evaluations of $f(\omega^j)$, the Greengard-Rokhlin algorithm is much more efficient than the Chebychev interpolation based technique. This algorithm was first designed for Coulomb and gravitational potentials evaluations (see [9]), and later subsequently improved, extended and applied to other areas. Moreover, as described below, it is well suited to the use of disk memory since it requires only a few number of memory access. In [22], Odlyzko presented this algorithm as a possible improvement to his technique but did not implement it.

Instead of computing the Taylor series of the complex variable z of $a_k/(z - b_k)$, the Greengard-Rokhlin algorithm consists in computing the series in the variable $1/(z-c)$ outside the pole c, by writing

$$\frac{a_k}{z - b_k} = \frac{a_k}{(z-c)\left(1 - \frac{b_k - c}{z - c}\right)} = \sum_{n=0}^{\infty} a_k (b_k - c)^n \frac{1}{(z-c)^{n+1}}.$$

The approximation is obtained by considering only the first V most significant terms in this sum (we will discuss the value of V later), thus giving

$$f_K(z) = \sum_{k\in K} \frac{a_k}{z-b_k} \approx \sum_{n=0}^{V-1} A_n(K,c) \frac{1}{(z-c)^{n+1}}$$

with

$$A_n(K,c) = \sum_{k\in K} a_k (b_k - c)^n. \tag{13}$$

The values of coefficients $A_n(K,c)$ when the pole c is moved satisfy a very nice property, that makes the Greengard-Rokhlin algorithm so efficient. We have the shift formula

$$A_n(K,d) = \sum_{j=0}^{n} \binom{n}{j} (c-d)^{n-j} A_j(K,c), \tag{14}$$

thus once the values $A_j(K,c)$ are known, the value of $A_n(K,d)$ can be computed with $O(n)$ operations only. Since n is not too large (in the practice, it is smaller than V which is less than 100), we save a lot of time compared to the use of definition formula (13) which is leads to much more expensive computations when the set K is very large.

The recursive process of Greengard-Rokhlin algorithm For a subset $K \subset \{k_0, k_0+1, \ldots, k_1\}$ of k indices and a subset $J \subset \{0, 1, \ldots, R-1\}$ of j indices, the Greengard-Rokhlin procedure, denoted by

$$\mathrm{GR}(J,K)$$

computes approximations of the contributions $f_K(\omega^j)$ for $j \in J$, where

$$f_K(z) = \sum_{k\in K} \frac{a_k}{z-b_k}, \qquad b_k = e^{i\beta_k}.$$

It also returns a pole c and the sequence of coefficients $A_n(K,c)$, defined in (13), for $0 \le n < V$.

The recursive process to compute $\mathrm{GR}(J,K)$ is as follows :

- First the extremal values $\beta_{\min}$ and $\beta_{\max}$ of the sequence $(\beta_k)_{k\in K}$ are computed. The pole c is chosen as $c = e^{\frac{i}{2}(\beta_{\min}+\beta_{\max})}$.
- If the number of elements in K or in J is "small" (say $\#K = O(V)$ or $\#J = O(1)$), then the contributions $f_K(\omega^j)$ for $j \in J$ are computed from the direct formula (for stability, formula (12) should also be used if needed). The values $A_n(K,c)$ for $0 \le n < V$ are computed directly with formula (13).
- Else, we define

  $$\beta' = \frac{\beta_{\min}+\beta_{\max}}{2}, \qquad L = \frac{\beta_{\max}-\beta_{\min}}{2}$$

  so that the elements $(\beta_k)_{k\in K}$ are in $[\beta'-L, \beta'+L]$. We then compute the subset J' of J made of j indices which satisfy $|\omega^j - e^{i\beta'}| \le \lambda |1-e^{iL}|$ (to fix ideas we can choose $\lambda = 2$, but the λ factor can be any constant > 1). Then the process is called recursively twice : first a call to $\mathrm{GR}(J', K_1)$ is made, with K_1 the subset of k indices defined by indices k in K such that $\beta_k < \beta'$, and then to $\mathrm{GR}(J', K_2)$ where $K_2 = K \setminus K_1$. The coefficients $A_n(K,c)$ for $0 \le n < V$ are then obtained with $A_n(K,c) = A_n(K_1,c) + A_n(K_2,c)$, where the coefficients $A_n(K_1,c)$ and $A_n(K_2,c)$ are obtained from $A_n(K_1,c_1)$ and $A_n(K_2,c_2)$ with formula (14). As for the values $f_K(\omega^j)$, they are obtained as $f_{K_1}(\omega^j) + f_{K_2}(\omega^j)$ for $j \in J'$, or with the Taylor approximation formula

  $$f_K(\omega^j) \approx \sum_{n=0}^{V-1} A_n(K,c) \frac{1}{(\omega^j - c)^{n+1}} \tag{15}$$

  for $j \in J \setminus J'$.

The error in approximation formula used in the last step can be controlled. In fact, we have

$$|A_n(K,c)| \leq \sum_{k\in K} |a_k|\,|e^{i\beta_k} - c|^n \leq \left(\sum_{k\in K} |a_k|\right) |1 - e^{iL}|^n,$$

thus an error bound on the approximation (15) for $j \in J \setminus J'$ is given by

$$\sum_{n\geq V} |A_n(K,c)| \frac{1}{|\omega^j - c|^{n+1}} \leq \frac{1}{\lambda |1 - e^{iL}|} \left(\sum_{k\in K} |a_k|\right) \sum_{n\geq V} \frac{1}{\lambda^n}.$$

2.3.3 Comparisons between both approaches

The first approach, dedicated to computation of $Z(t)$ at not too large height, and described in 2.3.1, has the same complexity as the approach implemented by Odlyzko in [22], and is equal to

$$nk_1 \log R,$$

n being the degree of the interpolating Chebyshev polynomial. As for the Greengard-Rokhlin algorithm, its use gives a complexity of

$$k_1 V + V^2 R.$$

Since n and V are fixed value of the same magnitude (in fact n is smaller than V due to the best Chebychev interpolation, but by a constant factor; in our implementation, we have $n = 16$ and $V = 30$) and are fixed, the comparison between both complexities give a significant saving for Greengard Rokhlin approach when k_1 is significantly bigger than R. Since in the implementations, the number R, corresponding to the number of discretization points, will be bounded by practical limiting factors (total available memory, including disk space, and total timing) it means that for very large height (thus for very large values of k_1), the Greengard Rokhlin algorithm will be significantly faster.

This theoretical observation is confirmed by practical experiments done with our implementation, achieved for computations of the zeros of $Z(t)$ near the 10^{13}-th zero (in this case, $k_1 \approx 6.2 \times 10^5$). The table in figure 1 gives the total multi-evaluation time of $f(\omega^j)$ for both approaches, for different values of R.

R	First approach	Greengard Rokhlin
65,536	8.56 s	1.73 s
131,072	9.3 s	2.89 s
262,144	9.9 s	4.14 s
524,288	10.8 s	8.12 s
1,048,576	12.1 s	13.47 s
2,097,152	14.2 s	32.3 s
4,194,304	18 s	50.6 s
8,388,608	25.4 s	151 s

Figure 1: Comparison of timing of multi-evaluation of $f(\omega^j)$ for R values of j near height 10^{13}, between first approach, based on interpolating Chebychev polynomial and described in section 2.3.1, and Greengard Rokhlin algorithm. As expected, the Greengard Rokhlin approach is significantly faster when R is small compared to k_1.

The Greengard Rokhlin algorithm has another very important property that permits to deal with very large values of k_1 : the values a_k and b_k can be computed on demand, since they are needed only once, thus there is no need to store them, making this approach particularly suited to very large values of k_1. On the contrary, the first approach based on Chebychev interpolation needs on the average $\log R$ access to the values a_k and b_k, and since computing these values is not immediate (see below), it is better to store them. To overcome this limiting memory factor at huge heights, it appears that one should cut the summation $\sum_k a_k/(z - b_k)$ into a collection of smaller summations (and repeat the process for each sub-summations), thus increasing very significantly the total timing with the first approach.

2.3.4 Additional considerations for multi-evaluation of $F(t)$

We focus here on three particular important topics that permit to achieve properly the multi-evaluation of $F(t)$. The first subject is related to the computations of the values a_k and b_k of (9), for which some care is needed to ensure enough precision. The second topic is the FFT step needed after the multi-evaluation of $f(z)$ to retrieve the values $F(T_0 + j\delta)$. The third topic relates with an important optimization that permitted to decrease the multi-evaluation time by roughly 30% to 40%.

Computing coefficients of the function $f(z)$. In order to compute $f(z)$, defined in (9), one needs to compute properly the coefficients a_k and b_k, for which we recall the definition

$$a_k = \frac{e^{iT_0 \log k}}{k^{1/2}}(1 - e^{iR\delta \log k}), \qquad b_k = e^{i\delta \log k} \tag{16}$$

In our implementation, we just make use of classic `double` floating point data type of C/C++ language, which provides floating point numbers with a mantissa of 53-bits (thus a little more than 15 significant decimal digits). In our context, the values of T_0 and R are very large (the value of T_0 can be bigger than 10^{20}), so to get the values $T_0 \log k \pmod{2\pi}$ and $R\delta \log k \pmod{2\pi}$ at a relative double floating point precision, we need a special treatment. What follows permits to compute efficiently the sequence

$$(u_k)_{k_0 \le k \le k_1}, \qquad u_k = U \log k \pmod{2\pi}$$

at double precision, for a very large value of an integer U (say U of the order of 10^{20} or even more).

First, we developed in our implementation a small module of multiprecision arithmetic, permitting to compute with roughly 40 decimal digits of relative precision. This module has not been optimized but since we use it a small portion of the time, it was sufficient for our purpose. A single value u_K can be computed with our multiprecision arithmetic module, but at a relatively high cost, and the idea of our approach is to compute a large set of successive values (u_k) efficiently, given a precomputation of a reasonable cost. For a given index K, we compute, thanks to the multiprecision module, 32-bits integers A and B and double floating points values with maximal possible precision r_A and r_B such that

$$\frac{U}{K} = \frac{A}{2^{32}}2\pi + r_A \pmod{2\pi}, \qquad \frac{U}{2K^2} = \frac{B}{2^{32}}2\pi + r_B \pmod{2\pi},$$

with $0 \le r_A < 2\pi/2^{32}$ and $0 \le r_B < 2\pi/2^{32}$. Now, once u_K is computed, to get the values u_{K+h} we write

$$u_{K+h} = u_K + U\log(1 + h/K) = u_K + U\rho - U\frac{\rho^2}{2} + U\sum_{j\ge 3}(-1)^{j-1}\frac{\rho^j}{j}, \qquad \rho = \frac{h}{K},$$

which leads to the formula

$$u_{K+h} = u_K + \frac{Ah - Bh^2 \pmod{2^{32}}}{2^{32}}2\pi + r_A h - r_B h^2 - U\sum_{j\ge 3}(-1)^{j-1}\frac{\rho^j}{j} \pmod{2\pi}.$$

In order to keep a good precision with this formula, we choose to restrict on values of h such that $0 < h < 2^{16}$ and such that $U\rho^3/3 \le 1$, which is equivalent to

$$0 < h < \min\left(2^{16}, K(3/U)^{1/3}\right).$$

Performing FFT. The final step to obtain a discretization of $F(t)$ with Odlyzko-Schönhage algorithm is to apply formula (8) to retrieve the values $(F(T_0 + j\delta))_{0\le j<R}$ from the sequence $(u_k)_{0\le k<R}$. We performed this FFT by applying two recursive level of a four-step FFT algorithm as described in [2], which is particularly suited to our context (very big size R of the FFT, and disk memory used). Thanks to this effective approach, the time spent in FFT step does not represent more than 5% to 10% of the total $F(t)$ multi-evaluation step time.

Decreasing the number of terms in the F summation. Odlyzko-Schönhage algorithm permits to perform multi-evaluation of the function

$$F(a,b;t) := \sum_{a<k\leq b} k^{-1/2+it}.$$

Instead of applying directly this technique on the function $F(k_0-1,k_1;t)$, we made use of a rearrangement in order to decrease the number of terms in the sum, that proved to be efficient in the practice. Separating terms of odd and even index in the summation, we get the identity

$$F(0,k_1;t) = \sum_{\substack{0<k\leq k_1\\ k \text{ odd}}} k^{-1/2+it} + \sum_{0<k\leq k_1/2} (2k)^{-1/2+it} = F_{\text{odd}}(0,k_1;t) + 2^{-1/2+it} F\left(0,\frac{k_1}{2};t\right), \tag{17}$$

where $F_{\text{odd}}(a,b;t)$ can be multi-evaluated by Odlyzko-Schönhage algorithm and is defined by

$$F_{\text{odd}}(a,b;t) = \sum_{\substack{a<k\leq b\\ k \text{ odd}}} k^{-1/2+it}.$$

The same identity (17) applied iteratively to $F(0,k_1/2;t)$ leads to

$$F(0,k_1;t) = F_{\text{odd}}(0,k_1;t) + 2^{-1/2+it}\left(F_{\text{odd}}\left(0,\frac{k_1}{2};t\right) + 2^{-1/2+it}F\left(0,\frac{k_1}{4};t\right)\right).$$

After two additional iterations, on $F(0,k_1/4;t)$ then on $F(0,k_1/8;t)$ we get

$$\begin{aligned} F(0,k_1;t) = F_{\text{odd}}(0,k_1;t) &+ 2^{-1/2+it}F_{\text{odd}}\left(0,\frac{k_1}{2};t\right) + 4^{-1/2+it}F_{\text{odd}}\left(0,\frac{k_1}{4};t\right)\\ &+ 8^{-1/2+it}F_{\text{odd}}\left(0,\frac{k_1}{8};t\right) + 16^{-1/2+it}F\left(0,\frac{k_1}{16};t\right), \end{aligned}$$

which finally writes as

$$\begin{aligned} F(0,k_1;t) = 16^{-1/2+it}F\left(0,\frac{k_1}{16};t\right) &+ \left(1+2^{-1/2+it}+4^{-1/2+it}+8^{-1/2+it}\right)F_{\text{odd}}\left(0,\frac{k_1}{8};t\right)\\ &+ \left(1+2^{-1/2+it}+4^{-1/2+it}\right)F_{\text{odd}}\left(\frac{k_1}{8},\frac{k_1}{4};t\right)\\ &+ \left(1+2^{-1/2+it}\right)F_{\text{odd}}\left(\frac{k_1}{4},\frac{k_1}{2};t\right) + F_{\text{odd}}\left(\frac{k_1}{2},k_1;t\right) \end{aligned} \tag{18}$$

Now, instead of applying directly Odlyzko-Schönhage algorithm to $F(k_0-1,k_1;t)$, we apply it to each of the five terms in the above identity. After, we combine the multi-evaluation of each term in the way described in formula (18) and use an additional substraction of $F(0,k_0-1;t)$ to recover multi-evaluation of $F(k_0-1,k_1;t)$. Since $F_{\text{odd}}(a,b;t)$ has $(b-a)/2$ terms in the summation, we see than the total number of needed summations using this technique is $k_1/16 + 1/2(k_1/8+k_1/8+k_1/4+k_1/2) = 9/16k_1$, so a saving of 43.75% compared to the k_1 summations needed in the direct use of the algorithm. But in addition here, we have the extra-cost of computing 5 FFT of size R instead of one, and combining multi-evaluation using (18). In the practice, because FFT is very fast, this extra-cost represented only a few percent of the total time at large height so this technique proved to be useful even if precise time saving has not be measured. The process could be of course refined, by applying more iterations with formula (17) or, more generally, by applying transformations in the spirit of Euler product but we expect that in the practice, it would probably not save significant enough time and would require and additional complicated implementation.

2.4 Band limited function interpolation

From a discretization of $F(t)$ on a regular grid, we now want to approximate any value of $F(t)$ thanks to an interpolation formula. We first state a general result about band-limited function like $F(t)$, which we will later use in our specific context.

2.4.1 A general result on band-limited function interpolation

It is known that for a band-limited function of the form

$$G(x) = \int_{-\tau}^{\tau} g(t) e^{ixt}\, dt,$$

then under smooth conditions, $G(x)$ is determined by its sampling at points $n\pi/\tau$ value with the "cardinal-series"

$$G(x) = \sum_{n=-\infty}^{+\infty} G\left(\frac{n\pi}{\tau}\right) \frac{\sin(\tau x - n\pi)}{\tau x - n\pi}.$$

However, this formula is not well suited to our context because its convergence is too slow. Instead we will make use of an interpolation formula at more dense points $n\pi/\beta$ with $\beta > \tau$, which leads to a much better convergence. Our approach is based on complex analysis techniques. It is different from the approach of Odlyzko in [22] and states results in a slightly different form.

Proposition 3 *Let $G(z)$ be a complex function of the complex variable $z = x + iy$ defined on the whole complex plane such that*

$$|G(z)| = O(e^{\tau|y|}), \qquad \tau > 0.$$

Let $\beta > \tau$, and let a kernel complex function $h(z)$ defined on the whole complex plane satisfying

$$h(0) = 1 \quad \textit{and} \quad |h(z)| = o(e^{\gamma|y|}), \qquad \tau + \gamma \leq \beta.$$

Then for any value of x we have

$$G(x) = \sum_{n=-\infty}^{+\infty} G\left(\frac{n\pi}{\beta}\right) h(x - n\pi/\beta) \frac{\sin(\beta x - n\pi)}{\beta x - n\pi}. \tag{19}$$

Proof : We start from the integral

$$\int G(z) \frac{\beta}{\sin(\beta z)} \frac{h(x-z)}{x-z}\, dz$$

where the contour is the circle $|z| = (n+1/2)\pi/\beta$ and letting $n \to \infty$, we obtain that its value is zero, thus the sum of all its residues is zero, which entails

$$G(x) \frac{\beta}{\sin(\beta x)} = \sum_{n=-\infty}^{+\infty} (-1)^n G\left(\frac{n\pi}{\beta}\right) \frac{h(x - n\pi/\beta)}{x - n\pi/\beta},$$

yielding the result. •

2.4.2 Application to interpolation of the function $F(t)$

We want to compute sums of the form

$$F(t) = \sum_{k=k_0}^{k_1} k^{-1/2+it}$$

thanks to an interpolation formula on a regular sampling of $F(t)$. Instead of interpolating directly $F(t)$, we interpolate an intermediate function which has a lower growth in the complex plane, defined by

$$G(t) = e^{-i\alpha t} F(t), \qquad \alpha = \frac{1}{2}(\log k_0 + \log k_1) \tag{20}$$

which satisfies

$$|G(z)| = O(e^{\tau|y|}), \qquad \tau = \frac{1}{2}(\log k_1 - \log k_0).$$

We then define the parameters $\beta = \lambda\tau$ (with $\lambda > 1$ being a fixed value, say $\lambda = 2$ for example) and $\gamma = \beta - \tau$. We choose the kernel function $h(t)$ in the form

$$h_M(t) = \left(\frac{\sin(\gamma t/M)}{\gamma t/M}\right)^M$$

with M a positive integer (we discuss this choice later). Now we are under condition of proposition 3, so we can use formula (19) to interpol $G(t)$ where we keep only the most significant terms of the summations, applied not to the values $G(n\pi/\beta)$ themselves but to approximations $\widehat{G}(n\pi/\beta)$ of these values obtained with the preceding algorithm of approximation of $F(t)$ on a regular grid. Our key formula consists in computing the value

$$\widehat{G}(x) = \sum_{n:|x-n\pi/\beta|<Mu0/\gamma} \widehat{G}\left(\frac{n\pi}{\beta}\right) h_M(x-n\pi/\beta)\frac{\sin(\beta x - n\pi)}{\beta x - n\pi}. \tag{21}$$

where $u_0 = 2.55$.

Error bound estimation of approximation formula. Bound estimation of approximation formula (21) is given in the following result.

Proposition 4 *Suppose that the approximations $\widehat{G}(n\pi/\beta)$ satisfy*

$$\left|\widehat{G}\left(\frac{n\pi}{\beta}\right) - G\left(\frac{n\pi}{\beta}\right)\right| < \epsilon$$

for integers n such that $|x - n\pi/\beta| < Mu_1/\gamma$ with $u_1 = 5.5$, and suppose that $M \geq 2$. Then the value $\widehat{G}(x)$, computed with formula (21), satisfies

$$\left|\widehat{G}(x) - G(x)\right| \leq A_{M,\beta,\gamma}\cdot\epsilon + B_{x,\beta,\gamma}\left(C_M + \frac{\gamma}{\beta}D_M\right) + 2\sqrt{k_1}\left(E_M + \frac{\gamma}{\beta}F_M\right).$$

where

$$A_{M,\beta,\gamma} = 2\sum_{0\leq n<Mu_0\beta/(\gamma\pi)} \frac{1}{(n+1/2)\pi} h_M\left(\frac{n\pi}{\beta}\right),$$

$$B_{x,\beta,\gamma} = \epsilon + \max_{n:|x-n\pi/\beta|<Mu_1/\gamma} |\hat{G}(n\pi/\beta)|,$$

and

$$C_M = 2I_M(Mu_0), \quad D_M = 2J_M(Mu_0), \quad E_M = 2I_M(Mu_1), \quad F_M = 2J_M(Mu_1)$$

with

$$I_M(u) = \frac{1}{\pi}\int_u^\infty \ell_M(t)\,dt, \quad J_M(u) = \frac{\ell_M(u)}{2} + \frac{1}{2}\int_u^\infty |\ell'_M(t)|\,dt, \quad \ell_M(t) = (\sin(t/M)/(t/M))^M/t$$

Proof : Starting from equation (21), we first write

$$\left|\widehat{G}(x) - G(x)\right| \leq X\cdot\epsilon + Y,$$

with

$$\begin{aligned} X &= \sum_{n:|x-n\pi/\beta|<Mu0} \left|h_M(x-n\pi/\beta)\frac{\sin(\beta x-n\pi)}{\beta x-n\pi}\right| \\ Y &= \sum_{n:|x-n\pi/\beta|\geq Mu0} \left|h_M(x-n\pi/\beta)\frac{\sin(\beta x-n\pi)}{\beta x-n\pi}\right| \cdot \left|G\left(x-\frac{n\pi}{\beta}\right)\right|. \end{aligned}$$

To obtain a bound on Y, we consider Euler-Maclaurin identity applied on a differentiable function $f(t)$ of the real variable t with rapid convergence to 0 at infinity

$$\sum_{n=n_0}^{+\infty} f(n) = \int_{n_0}^\infty f(t)\,dt + \frac{f(n_0)}{2} + \int_{n_0}^\infty \left(x - [x] - \frac{1}{2}\right) f'(t)\,dt \tag{22}$$

Starting from the inequality

$$\sum_{n\geq n_0} \left|h_M(x-n\pi/\beta)\frac{\sin(\beta x-n\pi)}{\beta x-n\pi}\right| \leq \frac{\gamma}{\beta}\sum_{n\geq n_0} \ell_M(n\pi\gamma/\beta - \gamma x)$$

and using identity (22) on the function $f(t) = \ell_M(t\pi\gamma/\beta - \gamma x)$ we obtain the upper bound

$$\sum_{n\ge n_0} \left| h_M(x - n\pi/\beta) \frac{\sin(\beta x - n\pi)}{\beta x - n\pi} \right| \le I_M\left(\frac{n_0\pi\gamma}{\beta} - \gamma x\right) + \frac{\gamma}{\beta} J_M\left(\frac{n_0\pi\gamma}{\beta} - \gamma x\right). \tag{23}$$

In the summation of Y the values of n for which $|x - n\pi/\beta| > Mu_1/\gamma$ satisfy $|G(n\pi/\beta)| < \sum_{k_0}^{k_1} k^{-1/2} < 2k_1^{1/2}$ and for other values of n, $|G(n\pi/\beta)| < \epsilon + |\widehat{G}(n\pi/\beta)| \le B_{x,\beta,\gamma}$. We deduce

$$Y \le B_{x,\beta,\gamma}\left(C_M + \frac{\gamma}{\beta} D_M\right) + 2\sqrt{k_1}\left(E_M + \frac{\gamma}{\beta} F_M\right).$$

Now we prove that $X \le A_{M,\beta,\gamma}$. Let y, $-1/2 < y \le 1/2$, such that $x = (n_0 + 1/2 + y)\pi/\beta$ for a (unique) integer n_0. If $|x - n\pi/\beta| < Mu_0/\gamma$, then n is necessarily of the form $n = n_0 - m$ with $-1 - m_{\max} < m < m_{\max}$, with $m_{\max} = Mu_0\beta/(\gamma\pi)$. Thus

$$X \le \sum_{m:-1-m_{\max}<m<m_{\max}} \left| h_M\left(\frac{(m + 1/2 + y)\pi}{\beta}\right) \frac{\sin((m + 1/2 + y)\pi)}{(m + 1/2 + y)\pi} \right|.$$

for convenience, we write this in the form

$$X \le \sum_{m:-1-m_{\max}<m<m_{\max}} g_m(y), \qquad g_m(y) = h_M\left(\frac{(m + y + 1/2)\pi}{\beta}\right) \frac{\cos(y\pi)}{(m + y + 1/2)\pi}.$$

For parity reasons, we have $g_{-m-1}(y) = g_m(-y)$, thus

$$X \le \sum_{0\le m<m_{\max}} c_m(y), \qquad c_m(y) = g_m(y) + g_m(-y).$$

The function $y \to h_M((m + y + 1/2)\pi/\beta)$ defined on $y \in [-1/2, 1/2]$ is decreasing when $0 \le m < m_{\max}$ thus its maximum is obtained for $y = -1/2$ and we deduce

$$c_m(y) \le h_M\left(\frac{m\pi}{\beta}\right) b_m(y), \qquad b_m(y) = \cos(y\pi)\left(\frac{1}{(m + y + 1/2)\pi} + \frac{1}{(m - y + 1/2)\pi}\right).$$

To prove that $X \le A_{M,\beta,\gamma}$, it now remains to prove that $b_m(y) \le b_m(0)$ for $-1/2 \le y \le 1/2$ when $0 \le m < m_{\max}$, and since $b_m(y)$ is even, it suffices to prove that $b_m'(y) \le 0$ for $0 \le y \le 1/2$. We have

$$b_m'(y) = \frac{(2m + 1)\cos(\pi y)}{\pi((m + 1/2)^2 - y^2)}\left(\frac{2y}{(m + 1/2)^2 - y^2} - \pi\tan(\pi y)\right),$$

thus on $[0, 1/2]$, $b_m'(y)$ has the sign of

$$a_m(y) = \frac{2y}{(m + 1/2)^2 - y^2} - \pi\tan(\pi y).$$

We have $a_m(y) \le a_0(y)$ and an easy study of the function $a_0(y)$ shows that $a_0(y) \le 0$ on $[0, 1/2]$, thus $b_m(y)$ is decreasing and the result follows. •

It is interesting to have the values of the involved parameters C_M, D_M, E_M and F_M for typical values of M. Here are some numerical upper values.

M	C_M	D_M	E_M	F_M
2	0.0165	0.033	4.39×10^{-3}	5.91×10^{-3}
4	4.65×10^{-4}	7.19×10^{-4}	3.84×10^{-5}	4.55×10^{-5}
6	1.71×10^{-5}	2.51×10^{-5}	4.62×10^{-7}	4.63×10^{-7}
8	6.86×10^{-7}	7.93×10^{-7}	6.26×10^{-9}	5.84×10^{-9}
10	2.87×10^{-8}	3.02×10^{-8}	8.97×10^{-11}	7.62×10^{-11}
12	1.24×10^{-9}	1.20×10^{-9}	1.33×10^{-12}	1.04×10^{-12}
16	2.37×10^{-12}	2.03×10^{-12}	3.08×10^{-16}	2.11×10^{-16}

We also give an idea of the values of other parameters in the proposition. Typical values of β and γ when computing around the 10^{12}-th zero of $\zeta(1/2 + it)$ are $\beta = 10$ and $\gamma = \beta/2$. For these values, we have

$$A_{4,\beta,\gamma} \simeq 1.922, \quad A_{8,\beta,\gamma} \simeq 2.111, \quad A_{12,\beta,\gamma} \simeq 2.225, \quad A_{16,\beta,\gamma} \simeq 2.307.$$

A typical value of k_1 in this zone is also $k_1 \sim 2 \times 10^5$; a value of $B_{x,\beta,\gamma}$ is very often smaller than 10, thus a typical bound obtained from the proposition in this range is

$$|\widehat{G}(x)-G(x)| \leq 1.922\epsilon+0.135 \quad (N=4), \qquad |\widehat{G}(x)-G(x)| \leq 2.111\epsilon+3.87\times 10^{-5} \quad (N=8),$$

$$|\widehat{G}(x)-G(x)| \leq 2.225\epsilon+3.9\times 10^{-8} \quad (N=12), \qquad |\widehat{G}(x)-G(x)| \leq 2.307\epsilon+6.5\times 10^{-11} \quad (N=16).$$

2.4.3 Practical considerations on interpolation implementation

The interpolation formula (21) needs a number of terms roughly equal to

$$Mu_0/\pi \times \beta/\gamma \simeq 0.81M\beta/\gamma = 0.81M\frac{\lambda}{\lambda-1}.$$

In order to decrease interpolation timing, one should use the smallest possible value of M that permits to reach a given error bound thanks to proposition 4. While isolating roots of the $Z(t)$ function, a typical need is getting the sign of $Z(t)$. In our implementation, each such evaluation is done first with the smallest value of $M = 2$, then the interpolation formula is used again for increasing higher values of M until the needed error bound (estimated with proposition 4) is reached. In this way, we highly decrease on average the cost of the use of the interpolation formula since the needed error is usually not too small. By comparison with a technique with a fixed value of $M = 16$ for example, we estimate that a factor of 3 or 4 has been saved in the interpolation computations. This is particularly important while evaluating $Z(t)$ at not too large height (say until the 10^{13}-th zero) because in this range, interpolation computations take approximately half of the total time.

Another trick we used to speed up the interpolation computations is the use of the iterative formula

$$\begin{aligned} \sin(x-(n+1)\pi/\beta) &= \sin(x-n\pi/\beta)c - \cos(x-n\pi/\beta)s \\ \cos(x-(n+1)\pi/\beta) &= \cos(x-n\pi/\beta)c + \sin(x-n\pi/\beta)s \end{aligned}$$

with $c = \cos(n\pi/\beta)$ and $s = \sin(n\pi/\beta)$, which permits to obtain the values of $\sin(x-(n+1)\pi/\beta)$ and $\cos(x-(n+1)\pi/\beta)$ from $\sin(x-n\pi/\beta)$ and $\cos(x-n\pi/\beta)$ faster than the direct evaluation of $\sin(x-(n+1)\pi/\beta)$, thus the successive values of the kernel function $h_M(x-n\pi/\beta)$ evaluate efficiently. This trick is the reason of our choice of the kernel function $h_M(x)$. In terms of implied speed of convergence, our kernel function is a little less efficient than the kernel function chosen by Odlyzko in [22]

$$h(u) = \frac{c}{\sinh c}\frac{\sinh(c^2-\epsilon^2u^2)^{1/2}}{(c^2-\epsilon^2u^2)^{1/2}}, \qquad \epsilon = \frac{\beta-\tau}{2},$$

with c a constant that was equal to 30 is most computations, but since the kernel function $h_M(u)$ evaluates faster, we globally save time.

Density of discretization. An important parameter is the value $\lambda > 1$ which is related to the density of discretization of the function $F(t) = \sum_{k_0 \leq k \leq k_1} k^{-1/2+it}$. Remember that the discretization is made with a regular step equal to

$$\delta = \frac{\pi}{\beta} = \frac{2\pi}{\lambda \log(k_1/k_0)},$$

whereas the cost of one interpolation from the discretization is proportional to $\lambda/(\lambda-1)$. In our implementation, the choice of λ was made in relation with the relative cost of the discretization of $F(t)$ and the interpolation. Until the 10^{13}-th zero for example, each of the discretization or interpolation part take approximately half of the time, and the value $\lambda = 2$ was chosen. As the height increases, the relative cost of the discretization becomes bigger and a smaller value of $\lambda > 1$ was taken. Around the 10^{22}-th zero for example, we have chosen the value $\lambda = 6/5$.

Another way to decrease the needed density of discretization was to increase the value of k_0. Until the 10^{13}-th zero, we have taken $k_0 = 6$. Around the 10^{22}-th zero, the choice $k_0 = 30$ was made. These values have been chosen with a heuristic based on rough timing experiments and are probably not optimal.

2.4.4 More accurate approximation of $Z(t)$

In some rare cases (especially between two very close zeros), it may be useful to compute $Z(t)$ more accurately to separate zeros of it. While using the band-limited function interpolation technique to compute $Z(t)$, this imply to compute the multi-evaluation of $F(t)$ at a high precision, which would have some bad impact on the total timing. We preferred to use a different technique : when the required precision on $Z(t)$ was higher than the available precision issued from the band-limited function interpolation, we computed $Z(t)$ directly with the classic direct use of the Riemann-Siegel formula (6). In this way, the Odlyzko-Schönhage algorithm is used with reasonable parameters so that almost all $Z(t)$ evaluations are computed precisely enough, thus controlling the total timing.

Nearly all the time in the direct computation of $Z(t)$ from the Riemann-Siegel is spent in the summation

$$\sum_{n=1}^{m} \frac{\cos(\theta(t) - t\log n)}{\sqrt{n}}, \qquad m = \left\lfloor \sqrt{\frac{t}{2\pi}} \right\rfloor. \tag{24}$$

Instead of using directly this form, we used an Euler-product form of the value

$$F(t) = \sum_{k=1}^{m} k^{-1/2+it},$$

and taking the real part of $e^{-i\theta(t)}F(t)$ gives the expected value. An Euler-product form of $F(t)$ (as described in [22]) is obtained by considering the product $P = 2 \times 3 \times \cdots \times p_h$ of the first h primes (with h small in the practice, say $h \leq 4$) and by considering the set Q of integers all of whose prime factors are smaller than p_h. It permits to reduce the summation to integers k relatively prime to P, with the formula

$$F(t) = \sum_{\substack{0<k\leq m \\ (k,P)=1}} k^{-1/2+it} s(m/k), \qquad s(a) = \sum_{\substack{0<\ell\leq a \\ \ell\in Q}} \ell^{-1/2+it}. \tag{25}$$

Since the $s(m/k)$ for $0 < k \leq m$ are the same for many k the summation above is much more efficient than the direct summation for $F(t)$, and the asymptotically expected factor is at most $\phi(P)/P = (1-1/2)(1-1/3) \times \cdots \times (1-1/p_h)$.

We implemented this technique, and as can be seen on the table of figure 2, we nearly save a factor 3 compared to the direct use of the summation (24) at large height.

Height index	Classic	$p_h = 2$	$p_h = 3$	$p_h = 5$	$p_h = 7$
10^{14}	1.19	0.92	0.82	0.76	0.77
10^{16}	7.86	5.98	4.91	4.45	4.36
10^{18}	60.48	39.92	32.18	28.8	27.6
10^{20}	473.50	299.79	238.90	210.08	197.31
10^{22}	4050.86	2501.24	1951.19	1668.51	1577.33
10^{24}	36909.16	22361.02	16842.16	14840.87	13584.88

Figure 2: Comparison of timing of a single evaluation of $Z(t)$ between classic summation (24) and faster formula (25) with 1, 2, 3 or 4 primes. The first column contains a zero index n near the evaluation (so evaluation was made near $t = g_n$ where g_n is the n-th Gram point). The timings are in seconds, computation was made on a Pentium m 1.6 Ghz.

In our program, notice that we always performed at least one very accurate evaluation of $Z(t)$ in this way per range where Odlyzko-Schönhage algorithm was used, in order to check correctness of the computation. Other needed very accurate evaluation of $Z(t)$ to separate zeros of it were very rare (say a few units per billion zero on average).

3 Verification of the Riemann Hypothesis until the 10^{13}-th zero

Our first family of computations consisted in verifying that until the 10^{13}-th zero, all zeros of the Riemann Zeta function lie on the critical line $\Re(s) = \frac{1}{2}$. In order to verify the RH, we

do not need precise computations abscissa of zeros, counting them by finding enough changes of sign of $Z(t)$ is sufficient. At higher ranges (see next section) we computed in addition the abscissa of zeros of Zeta (as Odlyzko did in [22]) in order to collect more statistics. The advantage of verifying the RH only relies in computational timing : while verifying the RH until the 10^{13}-th zero, we needed on average less than 1.2 evaluations of the $Z(t)$ function per zero to locate them, whereas to be able to compute abscissa of zeros in this range, we would have needed at least five or six times more evaluations of $Z(t)$. Since the verification of the RH on the first 10^{13} zeros took a little more than one year and a half of a single modern computer (and the same for verification), timing was a crucial aspect that motivates our choice of restricting on RH verification only in this range.

3.1 Gram points, Gram blocks

We recall the formula, already seen in (4)

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

with $\theta(t)$ and $Z(t)$ real valued functions. When $t \geq 7$, the $\theta(t)$ function is monotonic increasing; for $n \geq -1$, the n-th *Gram point* g_n is defined as the unique solution > 7 to

$$\theta(g_n) = n\pi.$$

The Gram points are as dense as the zeros of $\zeta(s)$ but are much more regularly distributed. The asymptotic formula (5) of $\theta(t)$ entails that

$$g_n \sim \frac{2n\pi}{\log(n)}.$$

Starting with the approximation

$$\frac{1}{2}Z(g_n) \approx \cos\theta(g_n) + \sum_{2\leq n\leq m} \frac{\cos(\theta(g_n) - g_n\log n)}{\sqrt{n}} = (-1)^n\left(1 + \sum_{2\leq n\leq m}\frac{\cos(g_n\log n)}{\sqrt{n}}\right)$$

we see that the leading term in expansion of $Z(g_n)$ has the sign of $(-1)^n$. This motivates *Gram's law*, which is the empirical observation that $Z(t)$ usually changes its sign in each *Gram interval* $G_n = [g_n, g_{n+1})$, that holds in Gram's computations [8]. It is known that this law fails infinitely often, but it is true a large proportion of cases (more than 70% at not too large height, more than 66% at larger height).

A Gram point g_n is called *good* if $(-1)^n Z(g_n) > 0$, and *bad* otherwise. A *Gram block* is an interval $[g_n, g_{n+k})$ such that g_n and g_{n+k} are good Gram points and g_{n+1}, ..., g_{n+k-1} bad Gram points. A Gram block is denoted by the notation $a_1a_2\dots a_k$ where k is called the *length* of the Gram block, and a_i denote the number of roots of $Z(t)$ in the Gram interval $[g_{n+i-1}, g_{n+i})$. As of today, no Gram interval has been found with more than 5 zeros, thus the notation is unambiguous.

Most Gram blocks of length k contain exactly k zeros (see below for the discussion about *violations of Rosser rule*); such Gram blocks are called *regular*. Regular Gram blocks must have a pattern of one of the following three forms

$$21\dots10, \qquad 01\dots12, \qquad 01\dots131\dots10.$$

where the notation $1\dots1$ refers to any string of consecutive 1, including zero length string.

Violations of Rosser rule

Another important empirical rule, known as *Rosser rule*, states that Gram blocks of length k contains at least k zeros. In fact, if this rule would be true, then it would imply the RH, which would imply that each Gram block of length k contains exactly k zeros. Violations of Rosser rule are quite rare. The first occurrence of such a violation is at index $n = 13,999,825$ and until the 10^{13}-th zero, there is just a little more than 32 violations of Rosser rule on average per million zeros. It is known that Rosser rule has to fail infinitely often.

If a Gram block of length k is an exception to Rosser rule, then its pattern of zeros must be of the form $01\ldots10$. To describe the exception, we must specify where the two missing zeros are. We use the notation

$$kXa_1\ldots a_m, \qquad X = L \text{ or } R \tag{26}$$

to describe an exception on a Gram block of length k where the missing zeros are on the left (for $X = L$) or on the right (for $X = R$), the pattern containing the missing zeros being $a_1\ldots a_m$ (moreover, this pattern is the smallest union of Gram block adjacent to the exception that contains the missing zeros). For example, $3L04$ denotes a violation of Rosser rule on a Gram block of length 3, the missing zeros being at its left. Globally, the pattern of zeros expressed by the notation is 04010.

We refer to (26) as the type of violation of Rosser rule, the value m being called the *length* of the excess block. Notice that the notation used for exception to Rosser rule is not unambiguous. When several contiguous violation of Rosser rule exists, they may overlap or missing zeros can be in the same Gram interval. When such very rare situation occurs (we found just it three times until the 10^{13}-th zero), we simply use the notation $Ma_1\ldots a_\ell$ where the pattern $a_1\ldots a_\ell$ is made of the minimal contiguous Gram blocks containing at least one violation to Rosser rule, and all the missing zeros. For example, the pattern $M00500$, first encountered at gram index $n = 3,680,295,786,518$, denotes a situation with two violations of Rosser rule ("00" and "00", Gram blocks with missing zeros) and a single Gram interval containing all the missing zeros (pattern "5").

3.2 Approach used to locate all zeros

To verify Riemann Hypothesis until the 10^{13}-th zero, we made use of the Odlyzko Schönhage approach presented in section 2, associated to the first variant presented in subsection 2.3.1, to compute the Riemann-Siegel Z-function. Evaluating the Z-function permits to find changes of sign of $Z(t)$, thus to locate some zeros of it, but may be not enough to verify the Riemann Hypothesis. We made use of the Turing's method presented below, that permits, once "enough" roots have been located on the critical line in a certain zone, to prove that all roots of $\zeta(s)$ lie on the critical line in this zone.

Regular Gram points and Turing's method

It is known that the number of zeros of $\zeta(\sigma + it)$ for $0 < \sigma < 1$ and $0 < t < T$, denoted by $N(T)$, satisfies

$$N(T) = 1 + \frac{\theta(T)}{\pi} + S(T),$$

with $S(t) = \frac{1}{\pi}\arg\zeta(\frac{1}{2} + it)$. Von Mangoldt proved that $S(T) = O(\log T)$. In 1924, Littlewood proved that $\int_0^T S(t)\,dt = O(\log T)$. Thus Littlewood result states that the average value of $S(t)$ is zero, whereas Von Mongoldt result proves that $S(t)$ is not too large. Since the zero value of $S(T)$ is the "standard", we say that a Gram point g_n is *regular* if $S(g_n) = 0$, which is equivalent to $N(g_n) = n + 1$. A regular Gram point is a good Gram point in the sense of the above definition.

One of the key point in RH verification is the ability to find regular Gram points. Once regular Gram points are found, it suffices to check that between them, the expected number of change of sign of $Z(t)$ occurs, in order to numerically check the RH in this zone.

A possibility to find regular Gram points would be to use Backlund's method, which consists in proving when possible that the real part of $\zeta(\sigma + iT)$ is never zero for $\frac{1}{2} \le \sigma \le \frac{3}{2}$. The disadvantage of this method is that it requires evaluations of $\zeta(s)$ out of the critical line. To overcome this difficulty we used Turing's method (see [7] for a complete description) which permits to prove that some Gram points are regular, by evaluating $Z(t)$ only. Turing approach is briefly described as follows : when locating zeros of $Z(t)$, a sequence (h_n) is found such that $(-1)^n Z(g_n + h_n) > 0$, the sequence $(g_n + h_n)$ being increasing, with h_n small and zero whenever possible. Turing showed that if $h_m = 0$ and if values of h_n for n near m are not too large, then g_m is a regular Gram point. More precisely, Turing obtained a quantitative version of Littlewood estimate

$$\left|\int_{t_1}^{t_2} S(t)\,dt\right| \le 2.30 + 0.128\log\frac{t_2}{2\pi},$$

from which he was able to prove that for all $k > 0$, we have

$$-1 - \frac{2.30 + 0.128 \log \frac{g_m}{2\pi} + \sum_{j=1}^{k-1} h_{m-j}}{g_m - g_{m-k}} \leq S(g_m) \leq 1 + \frac{2.30 + 0.128 \log \frac{g_{m+k}}{2\pi} + \sum_{j=1}^{k-1} h_{m+j}}{g_{m+k} - g_m}.$$

When $k > 0$ is found for which those estimates give $-2 < S(g_m) < 2$, then since for parity reason, $S(g_m)$ is always an even integer, we prove that $S(g_m) = 0$.

Locating zeros

To locate zeros of $Z(t)$ in a zone, we followed a heuristic approach based on the zeros statistics, in order to decrease as much as possible the number of evaluations needed to locate all zeros of $Z(t)$. The underlying idea was that each Gram interval is associated to a unique zero of $Z(t)$ in this Gram interval or very close to it. We first computed the value of $Z(t)$ at each Gram point (in fact the sign of $Z(t)$ is sufficient). This permitted to compute Gram blocks, and then missing changes of sign in Gram blocks were searched in most statistical frequent places order, with easy heuristics. Among the search techniques, we used the observation that under the RH, $|Z(t)|$ cannot have any relative minima between two consecutive zeros of $Z(t)$ (see [7] for a proof of it that holds under the RH for t not too small). This property gives a very powerful search method : if $a < b < c$ and $|Z(b)| < |Z(a)|$, $|Z(b)| < |Z(c)|$ with $Z(a)$, $Z(b)$ and $Z(c)$ having the same sign, then we should have at least two zeros between a and c (under the RH).

This simple approach permitted to find the pattern of most Gram blocks. When this simple approach did not work, it meant that we could have a violation of Rosser rule, so we looked in a neighborhood if a missing change of sign could occur. If yes, we had found an exception to Rosser rule, otherwise, we tried much aggressive techniques to look for the missed change of sign.

This approach permitted to locate quite easily most zeros of $Z(t)$ and since aggressive searches were performed only a very small fraction of the time, the average number of evaluations of $Z(t)$ per zero was only 1.193 until the 10^{13}-th zero, which is nearly optimal. As expected also, the average number of evaluations of $Z(t)$ per zero needed increases with the height of zeros : until zero number 2×10^9 for example, this average number of evaluations was just 1.174. Thus in a certain manner, and as "measured" by our internal indicator, the "complexity" of $Z(t)$ increases when t gets large.

3.3 Statistical data in RH verification until the 10^{13}-th zero

We now present statistical data that we generated while verifying the RH on the first 10^{13} zeros.

3.3.1 Computation information

Computation was distributed on several machines and in total, it took the equivalent of 525 days of a single modern computer in 2003 (Pentium 4 processor 2.4 Ghz), thus 220,000 zeros checked per second on average. Required memory was also classic for such computers (256 Mo were sufficient). The computation was made in different periods between April of 2003 and September of 2003, using spare time of several machines, and could be performed thanks to Patrick Demichel who could access to computer spare time and managed this distributed computation.

Computational results for the RH verification on the first 10^{13} zeros are based on proved inequality estimates of this paper. The computation was checked in too ways : first, after each application of Odlyzko-Schönhage algorithm in a certain range, evaluations at two different abscissas in this range were compared with the classic direct use of Riemann-Siegel formula. This check is in fact quite global, in the sense that a single evaluation using Odlyzko-Schönhage technique depends on all the result of the multi-evaluation of $f(z)$ (see section 2.3). Another check was done that consisted in re-launching all the computation with different parameters (we changed slightly a certain number of free parameters of our approach), and by checking that the same number of zeros were found in the same zones. This second check took an additional one year and a half time (equivalent timing of one modern computer).

3.3.2 Statistics

First of all, no exception to Riemann Hypothesis were found on the first 10^{13} zeros. As already discussed before, computations until the 10^{13}-th zero essentially consisted in computing the number of zeros in each Gram interval and not the zeros themselves.

Particular situations. Some particular situations have been observed that did not appear historically in previous ranges of RH verification.

- One Gram interval has been found which contains 5 zeros of the Zeta function (at index 3,680,295,786,520). All other Gram interval contained 4 zeros or less.
- Largest Gram block length found is 13, and first occurrence of Gram block of size 11, 12 and 13 have been found :
  - The first Gram block of size 11 is at Gram index 50,366,441,415.
  - The first Gram block of size 12 is at Gram index 166,939,438,596.
  - The first Gram block of size 13 is at Gram index 1,114,119,412,264.
- Three times, we found pairs of close violations of Rosser rule for which the missing zeros were merged in common Gram intervals :
  - The pattern $M00500$ was found at Gram index 3,680,295,786,518.
  - The pattern $M002400$ was found at Gram index 4,345,960,047,912.
  - The pattern $M004200$ was found at Gram index 6,745,120,481,067.
- The smallest known normalized difference δ_n between consecutive roots γ_n and γ_{n+1} of $Z(t)$, defined by
$$\delta_n = (\gamma_{n+1} - \gamma_n)\frac{\log(\gamma_n/(2\pi))}{2\pi},$$
was found at $\gamma_n = 1034741742903.35376$ (for index $n = 4,088,664,936,217$), with a value of $\delta \approx 0.00007025$. Non normalized difference between those roots is equal to $0.00001709\ldots$.

Closest found pairs of zeros. Close pairs of zeros are interesting because it corresponds to cases for which the RH is "nearly" false. Verifying the RH in zones where two zeros are very close is a particular situation (often described as the "Lehmer's phenomenon" since Lehmer was the first to observe such situations, see [7] for more details) that was detected in our implementation thanks to some simple heuristics. In such detected situations, we computed effectively the close zeros and their difference. In this way, even if all zeros abscissas were not computed, we were able to find a large number of close zeros. As the technique used to find them is based on a heuristic, some pairs of close zeros may have been missed. However, we estimate that for very close zeros (say those for which $\delta_n < 0.0002$) most (and probably all) pairs of close zeros have been found.

We recall our notations : the value γ_n denotes the abscissa of the n-th zero, the value δ_n denotes the normalized spacing at the n-th zero
$$\delta_n = (\gamma_{n+1} - \gamma_n)\frac{\log(\gamma_n/(2\pi))}{2\pi}.$$

To give an idea of the number of close pairs of zeros we may have missed with our simple heuristic, we recall that under the GUE hypothesis (see section 4.2), until the N-th zero, the expected number of pairs of consecutive zeros for which δ_n is less than a small value δ is asymptotic to
$$E(\delta, N) = N\frac{\pi^2}{9}\delta^3 + O(N\delta^5)$$
(see [22, p. 30] for example). In our case, $N = 10^{13}$, so $E(0.0001, N) \simeq 10.96$ and we found exactly 13 zeros such that $\delta_n < 0.0001$. We have $E(0.0002, N) \simeq 87.73$ and we found exactly 86 zeros such that $\delta_n < 0.0002$. So for such small values of δ_n, we are close to the GUE expectations. Higher values of δ show that our heuristic probably missed close pairs of zeros. For example, we found 1240 zeros for which $\delta_n < 0.0005$ whereas the GUE hypothesis expects $E(0.0005, N) \simeq 1370.78$ such zeros.

The table below lists statistics relative to all closest found pairs of zeros for which $\delta_n < 0.0001$ (it may not be extensive). The last column gives values of ϵ_n which is an error upper bound on the value $\gamma_{n+1} - \gamma_n$.

δ_n	$\gamma_{n+1} - \gamma_n$	γ_n	n	ϵ_n
0.00007025	0.00001709	1034741742903.35376	4,088,664,936,217	1.45E-08
0.00007195	0.00001703	2124447368584.39307	8,637,740,722,916	4.59E-08
0.00007297	0.00001859	323393653047.91290	1,217,992,279,429	1.29E-08
0.00007520	0.00001771	2414113624163.41943	9,864,598,902,284	1.69E-08
0.00008193	0.00002420	10854395965.14210	3,5016,977,795	4.46E-09
0.00008836	0.00002183	694884271802.79407	2,701,722,171,287	1.74E-08
0.00008853	0.00002133	1336685304932.84375	5,336,230,948,969	5.88E-08
0.00008905	0.00002127	1667274648661.65649	6,714,631,699,854	4.53E-08
0.00008941	0.00002210	693131231636.82605	2,694,627,667,761	3.19E-08
0.00009153	0.00002158	2370080660426.91699	9,677,726,774,990	1.38E-07
0.00009520	0.00002495	161886592540.99316	591,882,099,556	1.73E-08
0.00009562	0.00002367	664396512259.97949	2,578,440,990,347	4.09E-08
0.00009849	0.00002756	35615956517.47854	121,634,753,454	7.63E-09

Statistics on Gram blocks. First, statistics on Gram blocks are not completely rigorous since when the evaluation of $Z(t)$ at a Gram point $t = g_n$ gave a too small value compared to the error bound (so we were not able to decide the sign of $Z(t)$), we changed a little bit the value of g_n (this trick does not affect the RH verification). Thus the statistics in the table below contains a very small proportion of errors. However, it gives a very good idea of the repartition of Gram block until the 10^{13}-th zero.

Below is a table that contains the number of Gram blocks found between zero #10,002 and $\#10^{13} + 1$. Gram block of type I are those with pattern $21 \dots 10$ (except for Gram block of length 1 where it is just the pattern 1), type II corresponds to pattern $01 \dots 12$, type III corresponds to pattern $01 \dots 131 \dots 10$.

Length of Gram block	type I	type II	type III
1	6,495,700,874,143		
2	530,871,955,423	530,854,365,705	0
3	137,688,622,847	137,680,560,105	12,254,585,933
4	41,594,042,888	41,590,457,599	4,713,328,934
5	11,652,547,455	11,651,049,077	1,677,257,854
6	2,497,894,288	2,497,449,668	582,216,827
7	335,440,093	335,304,175	186,090,022
8	22,443,772	22,427,099	47,938,397
9	552,727	553,654	8,667,047
10	3,137	3,114	1,081,811
11	0	1	93,693
12	0	0	4,967
13	0	0	122

Violations of Rosser rules In our verification of the RH until the 10^{13}-th zero, we found 320,624,341 violations of Rosser rules (here again, the statistic is not completely rigorous, as explained above). So we have an average of about 32.06 violations of Rosser rules per million zero. Rosser rules fails more and more often with the height, as shown in the table below which contains the number of violations of Rosser rule (VRR) in different ranges of our computation. Table below also shows that the number of type of VRR (see (26)) also increases with the height.

Zero index range	Number of VRR	Number of types of VRR
$0 - 10^{12}$	20007704	121
$10^{12} - 2 \times 10^{12}$	26635210	147
$2 \times 10^{12} - 3 \times 10^{12}$	29517546	160
$3 \times 10^{12} - 4 \times 10^{12}$	31476295	159
$4 \times 10^{12} - 5 \times 10^{12}$	32970500	172
$5 \times 10^{12} - 6 \times 10^{12}$	34192167	186
$6 \times 10^{12} - 7 \times 10^{12}$	35211583	179
$7 \times 10^{12} - 8 \times 10^{12}$	36108621	184
$8 \times 10^{12} - 9 \times 10^{12}$	36893156	192
$9 \times 10^{12} - 10^{13}$	37611559	193

In total, we found 225 different types of violations of Rosser rules. The table below shows the most frequent encountered types on all the first 10^{13} zeros.

Type of VRR	Number of occurrence	Frequency
2L3	77146526	24.061%
2R3	77119629	24.053%
2L22	43241178	13.487%
2R22	43232794	13.484%
3L3	19387035	6.047%
3R3	19371857	6.042%
2L212	7992806	2.493%
2R212	7986096	2.491%
3L22	6644035	2.072%
3R22	6641646	2.071%
4L3	2326189	0.726%
4R3	2321577	0.724%
2R2112	716337	0.223%
2L2112	714976	0.223%
2L032	614570	0.192%
2R230	614407	0.192%
3L212	527093	0.164%
3R212	524785	0.164%
4L22	366441	0.114%
4R22	365798	0.114%
2L04	363861	0.113%
2R40	363174	0.113%

Among the more rare types, we found for example the patterns 7R410 (occurs once at index 2,194,048,230,633) and 2L011111114 (occurs twice). We found 17 different types that were encountered only once, 11 that were encountered just twice.

Large values of $Z(t)$. The largest value of $|Z(t)| = |\zeta(1/2 + it)|$ found until the 10^{13}-th zero was for $t = 2381374874120.45508$ for which we have $|Z(t)| \simeq 368.085$, but since no special treatment was made to find biggest values of $|Z(t)|$, bigger values probably exist and were missed in our computation.

4 Zeros computations of the Zeta function at very large height

The second type of computations we performed consisted in computing a large number of zeros at large height. This time, we did not restrict on RH verification, but we also approximated quite precisely all zeros in our ranges in order to get a larger collection of statistics. Our main goal here was to test the *GUE hypothesis* which conjectures a certain distribution of the spacing between the zeros of the zeta function. The GUE hypothesis is discussed below in section 4.2. As we will see, computations show a good agreement with this conjecture, and

moreover we have observed empirically the speed of the convergence toward the conjectured distribution.

We considered a collection of heights 10^n for all integer values of n between 13 and 24, and a set of two billion zeros was computed at each height. Our largest reached height 10^{24} is larger than the height reached before by Odlyzko in unpublished result (Odlyzko computed 50 billion zeros at height 10^{23}).

4.1 Approach to compute zeros at large height

Since the abscissas considered here are very large, we made use of the Greengard-Rokhlin algorithm (see section 2.3.2) to compute values of the $Z(t)$ function. As discussed in 2.3.3, this method is much more efficient in the large height context.

The approach consisted in locating first the zeros as done in the first family of computations (see section 3.2), and then to perform a few iterations to approximate the abscissa of each zero with a precision of about 10^{-9}. On average, we needed about 7.5 evaluations of the $Z(t)$ function per zero here, while only about 1.2 evaluations were needed per zero in the RH verification until the 10^{13}-th zero.

Managing precision control

We are dealing here with very large heights (largest height computations were made around the 10^{24}-th zero), making the precision management one of the key success factors. Since double precision storage only was used (thus a little more than 15 decimal digits of precision), the error bound on a sum like

$$\sum_{k=k_0}^{k_1} k^{-1/2} \cos(t \log k)$$

would be of the form

$$E = \sum_{k_0}^{k_1} \epsilon_k k^{-1/2}$$

where for all k, ϵ_k is the imprecision on the value of $\cos(t \log k)$. Due to our techniques in the computation of $t \log k$ modulo 2π, a typical precision for ϵ_k is $|\epsilon_k| < \epsilon$, say with $\epsilon = 10^{-12}$. Without any additional information, we would only deduce that the total error E is bounded by

$$|E| \leq \sum_{k_0}^{k_1} |\epsilon_k| k^{-1/2} \leq \epsilon \sum_{k_0}^{k_1} k^{-1/2} \sim 2k_1^{1/2}\epsilon. \tag{27}$$

Around the 10^{24}-th zero, the value of k_1 is around $k_1 \simeq 1.3 \times 10^{11}$, thus we would obtain $|E| < 0.73 \times 10^{-6}$. This error bound is too large in our context, since separations of some zeros frequently needs a higher precision. For performance reason, we obviously did not want to rely on multiprecision operations, so we needed to deal with our double precision storage. Since in the practice, the true error E is much smaller than (27), we preferred to use a statistically reasonable error bound. Based on the observation that ϵ_k can be seen as independent variables, taking any values between $-\epsilon$ and $+\epsilon$, the typical error bound on E has the form

$$|E| \leq \left(\sum_{k_0}^{k_1} (\epsilon k^{-1/2})^2 \right)^{1/2} \sim \epsilon (\log k_1)^{1/2}. \tag{28}$$

Around the 10^{24}-th zero, this gives an error of the order 5×10^{-12} instead of 0.73×10^{-6}, which is closer to the true error bound, and which gives enough precision to separate zeros of Zeta. Following this observation, in our large height context, anytime we add a sum of terms of the form

$$S = \sum_k f_k$$

each f_k having an imprecision bounded by ϵ_k, we expected for the error on S and error of the order

$$|E| = \left(\sum_k \epsilon_k^2 \right)^{1/2}$$

(multiplied by a certain security factor, equal to 10 in our implementation) instead of the very pessimistic classic bound $|E| \leq \sum_k \epsilon_k$. This kind of statistical error control was also used by Odlyzko in his large height computations (see [22, Section 4.6]). Thanks to this important error management, we have been able to control the precision with enough accuracy to separate zeros in our large height computations. Even if we did not use rigorous error bound but rather statistical one in our implementation, the computational results are thought to be accurate, as discussed below.

Computational correctness

Especially at very large height, controlling correctness of computational results is fundamental. Several ways to check the computational results were used. First, after each application of Odlyzko-Schönhage algorithm in a certain range, an evaluation at a certain abscissa in this range was compared with the classic direct use of Riemann-Siegel formula. As observed earlier, this check validates in some sense all the result of the multi-evaluation of $f(z)$ (see section 2.3). Another check was done that consisted in re-launching all the computation in some ranges with different parameters (some free parameters in multi-evaluation techniques were changed) and by computing the difference between computed zeros (see table in section 4.3.5 below for more information). Finally, as observed by Odlyzko in [22] the RH verification itself is also a check since a slight error anywhere in the evaluation of $Z(t)$ may lead to RH violations.

4.2 The GUE hypothesis

While many attempts to prove the RH had been made, a few amount of work has been devoted to the study of the distribution of zeros of the Zeta function. A major step has been done toward a detailed study of the distribution of zeros of the Zeta function by Hugh Montgomery [19], with the *Montgomery pair correlation conjecture*. Expressed in terms of the normalized spacing $\delta_n = (\gamma_{n+1} - \gamma_n)\frac{\log(\gamma_n/(2\pi))}{2\pi}$, this conjecture is that, for $M \to \infty$

$$\frac{1}{M}\#\{(n,k) : 1 \leq n \leq M, k \geq 0, \delta_n + \cdots + \delta_{n+k} \in [\alpha,\beta]\} \sim \int_\alpha^\beta 1 - \left(\frac{\sin \pi u}{\pi u}\right)^2 du. \quad (29)$$

In other words, the density of normalized spacing between non-necessarily consecutive zeros is $1-(\sin(\pi u)/\pi u)^2$. It was first noted by the Freeman Dyson, a quantum physicist, during a now-legendary short teatime exchange with Hugh Montgomery, that this is precisely the pair correlation function of eigenvalues of random hermitian matrices with independent normal distribution of its coefficients. Such random hermitian matrices are called the Gauss unitary ensemble (GUE). As referred by Odlyzko in [20] for example, this motivates the *GUE hypothesis* which is the conjecture that the distribution of the normalized spacing between zeros of the Zeta function is asymptotically equal to the distribution of the GUE eigenvalues. Under this conjecture, we might expect a stronger result than (29), that is

$$\frac{1}{M}\#\{(n,k) : N{+}1 \leq n \leq N{+}M, k \geq 0, \delta_n{+}\cdots{+}\delta_{n+k} \in [\alpha,\beta]\} \sim \int_\alpha^\beta 1{-}\left(\frac{\sin \pi u}{\pi u}\right)^2 du \quad (30)$$

with M not too small compared to N, say $M \geq N^\nu$ for some $\nu > 0$. Another result under the GUE hypothesis is about the distribution of the δ_n itself,

$$\frac{1}{M}\#\{n : N+1 \leq n \leq N+M, \delta_n \in [\alpha,\beta]\} \sim \int_\alpha^\beta p(0,u)\, du \quad (31)$$

where $p(0,u)$ is a certain probability density function, quite complicated to obtain (see (32) for an expression of it). As reported by Odlyzko in [22], we have the Taylor expansion around zero

$$p(0,u) = \frac{\pi^2}{3}u^2 - \frac{2\pi^4}{45}u^4 + \frac{\pi^6}{315}u^6 + \cdots$$

which under the GUE hypothesis entails that the proportion of δ_n less than a given small value δ is asymptotic to $(\pi^2/9)\delta^3 + O(\delta^5)$. Thus very close pairs of zeros are rare.

Previous computations by Odlyzko [20, 21, 22, 23], culminating with the unpublished result of computations at height 10^{23}, were mainly dedicated to the GUE hypothesis empirical

verifications. As observed by Odlyzko using different statistics, agreement is very good. Our goal here is to compute some of the statistics observed by Odlyzko relative to the GUE hypothesis, at height at each power of ten from 10^{13} to 10^{24}. Our statistics, systematically observed at consecutive power-of-ten heights, are also oriented to observe empirically how the distribution of the spacing between zeros of the Zeta function converges to the asymptotic expectation.

4.3 Statistics

4.3.1 Computation information

Computation was launched on spare time of several machines. Zeros were computed starting roughly from the 10^n-th zero for $13 \leq n \leq 24$. An amount of roughly 2×10^9 zeros was computed at each height. Physical memory requirement was less than 512 Mo, and in the case of large height (for height 10^{23} and 10^{24}), an amount of 12 Go of disk space was necessary.

Table below gives some indications of timing and the value of R used (see section 2.3). It is to notice that due to the difficulty to have some long spare times on the different computers used, we adapt values of R that is why it is not monotonous. Due also to different capacities of the machines, the amount of used memory were not always identical. Timings are not monotonous also but at least, the table is just here to fix idea about cost. Third and fourth columns relates to offset index, so the value 10^n should be added to have the absolute index of first or last zero. First and last zeros are always chosen to be Gram points proved regular with Turing's method (see section 3.2).

Height	Total timing in hours	offset index of first zero	offset index of last zero	Value of R
10^{13}	33.1	1	2×10^9	16777216
10^{14}	35.0	3	2×10^9	16777216
10^{15}	38.3	0	$2 \times 10^9 - 1$	8388608
10^{16}	49.5	1	$2 \times 10^9 - 1$	16777216
10^{17}	46.9	0	2×10^9	16777216
10^{18}	81.6	1	$2 \times 10^9 - 1$	33554432
10^{19}	65.9	0	$2 \times 10^9 + 1$	33554432
10^{20}	87.8	4	$2 \times 10^9 - 1$	33554432
10^{21}	139.9	0	$2 \times 10^9 - 1$	33554432
10^{22}	151.5	2	$2 \times 10^9 - 1$	134217728
10^{23}	219.0	100	$2 \times 10^9 - 1$	268435456
10^{24}	326.6	0	$2 \times 10^9 + 47$	268435456

Additional timing information relates to the efficiency of our implementation, using Odlyzko-Schönhage algorithm, compared to the direct evaluation of the Zeta function using Riemann-Siegel formula (6). At height 10^{24} for example, two third of the total time was spent in the multi-evaluation of $F(t)$ (see section 2.3) and a single evaluation of Zeta using the direct optimized evaluation of Riemann-Siegel formula (25) (we used it for verification) took 5% of the total time. So globally, the time needed to compute all the 2×10^9 zeros at height 10^{24} in our implementation is approximately equal to 20 evaluations of Zeta using the direct Riemann-Siegel formula. This proves the very high efficiency of the method.

4.3.2 Distribution of spacing between zeros

Statistics were done to observe numerically the agreement of asymptotic formulas (31) and (30). A first step is to be able to derive an expression for the density probability function $p(0,t)$. In [20], Odlyzko made use of a technique from Mehta and des Cloizeaux [17], that requires explicit computation of eigenvalues and eigenvectors of an integral operator on an infinite dimension functions space, then a complicated expression with infinite products and sums depending on these eigenvalues. As suggested by Odlyzko to the author, more modern and easier techniques are available today and Craig A. Tracy kindly transmitted those to the author (see [32]). The approach relies on the identity

$$p(0,s) = \frac{d}{ds^2}\left[\exp\left(\int_0^{\pi s} \frac{\sigma(x)}{x}\,dx\right)\right] \tag{32}$$

where σ satisfies the differential equation

$$(x\sigma'')^2 + 4(x\sigma' - \sigma)(x\sigma' - \sigma + (\sigma')^2) = 0$$

with the boundary condition $\sigma(x) \sim -x/\pi$ as $x \to 0$.

In our statistical study to check the validity of the GUE hypothesis, we observed the agreement of the empirical data with formulas (31) and (30) on each interval $[\alpha, \beta)$, with $\alpha = i/100$ and $\beta = (i+1)/100$ for integer values of i, $0 \leq i < 300$. In figure 3, in addition to the curve representing the density probability function $p(0,t)$, points were plotted at abscissa $(i+1/2)/100$ and coordinate

$$c_i = 100\frac{1}{M}\#\{n : N+1 \leq n \leq N+M, \delta_n \in [i/100, (i+1)/100]\},$$

for height $N = 10^{13}$ and number of zeros $M \simeq 2 \times 10^9$. As we can see the agreement is very good, whereas the graphic is done with the lowest height in our collection : human eye is barely able to distinguish between the points and the curve. That is why it is interesting to plot rather the density difference between empirical data and asymptotic conjectured behavior (as Odlyzko did in [23] for example). This is the object of figure 4, and this time what is plotted in coordinate is the difference

$$d_i = c_i - \int_{i/100}^{(i+1)/100} p(0,t)\,dt.$$

To make it readable, the graphic restricts on some family of height N even if the corresponding data were computed at all height. The values $I_i = \int_{i/100}^{(i+1)/100} p(0,t)\,dt$ were computed from formula (32) with Maple. It is convenient to notice that I_i can be computed as $p(0,t)$ in (32) but using one differentiation order only instead of two.

Even if oscillations in the empirical data appear because the sampling size of 2×10^9 zeros is a bit insufficient, we clearly see a structure in figure 4. First, the form of the difference at each height has a given form, and then, the way this difference decreases with the height can be observed.

Another interesting data is the agreement with Montgomery pair correlation conjecture (31) about normalized spacing between non-necessarily consecutive zeros. Here analogous graphics have been done, first with the distribution itself in figure 5 at height 10^{13}, then with difference of the asymptotic conjectured distribution and empirical data in figure 6. Again for readability in the graphic, we restricted to plot only data for a limited number of height. It is striking to observe here a better regularity in the form of the distribution difference, which is a sort of sinusoid put on a positive slope.

4.3.3 Violations of Rosser rule

The table below lists statistics obtained on violations of Rosser rule (VRR). As we should expect, more and more violations of Rosser rule occurs when the height increases. Special points are Gram points which are counted in a VRR, so equivalently, they are points that do not lie in a regular Gram block.

Height	VRR per million zeros	Number of types of VRR	Number of special points	Average number of points in VRR
10^{13}	37.98	68	282468	3.719
10^{14}	54.10	86	418346	3.866
10^{15}	72.42	109	581126	4.012
10^{16}	93.99	140	780549	4.152
10^{17}	117.25	171	1004269	4.283
10^{18}	142.30	196	1255321	4.411
10^{19}	168.55	225	1529685	4.538
10^{20}	197.28	270	1837645	4.657
10^{21}	225.80	322	2156944	4.776
10^{22}	256.53	348	2507825	4.888
10^{23}	286.97	480	2868206	4.997
10^{24}	319.73	473	3262812	5.102

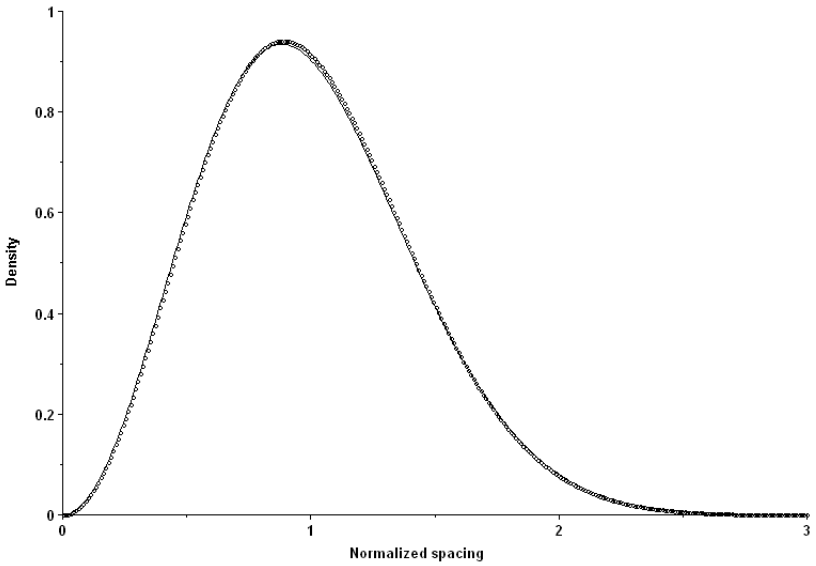

Figure 3: Probability density of the normalized spacing δ_n and the GUE prediction, at height 10^{13}. A number of 2×10^9 zeros have been used to compute empirical density, represented as small circles.

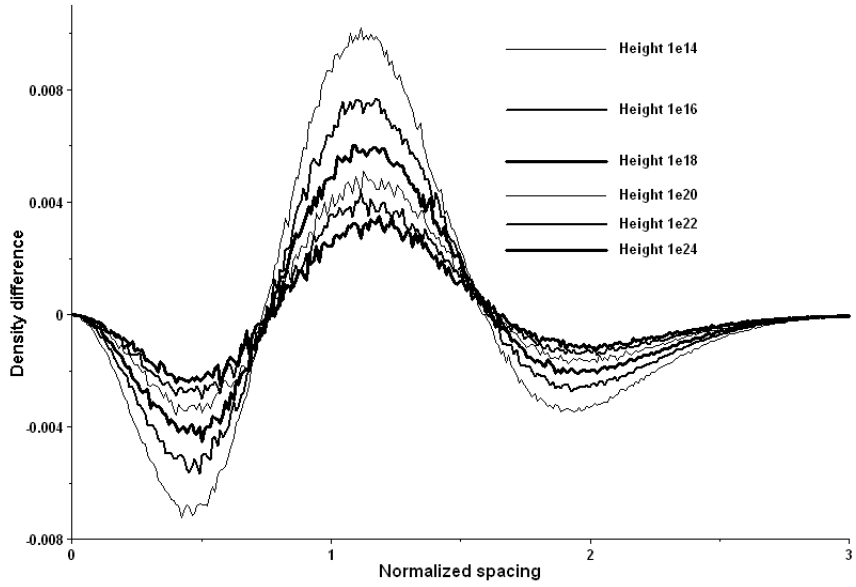

Figure 4: Difference of the probability density of the normalized spacing δ_n and the GUE prediction, at different height (10^{14}, 10^{16}, 10^{18}, 10^{20}, 10^{22}, 10^{24}). At each height, 2×10^9 zeros have been used to compute empirical density, and the corresponding points been joined with segment for convenience.

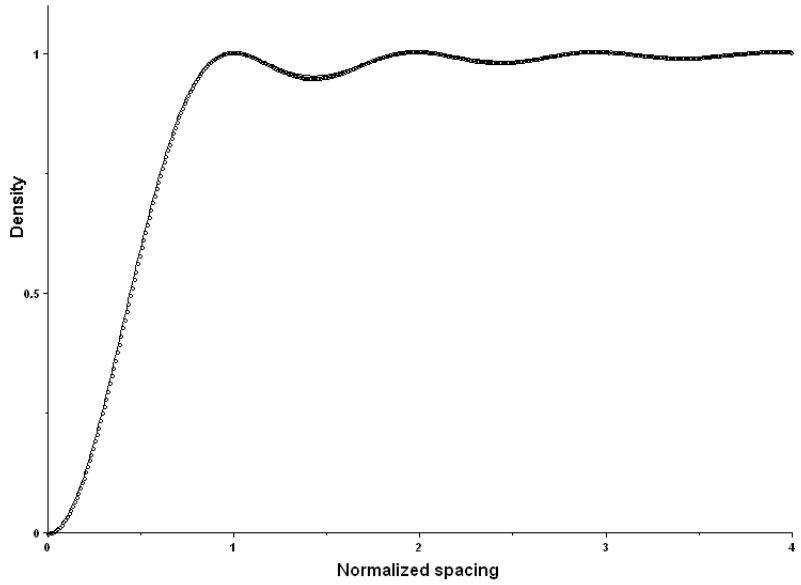

Figure 5: Probability density of the normalized spacing between non-necessarily consecutive zeros and the GUE prediction, at height 10^{13}. A number of 2×10^9 zeros have been used to compute empirical density, represented as small circles.

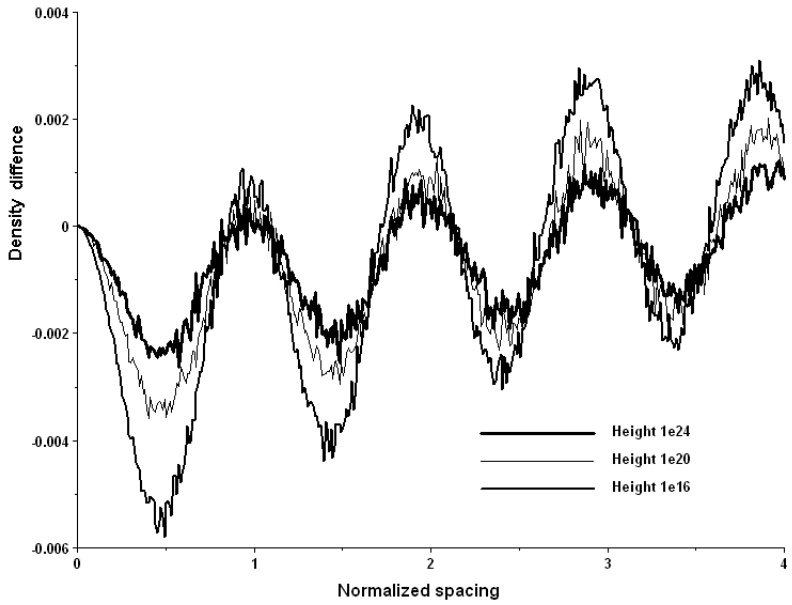

Figure 6: Difference of the probability density of the normalized spacing between non-necessarily consecutive zeros and the GUE prediction, at different height (10^{16}, 10^{20}, 10^{24}). At each height, 2×10^9 zeros have been used to compute empirical density, and the corresponding points been joined with segment for convenience.

4.3.4 Behavior of S(t)

The $S(t)$ function is defined in (2) and permits to count zeros with formula (3). It plays an important role in the study of the zeros of the Zeta function, because it was observed that special phenomenon about the zeta function on the critical line occurs when $S(t)$ is large. For example, Rosser rule holds when $|S(t)| < 2$ in some range, thus one needs to have larger values of $S(t)$ to find more rare behavior.

As already seen before, it is known unconditionally that

$$S(t) = O(\log t).$$

Under the RH, we have the slightly better bound

$$S(t) = O\left(\frac{\log t}{\log \log t}\right).$$

However, it is thought that the real growth of rate of $S(t)$ is smaller. First, it was proved that unconditionally, the function $S(t)/(2\pi^2 \log \log t)^{1/2}$ is asymptotically normally distributed. So in some sense, the "average" order of $S(t)$ is $(\log \log t)^{1/2}$. As for extreme values of $S(t)$; Montgomery has shown that under the RH, there is an infinite number of values of t tending to infinity so that the order of $S(t)$ is at least $(\log t/ \log \log t)^{1/2}$. Montgomery also conjectured that this is also an upper bound for $S(t)$. As described in section 4.3.6 with formula (33), the GUE suggests that $S(t)$ might get as large as $(\log t)^{1/2}$ which would contradict this conjecture.

As explained in [22, P. 28], one might expect that the average number of changes of sign of $S(t)$ per Gram interval is of order $(\log \log t)^{-1/2}$. This is to be compared with the last column of the table below, which was obtained thanks to the statistics on Gram blocks and violations of Rosser rule.

As it is confirmed in heuristic data in the table below, the rate of growth of $S(t)$ is very small. Since exceptions to RH, if any, would probably occur for large values of $S(t)$, we see that one should be able to reach much larger height, not reachable with today's techniques, to find those.

Height	Minimum of $S(t)$	Maximum of $S(t)$	Number of zeros with $S(t) < -2.3$	Number of zeros with $S(t) > 2.3$	Average number of change of sign of $S(t)$ per Gram interval
10^{13}	-2.4979	2.4775	208	237	1.5874
10^{14}	-2.5657	2.5822	481	411	1.5758
10^{15}	-2.7610	2.6318	785	760	1.5652
10^{16}	-2.6565	2.6094	1246	1189	1.5555
10^{17}	-2.6984	2.6961	1791	1812	1.5465
10^{18}	-2.8703	2.7141	2598	2743	1.5382
10^{19}	-2.9165	2.7553	3487	3467	1.5304
10^{20}	-2.7902	2.7916	4661	4603	1.5232
10^{21}	-2.7654	2.8220	5910	5777	1.5164
10^{22}	-2.8169	2.9796	7322	7359	1.5100
10^{23}	-2.8178	2.7989	8825	8898	1.5040
10^{24}	-2.9076	2.8799	10602	10598	1.4983

4.3.5 Estimation of the zeros approximation precision

As already discussed in section 4.1, a certain proportion of zeros were recomputed in another process with different parameters in the implementation and zeros computed twice were compared. Table below list the proportion of twice computed zeros per height, mean value of absolute value of difference and maximal difference.

Height	Proportion of zeros computed twice	Mean difference for zeros computed twice	Max difference for zeros computed twice
10^{13}	4.0%	5.90E-10	5.87E-07
10^{14}	6.0%	6.23E-10	1.43E-06
10^{15}	6.0%	7.81E-10	1.08E-06
10^{16}	4.5%	5.32E-10	7.75E-07
10^{17}	8.0%	5.85E-10	9.22E-07
10^{18}	7.5%	6.59E-10	1.88E-06
10^{19}	11.0%	5.15E-10	3.07E-06
10^{20}	12.5%	3.93E-10	7.00E-07
10^{21}	31.5%	5.64E-10	3.54E-06
10^{22}	50.0%	1.15E-09	2.39E-06
10^{23}	50.0%	1.34E-09	3.11E-06
10^{24}	50.0%	2.68E-09	6.82E-06

4.3.6 Extreme gaps between zeros

The table below lists the minimum and maximal values of normalized spacing between zeros δ_n and of $\delta_n + \delta_{n+1}$, and compares this with what is expected under the GUE hypothesis (see section 4.2). It can be proved that $p(0,t)$ have the following Taylor expansion around 0

$$p(0,u) = \frac{\pi^2}{3}u^2 - 2\frac{\pi^4}{45}u^4 + \cdots$$

so in particular, for small delta

$$\text{Prob}(\delta_n < \delta) = \int_0^\delta p(0,u)\,du \sim \frac{\pi^2}{9}\delta^3$$

so that the probability that the smallest δ_n are less than δ for M consecutive values of δ_n is about

$$1 - \left(1 - \frac{\pi^2}{9}\delta^3\right)^M \simeq 1 - \exp\left(-\frac{\pi^2}{9}\delta^3 M\right).$$

This was the value used in the sixth column of the table. The result can be also obtained for the $\delta_n + \delta_{n+1}$

$$\text{Prob}(\delta_n + \delta_{n+1} < \delta) \sim \frac{\pi^6}{32400}\delta^8,$$

from which we deduce the value of the last column.

Height	Mini δ_n	Maxi δ_n	Mini $\delta_n+\delta_{n+1}$	Maxi $\delta_n+\delta_{n+1}$	Prob min δ_n in GUE	Prob min $\delta_n+\delta_{n+1}$ in GUE
10^{13}	0.0005330	4.127	0.1097	5.232	0.28	0.71
10^{14}	0.0009764	4.236	0.1213	5.349	0.87	0.94
10^{15}	0.0005171	4.154	0.1003	5.434	0.26	0.46
10^{16}	0.0005202	4.202	0.1029	5.433	0.27	0.53
10^{17}	0.0006583	4.183	0.0966	5.395	0.47	0.36
10^{18}	0.0004390	4.194	0.1080	5.511	0.17	0.67
10^{19}	0.0004969	4.200	0.0874	5.341	0.24	0.18
10^{20}	0.0004351	4.268	0.1067	5.717	0.17	0.63
10^{21}	0.0004934	4.316	0.1019	5.421	0.23	0.50
10^{22}	0.0008161	4.347	0.1060	5.332	0.70	0.61
10^{23}	0.0004249	4.304	0.1112	5.478	0.15	0.75
10^{24}	0.0002799	4.158	0.0877	5.526	0.05	0.19

For very large spacing in the GUE, as reported by Odlyzko in [22], des Cloizeaux and Mehta [5] have proved that

$$\log p(0,t) \sim -\pi^2 t^2/8 \qquad (t \to \infty),$$

which suggests that

$$\max_{N+1 \le n \le N+M} \delta_n \sim \frac{(8\log M)^{1/2}}{\pi}. \tag{33}$$

This would imply that $S(t)$ would get occasionally as large as $(\log t)^{1/2}$, which is in contradiction with Montgomery's conjecture about largest values of $S(t)$, discussed in section 4.3.4.

4.3.7 Moments of spacings

The table below list statistical data about moments of the spacing $\delta_n - 1$ at different height, that is mean value of

$$M_k = (\delta_n - 1)^k,$$

together with the GUE expectations.

Height	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9
10^{13}	0.17608	0.03512	0.09608	0.05933	0.10107	0.1095	0.1719	0.2471
10^{14}	0.17657	0.03540	0.09663	0.05990	0.10199	0.1108	0.1741	0.2510
10^{15}	0.17697	0.03565	0.09710	0.06040	0.10277	0.1119	0.1759	0.2539
10^{16}	0.17732	0.03586	0.09750	0.06084	0.10347	0.1129	0.1776	0.2567
10^{17}	0.17760	0.03605	0.09785	0.06123	0.10407	0.1137	0.1789	0.2590
10^{18}	0.17784	0.03621	0.09816	0.06157	0.10462	0.1145	0.1803	0.2613
10^{19}	0.17805	0.03636	0.09843	0.06189	0.10511	0.1152	0.1814	0.2631
10^{20}	0.17824	0.03649	0.09867	0.06215	0.10553	0.1158	0.1824	0.2649
10^{21}	0.17839	0.03661	0.09888	0.06242	0.10595	0.1165	0.1836	0.2668
10^{22}	0.17853	0.03671	0.09906	0.06262	0.10627	0.1169	0.1842	0.2678
10^{23}	0.17864	0.03680	0.09922	0.06282	0.10658	0.1174	0.1850	0.2692
10^{24}	0.17875	0.03688	0.09937	0.06301	0.10689	0.1179	0.1859	0.2708
GUE	0.17999	0.03796	0.10130	0.06552	0.11096	0.1243	0.1969	0.2902

In the next table we find statistical data about moments of the spacing $\delta_n + \delta_{n+1} - 2$ at different height, that is mean value of

$$N_k = (\delta_n + \delta_{n+1} - 2)^k,$$

together with the GUE expectations.

Height	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9
10^{13}	0.23717	0.02671	0.16887	0.06252	0.2073	0.1530	0.3764	0.4304
10^{14}	0.23846	0.02678	0.17045	0.06301	0.2099	0.1550	0.3827	0.4388
10^{15}	0.23956	0.02688	0.17181	0.06349	0.2122	0.1568	0.3880	0.4458
10^{16}	0.24050	0.02700	0.17299	0.06396	0.2142	0.1585	0.3927	0.4523
10^{17}	0.24132	0.02713	0.17404	0.06446	0.2159	0.1601	0.3970	0.4583
10^{18}	0.24202	0.02726	0.17494	0.06488	0.2175	0.1614	0.4005	0.4630
10^{19}	0.24264	0.02740	0.17574	0.06530	0.2188	0.1627	0.4036	0.4672
10^{20}	0.24319	0.02753	0.17645	0.06569	0.2201	0.1639	0.4065	0.4713
10^{21}	0.24366	0.02766	0.17709	0.06609	0.2212	0.1651	0.4092	0.4753
10^{22}	0.24409	0.02778	0.17765	0.06643	0.2222	0.1660	0.4114	0.4780
10^{23}	0.24447	0.02790	0.17819	0.06679	0.2232	0.1671	0.4140	0.4821
10^{24}	0.24480	0.02801	0.17863	0.06709	0.2240	0.1679	0.4158	0.4846
GUE	0.249	0.03	0.185	0.073	0.237	0.185	0.451	0.544

The last table below is about mean value of $\log \delta_n$, $1/\delta_n$ and $1/\delta_n^2$.

Height	$\log \delta_n$	$1/\delta_n$	$1/\delta_n^2$
10^{13}	-0.101540	1.27050	2.52688
10^{14}	-0.101798	1.27124	2.53173
10^{15}	-0.102009	1.27184	2.54068
10^{16}	-0.102188	1.27235	2.54068
10^{17}	-0.102329	1.27272	2.54049
10^{18}	-0.102453	1.27308	2.54540
10^{19}	-0.102558	1.27338	2.54906
10^{20}	-0.102650	1.27363	2.54996
10^{21}	-0.102721	1.27382	2.54990
10^{22}	-0.102789	1.27401	2.54783
10^{23}	-0.102843	1.27415	2.55166
10^{24}	-0.102891	1.27427	2.55728
GUE	-0.1035	1.2758	2.5633

5 Acknowledgments

The author specially thanks Patrick Demichel who managed the distribution of the computation on several computers in order to make the RH verification on the first 10^{13} zeros possible. The author also thanks Andrew Odlyzko who provided some informations on the previous computations from his side, and Craig A. Tracy for the very precise and valuable informations about modern techniques to evaluate the $p(0,t)$ function.

References

[1] R. J. Backlund. Uber die nullstellen der Riemannschen zetafunktion. Dissertation, 1916. Helsingfors.

[2] David H. Bailey. Ffts in external or hierarchical memory. *Journal of Supercomputing*, 4(1):23–25, 1990.

[3] R. P. Brent. The first 40,000,000 zeros of $\zeta(s)$ lie on the critical line. *Notices of the American Mathematical Society*, (24), 1977.

[4] R. P. Brent. On the zeros of the Riemann zeta function in the critical strip. *Mathematics of Computation*, (33):1361–1372, 1979.

[5] J. des Cloizeaux and M. L. Mehta. Asymptotic behavior of spacing distributions for the eigenvalues of random matrices. *J. Math. Phys.*, 14:1648–1650, 1973.

[6] P. Dusart. *Autour de la fonction qui compte les nombres premiers*. PhD thesis, Universit de Limoges, 1998.

[7] H. M. Edwards. *Riemann's Zeta Function*. Academic Press, 1974.

[8] J. P. Gram. Note sur les zéros de la fonction de Riemann. *Acta Mathematica*, (27):289–304, 1903.

[9] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. of Computational Phys.*, 73:325–348, 1987.

[10] J. I. Hutchinson. On the roots of the Riemann zeta function. *Transactions of the American Mathematical Society*, 27(49-60), 1925.

[11] D. Bradley J. Borwein and R. Crandall. Computational strategies for the Riemann zeta function. available from the CECM preprint server, URL=http://www.cecm.sfu.ca/preprints/1999pp.html, CECM-98-118, 1999.

[12] H. J. J. te Riele J. van de Lune. On the zeros of the Riemann zeta function in the critical strip. iii. *Mathematics of Computation*, 41(164):759–767, October 1983.

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

[14] R. S. Leghman. Separation of zeros of the Riemann zeta-function. *Mathematics of Computation*, (20):523–541, 1966.

[15] D. H. Lehmer. Extended computation of the Riemann zeta-function. *Mathematika*, 3:102–108, 1956.

[16] D. H. Lehmer. On the roots of the Riemann zeta-function. *Acta Mathematica*, (95):291–298, 1956.

[17] M. L. Mehta and J. des Cloizeaux. The probabilities for several consecutive eigenvalues of a random matrix. *Indian J. Pure Appl. Math.*, 3:329–351, 1972.

[18] N. A. Meller. Computations connected with the check of Riemanns hypothesis. *Doklady Akademii Nauk SSSR*, (123):246–248, 1958.

[19] H. L. Montgomery. The pair correlation of zeroes of the zeta function. In *Analytic Number Theory*, volume 24 of *Proceedings of Symposia in Pure Mathematics*, pages 181–193. AMS, 1973.

[20] A. M. Odlyzko. On the distribution of spacings between zeros of the zeta function. *Mathematics of Computation*, 48:273–308, 1987.

[21] A. M. Odlyzko. The 10^{21}-st zero of the Riemann zeta function. note for the informal proceedings of the Sept. 1998 conference on the zeta function at the Edwin Schroedinger Institute in Vienna., Nov. 1988.

[22] A. M. Odlyzko. The 10^{20}-th zero of the Riemann zeta function and 175 million of its neighbors. Available at URL= http://www.dtc.umn.edu/~odlyzko/unpublished/index.html, 1992.

[23] A. M. Odlyzko. The 10^{22}-th zero of the Riemann zeta function. In M. van Frankenhuysen and M. L. Lapidus, editors, *Dynamical, Spectral, and Arithmetic Zeta Functions*, number 290 in Contemporary Math. series, pages 139–144. Amer. Math. Soc., 2001.

[24] A. M. Odlyzko and A. Schönhage. Fast algorithms for multiple evaluations of the Riemann zeta-function. *Trans. Amer. Math. Soc.*, (309), 1988.

[25] O. Ramaré and Y. Saouter. Short effective intervals containing primes. *Journal of Number Theory*, 98:10–33, 2003.

[26] J. B. Rosser. Explicit bounds for some functions of prime numbers. *Amer. J. Math.*, 63:211–232, 1941.

[27] J. B. Rosser and L. Schoenfeld. Sharper bounds for the Chebyshev functions $\theta(x)$ and $\psi(x)$. *Math. Comput.*, 29:243–269, 1975.

[28] J. B. Rosser, J. M. Yohe, and L. Schoenfeld. Rigorous computation and the zeros of the Riemann zeta-function. In *Information Processing*, number 68 in Proceedings of IFIP Congress, pages 70–76. NH, 1968.

[29] L. Schoenfeld. Sharper bounds for the Chebyshev functions $\theta(x)$ and $\psi(x)$. II. *Math. Comput.*, 30(134):337–360, 1976. Corrigenda in *Math. Comp.* **30** (1976), 900.

[30] E. C. Titchmarsh. The zeros of the Riemann zeta-function. In *Proceedings of the Royal Society of London*, volume 151, pages 234–255, 1935.

[31] E. C. Titchmarsh. *The theory of the Riemann Zeta-function*. Oxford Science publications, second edition, 1986. revised by D. R. Heath-Brown.

[32] C. A. Tracy and H. Widom. Introduction to random matrices. In G. F. Helminck, editor, *Geometric and Quantum aspects of integrale systems*, volume 424 of *Lecture Notes in physics*, pages 103–130, Berlin, Heidelberg, 1993. Springer.

[33] A. M. Turing. Some calculations of the Riemann zeta-function. In *Proceedings of the Royal Society of London*, number 3, pages 99–117, 1953.

[34] R. P. Brent J. van de Lune H. J. J. te Riele and D. T. Winter. On the zeros of the Riemann zeta function in the critical strip. ii. *Mathematics of Computation*, 39(160):681–688, October 1982.

[35] S. Wedeniwski. Zetagrid - computational verification of the Riemann hypothesis. In *Conference in Number Theory in Honour of Professor H.C. Williams*, Alberta, Canda, May 2003.

[36] Eric W. Weisstein. Riemann-siegel functions. available from the MathWorld site at URL=http://mathworld.wolfram.com/Riemann-SiegelFunctions.html.

A Appendix : Graphics of $Z(t)$ in particular zones

The following figures show the function $Z(t)$ in some particular zones. Vertical doted lines correspond to Gram points.

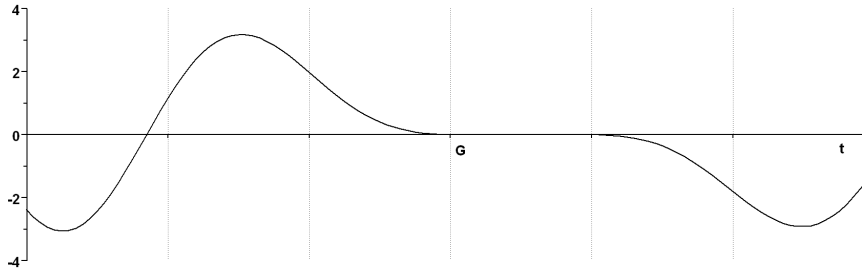

Figure 7: The function $Z(t)$ around the first Gram interval that contains 5 roots. The point G is the Gram point of index $3,680,295,786,520$.

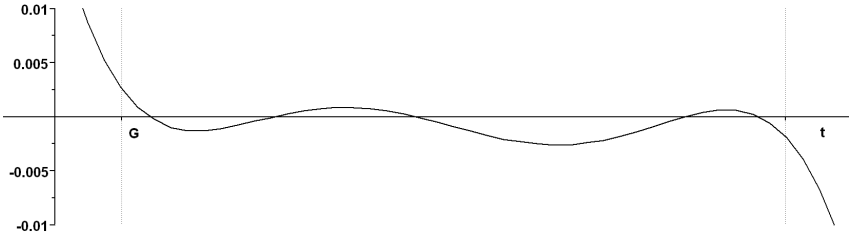

Figure 8: A zoom of previous figure focused on the Gram interval that contains 5 roots. The point G is the Gram point of index $3,680,295,786,520$.

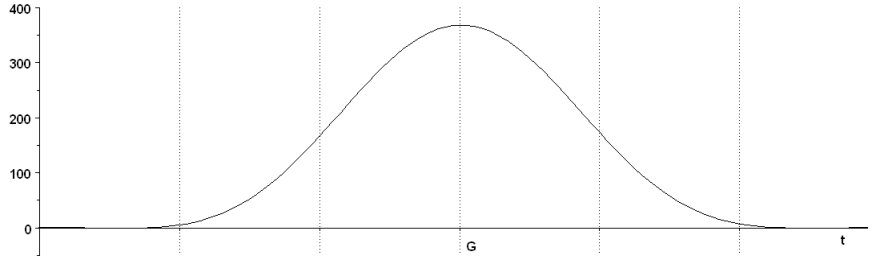

Figure 9: The function $Z(t)$ around its maximal value encountered on the 10^{13} zero, at Gram point of index $9,725,646,131,432$.