

New analytic algorithms in number theory

A. M. Odlyzko

AT&T Bell Laboratories

Murray Hill, NJ 07974

USA

1. Introduction

Several new number theoretic algorithms are sketched here. They all have the common feature that they rely on bounded precision computations of analytic functions. The main one of these algorithms is a new method, due to A. Schönhage and the author, of calculating values of the Riemann zeta function at multiple points. This method enables one to verify the truth of the Riemann Hypothesis (RH) for the first n zeros in what is expected to be $n^{1+o(1)}$ operations as $n \rightarrow \infty$, as opposed to about $n^{3/2}$ operations for earlier methods. This technique can also be extended to the evaluation of other zeta and L -functions.

Methods for the approximate computation of zeta and L -functions have been shown by J. C. Lagarias and the author to lead to fast algorithms for the exact computation of certain integer-valued number theoretic functions. It is shown below that these methods enable one to compute $\pi(x)$, the number of primes $\leq x$, in $x^{1/2+o(1)}$ operations as $x \rightarrow \infty$.

Numerical computations of zeros of the zeta function cannot ever prove the RH, but if there is a counterexample to the RH, such computations might find it. This was undoubtedly the main motivation for the computations that have been done. The first few nontrivial zeros were computed by hand by Riemann [Ed, Si], and further computations, by hand or with the help of electromechanical technology, were carried out in the early

1900's, culminating with the verification of the RH for the first 1041 zeros (i.e., the 1041 zeros in the upper half-plane $\text{Im}(s) > 0$ that are closest to the real axis $\text{Im}(s) = 0$) by Titchmarsh and Comrie [25]. After World War II, these computations were extended using electronic digital computers. The latest result in this direction is the verification of the RH for the first $1.5 \cdot 10^9$ zeros [16], a computation that involved about two months on a modern supercomputer, and used the same basic method for computing the zeta function that was employed by Titchmarsh and Comrie.

Computations of the zeros of the Riemann zeta function and related functions (such as Dirichlet L -functions and Epstein zeta functions) are also useful in providing evidence for and against various other conjectures, such as the Montgomery pair correlation conjecture [17, 18], and some related conjectures which predict that the zeros of $\zeta(s)$ ought to behave like eigenvalues of random hermitian matrices, (whose distribution is known relatively well, since they have been studied by physicists who use them to model energy levels in many-particle systems). For a discussion of these conjectures and the numerical evidence about them, see [17, 18, 19, 20]. Another, similar, motivation for computing zeros of zeta and L -functions comes from trying to understand better the differences between those functions, like the Riemann zeta function and Dirichlet L -function, that are expected to satisfy the RH, and those, such as the Epstein zeta functions, which are in most cases expected to violate the RH. See [2, 9] for some result on these problems. Other applications for approximate values of the zeta function arise in disproving certain conjectures, see [10, 21].

As will be explained briefly in Section 2, the problem of computing an approximate value of a zero of $\zeta(s)$, for example, and proving that this zero is on the critical line, can

be reduced to the problem of evaluating $\zeta(s)$ to a certain accuracy. In Section 3, the previous methods of computing $\zeta(s)$ will be presented. The fastest of them, the Riemann-Siegel formula, requires on the order of $t^{1/2}$ operations to evaluate $\zeta(1/2 + it)$ to within $\pm t^{-c}$, say, for $t > 1$, $c > 0$. In Section 4 new methods for evaluating $\zeta(s)$ at multiple points are presented. The fastest of them, due to A. Schönhage and the author [22], can compute any single value of $\zeta(1/2 + it)$ with $T \leq t \leq T + T^{1/2}$ to within $\pm T^{-c}$ in $T^{o(1)}$ operations (on numbers of $O(\log T)$ bits), provided a precomputation involving $T^{1/2 + o(1)}$ operations is carried out beforehand and $T^{1/2 + o(1)}$ storage locations are available. This method leads to an algorithm for numerically verifying the RH for the first n zeros in what is expected to be $n^{1 + o(1)}$ operations, as opposed to about $n^{3/2}$ operations using the older methods. These techniques extend to the computation of other zeta and L -functions.

The computation of zeros to a specified accuracy requires only the computations of approximate values of the appropriate function. It turns out that such approximate values can be used to compute exactly various number theoretic functions, such as $\pi(x)$, the number of primes $\leq x$. The reason for this is related to the ‘‘explicit formulas’’ of Guinand [8] and Weil [29], which relate arithmetic functions to zeros of the appropriate zeta and related functions. An even older formula of Landau [15] (see also [7]) says that for any fixed $y > 0$, as $T \rightarrow \infty$,

$$\sum_{0 < \text{Im}(\rho) < T} e^{\rho y} = \begin{cases} -\frac{T}{2\pi} \log p + O(\log T) & \text{if } y = \log p^m, \\ O(\log T) & \text{if } y \neq \log p^m, \end{cases}$$

where p denotes a prime and m a positive integer, and ρ runs over the zeros of the zeta

function. The formula above can be used to test integers for primality. This idea has occurred to many people, but unfortunately it does not seem to yield an efficient algorithm. However, modifications of such ideas can be used to compute relatively efficiently functions such as $\pi(x)$, as was shown by J. C. Lagarias and the author [13, 14]. Using the latest algorithm for evaluating the zeta function, it is possible to compute $\pi(x)$ in $x^{1/2 + o(1)}$ steps as $x \rightarrow \infty$, which is faster than the best known combinatorial algorithm [12], which requires around $x^{2/3}$ operations. This new algorithm for $\pi(x)$ and similar functions is sketched briefly in Section 5.

2. Zeta function and the numerical verification of the RH

The zeta function is defined for $\text{Re}(s) > 1$ by

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}, \quad (2.1)$$

but it can be extended by analytic continuation to an analytic function throughout $\mathbb{C} \setminus \{1\}$, and it has a first order pole at $s = 1$. If we define, as usual,

$$\xi(s) = \pi^{-s/2} \Gamma(s/2) \zeta(s), \quad (2.2)$$

then $\xi(s)$ satisfies the functional equation

$$\xi(s) = \xi(1-s), \quad s \in \mathbb{C} \setminus \{0, 1\}. \quad (2.3)$$

Since $\xi(s)$ is real for real s , we have $\xi(\bar{s}) = \overline{\xi(s)}$, and so for $t \in \mathbb{R}$,

$$\xi(1/2 + it) = \xi(1 - (1/2 + it)) = \xi(1/2 - it) = \overline{\xi(1/2 + it)}, \quad (2.4)$$

which implies that $\xi(1/2 + it) \in \mathbb{R}$. Therefore if we find two values t_1 and t_2 such that $\xi(t_1) < 0 < \xi(t_2)$ (and the errors in the computed values of $\xi(t_1)$ and $\xi(t_2)$ are

smaller than those values, so we can be certain that the two inequalities are valid), then we can conclude that there is a value t' , $t_1 < t' < t_2$, such that $\zeta(1/2 + it') = 0$. In this case we know that the zero is right on the critical line, not just in some neighborhood of it.

All the numerical verifications of the RH have relied on the use of the intermediate value theorem as sketched above. This theorem gives a lower bound for the number of zeros of the zeta function that are on the critical line up to some height. To assert that all of the zeros up to some height are on the critical line, one can use the principle of the argument to determine the total number of zeros up to that height and if that agrees with the number of zeros that have been located on the critical line, the verification is completed. In practice it is not even necessary to use the principle of the argument, since there is an elegant method of Turing [5, 20, 28], based on a theorem of Littlewood about the zeta function, which usually enables one to conclude that all the zeros up to some height are on the critical line based only on computations of $\xi(s)$ on that line.

It should be mentioned that the above strategy for numerically verifying the RH is not guaranteed to work, even if the RH is true, since there might be multiple zeros on the critical line. However, such zeros have not been encountered yet and are not expected to exist. In practice, using the sophisticated strategies for locating points t at which to evaluate $\xi(1/2 + it)$ that have been developed [16], it takes on average about 1.2 evaluation of $\xi(s)$ to establish that a zero is on the critical line.

The above discussion serves to reduce the problem of numerically verifying the RH to that of computing $\xi(s)$. The next section discusses that problem.

3. Methods for calculating the zeta function

For purposes of numerically verifying the RH or for computing $\pi(x)$ by the algorithm to presented in Section 5, it is sufficient to be able to compute $\zeta(s)$ to within $\pm |s|^{-c}$ for various constants $c > 0$. For some purposes (such as those of Section 5) it is necessary to calculate $\zeta(s)$ for $\text{Re}(s) > 1$, but to keep the presentation simple we will deal only with $\text{Re}(s) = 1/2$ here. (See [22] for more details.) Since $\Gamma(s/2)$ and $\pi^{-s/2}$ are very easy to compute accurately, computing $\xi(1/2 + it)$ is essentially the same as computing $\zeta(1/2 + it)$.

The simplest method for calculating $\zeta(s)$ is to apply the Euler-Maclaurin summation formula to (2.1). This gives an analytic continuation of $\zeta(s)$ to all of $\mathbb{C} \setminus \{1\}$, and enables one to compute $\zeta(s)$ anywhere to arbitrary accuracy by adjusting the parameters in the formula appropriately. However, this method is not very efficient, since to compute $\zeta(1/2 + it)$ to $\pm t^{-1}$, say, already takes about t steps. Still, this was the method that was used by the first few people to publish calculations of the zeros of the zeta function.

A much more efficient method for computing $\zeta(1/2 + it)$ was known to Riemann, and was used by him to compute the first few zeros. It is one of what Hardy and Littlewood called approximate functional equation, but it has the nice feature that the error term is given by an asymptotic expansion. Discovered in Riemann's unpublished papers by Siegel, it became known as the Riemann-Siegel formula [5, 23]. It has been used for all large-scale computations of zeros of the zeta function since the 1930's. (The only other method to have been proposed so far, that of Turing [27], was meant to be

used at relatively small heights, where it turned out not to be necessary.)

Let

$$\theta(t) = \arg \left[\pi^{-it/2} \Gamma(1/4 + it/2) \right], \quad (3.1)$$

$$Z(t) = \exp(i\theta(t)) \zeta(1/2 + it), \quad (3.2)$$

so that

$$Z(t) = \frac{\zeta(1/2 + it)}{|\pi^{-1/4 - it/2} \Gamma(1/4 + it/2)|}.$$

Also let

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

Then the Riemann-Siegel formula [5, 6, 11, 20, 23, 25, 26] says that for any $k \geq 0$,

$$Z(t) = 2 \sum_{n=1}^m n^{-1/2} \cos(t \log(n) - \theta(t)) \quad (3.3)$$

$$+ (-1)^{m+1} \tau^{-1/4} \sum_{j=0}^k \Phi_j(z) (-1)^j \tau^{-j/2} + R_k(\tau),$$

where

$$R_k(\tau) = O(\tau^{-(2k+3)/4}),$$

and the $\Phi_j(z)$ are certain entire functions. Explicit bounds for the $R_k(\tau)$ are known (the best ones being due to Gabcke [6]), and bounds for the error made in computing the $\Phi_j(x)$ are also known. Thus the entire difficulty in the computation of $Z(t)$ resides in the need to evaluate the sum of cosines in (3.3), which takes on the order of $t^{1/2}$ steps.

4. New methods for evaluating the zeta function

The Riemann-Siegel formula is still the fastest known method for computing a single value of $\zeta(1/2 + it)$ to medium accuracy ($\pm t^{-c}$ for some $c > 0$) for large t . Here we will show that if values of $\zeta(1/2 + it)$ at a large set of closely spaced t 's are desired, one can obtain much faster algorithms.

In the previous section we showed that the bottleneck in using the Riemann-Siegel formula is the need to evaluate the sum of about $t^{1/2}$ cosines. (In the Euler-Maclaurin formula, one has to evaluate a similar sum of about t cosines.) Note that

$$2 \sum_{n=1}^m n^{-1/2} \cos(t \log(n) - \theta(t)) = \operatorname{Re} e^{-i\theta(t)} \sum_{n=1}^m 2n^{-1/2} e^{it \log n},$$

so that it suffices to find an efficient method to compute the complex-valued function

$$f(t) = \sum_{n=1}^m 2n^{-1/2} e^{it \log n} \tag{4.1}$$

for the values of t that are of interest. We will count the number of elementary arithmetic operations on numbers of $O(\log t)$ digits. (See [22] for more on the model of computation that's used.) What makes the methods below work is that there is a lot of structure in sums of the form (4.1), and if many of them are to be computed, one can exploit this structure to lower the average cost of an evaluation.

The first observation to make is that if we are interested in computing $f(t)$ (and thus $\zeta(1/2 + it)$) for $T \leq t \leq T + T^\alpha$ for some $0 < \alpha \leq 1$, say, then it suffices to compute $f(t)$ and a few similar sums at an evenly spaced grid of points, $t = t_0, t_1, \dots, t_H$, where

$$t_j = T + j\delta \quad , \quad 0 \leq j \leq H = \lfloor T^{\alpha + \eta} \rfloor, \quad \delta = T^{-\eta} \quad ,$$

for some $\eta \in (0, 1/10)$. (Eventually η will go to zero.) The reason for this is that

$$f^{(k)}(t) = \sum_{n=1}^m 2 i^k n^{-1/2} (\log n)^k e^{it \log n} \quad , \quad (4.2)$$

so that

$$|f^{(k)}(t)| \leq 5(\log m)^k m^{1/2} \quad .$$

Therefore if we wish to compute $f(t)$ for some t , $T \leq t \leq T + T^\alpha$, to within $\pm T^{-c}$, we can use the Taylor series expansion around the nearest t_j , provided we have precomputed the values of $f^{(k)}(t_j)$ for $k \leq (c + 100)\eta^{-1}$, say. Since the derivatives $f^{(k)}(t)$ are of the same form as $f(t)$, we have reduced the problem to that of computing series of the form

$$g(t) = \sum_{n=1}^m a_n e^{it \log n} \quad (4.3)$$

at $t = t_j = T + j\delta$ for $0 \leq j \leq H$. We should note that m depends on t ,

$$m = \lfloor (2\pi)^{-1/2} t^{1/2} \rfloor.$$

The trivial way to compute $g(t)$ at each of the t_j takes about mH operations. One way to obtain a faster method is to use fast matrix multiplication methods. For simplicity, suppose that we are trying to compute $g(\delta k)$ for $0 \leq k < R^2$, where $\delta \leq 1$. Consider the matrix $C = (c_{jh})$, $0 \leq j, h \leq R-1$, where

$$c_{jh} = \begin{cases} a_h \exp(ij(R-1)\delta \log h) \quad , & \text{for } h \leq \beta_j \quad , \\ 0 \quad , & \text{for } h > \beta_j \quad , \end{cases} \quad (4.4)$$

where the β_j will be specified later. Also, let $D = (d_{hk}), 0 \leq h, k \leq R - 1$, be the matrix with

$$d_{hk} = \exp(i k \delta \log h) . \quad (4.5)$$

Then the matrix $CD = (b_{jk}), 0 \leq j, k \leq R - 1$, has

$$b_{jk} = \sum_{h=1}^{\beta_j} a_h \exp(i[j(R-1) + k] \delta \log h) . \quad (4.6)$$

If we select $\beta_j = \lfloor (2\pi)^{-1/2} j^{1/2} (R-1)^{1/2} \rfloor$, then b_{jk} will differ from $g(\delta(j(R-1) + k))$ by only a few terms, which can be easily computed. The trivial method for multiplying C and D takes about R^3 operations, which is not much better than the ordinary $O(R^3 \delta^{1/2})$ method of evaluating each sum of the form (4.6). However, if we use fast matrix multiplication methods, of which the latest one allows one to multiply two $n \times n$ matrices in time $O(n^{2.479})$ [24], we obtain a method with running time about $R^{2.479}$. Since in practice we would have $\delta = R^{-o(1)}$, $R = T^{1/2 + o(1)}$, this would let us compute all values of $\zeta(1/2 + it)$ for $0 \leq t \leq T$ in time $T^{o(1)}$ to within $\pm T^{-c}$ after an initial precomputation that requires $\leq T^{1.24}$ operations, and would thus be more efficient than using the Riemann-Siegel formula. (This method could also be modified to work on shorter intervals, but not as efficiently.) Unfortunately fast matrix multiplication methods are completely impractical (except possibly for Strassen's original $O(n^{2.807\dots})$ method for multiplying two $n \times n$ matrices [1]), so that the above technique is of purely theoretical significance.

An algorithm that is both theoretically efficient and appears to be practical can be obtained by a different approach. Suppose that we wish to compute $g(t_j), 0 \leq j \leq H$,

where $t_j = T + j\delta$, $0 < \delta < 1/10$, and $H \leq T^{1/2}$. Let $M = \lfloor (2\pi)^{-1/2} T^{1/2} \rfloor$, and

$$g^*(t) = \sum_{n=2}^M a_n e^{it \log n} .$$

Then $g(t_j)$ and $g^*(t_j)$ differ at most by two terms (for T large), and so it suffices to compute the $g^*(t_j)$ efficiently. Choose $N = 2^r$ so that $H < N \leq 2H$, and let

$$\omega = \exp(2\pi i/N) .$$

Let

$$h(k) = \sum_{j=0}^{N-1} g^*(t_j) \omega^{jk} , \quad 0 \leq k < N .$$

The inverse Fourier transform gives

$$g^*(t_j) = \frac{1}{N} \sum_{k=0}^{N-1} h(k) \omega^{-jk} , \quad 0 \leq j \leq H ,$$

so if we can compute all the $h(k)$ accurately, we will be able to recover the $g^*(t_j)$ in $O(N \log N)$ operations using the Fast Fourier Transform [1]. (All the numbers that come up in this procedure are of moderate size, so the required precision does not cause too much of a problem.)

The $h(k)$ can be rewritten as

$$\begin{aligned} h(k) &= \sum_{n=2}^M a_n e^{T \log n} \sum_{j=0}^{N-1} e^{ij \delta \log n} \omega^{jk} \\ &= \sum_{n=2}^M \frac{a_n e^{iT \log n}}{1 - \omega^k e^{i\delta \log n}} = \sum_{n=2}^M \frac{-a_n e^{-i(\delta - T) \log n}}{\omega^k - e^{-i\delta \log n}} . \end{aligned}$$

(We are assuming here for simplicity that $\omega^k - \exp(-i\delta \log n)$ is not too small for all

k and n .

Those n for which the above difference is small or even zero for some k have to be treated separately, cf. [22].) Thus evaluating $h(k)$ for $0 \leq k \leq N-1$ is no harder than evaluating a rational function of the type

$$h^*(z) = \sum_{n=1}^N \frac{\alpha_n}{z - \beta_n} \quad (4.7)$$

at the points $z = \omega^k$, $0 \leq k \leq N-1$, where the α_n and β_n can complex numbers with $|\beta_n| = 1$, the α_n are not too large, and $|\omega^k - \beta_n|$ is not too small for all k and n . It is easy to show functions of the form (4.7) can be evaluated in $O(N (\log N)^2)$ operations over some finite fields, by employing Fast Fourier Transforms methods. What is more surprising is evaluations in $O(N (\log N)^2)$ operations on numbers of $O(\log N)$ bit, can also be performed. This was shown by A. Schönhage, and is presented in [22]. The basic idea is to use Taylor series expansions of $h^*(z)$ around a small set of points. A set of functions $h_{p,q}(z)$ is defined, each of which consists of some of the terms in (4.7), such that the Taylor series of each $h_{p,q}(z)$ around a certain point converges in a relatively large region, and such that for each k , $h^*(\omega^k)$ can be written as a sum of a subset of the $h_{p,q}(\omega^k)$. Full details can be found in [22].

5. Analytic algorithms for arithmetic functions

In this section we will sketch the analytic algorithms for certain arithmetic function that have been developed recently. For simplicity of presentation, we will restrict ourselves to the problem of computing $\pi(x)$. Full details and extensions to other functions can be found in [14].

In spite of extensive interest in computing $\pi(x)$, until recently no algorithm with proved running time of $O(x^{1-\varepsilon})$ for some $\varepsilon > 0$ was known. (See [12, 13] for historical references.) In [12] a modification of the Meissel-Lehmer algorithm, which uses combinatorial sieving, was developed, which runs in time $x^{2/3+o(1)}$ and space $x^{1/3+o(1)}$. This algorithm has been implemented and has been used to compute values of $\pi(x)$ for various x up to $x = 4 \cdot 10^{16}$. At about the same time, a new analytic algorithm for computing $\pi(x)$ was developed [13, 14]. It relies on numerical integration of analytic transforms involving the zeta function. When one uses the Euler-Maclaurin formula to compute the zeta function, the resulting algorithm requires time $x^{2/3+o(1)}$ and space $x^{o(1)}$. With the Riemann-Siegel formula, the running time decreases to $x^{3/5+o(1)}$ and the space requirement stays at $x^{o(1)}$. With the use of the new algorithm described in Section 4, time drops to $x^{1/2+o(1)}$, but space required becomes $x^{1/4+o(1)}$.

The analytic algorithm for $\pi(x)$ is based on an old formula of Riemann, namely that

$$\pi(x) + \sum_{j=2}^{\infty} \frac{1}{j} \pi(x^{1/j}) = \frac{1}{2\pi i} \int_{2-i\infty}^{2+i\infty} \frac{x^s}{s} \log \zeta(s) ds \quad (5.1)$$

for $x \neq p^m$ for any prime p and $m \in \mathbb{Z}^+$. (If $x = p^m$, $(2m)^{-1}$ must be subtracted from the left side of (5.1).) One of the things which make the analytic algorithm work is that $\pi(x)$ is always an integer, and so it suffices to compute it to within $\pm 1/3$, say. The sum of the terms on the left side of (5.1) with $j \geq 2$ can easily be computed to within $\pm 1/10$ in about $x^{1/2}$ steps. (Even fewer are required if one uses the algorithm recursively.) Thus if we could compute the integral on the right side of (5.1) efficiently to within $\pm 1/10$, say, we would compute $\pi(x)$ rapidly. Unfortunately the integral in (5.1) is not even absolutely convergent, and no way has been found to compute it fast.

To overcome the problem of slow convergence of the integral in (5.1), we use a different formula, namely

$$\sum_{p \leq x} c(p) + \sum_{\substack{p,m \\ p^m \leq x \\ m \geq 2}} \frac{1}{m} c(p^m) = \frac{1}{2\pi i} \int_{2-i\infty}^{2+i\infty} F(s) \log \zeta(s) ds, \quad (5.2)$$

where

$$c(u) = \frac{1}{2\pi i} \int_{2-i\infty}^{2+i\infty} F(s) u^{-s} ds, \quad (5.3)$$

and where this formula holds whenever $F(s)$ is a sufficiently nice function. ($F(s)$ analytic in $\text{Re}(s) > 1$, $|F(s)| = |s|^{-2}$ as $|s| \rightarrow \infty$, is sufficient but not necessary, for example.) We would like to choose $F(s)$ so that (5.2) behaves like (5.1); i.e., $c(p) = 1$ for $p \leq x$ and $c(p) = 0$ for $p > x$. Unfortunately, the uniqueness theorem for Mellin transforms says that is impossible unless $F(s) = s^{-1} x^s$. We therefore select a somewhat different $F(s)$, in which the sum on the left side of (5.2) is relatively close to that on the left side of (5.1). To be precise, we select $F(s)$ so that

- (i) $F(2 + it) \rightarrow 0$ rapidly as $|t| \rightarrow \infty$, so that the integral in (5.2) is approximated well by a finite integral

$$\frac{1}{2\pi i} \int_{2-iT}^{2+iT} F(s) \log \zeta(s) ds; \quad (5.4)$$

- (ii) Individual values of $F(s)$ can be computed rapidly, so that the integral above can be evaluated by numerical integration;
- (iii) $c(p) = 1$ for primes $p \leq x - y$, $c(p) = 0$ for primes $p \geq x$, and $0 \leq c(p) \leq 1$

for primes $p \in (x - y, x)$;

(iv) Individual values of $c(k)$ can be computed efficiently.

Such $F(s)$ can be found for any y and T with the property that $yT \geq x^{1 + o(1)}$, see [13, 14]. (The requirement that $yT \geq x^{1 + o(1)}$ is caused by the uncertainty principle of Fourier analysis, which says that a function and its transform cannot both decrease too rapidly.) The contribution of the sum

$$\sum_{\substack{p, m \\ p^m \leq x \\ m \geq 2}} \frac{1}{m} c(p^m)$$

can be evaluated to within $\pm 1/100$ in $x^{1/2 + o(1)}$ operations. The difference

$$\pi(x) - \sum_{p \leq x} c(p) = \sum_{x - y < p \leq x} (1 - c(p))$$

can be evaluated to within $\pm 1/100$ by finding all the primes in $[x - y, x]$ and computing their contribution, and this can be done in $yx^{o(1)}$ operations. Finally, the evaluation of the integral (5.4) can be reduced to evaluating $\zeta(2 + it)$ at a grid of points between $-T$ and T , spaced $x^{-o(1)}$ apart, and by the algorithm of Section 4 this can be done in $T x^{o(1)}$ operations. Thus the running time of the algorithm is

$$x^{1/2 + o(1)} + y x^{o(1)} + T x^{o(1)},$$

and selecting $y = T = x^{1/2 + o(1)}$, we obtain the desired running time bound. Full details are presented in [14].

REFERENCES

1. A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, 1974.
2. E. Bombieri and D. A. Hejhal, manuscript in preparation.
3. D. Davies, An approximate functional equation for Dirichlet L -functions, *Proc. Royal Soc. Ser. A* 284 (1965), 224-236.
4. M. Deuring, Asymptotische Entwicklungen der Dirichletschen L -Reihen, *Math. Annalen* 168 (1967), 1-30.
5. H. M. Edwards, *Riemann's Zeta Function*, Academic Press, 1974.
6. W. Gabcke, Neue Herleitung und explizite Restabschätzung der Riemann-Siegel-Formel, Ph.D. Dissertation, Göttingen 1979.
7. S. M. Gonek, A formula of Landau and mean values of $\zeta(s)$, pp. 92-97 in *Topics in Analytic Number Theory*, S. W. Graham and J. D. Vaaler, eds., Univ. Texas Press, 1985.
8. A. P. Guinand, A summation formula in the theory of prime numbers, *Proc. London Math. Soc.* (2) 50 (1948), 107-119.
9. D. A. Hejhal, Zeros of Epstein zeta functions and supercomputers, *Proc. 1986 Intern. Congress Math.*, to appear.
10. A. E. Ingham, On two conjectures in the theory of numbers, *Amer. J. Math.* 64 (1942), 313-319.

11. A. Ivic, *The Riemann Zeta-function*, Wiley, 1985.
12. J. C. Lagarias, V. S. Miller, and A. M. Odlyzko, Computing $\pi(x)$: the Meissel-Lehmer method, *Math. Comp.*, 44 (1985), 537-560.
13. J. C. Lagarias and A. M. Odlyzko, New algorithms for computing $\pi(x)$, pp. 176-193 in *Number Theory: New York 1982*, D. V. Chudnovsky, G. V. Chudnovsky, H. Cohn, and M. B. Nathanson, eds., Lecture Notes in Mathematics #1052, Springer-Verlag, 1984.
14. J. C. Lagarias and A. M. Odlyzko, Computing $\pi(x)$: An analytic method, *J. Algorithms*, to appear.
15. E. Landau, Uber die Nullstellen der Zetafunktion, *Math. Ann.* 71 (1911), 548-564.
16. 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.
17. H. L. Montgomery, The pair correlation of zeros of the zeta function, Proc. Symp. Pure Math. 24, A.M.S., Providence 1973, 181-193.
18. H. L. Montgomery, Distribution of zeros of the Riemann zeta function, *Proc. Int. Congress Math. Vancouver* (1974), 379-381.
19. A. M. Odlyzko, On the distribution of spacings between zeros of the zeta function, *Math. Comp.*, to appear.
20. A. M. Odlyzko, On the distribution of zeros of the Riemann zeta function: Conjectures and computations, manuscript in preparation.

21. A. M. Odlyzko and H. J. J. te Riele, Disproof of the Mertens conjecture, *J. reine angew. Math.* 357 (1985), 138-160.
22. A. M. Odlyzko and A. Schönhage, Fast algorithms for multiple evaluations of the Riemann zeta function, to be published.
23. 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-Verlag, 1966, Vol. 1, pp. 275-310.
24. V. Strassen, Relative bilinear complexity and matrix multiplication, to be published.
25. E. C. Titchmarsh, The zeros of the Riemann Zeta-function, *Proc. Royal Soc. London* 151 (1935), 234-255 and 157 (1936), 261-263.
26. E. C. Titchmarsh, *The Theory of the Riemann Zeta-Function*, Oxford Univ. Press, 1951.
27. A. M. Turing, A method for the calculation of the zeta-function, *Proc. London Math. Soc. ser. 2*, 48 (1943), 180-197.
28. A. M. Turing, Some calculations of the Riemann Zeta-function, *Proc. London Math. Soc.* (3) 3 (1953), 99-117.
29. A. Weil, Sur les "formules explicites" de la theorie des nombres premiers, *Comm. Sem. Math. Univ. Lund*, tome supplementaire (1952), 252-265.