Computing the Summation of the Möbius Function

Marc Deléglise and Joël Rivat

CONTENTS

We describe an elementary method for computing isolated values of $M(x) = \sum_{n\le x} \mu(n)$, where μ is the Möbius function. The complexity of the algorithm is $O(x^{2/3}(\log\log x)^{1/3})$ time and $O(x^{1/3}(\log\log x)^{2/3})$ space. Certain values of $M(x)$ for x up to 10^{16} are listed: for instance, $M(10^{16}) = -3195437$.

1. INTRODUCTION

Möbius [1832] was the first to study the function $\mu(n)$, defined for positive integers n by

- $\mu(1) = 1$,
- $\mu(n) = 0$ if n has a squared prime factor;
- $\mu(p_1 \dots p_k) = (-1)^k$ if all the primes $p_1, \dots, p_k$ are different.

Mertens [1897] introduced the summation function

$$M(x) = \sum_{n\le x} \mu(n),$$

which is defined for all real $x \ge 0$. He verified that $|M(x)| \le \sqrt{x}$ for $x < 10000$, and conjectured that this inequality holds for any x. Von Sterneck [1912] verified this up to 500,000. (The Riemann Hypothesis implies the weaker conjecture $|M(x)| = O(x^{1/2+\varepsilon})$ for all $\varepsilon > 0$.)

However, Odlyzko and te Riele disproved the Mertens conjecture when they showed [1985] that

$$\liminf_{x\to+\infty} \frac{M(x)}{\sqrt{x}} < -1.009, \quad \limsup_{x\to+\infty} \frac{M(x)}{\sqrt{x}} > 1.06.$$

Pintz [1987] made this result effective, proving that there exist values of $x < \exp(3.21\times10^{64})$ such that $|M(x)| > \sqrt{x}$.

The first value of x for which $|M(x)| > \sqrt{x}$ is still unknown, but Dress [1993] has verified that it exceeds 10^{12}. He also proposed in his paper a

1058-6458/96 \$0.50 per page

method for computing an isolated value of $M(x)$, using $O(x^{3/4}\log^{1/2}x)$ time and $O(x^{1/2})$ space.

Lagarias and Odlyzko [1987] proposed an analytic method for computing $\pi(x)$ (the number of primes not greater than x) in $O(x^{1/2+\varepsilon})$ time and $O(x^{1/4+\varepsilon})$ space. They mentioned that their algorithm could be adapted for computing $M(x)$. To our knowledge, nobody has tried to compute $\pi(x)$ or $M(x)$ using their method yet.

In this paper we explain another method for computing an isolated value of $M(x)$ using

$$O(x^{2/3}(\log\log x)^{1/3})$$

time and $O(x^{1/3}(\log\log x)^{2/3})$ space. Our method is elementary, and was inspired by [Lehman 1960]. We give a table of certain values of $M(x)$ for x up to 10^{16}, and also some computation times.

2. A COMBINATORIAL IDENTITY

For completeness we recall some classical results concerning the Möbius function. Our goal is to obtain Lemma 2.1 below, which is essentially derived from [Lehman 1960, p. 314].

It follows immediately from the definition that $\mu(n)$ is a multiplicative function. Next, we have the *Möbius inversion formula*

$$\sum_{d\mid n}\mu(d)=\begin{cases}1 & \text{if } n=1,\\ 0 & \text{otherwise.}\end{cases}$$

This is obvious for $n=1$. For $n>1$, we write $n=p_1^{a_1}\cdots p_k^{a_k}$ with $k\ge 1$, and obtain

$$\begin{aligned}\sum_{d\mid n}\mu(d)&=1+\sum_i\mu(p_i)+\sum_{i,j}\mu(p_ip_j)+\cdots\\ &=1-k+\binom{k}{2}-\binom{k}{3}+\cdots=(1-1)^k=0.\end{aligned}$$

The inversion formula easily implies, for $x\ge 1$, that $\sum_{n\le x}M(x/n)=1$. Indeed,

$$\sum_{n\le x}M\Bigl(\frac{x}{n}\Bigr)=\sum_{n\le x}\sum_{d\le\frac{x}{n}}\mu(d)=\sum_{l\le x}\sum_{d\mid l}\mu(d)=\mu(1).$$

Lemma 2.1. *For $1\le u\le x$ we have the combinatorial identity*

$$M(x)=M(u)-\sum_{m\le u}\mu(m)\sum_{\frac{u}{m}<n\le\frac{x}{m}}M\Bigl(\frac{x}{mn}\Bigr)$$

Proof. We use the Möbius inversion formula, together with the equality $\sum_{n\le x}M(\frac{x}{n})=1$ with x replaced by x/m:

$$\sum_{m\le u}\mu(m)\sum_{n\le\frac{x}{m}}M\Bigl(\frac{x}{mn}\Bigr)=\sum_{m\le u}\mu(m)=M(u)$$

$$\begin{aligned}\sum_{m\le u}\mu(m)\sum_{n\le\frac{u}{m}}M\Bigl(\frac{x}{mn}\Bigr)&=\sum_{l\le u}M\Bigl(\frac{x}{l}\Bigr)\sum_{m\mid l}\mu(m)\\ &=M\Bigl(\frac{x}{1}\Bigr)=M(x).\end{aligned}$$

The result follows by writing $M(u)-M(x)$. □

3. OUTLINE OF THE METHOD

Observe that the sum in Lemma 2.1 has more than x terms, but these terms often have the same value. This comes from the general fact that, for $y>0$, the sequence $(\lfloor y/n\rfloor)_n$ takes at most $2\lfloor\sqrt{y}\rfloor+1$ different values:

- the values $\lfloor y/n\rfloor$ for $1\le n\le\lfloor\sqrt{y}\rfloor$,
- the values $0, 1, \ldots, \lfloor\sqrt{y}\rfloor$ corresponding to $n>\lfloor\sqrt{y}\rfloor$.

We apply this idea to split the sum in Lemma 2.1. For $1\le u\le\sqrt{x}$ we have $M(x)=M(u)-S_1(x,u)-S_2(x,u)$, with

$$S_1(x,u)=\sum_{m\le u}\mu(m)\sum_{\frac{u}{m}<n\le\sqrt{\frac{x}{m}}}M\Bigl(\frac{x}{mn}\Bigr),$$

$$S_2(x,u)=\sum_{m\le u}\mu(m)\sum_{\sqrt{\frac{x}{m}}<n\le\frac{x}{m}}M\Bigl(\frac{x}{mn}\Bigr).$$

For any $y>0$ of the form $y=x/m$, and any $k\le\sqrt{y}$, it is not difficult to compute the number of values of n with $\sqrt{y}<n\le y$ and $\lfloor y/n\rfloor=k$.

For both $S_1(x,u)$ and $S_2(x,u)$ the number of terms in the sum is at most

$$\sum_{m\le u}\sqrt{x/m}=O(\sqrt{xu}).$$

This summation will be done using a table of values of $\mu(n)$ for $1\le n\le u$ and a table of values of $M(n)$ for $1\le n\le x/u$.

We will see later that it is possible to build these tables in $O\big((x/u)\log\log(x/u)\big)$ time. By choosing $u=x^{1/3}(\log\log x)^{2/3}$, we get a total time of

$$O\Big(\frac{x}{u}\log\log x+\sqrt{xu}\Big)=O(x^{2/3}(\log\log x)^{1/3}).$$

The tabulation of $\mu(n)$ for $n\le u$ costs $O(u)$ space, which is acceptable. Unfortunately the tabulation of M would need $O(x/u)=O(x^{2/3})$ space, which is not available on current computers when $x\ge 10^{15}$. Hence we have to work by blocks of size $L\approx u$, as we will explain now.

4. TABULATING M BY BLOCKS

We suppose $L\ge u\ge x^{1/3}$ and we want to tabulate M for $a\le n<b=a+L$. Since we have $M(n)=M(n-1)+\mu(n)$, it suffices to know $M(a-1)$ and have a table of $\mu(n)$ for $a\le n<b$ in order to build a table of $M(n)$ for $a\le n<b$. The additional cost (from μ to M) is $O(L)$ time.

Hence it suffices to be able to tabulate $\mu(n)$ for $a\le n<b$. This must be achieved without the help of a table of primes up to b, which would be too big. The following algorithm uses only a table of primes up to $\sqrt{b}$:

Algorithm 4.1 (Tabulation of μ).
Input: bounds $b>a>0$.
Output: a table $t(n)$ of values of $\mu(n)$ for $a\le n<b$.

1. for each $n\in[a,b)$, set $t(n)=1$.
2. for each prime number $p\in\big[2,\sqrt{b}\,\big]$, do:
   - for each multiple $m\in[a,b)$ of p^2, set $t(m)=0$.
   - for each multiple $m\in[a,b)$ of p, multiply $t(m)$ by $-p$.
3. for each $n\in[a,b)$ such that $t(n)\ne 0$, do:
   - if $|t(n)|<n$, multiply $t(n)$ by -1.
   - if $t(n)>0$, set $t(n)=1$.
   - if $t(n)<0$, set $t(n)=-1$.

In order to use this algorithm for tabulating M by blocks up to x/u, we need a table of the prime numbers up to $\sqrt{x/u}$. Such a table is easy to build using Eratosthenes' sieve, a process that requires $O(\sqrt{x/u})$ space. The finished table takes $O\big(\sqrt{x/u}/\log(x/u)\big)$ space, by Chebyshev's Theorem. Since $\sqrt{x/u}\le L$, the space cost of tabulating M by blocks is $O(L)$. For each block $a\le n<b$ the number of operations we do is

$$O\Big(L+\sum_{p\le\sqrt{b}}\Big(1+\frac{L}{p^2}+\frac{L}{p}\Big)\Big)=O\Big(\frac{\sqrt{b}}{\log b}+L\log\log b\Big),$$

which is $O\big(\sqrt{x/u}+L\log\log(x/u)\big)$.

Hence the total cost for tabulating M by blocks of size L up to x/u is $O\big((x/u)\log\log(x/u)\big)$ time and $O(L)$ space.

5. COMPUTING $S_1(x,u)$ AND $S_2(x,u)$

For $1\le a\le b$ we have

$$a\le\frac{x}{mn}<b\iff\frac{x}{mb}<n\le\frac{x}{ma}.$$

We suppose we have tabulated $M(n)$ for an interval of size L, namely for $a_k\le n<a_{k+1}$ with $a_k=1+kL$, for some $k\le x/(uL)$. The number of terms of the sum over m and n corresponding to this block is

$$\sum_{m\le u}\Big(\min\Big(\Big\lfloor\frac{x}{ma_k}\Big\rfloor,\Big\lfloor\sqrt{\frac{x}{m}}\Big\rfloor\Big)-\max\Big(\Big\lfloor\frac{x}{ma_{k+1}}\Big\rfloor,\Big\lfloor\frac{u}{m}\Big\rfloor\Big)\Big).$$

The total number of terms we have to sum is obtained by summing the preceding expression over $k\le x/(uL)$. Reversing the order of summation shows that the total equals the sum over $m\le u$ of

$$\sum_{k\le\frac{x}{uL}}\Big(\min\Big(\Big\lfloor\frac{x}{ma_k}\Big\rfloor,\Big\lfloor\sqrt{\frac{x}{m}}\Big\rfloor\Big)-\max\Big(\Big\lfloor\frac{x}{ma_{k+1}}\Big\rfloor,\Big\lfloor\frac{u}{m}\Big\rfloor\Big)\Big).$$

n	10	11	12	13	14	15
$M(1\times 10^n)$	-33722	-87856	62366	599582	-875575	-3216373
$M(2\times 10^n)$	48723	-19075	-308413	127543	2639241	1011871
$M(3\times 10^n)$	42411	133609	190563	-759205	-2344314	5334755
$M(4\times 10^n)$	-25295	202631	174209	-403700	-3810264	-6036592
$M(5\times 10^n)$	54591	56804	-435920	-320046	4865646	11792892
$M(6\times 10^n)$	-56841	-43099	268107	1101442	-4004298	-14685733
$M(7\times 10^n)$	7917	111011	-4252	-2877017	-2605256	4195668
$M(8\times 10^n)$	-1428	-268434	-438208	-99222	3425855	6528429
$M(9\times 10^n)$	-5554	10991	290186	1164981	7542952	-12589671

TABLE 1. Values of $M(k\times 10^n)$.

The sequence $(\lfloor x/(ma_k)\rfloor)_k$ forms a subdivision of the interval $\lfloor u/m, \sqrt{x/m}\rfloor$ such that the above sum over k is at most $\sqrt{x/m}$. Hence the cost of computing $S_1(x,u)$ is $O(\sqrt{xu})$ time and $O(L)$ space.

Turning now to $S_2(x,u)$, we start by defining $l(y,k) = \#\{n : \sqrt{y} < n \le y,\ \lfloor y/n\rfloor = k\}$. The computation of $l(y,k)$ clearly needs $O(1)$ time. We have

$$\begin{aligned} S_2(x,u) &= \sum_{m\le u}\mu(m)\sum_{k\le\sqrt{\frac{x}{m}}} M(k)\, l\Big(\frac{x}{m},k\Big)\\ &= \sum_{k\le\sqrt{x}} M(k)\sum_{m\le\min(u,x/k^2)}\mu(m)\, l\Big(\frac{x}{m},k\Big). \end{aligned}$$

This last sum is appropriate for use in a tabulation of M done by blocks of size L without any additional cost. Hence the computation of $S_2(x,u)$ can be achieved in $O(\sqrt{xu})$ time and $O(L)$ space.

We therefore get a total time cost of

$$O\big((x/u)\log\log x + \sqrt{xu}\big) = O(x^{2/3}(\log\log x)^{1/3}),$$

by choosing

$$u = x^{1/3}(\log\log x)^{2/3}.$$

The total space cost is $O(L)$ with $L \ge u$. In our program we chose $L = 4u$. (Our program is written in C++, compiled with GNU C/C++, and is 270 lines long. For more information, contact the second author.)

Tables 1 and 2 show some of the values obtained.

x	$M(x)$	time (s)
10^6	212	0.06
10^7	1037	0.28
10^8	1928	1.36
10^9	-222	6.74
10^{10}	-33722	32.74
10^{11}	-87856	160.99
10^{12}	62366	878.37
10^{13}	599582	4927.23
10^{14}	-875575	24048.91
10^{15}	-3216373	115614.87
10^{16}	-3195437	555276.59

TABLE 2. Values of $M(10^n)$ and computation times on a 64-bit DEC Alpha 3000 Model 300 with 96 Mbytes of memory.

REFERENCES

[Dress 1993] F. Dress, "Fonction sommatoire de la fonction de Möbius; 1. Majorations expérimentales", *Experimental Math.* **2** (1993), 93–102.

[Lagarias and Odlyzko 1987] J. Lagarias and A. Odlyzko, "Computing $\pi(x)$: an analytic method", *J. Algorithms* **8** (1987), 173–191.

[Lehman 1960] R. S. Lehman, "On Liouville's function", *Math. Comp.* **14** (1960), 311–320.

[Mertens 1897] F. Mertens, "Über eine zahlentheoretische Funktion", *Akad. Wiss. Wien Math.-Natur. Kl. Sitzungber. IIa* **106** (1897), 761–830.

[Möbius 1832] A. F. Möbius, "Über eine besondere Art von Umkehrung der Reihen", *J. reine angew. Math.* **9** (1832), 105–123.

[Odlyzko and te Riele 1985] A. Odlyzko and H. te Riele, "Disproof of the Mertens conjecture", *J. reine angew. Math.* **357** (1985), 138–160.

[Pintz 1987] J. Pintz, "An effective disproof of the Mertens conjecture", *Astérisque* **147-148** (1987), 325–333, 346.

[von Sterneck 1912] R. D. von Sterneck, "Die zahlentheoretische Funktion $\sigma(n)$ bis zur Grenze 500000", *Akad. Wiss. Wien Math.-Natur. Kl. Sitzungber. IIa* **121** (1912), 1083–1096.

Marc Deléglise, Institut Girard Desargues (URA 746), Département de Mathématiques, Université Lyon 1, 43 Boulevard du 11 Novembre 1918, 69622 Villeurbaine, France (deleglis@lmdi.univ-lyon1.fr)

Joël Rivat, Institut Girard Desargues (URA 746), Département de Mathématiques, Université Lyon 1, 43 Boulevard du 11 Novembre 1918, 69622 Villeurbaine, France (rivat@caissa.univ-lyon1.fr)

Received January 2, 1996; accepted April 12, 1996