Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

H.J.J. te Riele

Parallel processing in number-theoretical problems

Department of Numerical Mathematics Report NM-R9010 June

The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Research (N.W.O.).

Parallel Processing in Number-Theoretical Problems

H.J.J. te Riele
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Abstract

Three classical problems in number theory are discussed, viz., the Riemann hypothesis, the conjecture of Mertens, and the problem of finding the prime factors of a given large integer. It is shown that, thanks to the power of parallel and vector-processing, it has been possible to extend the state-of-knowledge concerning these problems by several orders of magnitude.

1980 Mathematics Subject Classification (1985 Revision): Primary: 11M26; Secondary: 11A51, 11Y05, 11Y35.
1987 CR Categories: C.1.2, F.2.1.
Keywords and Phrases: Riemann hypothesis, Mertens conjecture, factorization, parallel processing.
Note: This report will appear in the Proceedings of the Workshop on Parallel Processing, held at the Bhabha Atomic Research Centre, Bombay, India, February 7-9, 1990.

Report NM-R9010
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

1 Introduction

Until recently, like in the days of Hardy and Ramanujan, number theory was considered as part of pure mathematics. Nowadays, it is gradually becoming a branch of mathematics with applications in cryptology, physics, biology, digital information and computing [16]. Also, there is a growing interest in *algorithms*, like primality testing and factorization; together with the interest in parallel processing, this explains the rise of the new discipline which is called *Computational Number Theory.*

Many number-theoretical problems are easy to state, but often very difficult to solve. For example, try to write an even number $N \geq 6$ as a sum of two odd prime numbers: $6 = 3 + 3$, $8 = 3 + 5$, ..., $14 = 3 + 11 = 7 + 7$, ..., $100 = 3 + 97 = 11 + 89 = 17 + 83 = 29 + 71 = 41 + 59 = 47 + 53$, ... One soon suspects that this indeed is possible for every even $N \geq 6$, and, in fact, in *many* ways as N grows. Up to now, however, no one has been able even to give a proof that there exists an N_0 such that every even $N \geq N_0$ can be written as a sum of two primes *in at least one way.* Numerically, it has been verified that all even numbers $\leq 2 * 10^{10}$ (and ≥ 6) can be written as a sum of two odd prime numbers. Moreover, if we define $p(N)$ as the smallest prime p for which $N - p$ is prime, then we have found that $p(N) \leq 2029$ for $4 \leq N \leq 2 * 10^{10}$ (N even) and that $p(12,703,943,222) = 2029$ [3].

For many number-theoretical problems it is also very useful to collect numerical data. These can reveal certain properties of the numbers studied, and one could then try to prove these properties. However, collecting numerical data by hand can be a very tedious task and computers can be of great help here. Since the many cases to be considered are often computationally independent, *parallel* computers are also very suitable for the study of such problems.

In this paper we shall discuss three classical problems in number theory, viz., the Riemann hypothesis, the conjecture of Mertens, and the problem of finding the prime factors of a given large integer. We argue that thanks to the power of parallel (and vector-) processing it has been possible to extent the state of knowledge concerning these (and many other) number-theoretical problems by several orders of magnitude.

We like to emphasize here that we consider vector processing as a special form of parallel processing since vector computers are ideally suited to handle many independent processes which all have the same sequence of arithmetic operations, with different inputs.

2 The Riemann hypothesis

The Riemann zeta function is defined as $\zeta(s) := \sum_{n=1}^{\infty} n^{-s}$, $s = \sigma + it$, for $\sigma > 1$. Its definition can be extended to the *whole* complex plane, except $\sigma = 1$, by analytic continuation. It is known that $\zeta(s)$ has infinitely many complex

zeros in the (so-called critical) strip $0 \leq \sigma \leq 1$. In 1859, Bernhard Riemann expressed his belief that all these complex zeros have real part $\frac{1}{2}$, and up to now no one has been able to prove or disprove this. The truth of this so-called *Riemann hypothesis* would imply that the error term in the approximation of $\pi(x)$ (the number of primes $\leq x$) by $x/\log x$ is asymptotically equal to $x^{1/2}\log x$, as $x \to \infty$. This asymptotic behaviour is confirmed by counts of the function $\pi(x)$ for various values of $x \leq 4 * 10^{16}$ [6].

It is possible to check the Riemann hypothesis in a finite part of the critical strip as follows (here, we only give a rough description; for details, cf. [2]). The function

$$Z(t) := \exp[i\theta(t)]\zeta(\frac{1}{2} + it),$$

where

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

is *real* for real t (this follows from the functional equation of the zeta function), so that simple zeros of $\zeta(s)$ *on the critical line* can be located by finding sign changes of $Z(t)$. *Gram's law* says that there is one zero of $Z(t)$ between two consecutive (so-called) Gram points. The m-th Gram point g_m ($m \geq -1$) is defined as the unique solution in the interval $[7, \infty)$ of the equation $\theta(g_m) = m\pi$. Gram's law does not always hold, but "missing" zeros can always be found easily in adjacent Gram intervals. After finding N sign changes of $Z(t)$ in N consecutive Gram intervals, a result of Lehman can be used to prove that there are no *other* complex zeros of $\zeta(s)$ in the part of the critical strip which corresponds to these N consecutive Gram intervals.

For not too small values of t, the so-called Riemann-Siegel formula for $Z(t)$ can be used to evaluate this function:

$$Z(t) = 2\sum_{k=1}^{m} k^{-1/2}\cos[t\log(k) - \theta(t)] + \text{error terms},$$

where $m = \lfloor (t/2\pi)^{1/2} \rfloor$. The time to compute the function $\theta(t)$ and the error terms is small compared with the time to compute the sum in $Z(t)$ above.

In [8] very extensive computations are described by which the Riemann hypothesis was verified for the first $1.5 * 10^9$ complex zeros of $\zeta(s)$ on a CDC Cyber 205 vector computer. To that end, many hundreds of millions of evaluations of $Z(t)$ had to be done for $35 * 10^6 < t < 600 * 10^6$ (so that $2300 < m < 10000$). A table of cos-values and values of $\log(k)$ and $k^{-1/2}$ was precomputed. By writing the sum in $Z(t)$ as an inner product of two vectors of length m, one of them containing the $k^{-1/2}$-values and the other the $\cos[t\log(k) - \theta(t)]$-values (obtained by interpolation in the precomputed table of cos-values), this function could be evaluated very efficiently on the Cyber 205 vector computer.

Very recently, a fast method was developed by Odlyzko and Schönhage [12] for multiple evaluations of the Riemann zeta function. This method was implemented by Odlyzko on a Cray X-MP vector computer [10] to compute long

ranges of zeros of the Riemann zeta function near the 10^{20}-th and other zeros, and to verify the Riemann hypothesis in these ranges.

By splitting a given finite part of the critical strip in disjoint strips, each processor of a *parallel* computer can handle one of these strips independently, virtually without any overhead costs. Thus the Riemann hypothesis also lends itself easily to parallel processing.

3 The conjecture of Mertens

The conjecture of Mertens is concerned with the following function:

$$M(x) := \sum_{n \leq x} \mu(n)$$

where μ is the Möbius function defined by: $\mu(1) = 1$ and, for $n \geq 2$, $\mu(n) = (-1)^k$ if n is the product of k different primes and $= 0$ if $p^2|n$ for some prime p. Hence, $M(x)$ counts the *difference* between the number of squarefree positive integers $\leq x$ with an *even* and of those with an *odd* number of prime factors.

In 1897, Mertens published a 50-page table of $\mu(n)$ and $M(n)$ for $n = 1, 2, \ldots, 10000$[9]. On the basis of the numerical evidence in the table, Mertens concluded that the inequality

$$|M(x)| < x^{1/2}, x > 1,$$

is "very probable". This inequality is known as the *Mertens conjecture.*

It is not difficult to see that the truth of the Mertens conjecture implies the truth of the Riemann hypothesis: for $\sigma = \Re s > 1$, we have

$$\zeta^{-1}(s) = \sum_{n=1}^{\infty} \mu(n) n^{-s} = \sum_{n=1}^{\infty} [M(n) - M(n-1)] n^{-s}$$

$$= \sum_{n=1}^{\infty} M(n)[n^{-s} - (n+1)^{-s}] = \sum_{n=1}^{\infty} M(n) \int_n^{n+1} s x^{-s-1} dx$$

$$= s \sum_{n=1}^{\infty} \int_n^{n+1} M(x) x^{-s-1} dx = s \int_1^{\infty} M(x) x^{-s-1} dx,$$

since $M(x)$ is constant on each interval $[n, n+1]$. If the Mertens conjecture were true, then the last integral above would define a function analytic in $\sigma > 1/2$, and this would give an analytic continuation of the function $1/\zeta(s)$ to $\sigma > 1/2$. In particular, this would imply that $\zeta(s)$ has no zeros in the half-plane $\sigma > 1/2$, which is exactly the statement of the Riemann hypothesis.

In 1985, the Mertens conjecture was disproved [11]. This disproof was based on our ability to find a real number y with the property that the m numbers

$\alpha_i y - \beta_i$, $i = 1, \ldots, m$ are all very close to an integer multiple of 2π, *for m large enough.* The α_i and β_i are certain given numbers related to $\zeta(s)$ and its complex zeros. In fact, this is nothing but an inhomogeneous simultaneous diophantine approximation problem. In 1982, Lenstra, Lenstra and Lovász published a new algorithm [7], the so-called *Lattice Basis Reduction* or L^3 algorithm, by which it turned out to be possible to solve this approximation problem with sufficient accuracy. This algorithm works with vectors of length m, and $m = 70$ turned out to be large enough to obtain a disproof of the Mertens conjecture, viz., that there exists an x such that $M(x) > 1.06x^{1/2}$ and another x such that $M(x) < -1.009x^{1/2}$. Previous work [13], before the L^3 algorithm was invented, was performed with $m = 12$, and yielded $M(x) > 0.86x^{1/2}$. The use of vectors of length 70 in the L^3 algorithm makes it suitable for vector processing. For the disproof of the Mertens conjecture it was also necessary to compute the first 2000 complex zeros of the Riemann zeta function with an accuracy of at least 100 decimal digits. Although this extensive computational job was not carried out on a parallel computer, this could have been done very conveniently because the computation of each separate zero can be done independently.

4 Factorization of large integers

As is well-known now, the security of certain cryptographic systems, like the Rivest-Shamir-Adelman (RSA) public-key cryptosystem, depends on the difficulty of factoring the public keys in this system [15]. The discovery of such cryptosystems has stimulated the renewed interest in the factorization problem, and also in the (related) problem of proving primality of large integers. Several factorization algorithms are known and the best known algorithms have been improved considerably in recent years. Moreover, it turns out that some of these algorithms are very well suited to parallel processing. Recently, Richard Brent [1] has discussed several integer factorization algorithms and their parallel implementation.

Here, we shall briefly describe the two best known algorithms for the factorization of large integers, viz., the *multiple-polynomial quadratic sieve* (MPQS) algorithm and the *elliptic curve method* (ECM). The expected running time of ECM to find a prime divisor p of N is

$$\mathcal{O}(\exp(c_1(\log p \log\log p)^{1/2})), \text{ with } c_1 = 2^{1/2},$$

and for MPQS this is

$$\mathcal{O}(\exp(c_2(\log N \log\log N)^{1/2})), \text{ with } c_2 = 1.$$

This means that the running time of ECM depends on the *size* of the prime divisor p of N, whereas the running time of MPQS depends on the size of the number N to be factorized. Hence, one should first try ECM for some time,

depending on the computer used, and next one should try MPQS. Numerical experience with numbers $N < 10^{100}$ learns that MPQS is usually faster than ECM if N is the product of two primes which both exceed $N^{1/3}$.

MPQS belongs to a class of factorization algorithms which look for two integers X and Y such that

$$X^2 = Y^2 \bmod N.$$

Once such X and Y have been found there is a good chance that $\gcd(X-Y, N)$ is a nontrivial factor of N. In order to find such an (X, Y)-pair, one tries to find triples (U_i, V_i, W_i) such that

$$U_i^2 = V_i^2 W_i \bmod N,$$

where the W_i have prime divisors $\leq B$ for some suitably chosen B. In MPQS the W_i are values of one or more quadratic polynomials with integer coefficients, which are easy to factorize by means of *sieving*. Once sufficiently many of the above relations between U_i, V_i and W_i have been found, they can be combined by a Gaussian elimination process to yield the required (X, Y)-pair.

The largest number factorized by means of MPQS on a single-CPU (vector-) computer is the 92-digit number $(6^{131}-1)/(5 \cdot 263 \cdot 3931 \cdot 6551)$; it has two prime factors: one of 34 and one of 59 decimal digits [14].

MPQS is ideally suited to parallel processing. Different polynomials can be processed on different processors, and there is extremely little communication between processors. Each processor generates (U, V, W)-relations and sends them to a host processor. After the generation of sufficiently many relations, the host processor performs the Gaussian elimination (which takes little time compared to the generation of the relations) and does the gcd-computation. A.K. Lenstra and M.S. Manasse have organized such a parallel factorization project on a world-wide scale [4]. Anyone with access to electronic mail and a C-compiler can contribute to this project. He/she receives a copy of the program for the generation of the $(U, V, W,)$-triples, together with a unique set of polynomials. Several numbers larger than 10^{100} were factorized in this way, the largest having 106 decimal digits.

Recently, Lenstra, Lenstra, Manasse and Pollard [5] have announced the factorization of even larger numbers by the so-called *Number Field Sieve*. This method, unlike MPQS, takes advantage of the special form of the number. Like MPQS, it collects relations and is ideally suited for parallel processing. The largest number presently factorized by the Number Field Sieve is the 138-digit number $(2^{457}+1)/3$, which has two prime divisors: one of 49 and one of 89 decimal digits. Whether the Number Field Sieve can be adapted to work for general numbers (like MPQS) remains an interesting open question.

In ECM one randomly selects an elliptic curve mod N to factorize N, and a point X in the group of points of this elliptic curve. Using group operations, one computes $K \cdot X$, where K is the product of all prime powers $\leq M$, and

where M is some suitably chosen number. If in this process we find the group identity, then we have found a factor of N. The group identity will be found if the order of the group has only prime factors $< M$. If this process does not succeed, one selects another elliptic curve and repeats the same computation with this new curve. This is continued, until a factor of N has been found or until one decides to try MPQS.

Like MPQS and the Number Field Sieve, ECM is also very suited to parallel processing because each elliptic curve trial can be processed on a different processor.

The current ECM-record was established very recently by A.K. Lenstra and M.S. Manasse who found a 38-digit prime factor of $(11^{118}+1)/(2\cdot 61\cdot 1291321673)$ (which is a composite number of 112 decimal digits).

References

[1] R.P. BRENT, *Parallel Algorithms for Integer Factorization,* Research Rept. CMA-R49-89, The Australian National University, Canberra, Oct. 1989.

[2] H.M. EDWARDS, *Riemann's Zeta Function,* Academic Press, 1974.

[3] A. GRANVILLE, J. VAN DE LUNE & H.J.J. TE RIELE, Checking the Goldbach conjecture on a vector computer, pp. 423-433 in: R.A. MOLLIN (ED.), *Number Theory and Applications,* Kluwer, 1989.

[4] A.K. LENSTRA & M.S. MANASSE, Factoring by electronic mail, *preprint,* June 1989.

[5] A.K. LENSTRA, H.W. LENSTRA, JR., M.S. MANASSE & J.M. POLLARD, The number field sieve, *preprint,* December 1989.

[6] J.C. LAGARIAS, V.S. MILLER & A.M. ODLYZKO, Computing $\pi(x)$: The Meissel-Lehmer Method, *Math. Comp.,* **44**(1985), pp. 537-560.

[7] A.K. LENSTRA, H.W. LENSTRA, JR. & L. LOVÁSZ, Factoring polynomials with rational coefficients, *Math. Ann.,* **261**(1982), pp.515-534.

[8] J. VAN DE LUNE, H.J.J. TE RIELE & D.T. WINTER, On the zeros of the Riemann zeta function in the critical strip. IV, *Math. Comp.,* **46**(1986), pp. 667-681.

[9] F. MERTENS, Über eine zahlentheoretische Funktion, *Sitzungsberichte Akad. Wiss. Wien IIa,* **106**(1897), pp. 761-830.

[10] A.M. ODLYZKO, The 10^{20}-th zero of the Riemann zeta function and 79 million of its neighbors, *preprint,* Jan. 1989.

[11] A.M. ODLYZKO & H.J.J. TE RIELE, Disproof of the Mertens conjecture, *J. reine angew. Math.*, **357**(1985), pp. 138-160.

[12] A.M. ODLYZKO & A. SCHÖNHAGE, Fast algorithms for multiple evaluations of the Riemann zeta function, *Trans. Amer. Math. Soc.*, **309**(1988), pp. 797- 809.

[13] H.J.J. TE RIELE, Computations concerning the conjecture of Mertens, *J. reine angew. Math.*, **311/312**(1979), pp. 356-360.

[14] H.J.J. TE RIELE, W.M. LIOEN & D.T. WINTER, Factoring with the quadratic sieve on large vector computers, *J. Comp. Appl. Math.*, **27**(1989), pp. 267-278.

[15] R.L. RIVEST, A. SHAMIR & L. ADELMAN, A method for obtaining digital signatures and public-key cryptosystems, *Comm. ACM*, **21**(1978), pp. 120-126.

[16] M.R. SCHROEDER, *Number Theory in Science and Communication*, Springer-Verlag, Berlin etc., 1984.