Fast algorithms for multiple evaluations of the Riemann zeta function

A. M. Odlyzko
AT&T Bell Laboratories
Murray Hill, NJ 07974
USA

A. Schönhage
Mathematisches Institut
Universität Tübingen
7400 Tübingen 1
West Germany

ABSTRACT

The best currently known algorithm for evaluating the Riemann zeta function, $\zeta(\sigma + it)$, with σ bounded and t large to moderate accuracy (within $\pm\, t^{-c}$ for some $c > 0$, say) is based on the Riemann-Siegel formula and requires on the order of $t^{1/2}$ operations for each value that is computed. New algorithms are presented in this paper which enable one to compute any single value of $\zeta(\sigma + it)$ with σ fixed and $T \le t \le T + T^{1/2}$ to within $\pm\, t^{-c}$ in $O(t^{\varepsilon})$ operations on numbers of $O(\log t)$ bits for any $\varepsilon > 0$, for example, provided a precomputation involving $O(T^{1/2 + \varepsilon})$ operations and $O(T^{1/2 + \varepsilon})$ bits of storage is carried out beforehand. These algorithms lead to methods for numerically verifying the Riemann Hypothesis for the first n zeros in what is expected to be $O(n^{1 + \varepsilon})$ operations (as opposed to about $n^{3/2}$ operations for the previous method), as well as improved algorithms for the computation of various arithmetic functions, such as $\pi(x)$. The new zeta function algorithms use the Fast Fourier Transform and a new method for the evaluation of certain rational functions. They can also be applied to the evaluation of L-functions, Epstein zeta functions, and other Dirichlet series.

Fast algorithms for multiple evaluations
of the Riemann zeta function

A. M. Odlyzko
AT&T Bell Laboratories
Murray Hill, NJ 07974
USA

A. Schönhage
Mathematisches Institut
Universität Tübingen
7400 Tübingen 1
West Germany

1. Introduction

Some of the algorithms for computing the Riemann zeta function and actual computations have dealt with values of the zeta function at positive integers, while others dealt with very accurate determinations of small zeros of this function. However, the most extensive computations have been those directed at verifying the Riemann Hypothesis (RH) for large sets of zeros, culminating in the recent calculations that established the truth of the RH for the first $1.5 \cdot 10^9$ zeros [17] using over a thousand hours on a modern supercomputer. These calculations involved computing values of the zeta function $\zeta(1/2 + it)$ for t real and very large but only to medium accuracy (roughly $\pm\, t^{-2}$). Other calculations of actual values of the zeros of the zeta function at even greater heights, designed to test conjectures about distribution of spacings between consecutive zeros [18, 19], required only slightly greater accuracy. Finally, new algorithms have recently been invented [16] for the computation of arithmetical functions such as $\pi(x)$, the number of primes $\leq x$, that are more efficient than any known combinatorial methods, and which require values of $\zeta(s)$ to be computed at points $s = \sigma + it$, for σ fixed (typically $\sigma = 2$) and t large, but again only to accuracy $\pm\, t^{-c}$

for various $c > 0$. (For a fuller description of these and other computations of the zeta function, see [19].) In this paper we will present improved algorithms for computing such ‘‘medium accuracy’’ values.

The Riemann zeta function is defined for $s = \sigma + it$ by

$$\zeta(s) = \sum_{n=1}^{\infty} n^{-s} \tag{1.1}$$

for $\sigma > 1$, and by analytic continuation can be extended to an analytic function of s for all $s \neq 1$ [7, 13, 22]. The definition (1.1) suggests the idea of using the Euler-Maclaurin summation formula [12; Eq. 23.1.30] to evaluate $\zeta(s)$, and one easily obtains, for any positive integers m and n,

$$\zeta(s) = \sum_{j=1}^{n-1} j^{-s} + \frac{1}{2} n^{-s} + \frac{n^{1-s}}{s-1} + \sum_{k=1}^{m} T_{k,n}(s) + E_{m,n}(s) , \tag{1.2}$$

where

$$T_{k,n}(s) = \frac{B_{2k}}{(2k)!} n^{1-s-2k} \prod_{j=0}^{2k-2} (s+j) ,$$

$B_2 = 1/6,\ B_4 = -1/30 , \ldots ,$ are the Bernoulli numbers, and

$$|E_{m,n}(s)| < \left| \frac{s + 2m + 1}{\sigma + 2m + 1} T_{m+1,n}(s) \right| . \tag{1.3}$$

The formula (1.2) with the estimate (1.3) can easily be shown to hold for any $\sigma > -(2m+1)$. By taking m and n large enough (and using sufficient accuracy in basic arithmetic routines), any value of $\zeta(s)$ can be computed to any desired accuracy by this formula. All calculations of zeros of the zeta function that were published before 1930 relied on this method. Its advantages include the ease of estimating the error term. (This is the main reason this formula is still used for very accurate computations of $\zeta(s)$ for s small, cf. [19].) Its main disadvantage is its inefficiency. For t large and $\sigma = 1/2$, say, it is necessary to take n on the order of t to obtain any kind of accuracy. This is due to the fact that the Euler-Maclaurin formula basically approximates each term k^{-s}, $k \geq n$, by

$$\int_k^{k+1} u^{-s} \, du = \frac{(k+1)^{1-s} - k^{1-s}}{1-s} = \frac{k^{1-s}}{1-s} \left[(1 + \frac{1}{k})^{1-s} - 1 \right] ,$$

and for t much larger than k this is not a good approximation, and cannot be improved easily even by taking higher degree approximations (which is where the terms with Bernoulli numbers come from).

A method for computing $\zeta(s)$ that is much more efficient than the Euler-Maclaurin formula (1.2) was discovered around 1932 in Riemann's unpublished papers by C. L. Siegel [21]. This formula [21; Eq. (32)], now universally referred to as the Riemann-Siegel formula, is presented in Section 2. Roughly speaking, it enables one to compute $\zeta(\sigma + it)$ for t large and σ bounded to within $\pm\, t^{-c}$ for any constant c in about $t^{1/2}$ steps. (Since $\zeta(\bar{s}) = \overline{\zeta(s)}$, we will always assume that $t > 0$.) The Riemann-Siegel formula is the fastest method for computing the zeta function to moderate accuracy that is currently known, and has been used for all large scale computations since the 1930's.

Only one other method, besides the Euler-Maclaurin and Riemann-Siegel ones, seems to have been proposed for computing $\zeta(s)$ to moderate accuracy at large heights, namely the one due to Turing [24]. It was designed to provide higher accuracy than was guaranteed by the crude bounds on the remainder term in the Riemann-Siegel formula that were available at that time, and at the same time be more efficient than the Euler-

Maclaurin formula. However, very good estimates for remainder terms in the Riemann-Siegel formula are now available [8], which seem to make Turing's method unnecessary.

In this paper we propose new methods of computing the zeta function and related functions which are much faster than the Riemann-Siegel formula method when many values at closely spaced points are needed. We will obtain upper bounds for the number of arithmetic operations (multiplication, addition, division, subtraction) on numbers of $O(\log T)$ bits that our algorithms use. Our main result is as follows.

Theorem 1.1. Given any positive constants $\delta, \sigma,$ *and* c_1, *there is an effectively computable constant* $c_2 = c_2(\delta, \sigma, c_1)$ *and an algorithm that for every* $T > 0$ *will perform* $\leq c_2 T^{1/2 + \delta}$ *elementary arithmetic operations on numbers of* $\leq c_2 \log T$ *bits using* $\leq c_2 T^{1/2 + \delta}$ *bits of storage, and will then be capable of computing any value* $\zeta(\sigma + it)$, $T \leq t \leq T + T^{1/2}$, *to within* $\pm T^{-c_1}$ *in* $\leq c_2 T^{\delta}$ *operations using the precomputed values.*

The algorithms referred to in the theorem are described and analyzed in sections 2-4. They offer no advantage over the Riemann-Siegel formula when a single value of $\zeta(s)$ is desired, and in fact they suffer from the disadvantage of requiring about $T^{1/2}$ bits of storage, whereas the Riemann-Siegel formula requires $O(T^{\varepsilon})$ storage. (The storage requirements of the new algorithms can be lowered at the cost of decreasing the range in which values can subsequently be computed rapidly, as will be explained in Section 5.) On the other hand, in typical applications, such as verifying the RH for a block of zeros or computing $\pi(x)$ by the method of [16], one needs at least $T^{1/2}$ values of $\zeta(\sigma + it)$, $t \in [T, T + T^{1/2}]$ for all large T, and the new algorithms allow one to compute these values in average time $O(T^{\varepsilon})$ for all $\varepsilon > 0$ (but with the use of $O(T^{1/2 + \varepsilon})$ bits of storage).

The best currently known strategy [4, 7, 17, 19] for verifying the RH for the first N zeros involves finding N sign changes of the real function $Z(t)$ that is defined by

$$Z(t) = e^{i\theta(t)} \zeta(1/2 + it) , \tag{1.4}$$

$$\theta(t) = \arg[\pi^{-it/2} \Gamma(1/4 + it/2)] , \tag{1.5}$$

in the subinterval of $[10, \infty)$ where $\theta(t) \lesssim N\pi$, i.e., $t \lesssim 2\pi N(\log N)^{-1}$. This requires at least $N + 1$ evaluations of $\zeta(1/2 + it)$, and in practice about $6N/5$ evaluations suffice [17, 19]. With the Euler-Maclaurin formula, verification of the RH for the first N zeros therefore takes in practice $O(N^{2 + \varepsilon})$ operations, and in any case $\geq c_3 N^2 (\log N)^{-1}$ operations. With the Riemann-Siegel formula, this same verification requires in practice $O(N^{3/2 + \varepsilon})$ operations, and at least $\geq c_4 N^{3/2} (\log N)^{-1/2}$ operations. With our new algorithms, we expect that this task can be accomplished in $O(N^{1 + \varepsilon})$ operations using $O(T^{1/2 + \varepsilon})$ bits of storage, for every $\varepsilon > 0$, which is essentially best possible for the basic strategy that is currently used. We cannot rigorously prove this running time bound because for all we know, counterexamples to the RH might occur, or else very close pairs of zeros or multiple zeros might cause the current verification strategy to fail, cf. [19].

In the case of algorithms for computing $\pi(x)$, the best currently known combinatorial algorithm [14] requires $O(x^{2/3 + \varepsilon})$ operations and $O(x^{1/3 + \varepsilon})$ bits of storage. A totally different algorithm, proposed in [16], uses numerical integration of analytic transforms involving the zeta function to compute $\pi(x)$. If one employs the Euler-Maclaurin formula to compute $\zeta(s)$, this algorithm runs in time $O(x^{2/3 + \varepsilon})$ and space $O(x^{\varepsilon})$. The

Riemann-Siegel formula lowers the running time to $O(x^{3/5 + \varepsilon})$ and keeps the storage requirement to $O(x^{\varepsilon})$. The new algorithms presented in this paper lead to an algorithm for computing $\pi(x)$ that uses $O(x^{1/2 + \varepsilon})$ operations and $O(x^{1/4 + \varepsilon})$ bits of storage. Similar running time improvements can be obtained for other algorithms, such as those for computing $M(x)$, the summatory function of the Möbius μ-function [16].

The basic idea behind our new algorithm comes from the fact that the bulk of the work in computing $\zeta(\sigma + it)$ or other Dirichlet series occurs in evaluating sums of the form

$$g(t) = \sum_{k=1}^{M} d_k \, k^{-it} \, , \tag{1.6}$$

see Section 2. If $g(t)$ and several of its derivatives (which are of the same general form) are known at a set of regularly spaced t's in some interval, though, $g(t)$ can be computed at any point of that interval by using Taylor series expansions around the nearest grid point (Section 2). On the other hand, the problem of evaluating $g(t)$ at an evenly spaced set of t's can be transformed, using the Fast Fourier Transform (FFT), to the problem of evaluating a rational function of the form

$$\sum_{k=1}^{n} \frac{a_k}{z - b_k} \tag{1.7}$$

at the n n-th roots of unity (Section 3). In Section 4 we show how to rapidly evaluate such rational functions at multiple points.

The basic method outlined above is very general. The specific algorithms referred to in Theorem 1.1 and described in this paper use the Riemann-Siegel formula as a starting point. However, they could also be used with other methods. For example, if we had to rely on the Euler-Maclaurin formula, the precomputation time and storage requirements in Theorem 1.1 would both go up to $O(T^{1 + \delta})$, but individual values of $\zeta(\sigma + it)$ could again be computed in time $O(T^{\delta})$. (The requirement $t \varepsilon [T, T + T^{1/2}]$ would be replaced by $t \varepsilon [T , 2T]$.) In Section 5 we will mention some of the applications of our basic method to the computation of other Dirichlet series, as well as some of the time-space tradeoffs that are possible with our algorithms.

The presentation of the basic method in sections 2-4 is limited to the zeta function in order to bring out the basic ideas without obscuring them with excessive notation. The various factors of T^{δ} and the like in our estimates could be replaced by powers of log T, but we have chosen not to do so in order to keep the presentation as simple as possible. Actual implementations of our algorithms, which we feel are likely to be practical, would require substantial modifications of the algorithms as outlined below and extensive work on good explicit estimates of various error terms.

2. Reduction to an evenly spaced grid

In this section we show that the evaluation of $\zeta(\sigma + it)$ for t in a short interval can be reduced to that of evaluating several simple exponential sums at an evenly spaced grid of points. Because of the functional equation of the zeta function, we can restrict our attention to $\sigma \geq 1/2,\ t > 0$. We start out by recalling the Riemann-Siegel formula [21], which we write in the more standard notation of [7, 13, 22], where this formula is proved for $0 \leq \sigma \leq 1$ only. (We could further simplify our presentation by restricting ourselves to $\sigma = 1/2$, in which case a nice proof with very good estimates of the error terms has been obtained recently by Gabcke [8]. However, the analytic algorithms for computing

$\pi(x)$ presented in [16] require the use of $\sigma > 1/2$.) Let $s = \sigma + it$, where $1/2 \le \sigma \le 2$ is fixed, $t > 2\pi$, $N \varepsilon Z^+$, and $t \ge c_5 N$ for a certain fixed $c_5 > 0$. Further let $m = \lfloor (2\pi)^{-1/2} t^{1/2} \rfloor$, and for v being the fractional part of $t^{1/2} (2\pi)^{-1/2}$, let

$$S_N(s) = \sum_{n=0}^{N-1} \sum_{r \le n/2} \frac{n!\, i^{r-n}}{r!\,(n-2r)!\, 2^n} \left[\frac{2}{\pi}\right]^{n/2-r} a_n(s)\, \Psi^{(n-2r)}(2v) , \tag{2.1}$$

with

$$\Psi(u) = \frac{\cos \pi(\frac{1}{2} u^2 - u - \frac{1}{8})}{\cos \pi u} , \tag{2.2}$$

and with the coefficients $a_n(s)$ given by the recurrence

$$a_0(s) = 1,\ \ a_1(s) = \frac{\sigma-1}{\sqrt{t}} \ ,\ a_2(s) = \frac{(\sigma-1)\,(\sigma-2)}{2t} , \tag{2.3}$$

$$(n+1) t^{1/2} a_{n+1}(s) = (\sigma - n - 1) a_n(s) + i a_{n-2}(s) \quad \text{for } n \ge 2 .$$

The $a_n (s)$ satisfy the bound [21]

$$|a_n(s)| \le c_9\, t^{-n/6} . \tag{2.4}$$

We then have

$$\zeta(s) = \sum_{k=1}^{m} k^{-s} + \chi(s) \sum_{k=1}^{m} k^{s-1} \tag{2.5}$$

$$+ (-1)^{m-1} (2\pi t)^{(s-1)/2} \exp(-i\pi(s-1)/2 - it/2 - i\pi/8)\, \Gamma(1-s)\, T_N(s) ,$$

where

$$\chi(s) \ = \ 2^s \ \pi^{s-1} \ \sin(\frac{\pi s}{2}) \ \Gamma(1-s) \ , \tag{2.6}$$

$$| T_N(s) \ - \ S_N(s) | \le c_6 \left\{ e^{-c_7 t} \ + \ (c_8 N/t)^{N/6} \right\} \tag{2.7}$$

for some effectively computable constants $c_6, \ c_7, \ c_8 \ > \ 0$.

We first show that the last term in (2.5) can be computed efficiently. First of all, by Stirling's formula [12; Eq. 6.1.42],

$$\log \ \Gamma(1-s) \ = \ (1/2-s) \ \log \ (1-s) \ - \ (1-s) \ + \ \frac{1}{2} \ \log(2\pi) \tag{2.8}$$

$$+ \ \sum_{h=1}^{M} \ \frac{B_{2h}}{2h \ (2h \ - \ 1)} \ (1-s)^{1-2h} \ + \ O(t^{-1-2M}) \ ,$$

where we choose $M \ = \ \lfloor c_1 \ + \ 10 \rfloor$, and the constant implied by the O-notation is (as will be the case for all other such constants) dependent only on $\sigma, \ \delta$, and c_1. Hence we can compute $\log \ \Gamma(1-s)$ using (2.8) to an accuracy of $O(t^{-c_1-15})$ in $O(t^{\delta})$ arithmetic operations. Similarly, we can compute

$$w \ = \ - \ i\pi(s-1)/2 \ - \ it/2 \ - \ i\pi/8$$

to within $O(t^{-c_1-10})$ in $O(t^{\delta})$ operations. Since [12; Eq. 6.1.45]

$$| \Gamma(1-s) | \ = \ O(t^{1/2-\sigma} \ e^{-\pi t/2})$$

(a result that follows from (2.8) also), we have

$$| (2\pi t)^{(s-1)/2} \ e^w \ \Gamma(1-s) | \ = \ O(t^{-\sigma/2}) \ ,$$

and we can compute it to an accuracy of $O(t^{-c_1-10})$ in $O(t^{\delta})$ operations.

Next we consider $S_N(s)$, where we select $N = \lfloor 6c_1 + 100 \rfloor$. Note that $\Psi(u)$ is actually an entire function, and its Taylor series expansion around $u = 1/2$ can be used to evaluate $\Psi(u)$ and its first N derivatives in a neighborhood of $u = 1/2$ (and similarly in a neighborhood of $u = -1/2$), while away from $u = \pm 1/2$, we can differentiate the expansion (2.5) and evaluate individual terms. Each of the first N derivatives takes $O(t^{\delta})$ operations to compute to within $O(t^{-c_1 - 10})$. Finally, the $a_n(s)$ for $0 \le n \le N-1$ can be computed by the recurrence (2.6) to within $O(t^{-c_1 - 10})$ in $O(t^{\delta})$ operations. Combining all these estimates, we conclude that the entire term in (2.1) containing $T_N(s)$ can be computed to within $O(t^{-c_1 - 10})$ in $O(t^{\delta})$ operations (using $O(t^{\delta})$ bits of storage).

Stirling's formula (2.8) allows us also to compute $\chi(s)$ to within $O(t^{-c_1 - 10})$ in time $O(t^{\delta})$. Therefore, in order to compute $\zeta(s)$ to within $O(t^{-c_1 - 1})$, it suffices to be able to compute

$$f^*(t) = \sum_{k=1}^{m} k^{\alpha + it} \tag{2.9}$$

for $\alpha = -\sigma$ and $\alpha = \sigma - 1$ to within $O(t^{-c_1 - 10})$.

Suppose now that we are interested in computing values of $\zeta(\sigma + it)$ for $t \; \varepsilon \; [T, \; T + T^{1/2}]$. By the discussion above, we will be able to compute each such value to within $O(t^{-c_1 - 1})$ in $O(t^{\delta})$ additional operations if we can compute $f^*(t)$ to within $O(t^{-c_1 - 10})$, where in the definition (2.9) of $f^*(t)$, $m = \lfloor t^{1/2} \, (2\pi)^{-1/2} \rfloor$, and α ranges over two values $\varepsilon [-2, 1]$. If we now choose

$$H = \lfloor T^{1/2} (2\pi)^{-1/2} \rfloor \tag{2.10}$$

and define

$$f(t) = \sum_{k=2}^{H} k^{\alpha + it} , \tag{2.11}$$

then for $t \in [T, T + T^{1/2}]$, $f(t)$ and $f^*(t)$ differ by at most two terms, namely 1 and

$$(H+1)^{\alpha + it} ,$$

and this difference can be computed to within $O(T^{-c_1 - 10})$ in $O(T^{\delta})$ operations for any t.

Since

$$f^{(r)}(t) = i^r \sum_{k=2}^{H} (\log k)^r k^{\alpha + it} , \tag{2.12}$$

we have

$$|f^{(r)}(t)| \leq H^2 (\log H)^r$$

for any nonnegative integer r (recall that $\alpha \leq 1$). Let

$$G = \lfloor T^{(1 + \delta)/2} + 2 \rfloor , \tag{2.13}$$

$$\eta = T^{1/2} G^{-1} , \tag{2.14}$$

$$t_j = T + j \cdot \eta \ , \ 0 \leq j \leq G , \tag{2.15}$$

and compute the $f^{(r)}(t_j)$ for $0 \leq r < R = 10 \lceil c_1 + 10 \rceil \lceil \delta^{-1} + 10 \rceil$ (where $\lceil x \rceil$ denotes the least integer $\geq x$) to within $O(T^{-c_1 - 20})$ each. Then for any $t \in [T, T + T^{1/2}]$, if

$$|t - t_j| < \eta$$

(as happens for at least one t_j), the first R terms of the Taylor series expansion of $f(t)$ around t_j give the values of $f(t)$ to within $O(t^{-c_1 - 10})$, since the remaining terms are

$$\sum_{r=R}^{\infty} \frac{f^{(r)}\ (t_j)}{r!}\ (t - t_j)^r\ =\ O\left[\sum_{r=R}^{\infty} \frac{H^2\ (\log H)^r\ \eta^r}{r!}\right]\ =\ O(T^{-c_1 - 10})$$

for T sufficiently large (depending only on δ and c_1), and the first R terms are all $O(T)$.

We have so far proved the last part of Theorem 1.1, namely that if we compute the $f^{(r)}\ (t_j)$ for $0 \le r < R$ and $0 \le j \le G$ to within $O(T^{-c_1 - 10})$ each (where α in (2.12) ranges over the two values $\alpha = -\sigma$ and $\alpha = \sigma - 1$), then any additional value of $\zeta(\sigma + it)$ for $t\ \varepsilon\ [T,\ T + T^{1/2}]$ can be computed in $O(T^{\delta})$ operations to within $O(T^{-c_1 - 1})$ for $T \ge T_0\ (\sigma,\ \delta,\ c_1)$. Therefore for $T \ge T_1(\sigma,\ \delta,\ c_1)$, we can compute $\zeta(\sigma + it)$ to within $\pm\ T^{-\ c_1}$ in $O(T^{\delta})$ operations. Furthermore, for $t < T_1$, each value of $\zeta(\sigma + it)$ can be computed in time $\le c_{10}$ (depending on σ, δ, and c_1 above) to the desired accuracy by means of the Euler-Maclaurin formula. Note that storage of all the $f^{(r)}\ (t_j)$ takes $O(T^{1/2 + \delta})$ bits. Therefore to complete the proof of Theorem 1.1, we only have to show how to evaluate all the $f^{(r)}\ (t_j)$ in time $O(T^{1/2\ +\ \delta})$ and space $O(T^{1/2\ +\ \delta})$.

3. Application of the Fast Fourier Transform

In this section we will show that if $\varepsilon > 0$ and $c_{11} > 1$ are constants and

$$g(t)\ =\ \sum_{k=2}^{H} d_k\ k^{it}\ , \tag{3.1}$$

where $|d_k| \le H^{c_{11}}$, and the values of the d_k are known to within $O(H^{-100\,c_{11}})$, then we can compute $g(0), g(\theta), \ldots, g(\theta G)$ for $G \le 2H^{1+\varepsilon}$, $0 < \theta < 1$, each to within $O(H^{-c_{11}})$, using $O(H^{1+2\varepsilon})$ operations and $O(H^{1+2\varepsilon})$ bits of storage. (The constants implied by the O-notation are to depend only on ε and c_{11}.) To do this, select n to be a power of 2, $n = 2^h$, such that

$$2H^{1+\varepsilon} + 2 \le n = 2^h \le 4H^{1+\varepsilon} + 4 ,$$

and let

$$\omega = e^{2\pi i/n} , \tag{3.2}$$

$$h(j) = \sum_{m=0}^{n-1} g(m\theta)\,\omega^{mj} \quad , \quad 0 \le j \le n-1 . \tag{3.3}$$

The inverse of this discrete Fourier transform yields

$$g(m\theta) = \frac{1}{n} \sum_{j=0}^{n-1} h(j)\,\omega^{-mj} , \; 0 \le m \le n-1 . \tag{3.4}$$

If we compute the $h(j)$ to within $O(H^{-8c_{11}})$, then, since the $h(j)$ are all $O(H^{4\,c_{11}})$, we will be able to compute the $g(m\theta)$ to within $O(H^{-3c_{11}})$ using the Fast Fourier Transform (FFT) [1] in $O(n \log n)$ arithmetic operations on numbers of $O(\log n)$ bits; i.e., a total of $O(H^{1+2\varepsilon})$ operations using $O(H^{1+2\varepsilon})$ bits of storage.

We now consider the computation of the $h(j)$. By (3.1) and (3.3),

$$h(j) = \sum_{k=2}^{H} d_k\, w_k(j) , \tag{3.5}$$

where

$$w_k(j) \;=\; \sum_{m=0}^{n-1} k^{im\theta}\ \omega^{mj}\ . \tag{3.6}$$

Take any k, $2 \le k \le H$. If there is some j (necessarily unique), say $j = J$, such that

$$|1 - k^{i\theta}\ \omega^J| < H^{-20c_{11}}\ , \tag{3.7}$$

then

$$w_k(J) \;=\; n + O(H^{-10c_{11}})\ ,$$

$$w_k(j) \;=\; O(H^{-10c_{11}}) \quad \text{for} \quad 0 \le j \le n-1,\ j \ne J\ .$$

If there is no J that satisfies (3.7), then

$$w_k(j) \;=\; \frac{1 - k^{in\theta}}{1 - k^{i\theta}\ \omega^j} \;=\; \frac{k^{-i\theta}\ (k^{in\theta} - 1)}{\omega^j - k^{-i\theta}}\ .$$

Let

$$f(z) \;=\; \sum_{k=2}^{H} \frac{a_k}{z - k^{-i\theta}}\ , \tag{3.8}$$

where $a_k = 0$ if there is some J, $0 \le J \le n-1$ satisfying (3.7), and

$$a_k \;=\; d_k\ k^{-i\theta}\ (k^{in\theta} - 1)$$

otherwise. If we can now evaluate $f(\omega^j)$ for $0 \le j \le n-1$, all to within $O(H^{-8c_{11}})$, and in $O(H^{1+2\varepsilon})$ operations in $O(H^{1+2\varepsilon})$ bits of storage, then we can compute all the $h(j)$ to a similar accuracy with the same running time and space bound by making a single pass through all the k's to identify those for which (3.7) is satisfied for some J, and adding to the already computed value of $f(\omega^J)$ the number $d_k\ n$. Thus to complete the presentation and analysis of our algorithm it remains to show how rational functions such as (3.8) can be evaluated rapidly and accurately.

4. Rational function evaluation

In this section we prove the auxiliary result that is needed to complete the description of our algorithms for evaluating the zeta function. This result is stated in a form that is much less general than possible. In fact, the basic method of proof we utilize can be generalized to produce a $O(n^{1+\varepsilon})$ operations algorithm for the following problem, which has been circulating under the name of the ‘‘Trummer problem’’ [11].

Given complex numbers x_j *and* a_j, $1 \le j \le n$, $|x_j| \le 1$ *and* $|a_j| \le 1$ *for all j, and* $|a_j - a_k| \ge n^{-c}$ *for some* $c > 0$ *and all* $j \ne k$, *and a constant* c', *compute*

$$b_k = \sum_{j \ne k} \frac{x_j}{a_j - a_k} \quad , \quad 1 \le k \le n \; ,$$

to within $\pm\, n^{-c'}$.

This problem arises in research on the complexity of conformal mappings [23] and was first formulated explicitly by Golub [11] in a form similar to that above, but without the restriction that $|a_j - a_k| \ge n^{-c}$ for $j \ne k$, with the question being whether all the b_k can be computed in fewer than n^2 multiplications. By a result of [9], both versions of this problem are equivalent to a rational function evaluation problem similar to the one we have to solve. In the algebraic model, where arithmetic operations are counted at unit cost (as if infinite precision were possible, which is realistic in the case of small finite field computations, but not in general), it can be shown that this task can be carried out in $O(n\,(\log n)^2)$ operations (cf. [10]). Our method yields a $O(n\,(\log m)^{c''})$ bit operations algorithm for the finite precision version stated above.

The method used to evaluate the rational function $f(x)$ given by (3.8) at the n-th roots of unity is based on Taylor series expansions around a small set of points. It is impossible to use such expansions of the function $f(x)$ itself, because the radii of convergence would in general be too small. What we show, though, is that there is a way to form functions $f_{p,q}(z)$, each consisting of some of the summands in (3.8), such that the Taylor series of each converges in a wide region, and such that for each j, $f(z_j)$ can be written as a sum of a subset of the $f_{p,q}(z_j)$, where each of the $f_{p,q}(z)$ that appears in the sum converges rapidly at $z = z_j$.

Theorem 4.1. Given any constant $c \geq 1$, there is a constant c' such that if a_k, $1 \leq k \leq n$, are complex numbers given to an accuracy of $\pm n^{-10c}$ with $|a_k| \leq 1$, and β_k, $1 \leq k \leq n$ are real numbers with $0 \leq \beta_k < n$, given also to an accuracy of $\pm n^{-10c}$, and moreover if $|\beta_k - j| \geq n^{-8c}$ for $1 \leq k \leq n$ and all integers j, then one can evaluate the function

$$f(z) = \sum_{k=1}^{n} \frac{a_k}{z - b_k} ,$$

$$b_k = \exp(2\pi i \beta_k / n), \quad 1 \leq k \leq n ,$$

at all the points

$$z_j = \exp(2\pi i j/n), \quad 0 \leq j \leq n-1 ,$$

to within $\pm n^{-c}$ using $\leq c' n (\log n)^2$ arithmetic operations with complex numbers of $\leq c' \log n$ bits in $\leq c' n (\log n)^2$ bits of storage.

Proof. Let $< x >$ denote the nearest integer to x,

$$<x> \; = \; \lfloor x + 1/2 \rfloor \,,$$

and let $\|x\|_n$ denote the ‘‘cyclic distance’’ on $\mathbb{R}/(n\,Z)$;

$$\|x\|_n \;=\; \min_k \; |x - kn| \;.$$

For $1 \le k \le n$ and $0 \le j$, define

$$T(k,j) \;=\; \max \{r\colon \text{ for some } s \ge 0, \; \|j - s3^r\|_n \le (3^r - 1)/2 \text{ and } \|\beta_k - s3^r\|_n \ge 3^r - 1\} \,,$$

and for nonnegative integers p and q, $p < n$, $3^q \le n/2 + 1$, $p3^q \le n-1 + (3^q - 1)/2$,

$$I_{p,q} \;=\; \{k\colon 1 \le k \le n, \; T(k, p3^q) \;=\; q\} \;.$$

Note that if $T(k,j) = r$, then for some s, $3^r - 1 \le \|\beta_k - s3^r\|_n \le n/2$, so we only need to consider q with $3^q \le n/2 + 1$. (The determination of $T(k,j)$ should be based on the known $O(\log n)$ bits of β_k; due to the high precision of the given values of β_k, any mistakes made here will not perceptibly affect the final answer.) For fixed k and q, $T(k, p3^q) = q$ implies

$$\|\beta_k \;-\; <p/3>\; 3^{q+1}\|_n \;<\; 3^{q+1} - 1 \;,$$

which is possible for at most 6 values of p with $0 \le p \le n-1$, $p3^q \le n-1 + (3^q - 1)/2$. Therefore each k belongs to at most 10 log n of the $I_{p,q}$, and they can be determined fast.

Define

$$f_{p,q}\,(z) \;=\; \sum_{k \,\varepsilon\, I_{p,q}} \frac{a_k}{z - b_k} \;.$$

We note first that for any j, $0 \le j \le n-1$, $Q = \left\lfloor \log_3 \, (n/2+1) \right\rfloor$,

$$f(z_j) = \sum_{q=0}^{Q} \sum_{\substack{k=1 \\ T(k,j) = q}} \frac{a_k}{z_j - b_k} = \sum_{q=0}^{Q} f_{<j3^{-q}>,q}\ (z_j)\ ,$$

since in this range,

$$I_{<j3^{-q}>,q} = \{k\colon 1 \le k \le n,\ \ T(k, <j3^{-q}>3^q) = q\}$$
$$= \{k\colon 1 \le k \le n,\ \ T(k,j) = q\}\ .$$

Now to evaluate a function $f_{p,q}\ (z)$ at all points z_j with $<j3^{-q}> = p$, we use a power series expansion around the point

$$z_{p,q} = \exp(2\pi i p 3^q / n)\ .$$

Note that $<j3^{-q}> = p$ is equivalent to $\|j - p3^q\|_n \le (3^q - 1)/2$, while $k \ \varepsilon\ I_{p,q}$ implies $\|\beta_k - p3^q\|_n \ge 3^q - 1$, so that

$$\left| \frac{z_j - z_{p,q}}{b_k - z_{p,q}} \right| \le \frac{1}{2 \cos \frac{\pi}{4}} = 2^{-1/2}\ .$$

Therefore for $R \ge 1000c \log n$, say,

$$\frac{a_k}{z_j - b_k} = \frac{a_k}{z_{p,q} - b_k}\ (1 - \frac{z_{p,q} - z_j}{z_{p,q} - b_k})^{-1}$$
$$= \sum_{r=0}^{R} \frac{a_k}{(z_{p,q} - b_k)^{r+1}}\ (z_{p,q} - z_j)^r + O(n^{-100c})\ ,$$

and if we choose some scaling factor ξ_q close to $2\pi 3^q/n$, possibly a power of 2, then we will have

$$f_{p,q}\ (z_j) = \sum_{r=0}^{R} A_{p,q,r} \left[\frac{z_{p,q} - z_j}{\xi_q} \right]^r + O(n^{-50c})\ ,$$

where

$$A_{p,q,r} \ = \sum_{k \,\varepsilon\, I_{p,q}} \frac{a_k \ \xi_q^r}{(z_{p,q} \ - \ b_k)^{r+1}} \ .$$

Hence we will obtain

$$f(z_j) \ = \ \sum_{r=0}^{R} \ \sum_q \ A_{<j3^{-q}>,q,r} \ \xi_q^{-r} \ (z_{p,q} \ - \ z_j)^r \ + \ O(n^{-40c}) \ .$$

If we compute the $A_{p,q,r}$ to within $\pm \ n^{-20c}$, say, which takes $O(n \ (\log \ n)^2)$ operations on complex numbers of $O(\log \ n)$ bits and $O(n \ (\log \ n)^2)$ bits of storage (all constants implied by the O-notation depending on c only), then we can compute each individual $f(z_j)$ to within $O(n^{-20c})$ in $O((\log \ n)^2)$ additional operations.

5. Extensions of basic results

The basic method presented in the preceding sections can be extended in several ways. In particular, it is possible to partially overcome the main disadvantage of the algorithm of Theorem 1.1, namely large space requirement. By breaking up the exponential sums of the form (2.11) into smaller sums, it is easy to obtain the following generalization of Theorem 1.1.

Theorem 5.1. For any $a \ \varepsilon \ [0, \ 1/2]$ *and any constants* δ, σ, *and* c_1, *there is an effectively computable constant* $c'_2 \ = \ c'_2 \ (\delta, \ \sigma, \ c_1, \ a)$ *and an algorithm that for every* $T \ > \ 0$ *will perform* $\leq c'_2 \ T^{1/2+\delta}$ *operations on numbers of* $\leq c'_2 \ \log \ T$ *bits using* $\leq c'_2 \ T^{a+\delta}$ *bits of storage and will then be capable of computing any value* $\zeta(\sigma \ + \ it)$ *for* $T \leq t \leq T \ + \ T^a$ *to within* $\pm \ T^{-c_1}$ *in* $\leq c'_2 \ T^{\delta}$ *operations using the precomputed values.*

The Riemann-Siegel formula was the starting point of our method and was instrumental in keeping down the precomputation time and space. However, the same basic idea could be used with the Euler-Maclaurin formula. In the case of Dirichlet *L*-functions, it could be employed either with the Euler-Maclaurin formula or the generalized Riemann-Siegel formula of Davies [5] and Deuring [6].

A somewhat more interesting application of our method is to the computation of Epstein zeta functions [3, 15]. There are formulas for them in which the main contribution to the evaluation of one of these function at s comes either from a sum of $K_{1/2-s}(z)$ for various values of z [20], where

$$K_s(z) = \frac{1}{2} \int_0^\infty \exp\{-\frac{z}{u}(u + u^{-1})\} u^{s-1} du$$

is the modified Bessel function of the second kind, or else from a sum of $G(s,z)$ for various values of z [2], where

$$G(s,z) = \int_1^\infty u^{s-1} e^{-zu} du$$

is a modified incomplete gamma function. It is again easy to reduce the basic problem of computing these sums of integrals to that of computing the first few derivatives of such sums at an evenly spaced grid of points. If $s = \sigma + iT + i\eta j$, $0 \le j \le n-1$, though (cf. Section 2), then for ω an n-th root of unity (cf. Section 3),

$$\sum_{j=0}^{n-1} G(\sigma + iT + i\eta j,\, z)\, \omega^j = \int_1^\infty e^{-zu} u^{\sigma - 1 + iT} \frac{1 - u^{i\eta n}}{1 - u^{i\eta}\omega} du ,$$

and similar expressions arise when we consider sums of $K_{1/2-s}(z)$. When this is summed over the various values of z, we reduce to the problem of evaluating integrals with respect to u involving rational functions $F(\omega, u^{i\eta})$ of two variables, where we need to evaluate these integrals at all n-th roots of unity ω. It appears that this could often be done fast using the method of Section 4.

Our basic method might very well be practical for computing $\pi(x)$ and other arithmetical functions, using the algorithms of [16]. However, very substantial work would be required to actually put it in practice, since good error bounds would have to be obtained for the remainders in the Riemann-Siegel formula away from the critical line, and good contours of integration, kernels, and quadrature rules would have to be chosen (cf. [16]). On the other hand, since the Riemann-Siegel formula is much simpler on the critical line $\sigma = 1/2$, and very good estimates for it are known there [8], it would be much easier to implement our method there for the purpose of computing zeros of the zeta function.

References

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, 1974.

[2] P. T. Bateman and E. Grosswald, On Epstein's zeta function, *Acta Arith.*, *9* (1964), 365-384.

[3] E. Bombieri and D. Hejhal, manuscript in preparation.

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

[5] D. Davies, An approximate functional equation for Dirichlet *L*-functions, *Proc. Royal Soc. Ser. A 284* (1965), 224-236.

[6] M. Deuring, Asymptotische Entwicklungen der Dirichletschen L-Reihen, *Math. Ann.*, *168* (1967), 1-30.

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

[8] W. Gabcke, Neue Herleitung und explicite Restabschätzung der Riemann-Siegel-Formel, Ph.D. Dissertation, Göttingen, 1979.

[9] N. Gastinel, Inversion d'une matrice généralisant la matrice de Hilbert, *Chiffres 3* (1960), 149-152.

[10] A. Gerasoulis, M. D. Grigoriadis, and Liping Sun, A fast algorithm for Trummer's problem, report LCSR-TR-77, Computer Science Dept., Rutgers Univ.

[11] G. Golub, Trummer's problem, *SIGACT News 17* (1985), no. 2, p. 17.2.12.

[12] *Handbook of Mathematical Functions*, M. Abramowitz and I. A. Stegun, eds., National Bureau of Standards, 9th printing, 1970.

[13] A. Ivic, *The Riemann Zeta-function*, Wiley, 1985.

[14] J. C. Lagarias, V. S. Miller, and A. M. Odlyzko, Computing $\pi(x)$: The Meissel-Lehmer method, *Math. Comp.*, *44* (1985), 537-560.

[15] J. C. Lagarias and A. M. Odlyzko, On computing Artin *L*-functions in the critical strip, *Math. Comp.*, *33* (1979), 1081-1095.

[16] J. C. Lagarias and A. M. Odlyzko, Computing $\pi(x)$: An analytic method, *J. Algorithms*, to appear.

[17] 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. IV., *Math. Comp. 46* (1986), 667-681.

[18] A. M. Odlyzko, On the distribution of spacings between zeros of the zeta function, *Math. Comp.*, to appear.

[19] A. M. Odlyzko, Distribution of zeros of the Riemann zeta function: Conjectures and computations, manuscript in preparation.

[20] A. Selberg and S. Chowla, On Epstein's zeta function, *J. reine angew. Math.*, *227* (1967), 86-110.

[21] C. L. Siegel, Über Riemanns Nachlass zur analytischen Zahlentheorie, *Quellen und Studien zur Geschichte der Math. Astr. Phys. 2* (1932), 45-80. Reprinted in C. L. Siegel, *Gesammelte Abhandlungen*, Springer, 1966, Vol. 1, pp. 275-310.

[22] E. L. Titchmarsh, *The Theory of the Riemann Zeta-function*, Oxford Univ. Press, 1951.

[23] M. R. Trummer, An efficient implementation of a conformal mapping method using the Szegö kernel, *SIAM J. Num. Anal.*, to appear.

[24] A. M. Turing, A method for the calculation of the zeta-function, *Proc. London Math. Soc. ser. 2*, *48* (1943), 180-197.