NUMERICAL VERIFICATION OF LITTLEWOOD’S BOUNDS FOR $|L(1, \chi)|$

ALESSANDRO LANGUASCO

Abstract. Let $L(s, \chi)$ be the Dirichlet $L$-function associated to a non trivial primitive Dirichlet character $\chi$ defined mod $q$, where $q$ is an odd prime. In this paper we introduce a fast method to compute $|L(1, \chi)|$ using the values of Euler’s $\Gamma$ function. We also introduce an alternative way of computing $\log \Gamma(x)$ and $\psi(x) = \Gamma'/\Gamma(x)$, $x \in (0, 1)$. Using such algorithms we numerically verify the classical Littlewood bounds and the recent Lamzouri-Li-Soundararajan estimates on $|L(1, \chi)|$, where $\chi$ runs over the non trivial primitive Dirichlet characters mod $q$, for every odd prime $q$ up to $10^7$. The programs used and the results here described are collected at the following address http://www.math.unipd.it/~languasc/Littlewood_ineq.html.

1. Introduction

Let $q$ be an odd prime, $\chi$ be a Dirichlet character mod $q$ and $L(s, \chi)$ be the associated Dirichlet $L$-function. The goal of this paper is to introduce a fast algorithm to compute the values of $|L(1, \chi)|$ for every non trivial primitive Dirichlet character $\chi$ defined mod $q$ and, using such a new method, to numerically study a generalisation of the classical bounds of Littlewood [14] for $|L(1, \chi_d)|$, where $\chi_d$ is a quadratic Dirichlet character. Assuming the Riemann Hypothesis for $L(s, \chi_d)$ holds, in 1928 Littlewood proved, for $d$, $m \geq 2$, that

$$\left(\frac{12e^\gamma}{\pi^2}(1 + o(1))\log \log |d|\right)^{-1} < L(1, \chi_d) < 2e^\gamma(1 + o(1))\log \log |d|$$

as $d$ tends to infinity, where $\gamma$ is the Euler-Mascheroni constant. In 1973 Shanks [18] numerically studied the behaviour of the upper and lower Littlewood indices defined as

$\text{ULI}(d, \chi_d) := \frac{L(1, \chi_d)}{2e^\gamma \log \log |d|}$ and $\text{LLI}(d, \chi_d) := L(1, \chi_d)\frac{12e^\gamma}{\pi^2} \log \log |d|$.

for several small discriminants $d$. Such computations were extended by Williams-Broere [19] in 1976 and by Jacobson-Ramachandran-Williams [7] in 2006.

Recently Lamzouri-Li-Soundararajan [10, Theorem 1.5] proved an effective form of Littlewood’s inequalities: assuming the Generalised Riemann Hypothesis holds, for every integer $q \geq 10^{10}$ and for every non trivial primitive character $\chi$ mod $q$, they obtained that

$$|L(1, \chi)| \leq 2e^\gamma \left(\log \log q - \log 2 + \frac{1}{2} + \frac{1}{\log \log q}\right)$$

and

$$\frac{1}{|L(1, \chi)|} \leq \frac{12e^\gamma}{\pi^2} \left(\log \log q - \log 2 + \frac{1}{2} + \frac{1}{\log \log q} + \frac{14}{\log q}\right).$$

Using our method we will compute the values of $|L(1, \chi)|$ for every non trivial primitive Dirichlet character $\chi$ defined mod $q$, for every odd prime $q$ up to $10^7$. This largely extends...
previous results. Moreover, letting
\[ M_q := \max_{\chi \neq \chi_0} |L(1, \chi)|, \quad m_q := \min_{\chi \neq \chi_0} |L(1, \chi)|, \]
\[ f(q) := \log \log q - \log 2 + 1/2 + 1/\log \log q, \quad g(q) := f(q) + 14(\log \log q)/\log q, \]
we obtain the following

**Theorem 1.** Let 3 ≤ q ≤ 10^7, q be a prime number and M_q be defined in (4). Then we have 0.604599 . . . = M_3 ≤ M_q ≤ M_{4305479} = 6.399873 . . . Moreover, we also have
\[ 0.325 \cdot 2e^\gamma f(q) < M_q < 0.62 \cdot 2e^\gamma f(q), \]
where the lower bound holds just for q ≥ 79, and
\[ 0.4 < \max_{\chi \neq \chi_0} ULI(q, \chi) < 0.66, \]
where the upper bound holds just for q ≥ 5.

We also have an analogous result on m_q.

**Theorem 2.** Let 3 ≤ q ≤ 10^7, q be a prime number and m_q be defined in (4). Then we have 0.198814 . . . = m_{991027} ≤ m_q ≤ m_{11} = 0.618351 . . . Moreover, we also have
\[ \frac{\pi^2}{12e^\gamma} \frac{2.35}{g(q)} < m_q < \frac{\pi^2}{12e^\gamma} \frac{5}{g(q)}, \]
where the upper bound holds just for q ≥ 953, and
\[ 1.13 < \min_{\chi \neq \chi_0} LLI(q, \chi) < 2, \]
where the lower bound holds just for q ≥ 373.

Theorems 1-2 are in agreement with Littlewood’s bounds in (1) and the Lamzouri-Li-Soundararajan estimates in (2)-(3).

The paper is organised as follows: in Section 2 we will see how to compute |L(1, \chi)| using the values of Euler’s \(\Gamma\) function and the Fast Fourier Transform algorithm; we will also describe the actual computation we performed and how Theorems 1-2 are obtained. In Sections 3-4 we will see how to efficiently evaluate \(\log \Gamma(x)\) and \(\psi(x) = \Gamma'/\Gamma(x)\), for \(x \in (0, 1)\) using precomputed values of the Riemann zeta-function at positive integers. After the bibliography we will also insert some tables and figures (the scatter plots were obtained using GNUPLOT, v.5.2, patchlevel 8).

**Acknowledgements.** I wish to thank Luca Righi (University of Padova) for his help in developing the C language implementation of the algorithms described in Sections 3-4.

2. Computation of |L(1, \chi)| and Proofs of Theorems 1-2

Recall that q is an odd prime and let \(\chi\) be a primitive non trivial Dirichlet character mod q. The values of |L(1, \chi)| can be computed in two different ways. Recalling eq. (3.1) of [2], we have
\[ L(1, \chi) = -q^{-1} \sum_{a=1}^{q-1} \chi(a) \psi(a/q), \]
so that
\[ |L(1, \chi)| = \frac{1}{q} \left| \sum_{a=1}^{q-1} \chi(a) \psi \left( \frac{a}{q} \right) \right|, \]
where \(\psi(x) = \Gamma'/\Gamma(x)\) is the digamma function and \(\Gamma\) is Euler’s function. As we will see later, for computational purposes it is in fact more efficient to distinguish between the parity
of the Dirichlet characters. If $\chi$ is an even character we have, see, e.g., Cohen [1, proof of Proposition 10.3.5], that $L(1, \chi) = 2\tau(\chi)q^{-1} \sum_{a=1}^{q-1} \chi(a) \log(\Gamma(a/q))$, where the Gauß sum $\tau(\chi) := \sum_{a=1}^{q} \chi(a) e(a/q)$. $e(x) := \exp(2\pi ix)$, verifies $|\tau(\chi)| = q^{1/2}$. Hence

$$|L(1, \chi)| = \frac{2}{q^{1/2}} \left| \sum_{a=1}^{q-1} \chi(a) \log\left(\Gamma\left(\frac{a}{q}\right)\right) \right| \quad (\chi \text{ even}). \quad (7)$$

Moreover, if $\chi$ is an odd character, we have, see, e.g., Cohen [1, Corollary 10.3.2], that $L(1, \chi) = -w(\chi)\pi q^{-1/2}B_{1, \chi}$, where $w(\chi) = \tau(\chi)/q^{1/2}$ and $B_{1, \chi} := q^{-1} \sum_{a=1}^{q-1} a\chi(a)$ is the first $\chi$-Bernoulli number. Hence $|w(\chi)| = 1$ and

$$|L(1, \chi)| = \frac{\pi}{q^{3/2}} \left| \sum_{a=1}^{q-1} a\chi(a) \right| \quad (\chi \text{ odd}). \quad (8)$$

We will use the formulae (7)-(8) because in half of the cases we don’t need any special function, while in (6) we need to evaluate the digamma function at $q - 1$ points. Moreover, in both the equations (7)-(8) we can embed a decimation in frequency strategy in the Fast Fourier Transform (FFT) algorithm used to perform the sum over $a$, see subsection 2.2. Using the algorithm described in Section 3, see also Remark 1, the needed set of Gamma-function values can be computed with a precision of $n$ binary digits with a cost of $O(qn)$ floating point products, plus the cost of computing $(q - 1)/2$ values of the logarithm function. Hence, recalling also that the computational cost of the FFT algorithm of length $q$ is $O(q \log q)$ floating point products, the total cost for computing $|L(1, \chi)|$ with a precision of $n$ binary digits is then $O(q(n + \log q))$ floating point products plus the cost of computing $(q - 1)/2$ values of the logarithm function. So far, this is the fastest algorithm to compute $|L(1, \chi)|$.

We now proceed to describe our computational strategy. Defining

$$M_q^{\text{odd}} := \max_{\chi \text{ odd}} |L(1, \chi)|, \quad M_q^{\text{even}} := \max_{\chi \neq \chi_0} |L(1, \chi)|, \quad m_q^{\text{odd}} := \min_{\chi \text{ odd}} |L(1, \chi)|, \quad m_q^{\text{even}} := \min_{\chi \neq \chi_0} |L(1, \chi)|,$$

we will obtain $M_q, m_q$ as defined in (4), using (7)-(8), $M_q = \max(M_q^{\text{odd}}, M_q^{\text{even}})$ and $m_q = \min(m_q^{\text{odd}}, m_q^{\text{even}})$.

2.1. Computations trivially summing over $a$ (slower, more decimal digits available). In practice we first computed a few values of $M_q$ and $m_q$ using PARI/GP, v. 2.11.4, since it has the ability to generate the Dirichlet $L$-functions (and many other $L$-functions). This can be done with few instructions of the gp scripting language. Such a computation has a linear cost in the number of calls of the $\text{lffun}$ function of PARI/GP and it is, at least on our Dell OptiPlex desktop machine, slower than using (7)-(8). So we also implemented such formulae in PARI/GP and we were able to get the values of $M_q, m_q$ for every $q$ prime, $3 \leq q \leq 1000$, with a precision of 30 decimal digits (see Tables 1 and 2) in less than 17 seconds of computation time for each table. The machine we used was a Dell OptiPlex-3050, equipped with an Intel i5-7500 processor, 3.40GHz, 16 GB of RAM and running Ubuntu 18.04.2.

2.2. Building the FFT approach. As $q$ becomes large, the time spent in summing over $a$ dominates the overall computational cost. So we implemented the use of the FFT by using the $\text{fftw}$ [3] library in our C programs. We see now how to do so.
In both (7) and (8) we remark that, since \( q \) is prime, it is enough to get \( g \), a primitive root of \( q \), and \( \chi_1 \), the Dirichlet character mod \( q \) given by \( \chi_1(g) = e^{2\pi i j q^{-1}} \), to see that the set of the non-trivial characters mod \( q \) is \( \{ \chi_j' : j = 1, 2, \ldots, q - 2 \} \). Hence, if, for every \( k \in \{ 0, \ldots, q - 2 \} \), we denote \( g^k \equiv a_k \in \{ 1, \ldots, q - 1 \} \), every summation in (7) and (8) is of the type \( \sum_{k=0}^{q-2} e^{-2\pi i j k (q-1)} f(a_k/q) \), where \( j \in \{ 1, \ldots, q - 2 \} \) is odd and \( f \) is a suitable function. As a consequence, such quantities are the Discrete Fourier Transform (DFT) of the sequence \( \{ f(a_k/q) : k = 0, \ldots, q - 2 \} \). This idea was first formulated by Rader [17] and it was used in [2, 11, 12, 13] to speed-up the computation of similar quantities via the use of Fast Fourier Transform dedicated software libraries.

In this case we can also use the decimation in frequency strategy. Let \( f \) be a function that assumes real values. Following the line in Section 4.1 of [11], letting \( e(x) := \exp(2\pi i x) \), \( m = (q - 1)/2 \), for every \( j = 0, \ldots, q - 2 \), \( j = 2t + \ell \), \( t, \ell \in \{ 0, 1 \} \) and \( t \in \mathbb{Z} \), we have that

\[
\sum_{k=0}^{q-2} e\left( -\frac{jk}{q-1} \right) f\left( \frac{ak}{q} \right) = \sum_{k=0}^{m-1} e\left( -\frac{tk}{m} \right) e\left( -\frac{\ell k}{q-1} \right) f\left( \frac{a_{k+m}}{q} \right) + (-1)^{\ell} f\left( \frac{a_{k+m}}{q} \right),
\]

where \( t = 0, \ldots, m - 1 \). Letting

\[
b_k := f\left( \frac{ak}{q} \right) + f\left( \frac{a_{k+m}}{q} \right) \quad \text{and} \quad c_k := e\left( -\frac{k}{q-1} \right) f\left( \frac{a_{k+m}}{q} \right) - f\left( \frac{a_k}{q} \right),
\]

we can rewrite the previous formula (recall that \( j = 2t + \ell \), \( t, \ell \in \{ 0, 1 \} \) and \( t = 0, \ldots, m - 1 \) as

\[
\sum_{k=0}^{q-2} e\left( -\frac{jk}{q-1} \right) f\left( \frac{ak}{q} \right) = \begin{cases} 
\sum_{k=0}^{m-1} e\left( -\frac{tk}{m} \right) b_k & \text{if } \ell = 0 \\
\sum_{k=0}^{m-1} e\left( -\frac{tk}{m} \right) c_k & \text{if } \ell = 1.
\end{cases}
\]

Since we just need the sum over the odd Dirichlet characters for \( f(x) = x \) and over the even Dirichlet characters for \( f(x) = \log \Gamma(x) \), in this way we can evaluate an FFT of length \((q - 1)/2\), instead of \( q - 1 \), applied on a suitably modified sequence according to (9)-(10). Clearly this represents a gain in both speed and memory usage in running the actual computer program.

In the case \( f(x) = \log \Gamma(x) \) we can simplify the form of \( b_k = \log \Gamma(a_k)/q + \log \Gamma(a_{k+m})/q \), where \( m = (q - 1)/2 \) and \( k = 0, \ldots, m - 1 \), in the following way. Recalling \( \langle g \rangle = \mathbb{Z}_q^* \), \( a_k \equiv g^k \mod q \) and \( g^m \equiv q - 1 \mod q \), we can write that \( a_{k+m} \equiv q - a_k \mod q \) and hence \( \log \Gamma(a_{k+m}/q) = \log \Gamma((q - a_k)/q) = \log \Gamma(1 - a_k/q) \). Using the well-known reflection formula \( \Gamma(1 - x) = \pi / \sin(\pi x) \), we obtain

\[
\log \Gamma\left( \frac{a_k}{q} \right) + \log \Gamma\left( \frac{a_{k+m}}{q} \right) = \log \Gamma\left( \frac{a_k}{q} \right) + \log \Gamma\left( 1 - \frac{a_k}{q} \right) = \log \pi - \log \left( \sin\left( \frac{\pi a_k}{q} \right) \right),
\]

for every \( k = 0, \ldots, m - 1 \). Inserting the last relation in the definition of \( b_k \) in (9) and remarking that, by orthogonality, the constant term \( \log \pi \) is negligible, we can replace in the actual computation the Gamma function with the \( \log(\sin(\cdot)) \) one. Since in our application we will have \( a_i/q \in (0, 1) \), we also developed our own alternative implementation of \( \log \Gamma(x) \), \( x \in (0, 1) \), see Section 3.

In the case \( f(x) = x \), it is easier to obtain a simplified form of \( c_k \) as defined in (9). Using again \( \langle g \rangle = \mathbb{Z}_q^* \), \( a_k \equiv g^k \mod q \) and \( g^m \equiv q - 1 \mod q \), we can write that \( a_{k+m} \equiv q - a_k \mod q \); hence \( a_k - a_{k+m} = a_k - (q - a_k) = 2a_k - q \), so that in this case, for every \( k = 0, \ldots, m - 1, m = (q - 1)/2 \), we obtain

\[
c_k = e\left( -\frac{k}{q-1} \right) \left( 2\frac{a_k}{q} - 1 \right).
\]
2.3. Computations summing over a via FFT (much faster, less decimal digits available). Using the setting explained in the previous subsection, we were able to compute, using the long double precision (80 bits) of the C programming language, the values of $M_q$ and $m_q$ for every prime $3 \leq q \leq 10^7$ and we provide here the scatter plots of such values and of their normalisations, see Figures 1-12. The data were obtained in about 57 days of computation time on the Dell OptiPlex machine mentioned before.

The actual FFTs were performed using the FFTW [3] software library. The PARI/GP scripts and the C programs used and the computational results obtained are available at the following web address: http://www.math.unipd.it/~languasc/Littlewood_ineq.html.

2.4. Proof of Theorems 1-2. Theorems 1-2 follow by analysing, using suitable programs written in python, the data computed in subsection 2.3 and collected in two comma-separated values (csv) files. We obtain that the inequalities in the statements of Theorems 1-2 hold and that the minimal value for $M_q$ is $0.604599 \ldots$ attained at $q = 3$ and the maximal one is $6.399873 \ldots$ attained at $q = 4305479$. The minimal value for $m_q$ is $0.198814 \ldots$ attained at $q = 991027$ and the maximal one is $0.618351 \ldots$ attained at $q = 11$.

The output of such python programs are available at the web page: http://www.math.unipd.it/~languasc/Littlewood_ineq.html. A few plots representing Theorems 1-2 are given in Figures 1-2 and 7-8.

3. An alternative algorithm to compute $\log \Gamma(x)$, $x \in (0, 1)$

We describe here an alternative way of computing $\log \Gamma(x)$, $x \in (0, 1)$, which is based on the well-known Euler formula (see, e.g., Lagarias [9, section 3]):

$$\log \Gamma(x) = \gamma(1-x) + \sum_{k=2}^{+\infty} \frac{\zeta(k)}{k}(1-x)^k,$$

where $\zeta(s)$ is the Riemann zeta-function.

We follow the argument used in Languasco-Righi [13] to study the Ramanujan-Deninger Gamma function $\Gamma_1(x)$. We immediately remark that the series in (12) absolutely converges for $x \in (0, 2)$; this fact and the well-known relation

$$\log \Gamma(1+x) = \log \Gamma(x) + \log x, \quad x > 0,$$

let us obtain $\log \Gamma(x)$, $x \in (0, 1)$, in two different ways. Recalling $\log \Gamma(1) = 0$ and $\log \Gamma(1/2) = (\log \pi)/2$, we also remark that, letting $n \in \mathbb{N}$, $n \geq 2$, for every $x \in (0, 2)$ there exists $r = r_T(x, n) \geq 2$ such that

$$\left| \sum_{k=r+1}^{+\infty} \frac{\zeta(k)}{k} (1-x)^k \right| < \frac{\zeta(3)}{3} \sum_{k=r+1}^{+\infty} |1-x|^k < 0.41 \frac{|1-x|^{r+1}}{1-|1-x|} < 2^{-n-1}.$$  \hspace{1cm}(14)

A straightforward computation reveals that we can choose

$$r_T(x, n) = \left\lceil \frac{(n+1) \log 2 + |\log(1-|1-x|)|}{|\log |1-x||} \right\rceil - 1,$$

where we denoted as $[y]$ the least integer greater than or equal to $y \in \mathbb{R}$. 

3.1. The shifting trick for $\log \Gamma(x)$, $x \in (0, 1)$. Clearly $r_{\Gamma}(x, n)$ becomes larger as $|1 - x|$ increases. So when $x$ is close to zero we will evaluate $\log \Gamma$ at $1 + x$ via (13). In the following we will refer to this idea as the shifting trick. This way we will always use the best convergence interval, $x \in (1/2, 3/2)$, we have for the series in (12); we also remark that $r_{\Gamma}(x, n) \leq r_{\Gamma}(1/2, n) = r_{\Gamma}(3/2, n) = n + 1$ for every $x \in (1/2, 3/2)$. Summarising, using (12) and (14), for $x \in (1/2, 1)$ we have that there exists $\theta = \theta(x) \in (-1/2, 1/2)$ such that

$$\log \Gamma(x) = \gamma (1 - x) + \sum_{k=2}^{+\infty} \frac{\zeta(k)}{k} (1 - x)^k = \gamma (1 - x) + \sum_{k=2}^{r_{\Gamma}(x, n)} \frac{\zeta(k)}{k} (1 - x)^k + |\theta| 2^{-n}.$$  \hspace{0.5cm} (15)

We also remark that for $x \in (1/2, 1)$, we have

$$r_{\Gamma}(x, n) = \left\lceil \frac{(n + 1) \log 2 + |\log x|}{\log(1 - x)} \right\rceil - 1 \leq n + 1.$$ \hspace{0.5cm} (16)

Moreover, using (12)-(14), for $x \in (0, 1/2)$ we have that there exists $\eta = \eta(x) \in (-1/2, 1/2)$ such that

$$\log \Gamma(x) = -\log x - x \gamma x + \sum_{k=2}^{+\infty} \frac{(-1)^k \zeta(k)}{k} x^k = -\log x - x \gamma x + \sum_{k=2}^{r'_{\Gamma}(x, n)} \frac{(-1)^k \zeta(k)}{k} x^k + |\eta| 2^{-n},$$

where

$$r'_{\Gamma}(x, n) := r_{\Gamma}(1 + x, n) = \left\lceil \frac{(n + 1) \log 2 + |\log(1 - x)|}{\log x} \right\rceil - 1 \leq n + 1.$$  \hspace{0.5cm} (17)

Since the needed $\zeta$-values can be precomputed and stored with the desired precision (using, for example, PARI/GP), the formulae in (15)-(16) allow us to compute $\log \Gamma(x)$, $x \in (0, 1)$, with a precision of $n$ binary digits using at most $n + 1$ summands; moreover, they also reveal that such a task is, from a computational point of view, essentially as difficult as computing $\log x$ when $x$ is close to $0$.

**Remark 1 (Computational cost).** The estimates $r_{\Gamma}(x, n), r'_{\Gamma}(x, n) \leq n + 1$ for every $x \in (1/2, 1)$ and, respectively, $x \in (0, 1/2)$, imply that $\log \Gamma(x)$, $x \in (0, 1)$ can be obtained with a $n$-bit precision using at most $n + 1$ summands. The summation is performed combining the “pairwise summation” [6] algorithm with Kahan’s [8] method (the minimal block for the pairwise summation algorithm is summed using Kahan’s method) to have a good compromise between precision, computational cost and execution speed. Hence the cost of computing $\log \Gamma(x)$, $x \in (1/2, 1)$ is $\Theta(n)$ floating point products and $\Theta(n)$ floating point summations with a precision of $n$ binary digits; for $x \in (0, 1/2)$ we have the same plus the cost of computing $\log x$.

In the particular case in which $x = a/q$ and $a$ runs over $1, \ldots, q - 1$, the total cost to obtain the values $\{\log \Gamma(a/q) : a = 1, \ldots, q - 1\}$, each one with a precision of $n$ binary digits, is then $\Theta(qn)$ floating point products, plus the cost of computing $(q - 1)/2$ values of the logarithm function.

**Remark 2 (Computation in the whole real axis).** It is clear that using (13) and (15)-(16) we can compute $\log \Gamma(x)$ for every $x > 0$ as follows. For every $x > 0$, we denote as $\lfloor x \rfloor$ the integral part of $x$ and as $\{ x \} = x - \lfloor x \rfloor$ the fractional part of $x$. Hence we obtain:

i) $\log \Gamma(1) = \log \Gamma(2) = 0$ and $\log \Gamma(m) = \sum_{k=2}^{m-1} \log k$ for every $m \in \mathbb{N}, m \geq 3$;

ii) for $x > 1$, $x \notin \mathbb{N}$, we compute $\log \Gamma(x)$ as $\log \Gamma(x) = \log \Gamma(\{ x \}) + \sum_{k=0}^{\lfloor x \rfloor - 1} \log(\{ x \} + k)$;

iii) $\log \Gamma(1/2) = (\log \pi)/2$;

iv) for $x \in (0, 1/2)$, we compute $\log \Gamma(x)$ as in (16);

v) for $x \in (1/2, 1)$, we compute $\log \Gamma(x)$ as in (15).
Even if we are mainly interested in working with \( x \in (0, 1) \) we recall that for \( x \) large it might be more convenient to implement Stirling’s formula for \( \log \Gamma(x) \).

**Remark 3 (Enlarging the convergence radius).** We remark, even if it is not useful in our application, that the size of the convergence interval in (12) can be doubled by isolating the Taylor series at 1 of \( \log x - (x - 1) \) in (12) thus getting

\[
\log \Gamma(x) = -\log x + (\gamma - 1)(1 - x) + \sum_{k=2}^{+\infty} \frac{\zeta(k) - 1}{k}(1 - x)^k. \tag{17}
\]

Using the well-known estimate \(|\zeta(k) - 1| < 2^{1-k}\) for every \( k \in \mathbb{N}, k \geq 3 \), it is easy to prove that the series in (17) converges for every \( x \in (-1, 3) \).

**Remark 4 (Computation in the complex plane).** 1) It seems that the argument leading to (15)-(16) is not usually implemented in the most used software libraries or Computer Aided Systems (CAS) for Mathematics probably because the shifting trick used before can be directly generalised to complex variables only in a thin horizontal strip around the positive part of the real axis, see the next point of this remark. In fact, many software libraries and CAS usually implement the computation of \( \log \Gamma(z) \), \( z \in \mathbb{C}, z \neq -n, n \in \mathbb{N} \), using the Lanczos approximation thus following the setting of Press et al. [16].

2) A possible complex strip can be built combining (13), which in fact holds for any argument \( z \in \mathbb{C}, z \neq -n, n \in \mathbb{N} \), together with the complex power series contained in the following formula

\[
\log \Gamma(z) = \gamma(1 - z) + \sum_{k=2}^{+\infty} \frac{\zeta(k)}{k}(1 - z)^k \tag{18}
\]

which generalises (12) to the region \(|z - 1| < 1\). We can start from the rectangle \( 1/2 \leq \Re(z) < 3/2, |\Im(z)| \leq 1/4 \), since for every \( z \) in this region less than \( 1.2 \cdot (n + 1) + 4 \) terms are sufficient to have a precision of \( n \) binary digits in computing a truncation of the series in (18). We also remark here that such a strip can be vertically enlarged using Gauß’ multiplication theorem in the following form

\[
\log \Gamma(mz) = \frac{1-m}{2} \log(2\pi) + \left( mz - \frac{1}{2} \right) \log m + \sum_{j=0}^{m-1} \log \Gamma \left( z + \frac{j}{m} \right),
\]

where \( m \in \mathbb{N}, m \geq 1 \).

### 3.2. Reflection formulae.

We now remark that using (15)-(16) to compute \( \log \Gamma(x) + \log \Gamma(1-x) \), \( x \in (0, 1) \), the odd summands of the series will vanish and something similar happens in computing \( \log \Gamma(x) - \log \Gamma(1-x) \). We summarise the situation in the following

**Proposition 1.** Let \( x \in (0, 1), x \neq 1/2, n \in \mathbb{N}, n \geq 2, r_1(x,n) = \left\lceil \frac{\log(1-x)}{\log(x)} \right\rceil - 1 \right\rceil/2 \) and \( r_2(x,n) = \left\lfloor \frac{\log(x)}{\log(1-x)} \right\rfloor - 1 \right\rceil/2 \). Using (15) and (16), we have that there exists \( \theta = \theta(x) \in (-1/2, 1/2) \) such that for \( 0 < x < 1/2 \) we have

\[
\log \Gamma(x) + \log \Gamma(1-x) = -\log x + \sum_{\ell=1}^{r_1} \frac{\zeta(2\ell)}{\ell} x^{2\ell} + |\theta|2^{-n}, \tag{19}
\]
we compared the practical running times of computing

\[ \log \Gamma(x) - \log \Gamma(1 - x) = - \log x - 2\gamma x - 2 \sum_{\ell=1}^{r_1} \frac{\zeta(2\ell + 1)}{2\ell + 1} x^{2\ell+1} + |\theta|2^{-n}, \]  

(20)

and for \(1/2 < x < 1\) we have

\[ \log \Gamma(x) + \log \Gamma(1 - x) = -\log(1 - x) + \sum_{\ell=1}^{r_2} \frac{\zeta(2\ell)}{\ell} (1 - x)^{2\ell} + |\theta|2^{-n}, \]  

(21)

\[ \log \Gamma(x) - \log \Gamma(1 - x) = \log(1 - x) + 2\gamma(1 - x) + 2 \sum_{\ell=1}^{r_2} \frac{\zeta(2\ell + 1)}{2\ell + 1} (1 - x)^{2\ell+1} + |\theta|2^{-n}. \]  

(22)

**Proof.** Assume that \(0 < x < 1/2\); we compute \(\log \Gamma(x)\) with the infinite series in (16) and \(\log \Gamma(1 - x)\) with the infinite series in (15). Since they absolutely converge, their sum is obtained with the series having as summands the sum of their coefficients. Arguing as in (14) and remarking that \(r_1(x, n) = r_1(1 - x, n)/2 = r_1'(x, n)/2\), we immediately have that (19) holds since the odd summands vanish. Assume that \(1/2 < x < 1\); in this case we compute \(\log \Gamma(x)\) with the infinite series in (15) and \(\log \Gamma(1 - x)\) with the infinite series in (16). Since they absolutely converge, their sum is obtained with the series having as summands the sum of their coefficients. Arguing as in (14) and remarking that \(r_2(x, n) = r_2(1 - x, n)/2 = r_2'(1 - x, n)/2\), we immediately have that (21) holds since the odd summands vanish. The derivation of (20) and (22) is similar.

This completes the proof. \(\square\)

It is worth mentioning that the right hand side in (21) can be obtained from the one in (19) formally replacing \(x\) with \(1 - x\); and that, also changing of sign, the same holds for (20) and (22). Using \(\Gamma(x)\Gamma(1 - x) = \pi/\sin(\pi x)\), Proposition 1 also immediately gives a way of writing \(\sin(\pi x)\) in term of logs and values of the Riemann zeta-function at positive even integers, see Remark 5 below. Comparing with (11), the use of Proposition 1 in our application is particularly efficient for the following reasons:

- the cancellation of the odd terms we have in (19) and (21) leads to gain a factor of 2 in the computational cost since we just need to use half of the summands (the ones with even indices); a similar remark applies to (20) and (22) too;
- in (19) and (21) the number of summands is \(\leq (n + 1)/2\); hence to have a precision of \(n\) bits we just need less than \((n + 1)/2\) summands (assuming the logarithm function can be evaluated with the same precision); a similar remark applies to (20) and (22) too;
- in (19) and (21) just the values of the Riemann zeta-function at positive even integers are required and for them we can use the well-known exact formulæ involving the Bernoulli numbers \(B_k\): \(\zeta(2\ell) = (-1)^{\ell+1}\frac{B_{2\ell}(2\pi)^{2\ell}}{2(2\ell)!}\), for every \(\ell \in \mathbb{N}, \ell \geq 1\), where the Bernoulli numbers \(B_k\) are defined as the coefficients of the following series expansion: \(\frac{t}{e^t-1} = \sum_{k=0}^{+\infty} B_k \frac{t^k}{k!}, \ |t| < 2\pi\).

3.3. **Comparing running times.** We implemented (15)-(16) and the formulæ of Proposition 1 both in the scripting language of PARI/GP and in the C programming language. In the first case (PARI/GP and gp2e), using a precision of 128 bits, i.e., letting \(n = 128\), we compared the practical running times of computing \(\log \Gamma(g^k/q)\), \(k = 0, \ldots, q - 2\), for \(q = 10007, 305741, 6766811, 10000019, 28227761\), \(g\) being a fixed primitive root of \(q\). In all these cases the use of (15)-(16) improved the total running times by a 40\% factor with respect to the ones obtained using the predefined functions of PARI/GP. Further improvements can be obtained using Proposition 1 if the particular application we are working on allows its use.
In the second case (C programming language), we repeated the computation previously described and we then compared the running times of our implementation of (15)-(16) and of the long double precision version of \( \log \Gamma \) defined in the C language (the 1gamma1 function). We clearly used a precision of 80 bits (\( n = 80 \)). In this case our functions are slower of a factor 2.5 than 1gamma1, while, for the formulae of Proposition 1, our functions are slower of a factor 1.3 with respect to 1gamma1. In both cases a low-level implementation of our results might lead to a different outcome.

**Remark 5 (A digression on \( \sin u, \pi \) and \( \gamma \)).** 1) A straightforward computation which uses\( \Gamma(x)\Gamma(1 - x) = \pi / \sin(\pi x) \) and Proposition 1 immediately gives the well-known formula

\[
\sin(\pi x) = \pi x \exp \left( -\sum_{\ell=1}^{+\infty} \frac{\zeta(2\ell)}{\ell} x^{2\ell} \right), \quad (0 < x \leq \frac{1}{2}),
\]

which, combined with the parity of the sin-function and \( \sin(0) = 0 \), can also be extended to the whole interval \(-1/2 < x \leq 1/2\). Equation (23), combined with the Bernoulli numbers definition, gives also

\[
\sin u = u \exp \left( \sum_{\ell=1}^{+\infty} (-1)^{\ell} \frac{2^{2\ell - 1}}{\ell(2\ell)!} B_{2\ell} u^{2\ell} \right),
\]

for every \( u \in (-\pi/2, \pi/2) \).

2) Computation of \( \pi \). As a matter of curiosity, since we know that there are faster algorithms for this task, we remark that, computing \( \log \Gamma(x) + \log \Gamma(1 - x) \) at \( x = 1/2 \) with (15)-(16), we obtain

\[
\log \pi = \log 2 + \sum_{\ell=1}^{+\infty} \frac{\zeta(2\ell)}{\ell 4^\ell},
\]

which can be used to compute \( \pi \); in fact, a straightforward implementation using the scripting language of PARI/GP let us compute 10 000 decimal digits of \( \pi \) in about 3 seconds and 277 milliseconds while, for getting 1 000 decimal digits we just needed 19 milliseconds on the Dell OptiPlex machine previously mentioned.

3) Computation of \( \gamma \). As a matter of curiosity, since we know that there are faster algorithms for this task, we remark that, computing \( \log \Gamma(x) - \log \Gamma(1 - x) \) at \( x = 1/2 \) with (15)-(16), we obtain the following result (first obtained by Stieltjes in 1887):

\[
\gamma = \log 2 - \sum_{\ell=1}^{+\infty} \frac{\zeta(2\ell + 1)}{(2\ell + 1)4^\ell}.
\]

Such last formula can be clearly used to compute \( \gamma \); in fact, a straightforward implementation using the scripting language of PARI/GP let us compute 10 000 decimal digits of \( \gamma \) in about 5 minutes, 19 seconds and 255 milliseconds while, for getting 1 000 decimal digits we just needed 167 milliseconds on the Dell OptiPlex machine mentioned before.

4. **An alternative algorithm to compute** \( \psi(x), x \in (0, 1) \)

Here we apply to the digamma function \( \psi(x) = \Gamma'/\Gamma(x), x \in (0, 1) \), the same argument used in Section 3. The starting point is the well-known Euler formula (see, e.g., Lagarias [9, section 3]):

\[
\psi(x) = -\gamma - \sum_{k=2}^{+\infty} \frac{\zeta(k)(1-x)^{k-1}}{k}.
\]

(24)
We immediately remark that the series in (24) absolutely converges for \(x \in (0, 2)\); this fact and the well-known relation
\[
\psi(1 + x) = \psi(x) + \frac{1}{x}
\]  
(25)
let us obtain \(\psi(x)\), \(x \in (0, 1)\), in two different ways. Recalling \(\psi(1) = -\gamma\) and \(\psi(1/2) = -2 \log 2 - \gamma\), we also remark that, letting \(n \in \mathbb{N}\), \(n \geq 2\), for every \(x \in (0, 2)\) there exists \(r = r_\psi(x, n) \geq 2\) such that
\[
\left| \sum_{k=r+1}^{+\infty} \zeta(k)(1 - x)^{k-1} \right| < \zeta(3) \sum_{k=r+1}^{+\infty} |1 - x|^{k-1} = 1.21 \frac{|1 - x|^r}{1 - |1 - x|} < 2^{-n+1}. 
\]  
(26)
A straightforward computation reveals that we can choose
\[
r_\psi(x, n) = \left\lceil \frac{(n + 2) \log 2 + \left|\log(1 - |1 - x|)\right|}{\log |1 - x|} \right\rceil.
\]

4.1. **The shifting trick for** \(\psi(x)\), \(x \in (0, 1)\). As for \(r_\gamma(x, n)\), we clearly have that \(r_\psi(x, n)\) becomes larger as \(|1 - x|\) increases. We also remark that, using (25), we can exploit the shifting trick in this case too. This way we will always use the best convergence interval, \(x \in (1/2, 3/2)\), we have for the series in (12); we also remark that \(r_\psi(x, n) \leq r_\psi(1, 2, n) = r_\psi(3/2, n) = n + 3\) for every \(x \in (1/2, 3/2)\). Summarising, using (24) and (26), for \(x \in (1/2, 1)\) we have that there exists \(\theta = \theta(x) \in (-1/2, 1/2)\) such that
\[
\psi(x) = -\gamma - \sum_{k=2}^{+\infty} \zeta(k)(1 - x)^{k-1} = -\gamma - \sum_{k=2}^{r_\psi(x, n)} \zeta(k)(1 - x)^{k-1} + |\theta|2^{-n}. 
\]  
(27)
We also remark that for \(x \in (1/2, 1)\), we have \(r_\psi(x, n) = \left\lceil \frac{(n+2) \log 2 + \left|\log x\right|}{\log |1-x|} \right\rceil \leq n + 3\). Moreover, using (24)-(26), for \(x \in (0, 1/2)\) we have that there exists \(\eta = \eta(x) \in (-1/2, 1/2)\) such that
\[
\psi(x) = -\frac{1}{x} - \gamma - \sum_{k=2}^{+\infty} (-1)^{k-1} \zeta(k)x^{k-1} = -\frac{1}{x} - \gamma - \sum_{k=2}^{r'_\psi(x, n)} (-1)^{k-1} \zeta(k)x^{k-1} + |\eta|2^{-n}, 
\]  
(28)
where \(r'_\psi(x, n) := r_\psi(1 + x, n) = \left\lceil \frac{(n+2) \log 2 + \left|\log(1-x)\right|}{\log x} \right\rceil \leq n + 3\).

We also remark that the series in the middle of (27)-(28) can also be obtained from the ones in (15)-(16) by differentiation. Since the needed \(\zeta\)-values can be precomputed and stored with the desired precision (using, for example, PARI/GP), the formulae on the right hand sides of (27)-(28) allow us to compute \(\psi(x), x \in (0, 1),\) with a precision of \(n\) binary digits using at most \(n + 3\) summands; moreover they also reveal that, from a computational point of view, such a task is essentially as difficult as computing \(1/x\) when \(x\) is close to 0.

**Remark 6 (Computation in the whole real axis).** It is clear that using (25) and (27)-(28) we can compute \(\psi(x)\) for every \(x > 0\) as follows. For every \(x > 0\), we denote as \([x]\) the integral part of \(x\) and as \(\{x\} = x - [x]\) the fractional part of \(x\). Hence we obtain:

i) \(\psi(1) = -\gamma\) and \(\psi(m) = -\gamma + \sum_{k=1}^{m-1} 1/k\) for every \(m \in \mathbb{N}, m \geq 2;\)
ii) for \(x > 1, x \notin \mathbb{N}\), we compute \(\psi(x)\) as \(\psi(x) = \psi([x]) + \sum_{k=0}^{\{x\}-1} 1/([x] + k);\)
iii) \(\psi(1/2) = -2 \log 2 - \gamma;\)
iv) for \(x \in (0, 1/2)\), we compute \(\psi(x)\) as in (28);
v) for \(x \in (1/2, 1)\), we compute \(\psi(x)\) as in (27).

Even if we are mainly interested in working with \(x \in (0, 1)\) we recall that for \(x\) large it might be more efficient to implement an asymptotic formula for \(\psi(x)\).
Remark 7 (Enlarging the convergence radius). The size of the convergence interval in (24) can be doubled by isolating the Taylor series at 1 of \(1/x - 1\) in (24) thus getting
\[
\psi(x) = -\frac{1}{x} - \gamma + 1 - \sum_{k=2}^{+\infty} (\zeta(k) - 1)(1 - x)^{k-1}.
\] (29)

Using the well-known estimate \(|\zeta(k) - 1| < 2^{1-k}\) for every \(k \in \mathbb{N}, k \geq 3\), it is easy to prove that the series in (29) converges for every \(x \in (-1, 3)\).

Remark 8 (Computation in the complex plane). 1) It seems that the argument leading to (27)-(28) is not usually implemented in the most used software libraries or Computer Aided Systems for Mathematics for the same reasons we discussed for the log \(\Gamma\)-function.

2) We argue analogously as we did for the log \(\Gamma\)-function. Using the formula
\[
\psi(z) = -\gamma - \sum_{k=2}^{+\infty} \zeta(k)(1 - z)^{k-1}
\] (30)

which generalises (24) to the region \(|z - 1| < 1\), and (25), which in fact holds for any argument \(z \in \mathbb{C}, z \neq -n, n \in \mathbb{N}\), we can build a possible complex strip starting from the rectangle \(1/2 \leq \Re(z) < 3/2, |\Im(z)| \leq 1/4\), in which less than \(1.2 \cdot (n + 2) + 4\) terms are sufficient to have a precision of \(n\) binary digits in computing a suitable truncation of the series in (30). Moreover, such a strip can be vertically enlarged using the following form of Gauß’ multiplication theorem
\[
\psi(mz) = \log m + \frac{1}{m} \sum_{j=0}^{m-1} \psi\left(z + \frac{j}{m}\right),
\]

where \(m \in \mathbb{N}, m \geq 1\).

4.2. Reflection formulae. We now remark that in using (27)-(28) to compute \(\psi(x) - \psi(1 - x)\), \(x \in (0, 1)\), the odd summands in the series will vanish (and in the corresponding series for \(\psi(x) + \psi(1 - x)\) the even summands will be discarded too). We summarise the situation in the following

Proposition 2. Let \(x \in (0, 1)\), \(x \neq 1/2\), \(n \in \mathbb{N}\), \(n \geq 2\), \(r_1(x, n) = \lceil (n+2)\log 2 + \log(1-x) \rceil / 2\) and \(r_2(x, n) = \lceil (n+2)\log 2 + \log x \rceil / 2\). Using (27)-(28), we have that there exists \(\theta = \theta(x) \in (-1/2, 1/2)\) such that for \(0 < x < 1/2\) we have
\[
\psi(x) - \psi(1 - x) = -\frac{1}{x} + 2 \sum_{\ell=1}^{r_1} \zeta(2\ell)x^{2\ell-1} + |\theta|2^{-n},
\] (31)
\[
\psi(x) + \psi(1 - x) = -2\gamma - \frac{1}{x} - 2 \sum_{\ell=1}^{r_1} \zeta(2\ell + 1)x^{2\ell} + |\theta|2^{-n},
\] (32)

and for \(1/2 < x < 1\) we have
\[
\psi(x) - \psi(1 - x) = \frac{1}{1 - x} - 2 \sum_{\ell=1}^{r_3} \zeta(2\ell)(1 - x)^{2\ell-1} + |\theta|2^{-n},
\] (33)
\[
\psi(x) + \psi(1 - x) = -2\gamma - \frac{1}{1 - x} - 2 \sum_{\ell=1}^{r_3} \zeta(2\ell + 1)(1 - x)^{2\ell} + |\theta|2^{-n}.
\] (34)
Proof. Assume that $0 < x < 1/2$; we compute $\psi(x)$ with the series in (28) and $\psi(1-x)$ with the series in (27). Since they absolutely converge, their sum is obtained with the series having as summands the sum of their coefficients. Arguing as in (26) and remarking that $r_1(x, n) = r_\psi(1-x, n)/2 = r_\psi'(x, n)/2$, we immediately have that (31) holds since the odd summands vanish. Assume that $1/2 < x < 1$; in this case we compute $\psi(x)$ with the series in (27) and $\psi(1-x)$ with the series in (28). Since they absolutely converge, their sum is obtained with the series having as summands the sum of their coefficients. Arguing as in (26) and remarking that $r_2(x, n) = r_\psi(x, n)/2 = r_\psi'(1-x, n)/2$, we immediately have that (33) holds since the odd summands vanish. The derivation of (32) and (34) is similar. This completes the proof. □

It is worth mentioning that the right hand side in (33) can be obtained from the one in (31) formally replacing $x$ with $1-x$ and changing sign; and that, without the change of sign, the same holds for (32) and (34). Using $\psi(1-x) - \psi(x) = \pi \cot(\pi x)$, Proposition 2 also immediately gives a way of writing $\cot(\pi x)$ in term of logs and values of the Riemann zeta-function at positive even integers, see Remark 9 below. The use of Proposition 2 is particularly efficient for the same reasons we already described for Proposition 1; we just need to remark that the number of summands in this case is $\leq (n+3)/2$.

Remark 9 (A digression on $\cot u$). A straightforward computation which uses $\psi(1-x) - \psi(x) = \pi \cot(\pi x)$ and Proposition 2 immediately gives the well-known formula

$$\cot(\pi x) = \frac{1}{\pi x} - 2 \sum_{\ell=1}^{+\infty} \frac{\zeta(2\ell)}{\ell} x^{2\ell-1} \quad (0 < x \leq \frac{1}{2})$$

which, combined with the parity of the cotangent function, can also be extended to $-1/2 < x \leq 1/2$, $x \neq 0$.

References

[1] H. Cohen, Number Theory. Volume II: Analytic and Modern Tools, Graduate Texts in Mathematics, vol. 240, Springer, 2007.

[2] K. Ford, F. Luca, P. Moree, Values of the Euler $\phi$-function not divisible by a given odd prime, and the distribution of Euler-Kronecker constants for cyclotomic fields, Math. Comp. 83 (2014), 1447–1476.

[3] M. Frigo, S. G. Johnson, The Design and Implementation of FFTW3, Proceedings of the IEEE 93(2), 216–231 (2005). The C library is available at http://www.fftw.org.

[4] A. Granville, K. Soundararajan, Upper bounds for $|L(1, \chi)|$, Quart. J. Math. 53 (2002), 265–284.

[5] A. Granville, K. Soundararajan, The distribution of values of $L(1, \chi_d)$, Geom. Funct. Anal. 13 (5), 992–1028, (2003)

[6] N. J. Higham, The accuracy of floating point summation, SIAM Journal on Scientific Computing, 14 (1993), 783–799.

[7] M. J. Jacobson Jr., S. Ramachandran, H. C. Williams, Numerical results on class groups of imaginary quadratic fields, Algorithmic number theory, Lecture Notes in Comput. Sci., vol. 4076, Springer, Berlin, 2006, pp. 87–101.

[8] W. Kahan, Further remarks on reducing truncation errors, Communications of the ACM 8 (1965), page 40.

[9] J. C. Lagarias, Euler’s constant: Euler’s work and modern developments, Bull. Amer. Math. Soc. 50 (2013), 527–628.
[10] Y. Lamzouri, X. Li, K. Soundararajan, Conditional bounds for the least quadratic non-residue and related problems, Math. Comp. 84 (2015), 2391–2412. Corrigendum ibid., Math. Comp. 86 (2017), 2551–2554.

[11] A. Languasco, Efficient computation of the Euler-Kronecker constants for prime cyclotomic fields, Arxiv, 2019, http://arxiv.org/abs/1903.05487, submitted.

[12] A. Languasco, P. Moree, S. Saad Eddin, A. Sedunova, Computation of the Kummer ratio of the class number for prime cyclotomic fields, Arxiv (2019), http://arxiv.org/abs/1908.01152.

[13] A. Languasco, L. Righi, A fast algorithm to compute the Ramanujan-Deninger Gamma-function and some number-theoretic applications, Arxiv, 2020, http://arxiv.org/abs/2005.10046, submitted.

[14] J. E. Littlewood, On the class number of the corpus $P(\sqrt{-k})$, Proc. London Math. Soc. 27 (1928), 358–372.

[15] The PARI Group, PARI/GP version 2.11.4, Bordeaux, 2020. Available from http://pari.math.u-bordeaux.fr/.

[16] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, Numerical recipes. The art of scientific computing, third edition, Cambridge U.P., 2007.

[17] C. M. Rader, Discrete Fourier transforms when the number of data samples is prime, Proc. IEEE 56 (1968), 1107–1108.

[18] D. Shanks, Systematic Examination of Littlewood’s Bounds on $L(1, \chi)$, Proc. Sympos. Pure Math., vol. 24, Amer. Math. Soc., 1973, pp. 267–283.

[19] H. C. Williams, J. Broere, A computational technique for evaluating $L(1, \chi)$ and the class number of a real quadratic field, Math. Comp. 30 (1976), 887–893.

Alessandro Languasco Università di Padova, Dipartimento di Matematica, “Tullio Levi-Civita”, Via Trieste 63, 35121 Padova, Italy. e-mail: alessandro.languasco@unipd.it
TABLE 1. Values of $M_q$ for every odd prime up to 1000 with 30-digit precision; computed with PARI/GP, v. 2.11.4, with trivial summing over a. Total computation time: 16 sec., 539 milliseconds.
Table 2. Values of $m_q$ for every odd prime up to 1000 with 30-digit precision; computed with PARI/GP, v. 2.11.4, with trivial summarizing over $a$. Total computation time: 16 sec., 81 milliseecs.
Figure 1. Graphical representation of Theorem 1. The values of $M_q$, $q$ prime, $500 \leq q \leq 10^7$. The minimal value for $M_q$ is $0.604599\ldots$ attained at $q = 3$ and the maximal one is $6.399873\ldots$ attained at $q = 4305479$. The blue line represents $0.62L_2 f(q)$; the red one $0.325L_2 f(q)$, where $f(q)$ is defined in (5) and $L_2 := 2e^7$. The green line represents $0.66L_2 \log \log q$; the orange one $0.4L_2 \log \log q$. For $3 \leq q < 500$, see the next plot.

Figure 2. Graphical representation of Theorem 1. The values of $M_q$, $q$ prime, $3 \leq q \leq 500$. The blue line represents $0.62L_2 f(q)$; the red one $0.325L_2 f(q)$, where $f(q)$ is defined in (5) and $L_2 = 2e^7$. The green line represents $0.66L_2 \log \log q$; the orange one $0.4L_2 \log \log q$. 
NUMERICAL VERIFICATION OF LITTLEWOOD’S BOUNDS FOR $|L(1, \chi)|$

Figure 3. The values of $M'_q := M_q / f(q)$, $q$ prime, $500 \leq q \leq 10^7$, where $f(q)$ is defined in (5). The minimal value for $M'_q$ is 0.057396... attained at $q = 3$ and the maximal one is 2.206927... attained at $q = 4305479$ (the “second” maximal value is 2.192260... attained at $q = 3190151$). The blue line represents $0.62L_2$; the red one $0.325L_2$, where $L_2 = 2e^\gamma$. For $3 \leq q < 500$, see the next plot.

Figure 4. The values of $M'_q := M_q / f(q)$, $q$ prime, $3 \leq q \leq 500$, where $f(q)$ is defined in (5). The blue line represents $0.62L_2$; the red one $0.325L_2$, where $L_2 = 2e^\gamma$. 
Figure 5. The values of $M''_q := M_q / \log \log q$, $q$ prime, $500 \leq q \leq 10^7$. The minimal value for $M''_q$ is $1.492809 \ldots$ attained at $q = 13$ and the maximal one is $6.428641 \ldots$ attained at $q = 3$ (the “second” maximal value is $2.347506 \ldots$ attained at $q = 4305479$, marked in red in this plot). The green line represents $0.66L_2$; the orange one $0.4L_2$, where $L_2 = 2e^\gamma$. For $3 \leq q < 500$, see the next plot.

Figure 6. The values of $M''_q := M_q / \log \log q$, $q$ prime, $5 \leq q \leq 500$. The green line represents $0.66L_2$; the orange one $0.4L_2$, where $L_2 = 2e^\gamma$. $M''_3 = 6.428641 \ldots$ is not included in this plot.
Figure 7. Graphical representation of Theorem 2. The values of $m_q$, $q$ prime, $500 \leq q \leq 10^7$. The minimal value for $m_q$ is $0.198814 \ldots$ attained at $q = 991027$ and the maximal one is $0.618351 \ldots$ attained at $q = 11$. The blue line represents $5L_1/g(q)$; the red one $2.35L_1/g(q)$, where $g(q)$ is defined in (5) and $L_1 := \frac{\pi^2}{12e^2}$. The green line represents $2L_1/\log \log q$; the orange one $1.13L_1/\log \log q$. For $3 \leq q < 500$, see the next plot.

Figure 8. Graphical representation of Theorem 2. The values of $m_q$, $q$ prime, $500 \leq q \leq 10^7$. The blue line represents $5L_1/g(q)$; the red one $2.35L_1/g(q)$, where $g(q)$ is defined in (5) and $L_1 := \frac{\pi^2}{12e^2}$. The green line represents $2L_1/\log \log q$; the orange one $1.13L_1/\log \log q$. 
Figure 9. The values of $m'_q := m_q g(q)$, $q$ prime, $500 \leq q \leq 10^7$, where $g(q)$ is defined in (5). The minimal value for $m'_q$ is 1.088477 . . . attained at $q = 991027$ (the “second” minimal value is 1.134017 . . . attained at $q = 759827$) and the maximal one is 7.093329 . . . attained at $q = 3$. The blue line represents $5L_1$; the red one $2.35L_1$, where $L_1 = \frac{\pi^2}{12\zeta(2)}$. For $3 \leq q < 500$, see the next plot.

Figure 10. The values of $m'_q := m_q g(q)$, $q$ prime, $3 \leq q \leq 500$, where $g(q)$ is defined in (5). The blue line represents $5L_1$; the red one $2.35L_1$, where $L_1 = \frac{\pi^2}{12\zeta(2)}$. $m'_3 = 7.093329 . . .$ is not included in this plot.
Figure 11. The values of $m''_q := m_q \log \log q$, $q$ prime, $500 \leq q \leq 10^7$. The minimal value for $m''_q$ is 0.056861... attained at $q = 3$ (the “second” minimal value is 0.2048251... attained at $q = 5$) and the maximal one is 0.7445135... attained at $q = 19001$. The green line represents $2L_1$; the orange one $1.13L_1$, where $L_1 = \frac{\pi^2}{12e\gamma}$. The red point represents $m''_{991027} = 0.521914...$ For $3 \leq q < 500$, see the next plot.

Figure 12. The values of $m''_q := m_q \log \log q$, $q$ prime. The green line represents $2L_1$; the orange one $1.13L_1$, where $L_1 = \frac{\pi^2}{12e\gamma}$. 