A FAST ALGORITHM TO COMPUTE THE RAMANUJAN-DENINGER GAMMA-FUNCTION AND SOME NUMBER-THEORETIC APPLICATIONS

ALESSANDRO LANGUASCO AND LUCA RIGHI

Abstract. We introduce a fast algorithm to compute the Ramanujan-Deninger gamma function and its logarithmic derivative at positive values. Such an algorithm allows us to greatly extend the numerical investigations about the Euler-Kronecker constants $\theta_q$, $\theta_q^*$ and $M_q = \max_{x \neq x_0} |L'/L(1, \chi)|$, where $q$ is an odd prime, $\chi$ runs over the primitive Dirichlet characters mod $q$, $\chi_0$ is the principal Dirichlet character mod $q$ and $L(s, \chi)$ is the Dirichlet $L$-function associated to $\chi$. Using such algorithms we obtained that $\theta_{5004095563} = -0.16595399 \ldots$ and $\theta_{5004095563}^* = 13.89764738 \ldots$ thus getting a new negative value for $\theta_q$.

Moreover we also computed $\theta_q$, $\theta_q^*$ and $M_q$ for every prime $q$, $10^6 < q \leq 10^7$, thus extending the results in [18]. As a consequence we obtain that both $\theta_q$ and $\theta_q^*$ are positive for every odd prime $q$ up to $10^7$ and that $\frac{\log q}{\log 2} < M_q < \frac{5}{4} \log q$ for every prime $1531 < q \leq 10^7$. In fact the lower bound holds true for $q > 13$. The programs used and the results here described are collected at the following address [http://www.math.unipd.it/~languasc/Scomp-appl.html]

1. Introduction

We introduce a fast algorithm to compute the Ramanujan-Deninger gamma function and its logarithmic derivative at positive values. We then use such a new algorithm to efficiently compute $L'/L(1, \chi)$, where $\chi$ runs over the non-principal primitive Dirichlet characters mod $q$, $q$ is an odd prime and $L(s, \chi)$ is the Dirichlet $L$-function associated to $\chi$. Such a quantity is involved in several interesting number-theoretic problems like the evaluation of the Euler-Kronecker constants $\theta_q$ for the cyclotomic field $\mathbb{Q}(\zeta_q)$, $\zeta_q$ being a $q$-root of unity, the analogous problem for $\theta_q^*$ attached to $\mathbb{Q}(\zeta_q + \zeta_q^{-1})$, the maximal real subfield of $\mathbb{Q}(\zeta_q)$, and the study of the extremal values of $M_q = \max_{x \neq x_0} |L'/L(1, \chi)|$. We will give a detailed description of such problems in Section 4.

Following Deninger’s notation in [3], we introduce now the functions we will work on. The main object is the Ramanujan-Deninger Gamma function $\Gamma_1(x) := \exp(R(x))$, $x > 0$, where

$$ R(x) := -\frac{\partial^2}{\partial s^2} \zeta(s,x)|_{s=0}, $$

$\zeta(s,x)$ is the Hurwitz zeta function, $\zeta(s,x) = \sum_{n=0}^{\infty} (n+x)^{-s}$ for $\Re(s) > 1$ and it is meromorphically extended to $s \in \mathbb{C} \setminus \{1\}$. We recall that $\zeta(s,1)$ is Riemann’s zeta-function $\zeta(s)$. Using eq. (2.3.2) of [3], the $R$-function can be expressed for every $x > 0$ by

$$ R(x) = -\zeta''(0) - S(x), \quad (1) $$

$$ S(x) := 2\gamma_1 x + (\log x)^2 + \sum_{k=1}^{\infty} \left( (\log (k+x))^2 - (\log k)^2 - 2x \frac{\log k}{k} \right), \quad (2) $$

2010 Mathematics Subject Classification. Primary 33-04, 11-04; secondary 33E20, 11Y16, 11Y60.

Key words and phrases. Generalised Gamma functions, Euler-Kronecker constants, extremal values for the logarithmic derivative of Dirichlet $L$-functions, application of the Fast Fourier Transform.
where
$$\gamma_1 = \lim_{n \to +\infty} \left( \sum_{k=1}^{n} \frac{\log k}{k} - \frac{(\log n)^2}{2} \right), \quad \zeta''(0) = \frac{1}{2} \left( -\left( \log 2\pi \right)^2 - \frac{\pi^2}{12} + \gamma_1 + \gamma^2 \right)$$  \hspace{1cm} (3)

and $\gamma$ is the Euler-Mascheroni constant. We introduced the $S$-function because in the applications we will see in Section 4 below the constant term $\zeta''(0)$ will play no role and hence we may focus our attention just on the $S$-function. We also have $S(1) = 0$ and $R(1) = -\zeta''(0)$.

A key point to be able to obtain the following results is that $R(x)$ is the unique solution in $(0, +\infty)$ of the difference equation $R(x + 1) = R(x) + (\log x)^2$, with initial condition $R(1) = -\zeta''(0)$, which is convex in some interval $(A, +\infty)$, $A > 0$, see Theorem 2.3 of Deninger [13]. As a consequence the $S$-function too verifies a difference equation:

$$S(x + 1) = S(x) - (\log x)^2 \quad \text{for every } x > 0.$$  \hspace{1cm} (4)

Another important ingredient to be used to derive our results is the following alternative definition of $S(x)$, $x > 0$, which is implicitly contained in eq. (2.12) of Deninger [3]. It is the analogue for $S$ of Plana’s integral for $\log \Gamma$, where $\Gamma(s), s \in \mathbb{C} \setminus (-\mathbb{N})$, is Euler’s Gamma function:

$$S(x) = 2 \int_{0}^{+\infty} \left( (x - 1)e^{-t} + \frac{e^{-xt} - e^{-t}}{1 - e^{-t}} \right) \frac{\gamma + \log t}{t} \, dt.$$  \hspace{1cm} (5)

We introduce now the first derivative of $R(x)$, namely

$$\psi_1(x) := \frac{1}{2} R'(x) = \frac{1}{2} \frac{\Gamma'}{\Gamma}(x).$$  \hspace{1cm} (6)

The factor $1/2$ in (6) is needed because Deninger, in its definition of $R(x)$, used an extra factor 2 and we need now to remove it to connect $\psi_1$ with the results proved by other authors. We also recall that generalised $\psi$-functions of this kind occur in Ramanujan’s second notebook, see [11, Chapter 8, Entry 22]. Differentiating (2) we have

$$\psi_1(x) = -\gamma_1 - \frac{\log x}{x} - \sum_{k=1}^{+\infty} \left( \frac{\log(k + x)}{k + x} - \frac{\log k}{k} \right)$$

for $x > 0$. We also define

$$T(x) := \gamma_1 + \psi_1(x)$$  \hspace{1cm} (7)

so that $T(1) = 0$. As before, we introduced the function $T(x)$ because the constant term $\gamma_1$ in the definition of $\psi_1$ will play no role in the applications contained in Section 4 below. Moreover, since $S'(x) = 2(\gamma_1 - T(x))$, it follows that

$$T(x) = \gamma_1 - \frac{1}{2} S'(x) \quad \text{and} \quad T(x + 1) = T(x) + \frac{\log x}{x} \quad \text{for every } x > 0.$$  \hspace{1cm} (8)

In our applications, see Section 4 below, we will need to evaluate $S$ or $T$ at some rational points contained in $(0, 1)$. A possible solution, used in [18], is to use the \texttt{intnum} and \texttt{sumnum} functions of PARI/GP [23] to numerically evaluate (2), (5), and (7). Here we show how to largely reduce the cost of this computation by introducing a new algorithm to obtain such quantities. Denoting as $[y]$ the least integer greater than or equal to $y \in \mathbb{R}$ and defining the $m$-th harmonic number as

$$H_m = \sum_{j=1}^{m} \frac{1}{j},$$  \hspace{1cm} (9)
where \( m \in \mathbb{N}, m \geq 1 \), our starting point is the following Theorem \([1]\) in which formula \((10)\) was first proved \([1]\) in Dilcher \([5]\). 

**Theorem 1.** Let \( x \in (0, 2) \). Then

\[
S(x) = -2\gamma_1(1 - x) + 2 \sum_{k=2}^{\infty} \frac{\zeta(k)H_{k-1} + \zeta'(k)}{k}(1 - x)^k, \tag{10}
\]

where \( \gamma_1 \) is defined as in \([3]\). \( H_k \) is defined as in \([9]\). \( \zeta(\cdot) \) is the Riemann zeta-function and \( \zeta'(\cdot) \) is its first derivative. Moreover, letting \( x \in (0, 1) \cup (1, 2) \), \( n \in \mathbb{N}, n \geq 1 \) be fixed, and \( r_S(x, n) \in \mathbb{N}, \)

\[
r_S(x, n) := \left\lfloor \frac{(n + 2) \log 2 + |\log(1 - |1 - x|)|}{\log |1 - x|}\right\rfloor - 1, \tag{11}\]

we have that there exists \( \theta = \theta(x) \in (-1/2, 1/2) \) such that

\[
S(x) = -2\gamma_1(1 - x) + 2 \sum_{k=2}^{r_S(x,n)} \frac{\zeta(k)H_{k-1} + \zeta'(k)}{k}(1 - x)^k + |\theta|2^{-n}. \tag{12}
\]

We immediately remark that \((10)\) is the Taylor series centred at 1 of \( S(x) \); in particular this implies that

\[
S'(1) = 2\gamma_1 \quad \text{and} \quad S^{(k)}(1) = 2(-1)^k(k - 1)!(\zeta(k)H_{k-1} + \zeta'(k)) \quad (k \in \mathbb{N}, k \geq 2).
\]

Using Theorem 1 and (1), for \( x \in (0, 2) \) we trivially have

\[
R(x) = -\zeta''(0) + 2(1 - x)\gamma_1 - 2 \sum_{k=2}^{\infty} \frac{\zeta(k)H_{k-1} + \zeta'(k)}{k}(1 - x)^k
\]

and its corresponding truncated version. We already remarked that \((10)\) is equation (2.14) of Dilcher \([5]\) but in Theorem 1 we will prove it in a different way, i.e., starting from \([5]\), which in fact reveals that such an argument can be used for any function having an integral representation of Plana’s type like the one in \([5]\). Formulae \((11)-(12)\) of Theorem 1 are new.

Recalling \([4]\), the fact that Theorem 1 holds for every \( x \in (0, 2) \) means that every value of \( S(x), x \in (0, 1) \), can be computed in two different ways \([2]\). Moreover it is clear that \( r_S(x, n) \) becomes larger as \( |1 - x| \) increases. Hence, remarking that in our applications of Section 4 we are mainly interested in \( x \in (0, 1) \), if \( x \in (1/2, 1) \) we will directly compute \( S(x) \) using Theorem 1 while for \( x \in (0, 1/2) \) we will shift the problem using \([4]\) and use Theorem 1 in \( (1, 3/2) \). In the following we will refer to this procedure as the shifting trick. Such an argument leads to the following two corollaries.

**Corollary 1.** Let \( x \in (0, 1/2) \). We have that

\[
S(x) = (\log x)^2 + 2\gamma_1 x + 2 \sum_{k=2}^{\infty} (-1)^k \frac{\zeta(k)H_{k-1} + \zeta'(k)}{k} x^k. \tag{13}
\]
Letting further \( n \in \mathbb{N}, n \geq 1 \) be fixed and \( r'_S(x, n) \in \mathbb{N}, \)
\[
 r'_S(x, n) := r_S(1 + x, n) = \left\lfloor \frac{(n + 2) \log 2 + \lfloor \log(1 - x) \rfloor}{\log x} \right\rfloor - 1,
\]
where \( r_S(u, n) \) is defined in Theorem 1, we have that exists \( \eta = \eta(x) \in (-1/2, 1/2) \) such that
\[
 S(x) = (\log x)^2 + 2\gamma_1 x + 2 \sum_{k=2}^{r'_S(x, n)} (-1)^k \frac{\zeta(k)H_{k-1} + \zeta'(k)}{k} x^k + |\eta|2^{-n}. \tag{14}
\]
Recalling Remark (2.6) of Deninger [8] page 176, (1) and (3) we also obtain
\[
 S\left(\frac{1}{2}\right) = -R\left(\frac{1}{2}\right) - \zeta''(0) = \frac{1}{2} (\log \pi)^2 + \frac{\pi^2}{24} - \frac{\gamma_1 + \gamma^2}{2}. \tag{15}
\]
\begin{remark}
 As a matter of curiosity, since we are aware of the fact that much faster algorithms exist to compute \( \gamma_1 \), see Johansson-Blagouchine [13], we remark that evaluating twice \( S(1/2) \) using Theorem 1 (the first time directly and the second as \( S(3/2) + (\log 2)^2 \)), we have, by subtracting such formulae, that the summands having even indices vanish; thus we obtain
\[
 \gamma_1 = -\frac{1}{2} (\log 2)^2 + \sum_{\ell=1}^{n/2+1} \frac{\zeta(2\ell + 1)H_{2\ell} + \zeta'(2\ell + 1)}{(2\ell + 1)4\ell} + \tau 2^{-n+1},
\]
for some \( \tau \in (-1/2, 1/2) \). Such a result, similar to equation (3.10) of Dilcher [5], allows us to fast compute \( \gamma_1 \) with a precision of \( n \) bits using about \( n/2 \) summands. For example, using PARI/GP, we got \( \gamma_1 \) with a precision of 1 000 decimal digits within 1 minute and 1 seconds of computation time on a Dell OptiPlex-3050 machine (equipped with an Intel i5-7500 processor, 3.40GHz, 16 GB of RAM and running Ubuntu 18.04.5).

Combining (12), (14) and (15), we obtain a very fast way of computing \( S(x) \) for every \( x > 0 \). We will see more about this in Section 5 but we also summarise the situation in the following
\begin{corollary}
 We use the notation introduced in Theorem 1 and Corollary 1. Moreover, 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:
\begin{enumerate}
  \item \( S(1) = S(2) = 0 \) and \( S(m) = -\sum_{k=2}^{m-1} (\log k)^2 \) for every \( m \in \mathbb{N}, m \geq 3; \)
  \item for \( x > 1, x \notin \mathbb{N} \), we compute \( S(x) \) as \( S(\{x\}) = S([x]) - \sum_{k=2}^{\lfloor x \rfloor - 1} (\log(\lfloor x \rfloor + k))^2; \)
  \item \( S(1/2) = (\log \pi)^2/2 + \pi^2/24 - (\gamma_1 + \gamma^2)/2; \)
  \item for \( x \in (0, 1/2) \), we compute \( S(x) \) as in (14);
  \item for \( x \in (1/2, 1) \), we compute \( S(x) \) as in (12).
\end{enumerate}
\end{corollary}

The proof of Corollary 2 follows just collecting the information coming from Theorem 1, Corollary 1, equations (15) and (3).

Even if in our application we will always work with \( x \in (0, 1) \), we recall that, for \( x \) large, it might be useful to implement the Stirling-like formula proved in Theorem 2.11 of Deninger [3] which gives an asymptotic expression for \( R(x) \) and, a fortiori, for \( S(x) \).

Our second theorem is about the function \( T \) defined in (7). As for \( S(x) \), the starting point is the following

Theorem 2. Let \( x \in (0, 2) \). Using the notation introduced in Theorem 1 we have
\[
T(x) = \sum_{k=2}^{\infty} (\zeta(k)H_{k-1} + \zeta'(k))(1-x)^{k-1}. 
\] (16)

Moreover, letting \( x \in (0, 1) \cup (1, 2) \), \( n \in \mathbb{N} \), \( n \geq 1 \) be fixed, and \( r_T(x, n) \in \mathbb{N} \),
\[
r_T(x, n) := \left[ \min_r \left\{ r \geq 1 + \frac{(n + 2) \log 2 - \log |1-x| + \log \log r}{\log |1-x|} \right\} \right], 
\] (17)
we have that there exists \( \theta = \theta(x) \in (-1, 1) \) such that
\[
T(x) = \sum_{k=2}^{r_T(x, n)} (\zeta(k)H_{k-1} + \zeta'(k))(1-x)^{k-1} + |\theta| 2^{-n}. 
\] (18)

We immediately remark that (16) is the Taylor series centred at 1 of \( T(x) \); in particular this implies that
\[
T^{(k)}(1) = (-1)^k k! (\zeta(k+1)H_k + \zeta'(k+1)) \quad (k \in \mathbb{N}, k \geq 1). 
\]

Using Theorem 2 and (7), for \( x \in (0, 2) \) we trivially have
\[
\psi_1(x) = -\gamma_1 + \sum_{k=2}^{+\infty} (\zeta(k)H_{k-1} + \zeta'(k))(1-x)^{k-1} 
\]
and the corresponding truncated version.

Formula (16) is essentially the one in Entry 21(ii) on page 280 of [1] and it follows by differentiation from (10) of Theorem 1. The series in Theorem 2 clearly has a worst convergence speed than the one in Theorem 1 and this justifies the different bound on \( r_T(x, n) \) we have in (17) comparing with the one for \( r_S(x, n) \) in (11). Recalling (8), the fact that Theorem 2 holds for every \( x \in (0, 2) \) means that every value of \( T(x) \), \( x \in (0, 1) \), can be computed in two different ways and that the shifting trick can be used in this case too. Hence, if \( x \in (1/2, 1) \) we will directly compute \( T(x) \) using Theorem 2 while for \( x \in (0, 1/2) \) we will use (8) and Theorem 2 in (1, 3/2). This way we obtain the following two corollaries.

**Corollary 3.** Let \( x \in (0, 1/2) \). We have that
\[
T(x) = -\frac{\log x}{x} + \sum_{k=2}^{+\infty} (1)^{k-1}(\zeta(k)H_{k-1} + \zeta'(k))x^{k-1}. 
\] (19)

Letting further \( n \in \mathbb{N} \), \( n \geq 1 \) be fixed and \( r_T'(x, n) \in \mathbb{N} \),
\[
r_T'(x, n) := r_T(1+x, n) = \left[ \min_r \left\{ r \geq 1 + \frac{(n + 2) \log 2 - \log |x| + \log \log r}{\log |x|} \right\} \right], 
\]
where \( r_T(u, n) \) is defined in Theorem 2 we have that there exists \( \eta = \eta(x) \in (-1, 1) \) such that
\[
T(x) = -\frac{\log x}{x} + \sum_{k=2}^{r_T'(x, n)} (1)^{k-1}(\zeta(k)H_{k-1} + \zeta'(k))x^{k-1} + |\eta| 2^{-n}. 
\] (20)

\[ \text{We remark that the size of the convergence interval can be doubled by isolating the Taylor series at 1 of 2(\log x)/x and using the estimates on } |\zeta(n) - 1| \text{ of Lemma 3 below. We do not insert such an idea here, since the computation of such an extra-factor 2(\log x)/x leads, in our practical application, to a longer total running time.} \]
Recalling equation (7.14) of Dilcher [4] we also obtain

\[ T\left(\frac{1}{2}\right) = \gamma_1 + \psi_1\left(\frac{1}{2}\right) = (\log 2)^2 + 2\gamma \log 2. \]  

(21)

**Remark 2.** In this case too we remark that evaluating twice \( T(1/2) \) using Theorem [2](the first time directly and the second as \( T(3/2) + 2\log 2 \)), we have, by summing such formulae, that the summands having even indices vanish; thus we obtain

\[ \gamma = -\frac{1}{2} \log 2 + \frac{1}{2} + \frac{1}{2 \log 2} \sum_{\ell=1}^{n/2+4} \frac{\zeta(2\ell + 1)H_{2\ell} + \zeta'(2\ell + 1)}{4\ell} + \tau 2^{-n+2}, \]

for some \( \tau \in (-1/2, 1/2) \). Such a result allows us to fast compute \( \gamma \) with a precision of \( n \) bits using about \( n/2 \) steps. For example, using PARI/GP, we got \( \gamma \) with a precision of 1 000 decimal digits within 1 minute and 10 seconds of computation time on the Dell Optiplex machine previously mentioned. In this case too there exist much faster algorithms to perform such a computation, see again [13].

Combining (18), (20) and (21), we obtain a very fast way of computing \( T(x) \) for every \( x > 0 \). We will see more about this in Section 5 but we also summarise the situation in the following

**Corollary 4.** We use the notation introduced in Theorem [2] and Corollaries [2,3] We have:

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

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

iii) \( T(1/2) = (\log 2)^2 + 2\gamma \log 2 \);

iv) if \( x \in (0, 1/2) \), we compute \( T(x) \) as in (20);

v) if \( x \in (1/2, 1) \), we compute \( T(x) \) as in (18).

The proof of Corollary 4 follows just collecting the information coming from Theorem [2], Corollary [3], equations (21) and (8).

We finally remark that the shifting trick applies to any function which can be defined as the solution of a difference equation, like \( S \) and \( T \), and that it can be expressed via a power series whose convergence interval is twice as large than the step of the difference equation. Another classical example of such a phenomenon\(^4\) is the pair of functions given by \( \log \Gamma \) and \( \psi = \Gamma'\Gamma \), for which the analogues of the formulæ (10) and (16) were first proved by Euler, see, e.g., Section 3 of the beautiful survey of Lagarias [15]. But this also holds for further generalisations of Euler’s Gamma function like the ones studied by Dilcher in [5]; in fact \( S \) and \( T \) are the first and easier cases of such generalisations.

Here we are mainly interested in \( S \) and \( T \) because of the number-theoretic applications concerning the logarithmic derivative at 1 of Dirichlet \( L \)-functions, see Section 4 There we will examine how to compute such quantities in a fast way and the number-theoretic consequences we can infer from such data.

The paper is organised as follows: in Sections [2,3] we will respectively prove Theorems [1,2] In Section 4 we will describe the problems in which the use of \( S(x) \) and \( T(x) \) is relevant. In Section 5 we will discuss the computational costs and some of the implementation features of the formulæ in Theorems [1,2] with respect to the applications too. Finally, Section 6 is dedicated to show some figures about the applications described in Section 4.

\(^4\) We used it in [19] to numerically study Littlewood’s bounds on \( |L(1, \chi)| \).
Acknowledgements. The calculations here described in Section 5 were performed using the University of Padova Strategic Research Infrastructure Grant 2017: “CAPRI: Calcolo ad Alte Prestazioni per la Ricerca e l’Innovazione”, http://capri.dei.unipd.it. The authors would also like to thank the anonymous referees for their remarks and suggestions.

2. Proof of Theorem 1

We start with the following lemmas that might have some independent interest too.

Lemma 1. Let \( x > 0 \) be fixed, \( T(x) \) be defined as in (7) and \( \gamma_1 \) as in (3). Moreover let \( \gamma \) be the Euler-Mascheroni constant. Then we have

\[
T(x) = \gamma_1 - \int_0^{+\infty} \left( e^{-t} - \frac{t e^{-xt}}{1 - e^{-t}} \right) \frac{\gamma + \log t}{t} \, dt \tag{22}
\]

and

\[
\int_0^{+\infty} \left( e^{-t} - \frac{t e^{-t}}{1 - e^{-t}} \right) \frac{\gamma + \log t}{t} \, dt = \gamma_1. \tag{23}
\]

Proof. Using (5) and (8), a differentiation immediately gives (22). The second part follows from the first using \( T(1) = 0 \).

Lemma 2. Let \( x > 0 \) be fixed, \( S(x) \) be defined as in (2), \( T(x) \) be defined as in (7) and \( \gamma_1 \) as in (3). Let moreover \( \gamma \) be the Euler-Mascheroni constant. Then we have

\[
T(x) = \int_0^{+\infty} \left( e^{(1-x)t} - 1 \right) \frac{\gamma + \log t}{e^t - 1} \, dt \tag{24}
\]

and

\[
S(x) = -2(1-x)\gamma_1 + 2\int_0^{+\infty} \left( e^{(1-x)t} - 1 - (1-x)t \right) \frac{\gamma + \log t}{t(e^t - 1)} \, dt. \tag{25}
\]

Proof. Inserting (23) into (22) and performing a trivial computation on absolutely convergent integrals gives (24). Moreover, an algebraic manipulation on equation (5) immediately give

\[
S(x) = 2\int_0^{+\infty} (x-1) \left( e^{-t} - \frac{t e^{-t}}{1 - e^{-t}} \right) \frac{\gamma + \log t}{t} \, dt \\
+ 2\int_0^{+\infty} \left( (x-1)te^{-t} + e^{-xt} - e^{-t} \right) \frac{\gamma + \log t}{t} \, dt
\]

which is allowed since both integrals absolutely converge. Recalling Lemma 1 we see that the first integral is equal to \( 2(x-1)\gamma_1 \). Another algebraic manipulation on the second integral proves (25). We also remark that (24) can also be obtained by differentiation from (25) and (8).

We will also need the following elementary estimates.

Lemma 3. Let \( \gamma \) be the Euler-Mascheroni constant, \( \psi(s) = \Gamma'(s)/\Gamma(s) \) be the digamma function and let \( x > 0 \). Then

\[
\log x - \frac{1}{x} < \psi(x) < \log x.
\]

Moreover, for every \( k \in \mathbb{N} \), \( k \geq 3 \), we have \( \psi(k) + \gamma = H_{k-1} \), \( 1 + 2^{-k} < \zeta(k) < 1 + 2^{1-k} \) and, for \( k \in \mathbb{N} \), \( k \geq 4 \), also that

\[
-\frac{\log 2 + (2/3) \log 3}{2k} < \zeta'(k) < -\frac{\log 2}{2k}.
\]
We now prove the second part of Theorem 1. From now on we denote $E_k$ and $E_r$. Moreover, letting
$$\xi = \frac{1}{s+1},$$
for every $s$ the proof of Theorem 1 now starts from (25) of Lemma 2. Let $\gamma_0$ be a positive number. Hence (25) becomes
$$\xi(k) = \frac{\log 2}{2k} + \sum_{n=3}^{+\infty} \frac{\log n}{n^k} \leq \frac{\log 2}{2k} + \frac{\log 3}{3} \sum_{n=3}^{+\infty} \frac{1}{n^{k-1}} = \frac{\log 2}{2k} + \frac{\log 3}{3} (\zeta(k) - 1 - \frac{1}{2k-1})$$
and using the inequality $\zeta(m) < 1 + 2^{1-m}$, $m \in \mathbb{N}$, $m \geq 3$, previously proved.

The proof of Theorem 1 now starts from (25) of Lemma 2. Let $x \in (0,2)$ and, for every $t \in \mathbb{R}$, define $f(x,t) := e^{(1-x)t} - 1 - (1-x)t$. Hence (25) becomes
$$S(x) = -2(1-x)\gamma_1 + 2 \int_0^{+\infty} f(x,t) \frac{\gamma + \log t}{t(e^t - 1)} \, dt.$$ 
Writing the Taylor expansion at 0 of $f(x, \cdot)$, we can easily get that $f(x,t) = \sum_{k=2}^{+\infty} t^k (1-x)^k / k!$ which holds for every $t \in \mathbb{R}$ and $x \in (0,2)$. Hence
$$\int_0^{+\infty} f(x,t) \frac{\gamma + \log t}{t(e^t - 1)} \, dt = \sum_{k=2}^{+\infty} \frac{(1-x)^k}{k!} \int_0^{+\infty} \frac{t^{k-1}(\gamma + \log t)}{e^t - 1} \, dt$$
in which we exchanged the series and the integral signs by exploiting their absolute convergence. Let now $s \in \mathbb{C}$, $\Re(s) > 1$. Recalling the classical formula
$$\int_0^{+\infty} \frac{t^s-1}{e^t-1} \, dt = \Gamma(s)\zeta(s),$$
differentiating over $s$ we immediately get
$$\int_0^{+\infty} \frac{t^{s-1} \log t}{e^t-1} \, dt = \Gamma'(s)\zeta(s) + \Gamma(s)\zeta'(s).$$
Recalling that $\psi(s) = \Gamma'/\Gamma(s)$, Lemma 3 and $\Gamma(k) = (k-1)!$, by inserting (28)–(29) into (27) we obtain
$$\int_0^{+\infty} f(x,t) \frac{\gamma + \log t}{t(e^t - 1)} \, dt = \sum_{k=2}^{+\infty} \frac{\zeta(k)H_{k-1} + \zeta'(k)}{k} (1-x)^k.$$ 
Hence (20) immediately follows by inserting (30) into (26). This proves the first part of Theorem 1. We now prove the second part of Theorem 1. From now on we denote
$$\mathcal{L}(k) := \zeta(k)H_{k-1} + \zeta'(k).$$
Moreover, letting $r \in \mathbb{N}$, $r \geq 3$, we define
$$\Sigma_5(r,x) := \sum_{k=2}^{r} \frac{\mathcal{L}(k)}{k} (1-x)^k$$
and $E_5(r,x) := \sum_{k=r+1}^{+\infty} (\mathcal{L}(k)/k) (1-x)^k$. Hence from (10) we get
$$S(x) = -2(1-x)\gamma_1 + 2\Sigma_5(r,x) + 2E_5(r,x).$$
Let now \(n \geq 1\) be fixed. For every fixed \(x \in (0, 2), x \neq 1\), we will find \(r = r_S(x, n) \in \mathbb{N}\) such that \(|E_S(r, x)| < 0.25 \cdot 2^{-n}\). Using Lemma 3, we obtain, for \(k \geq r + 1 \geq 4\), that

\[
\frac{1}{k} \sum_{k=2}^{+\infty} \frac{\zeta(k)H_{k-1} + |\zeta'(k)|}{\zeta(k)} < \frac{(1 + 2^{-r})(\log(r + 1) + \gamma) + 0.72 \cdot 2^{-r}}{r + 1} < 0.76
\]

and hence, using the well-known formula about the sum of a geometric progression, we can write

\[
|E_S(r, x)| < 0.76 \sum_{k=r+1}^{+\infty} |1 - x|^k = 0.76 \frac{|1 - x|^{r+1}}{1 - |1 - x|}. \tag{33}
\]

We now look for \(r = r_S(x, n) \in \mathbb{N}, r \geq 3\), such that \(\frac{|1 - x|^{r+1}}{1 - |1 - x|} \leq 2^{-n-2}\). An easy computation reveals that

\[
r + 1 \geq \frac{(n + 2) \log 2 + |\log(1 - |1 - x|)|}{|\log |1 - x||} \tag{34}
\]

suffices. The second part of Theorem 1 then follows from (32)-(34).

3. Proof of Theorem 2

We already remarked that (16) follows via (8) from (10) but a direct proof can also be obtained starting from (24) and arguing as in the proof of Theorem 1. We now prove the second part of Theorem 2. Letting \(r \in \mathbb{N}, r \geq 3\), and recalling (31), we define

\[
\Sigma_T(r, x) := \sum_{k=2}^{r} \mathcal{L}(k)(1 - x)^{k-1}
\]

and \(E_T(r, x) := \sum_{k=r+1}^{+\infty} \mathcal{L}(k)(1 - x)^{k-1}\). Hence from (16) we get

\[
T(x) = \Sigma_T(r, x) + E_T(r, x). \tag{35}
\]

Let now \(n \geq 1\) be fixed. For every fixed \(x \in (0, 2), x \neq 1\), we will find \(r = r_T(x, n) \in \mathbb{N}\) such that \(|E_T(r, x)| < 2^{-n}\). Using Lemma 3 we have, for \(k \geq r + 1 \geq 4\), that

\[
|E_T(r, x)| < \frac{2}{|1 - x|} \sum_{k=r+1}^{+\infty} |1 - x|^k \log k. \tag{36}
\]

Assuming that \(r \geq 1/x\), we have that \(|1 - x|^k \log k\) is a decreasing sequence for \(k \geq r + 1\); hence a partial integration argument gives that

\[
|E_T(r, x)| < \frac{2}{|1 - x|} \int_{r}^{+\infty} |1 - x|^u \log u \, du < 2 \frac{|1 - x|^{r-1}}{|\log |1 - x||} \left(\log r + \frac{1}{r |\log |1 - x||}\right) \tag{37}
\]

in which we also assumed that \(r |\log |1 - x|| \geq 1\). We now look for \(r = r_T(x, n) \in \mathbb{N}\) such that \(\frac{|1 - x|^{r-1}}{|\log |1 - x||} \log r \leq 2^{-n-2}\). An easy computation reveals that

\[
r - 1 \geq \frac{(n + 2) \log 2 - \log |\log |1 - x|| + \log \log r}{|\log |1 - x||} \tag{38}
\]

suffices. The second part of Theorem 2 then follows from (35)-(38).

4. Applications

We briefly describe here some number-theoretic applications in which the use of \(S\) and \(T\) is relevant; we will heavily refer to [18] in which a more detailed presentation is given.
4.1. Computation of $L'/L(1, \chi)$. The main application in which it is important to know the values of $S(a/q)$, where $q$ is an odd prime and $a = 1, \ldots, q - 1$, is to evaluate the logarithmic derivative at 1 of the Dirichlet $L$-functions. Following the argument of Section 3 of [18] we have, for $\chi$ primitive and odd, that

$$
\frac{L'}{L}(1, \chi) = \gamma + \log(2\pi) + \frac{1}{B_{1,\chi}} \sum_{a=1}^{q-1} \overline{\chi}(a) \log \left( \Gamma \left( \frac{a}{q} \right) \right)
$$

(39)

and, for $\chi$ primitive and even, $\chi \neq \chi_0$, that

$$
\frac{L'}{L}(1, \chi) = -\log q - \frac{\sum_{a=1}^{q-1} \chi(a) T(a/q)}{\sum_{a=1}^{q-1} \chi(a) \psi(a/q)}.
$$

(40)

where $B_{1,\chi} := (\sum_{a=1}^{q-1} a \chi(a))/q$ is the first $\chi$-Bernoulli number, $L(s, \chi)$ denote the Dirichlet $L$-functions, $\chi$ run over the non-principal Dirichlet characters mod $q$ and $\chi_0$ is the principal Dirichlet character mod $q$. Using the values of $T(a/q)$ we can alternatively write, for every $\chi \neq \chi_0$, that

$$
\frac{L'}{L}(1, \chi) = -\log q - \frac{\sum_{a=1}^{q-1} \chi(a) T(a/q)}{\sum_{a=1}^{q-1} \chi(a) \psi(a/q)}.
$$

(41)

Ford, Luca and Moree [6] were the first to use (41) and the Fast Fourier Transform (FFT) method to compute $\mathcal{O}_q$, see §4.3 for its definition. To see how the summations over $a$ can be efficiently performed using the FFT method, see, e.g., Section 4 of [18], and for this approach the use of (39)-(40) leads to a faster algorithm than the one which uses (41) because in the former case a decimation in frequency strategy can be applied. This essentially means that just the $(q - 1)/2$ values of $S(a/q) + S(1 - a/q)$ are needed to perform the summation over $a$ in (40); combining this with the use of suitable reflection formulae for $S(x)$ lead to gain a factor 4 in the computational cost of generating the $S$-values with respect to the cost of generating the $T$-values, see also subsection 5.3.

It is a well-known fact, see, e.g., Corollary 10.18 of Montgomery-Vaughan [22], that the logarithmic derivative at 1 of the Dirichlet $L$-functions is connected with the distribution of non-trivial zeros of $L(s, \chi)$. In literature, see, e.g. Ihara-Murty-Shimura [12], it is interesting to study the extremal values of the logarithmic derivative at 1 of the Dirichlet $L$-functions under the assumption of the Generalised Riemann Hypothesis.

4.2. Extremal values of $L'/L(1, \chi)$. For every odd prime $q$ we define

$$
M_q^{\text{odd}} := \max_{\chi \text{ odd}} \left| \frac{L'}{L}(1, \chi) \right|, \quad M_q^{\text{even}} := \max_{\chi \text{ even}} \left| \frac{L'}{L}(1, \chi) \right|, \quad M_q = \max_{\chi \neq \chi_0} \left| \frac{L'}{L}(1, \chi) \right|.
$$

Hence we can compute $M_q = \max(M_q^{\text{odd}}, M_q^{\text{even}})$ using (39)-(40). Numerical values for $M_q$ were obtained in [18] for every odd prime $q \leq 10^6$. Such data are in agreement with the estimate proved by Ihara-Murty-Shimura [12] (please remark that our $M_q$ is denoted as $Q_m$ there) since they proved that $M_q \leq (2 + o(1)) \log \log q$ as $q$ tends to infinity, under the assumption of the Generalised Riemann Hypothesis. On the other hand, Lamzouri, in a personal communication with the first author, remarked that, by adapting the techniques in his paper [16], one can show that $M_q \geq (1 + o(1)) \log \log q$ for every sufficiently large prime $q$.

We will extend here the study of $M_q$ to the larger interval $q \leq 10^7$; we can do so because of the much faster algorithm to compute $S(a/q)$ presented here. A similar study for $m_q :=$
when worst cases for

Another interesting quantity related to we immediately get that the 

values of binary digits, with at most 

have a 

on 

Number of summands.

\[ \log_{q} \]

for 

\( \text{𝔊} \)

to the large bound Ford-Luca-Moree [6] (other two occurrences of that both these quantities should be positive. Such a conjecture for over by many others; we just recall here the papers by Ford-Luca-Moree [6] and Languasco [18]

in this cyclotomic case we have that the Dedekind zeta-function can be written as 

\[ \tilde{\zeta}(\chi) = \prod_{\chi \neq \chi_0} L(s, \chi), \]

where \( \zeta(s) \) is the Riemann zeta-function. By logarithmic differentiation, we immediately get that the Euler-Kronecker constant for the prime cyclotomic field \( \text{ℚ}(\zeta_q) \) is

\[ \tilde{\zeta}_q := \gamma + \sum_{\chi \neq \chi_0} \frac{L'}{L}(1, \chi). \]

An extensive study about the properties of \( \tilde{\zeta}_q \) and \( \tilde{\zeta}_q^{+} \) was started by Ihara [10, 11] and carried over by many others; we just recall here the papers by Ford-Luca-Moree [6] and Languasco [18] because they both have some computational results on \( \tilde{\zeta}_q \) and \( \tilde{\zeta}_q^{+} \).

For both \( \tilde{\zeta}_q \) and \( \tilde{\zeta}_q^{+} \) it is interesting to find their negative values since Ihara conjectured that both these quantities should be positive. Such a conjecture for \( \tilde{\zeta}_q \) was disproved by Ford-Luca-Moree [6] (other two occurrences of \( \tilde{\zeta}_q < 0 \) were detected in [18]). No negative values of \( \tilde{\zeta}_q^{+} \) are known so far. We will extend here the search for negative values of \( \tilde{\zeta}_q \) and \( \tilde{\zeta}_q^{+} \) to the large bound \( q \leq 10^7 \); in this way we also prove that there are no negative values for both \( \tilde{\zeta}_q \) and \( \tilde{\zeta}_q^{+} \) for every odd prime \( q \) up to \( 10^7 \). We also evaluate such quantities for some very large \( q \). We can do so because of the much faster algorithm to compute \( S(a/q) \) presented here.

5. Implementation

We discuss here some implementation features of the formulae in Corollaries 2 and 4. Since for \( x > 1 \) we can reduce the problem of evaluating \( S(x) \), or \( T(x) \), to a sum of a finite number of log-values plus \( S(\{x\}) \), or \( T(\{x\}) \), in this Section we assume that \( x \in (0, 1) \).

5.1. Number of summands. We already remarked in the Introduction that, from the estimates on \( r_S(x, n) \) and \( r_T(x, n) \) in Theorems 12, the number of needed terms we have to consider to have a \( n \)-bits digit precision result becomes arbitrarily large as \( x \rightarrow 0^+ \). To avoid this problem we can in practice use the formulae in Corollaries 2 and 4. In both corollaries it is clear that the worst cases for \( r_S(x, n) \) and \( r_S'(x, n) \) (and, respectively, for \( r_T(x, n) \) and \( r_T'(x, n) \)) are obtained when \( x \) approaches \( 1/2 \). Hence we can get any value of \( S(x) \), \( x \in (0, 1) \), with a precision of \( n \) binary digits, with at most \( n + 2 \) summands (assuming that the needed log and \( Z(k) \) values can
be obtained with the same precision). Analogously we can get any value of \( T(x) \), \( x \in (0, 1) \), with a precision of \( n \) binary digits, with at most \( n + 4 + \log \log(n + 4) \) summands (assuming that the needed \( \log \) and \( \mathcal{L}(k) \) values can be obtained with the same precision).

### 5.2. Precomputed coefficients.

In (12) and (18) we have a power series whose coefficients involve the values \( \mathcal{L}(k) \), \( k \in \mathbb{N} \), \( k \geq 2 \) (see (31) for the definition of \( \mathcal{L}(k) \)). Hence in both cases such values can be precomputed, stored and reused for any \( x \in (0, 1) \). Moreover, the estimates in Lemma 3 imply that \( |\zeta(k) - 1| < 10^{-200} \) for \( k \geq 160 \) and \( |\zeta''(k)| < 10^{-200} \) for \( k \geq 420 \). Hence, after about 420 terms just the contribution of \( H_{k-1} \) matters in (12) and (18).

So, after few hundreds terms, the problem of obtaining \( \mathcal{L}(k) \) reduces to being able to evaluate \( H_{k-1} \). We also remark that the computation of the needed first hundreds values of \( \zeta(k) \) and \( \zeta''(k) \) can be performed, for instance, using PARI/GP.

Another nice aspect we have in (12) and (18) is that the powers \((1 - x)^k\) can be computed by recurrence, starting from \((1 - x)^2\) and \(1 - x\), respectively. The same clearly holds for equations (14) and (20) too.

All these remarks also reveal that the tasks of evaluating \( S(x) \) and \( T(x) \) are essentially as difficult as evaluating \((\log x)^2\) and \((\log x)/x\), when \( x \) is close to 0.

### 5.3. Reflection formulae for \( S(x) \).

As mentioned in Section 4.1 and extensively explained in Section 4 of [18], the use of the FFT algorithm is important to efficiently compute \( \delta_q \), \( \delta_q^* \) and \( M_q \). In particular, using \( S(x) \), a decimation in frequency strategy can be implemented and hence it is important to have the following reflection formulae for \( S \).

We directly express such formulae using Theorem 1 and Corollary 1 or Corollary 2, even if similar ones which use (2) and (5) are also available (such formulae were in fact used in [18], see Section 4.2 there).

**Proposition 1.** Let \( x \in (0, 1) \), \( x \neq 1/2 \), \( n \in \mathbb{N} \), \( n \geq 2 \), \( r_1(x, n) = \lceil \frac{(n+2) \log 2 + \log(1+x)}{\log x} \rceil - 1 \)/2 and \( r_2(x, n) = \lfloor \frac{(n+2) \log 2 + \log(1+x)}{\log x} \rfloor - 1 \)/2. Using (31) and the notations of Theorem 1 and Corollary 2, we have that there exists \( \theta = \theta(x) \in (-1/2, 1/2) \) such that

\[
S(x) + S(1-x) = (\log x)^2 + 2 \sum_{\ell=1}^{r_1} \frac{\mathcal{L}(2\ell)}{\ell} x^{2\ell} + |\theta| 2^{-n}, \quad (0 < x < \frac{1}{2}),
\]

and

\[
S(x) + S(1-x) = (\log(1-x))^2 + 2 \sum_{\ell=1}^{r_2} \frac{\mathcal{L}(2\ell)}{\ell} (1-x)^{2\ell} + |\theta| 2^{-n}, \quad \left( \frac{1}{2} < x < 1 \right).
\]

**Proof.** Assume that \( 0 < x < 1/2 \); in this case we compute \( S(x) \) with (13) and \( S(1-x) \) with (10). Since the series absolutely converge, their sum is the series having as summands the sum of their coefficients. Arguing as in (33), remarking that \( r_1(x, n) = r_S(1-x, n)/2 = r'_S(x, n)/2 \) and recalling (31), we immediately have that (42) holds since the odd summands vanish. Assume that \( 1/2 < x < 1 \); in this case we compute \( S(x) \) with (10) and \( S(1-x) \) with (13). Arguing as for \( x \in (0, 1/2) \), remarking that \( r_2(x, n) = r_S(x, n)/2 = r'_S(1-x, n)/2 \) and recalling (31), we immediately have that (43) holds since the odd summands vanish. This completes the proof. \( \square \)

The corresponding series for (42)-(43) are

\[
S(x) + S(1-x) = (\log x)^2 + 2 \sum_{\ell=1}^{r_0} \frac{\mathcal{L}(2\ell)}{\ell} x^{2\ell}, \quad (0 < x < \frac{1}{2}),
\]
we have

As we said before, the use of Proposition 1, if possible, is particularly efficient. To compare
4 require to evaluate (12) over (18) in the applications but in this case we cannot use the decimation in frequency strategy, see
step by a factor of at least
evaluate such applications we can use a decimation in frequency strategy, Proposition 1 let us directly
floating point products each, this proves that the total computational cost of our applications is
are three Fourier Transforms of length
rational points less than
products, with a precision of
the total cost of evaluating (12) over (18) and (19) since we just need to use half of the
summands (the ones with even indices);
iii) in (42)-(43) the values of the Riemann \( \zeta \)-function at even integers and for
them we can use the well-known exact formulae 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 using the following series expansion:
\[
\frac{1}{e^t - 1} = \sum_{k=0}^{\infty} B_k \frac{t^k}{k!}, \quad |t| < 2\pi,
\]
see, e.g., Cohen’s book [2, chapter 9].

As we said before, the use of Proposition 1, if possible, is particularly efficient. To compare
the practical running times of using (42)-(43) with previous implementations, which used the
series/integral definitions of \( S(x) \), see (2) and (3), we compared the two PARI/GP scripts used
to obtain \( S(a/q) + S(1 - a/q) \) for every \( a = 1, \ldots, (q - 1)/2 \) when \( q = 305741, 6766811, 212634221 \). The gain in speed is huge, and it seems to improve as \( q \) becomes larger: we observed
that the use of (42)-(43) leads to a computation time for \( S(a/q) + S(1 - a/q) \) (with a precision
of 128 bits) for every \( a = 1, \ldots, (q - 1)/2 \) which is respectively about 405, 829, 1206 times
faster for the three primes mentioned before.

Further practical experiments confirmed such a computational time gain; we will see more on
this in the next subsections.

5.4. Computational costs for the problems of Section 4. The applications described in Section
4 require to evaluate (12) over \( x = a/q, a = 1, \ldots, q - 1 \). Using the estimates in subsection 5.1
we have \( r_S(x, n), r'_S(x, n) \leq n + 2 \) for every \( x \in (1/2, 1) \) and, respectively, \( x \in (0, 1/2) \). Hence
the total cost of evaluating (12) over \( x = a/q, \) for every \( a = 1, \ldots, q - 1 \), is \( \Theta (qn) \) floating point
products, with a precision of \( n \) binary digits, and \( \Theta (q) \) evaluation of the logarithm function at
rational points less than 1/2. Since the remaining part of the computations in our applications
are three Fourier Transforms of length \( \leq q - 1 \), see [18, Table 1], having a cost of \( \Theta (q \log q) \)
floating point products each, this proves that the total computational cost of our applications is
\( \Theta (q(n + \log q)) \) floating point products, with a precision of \( n \) binary digits. In practice, since
in such applications we can use a decimation in frequency strategy, Proposition 1 let us directly
evaluate \( S(a/q) + S(1 - a/q) \) for every \( a = 1, \ldots, (q - 1)/2 \) thus reducing of the cost of such a
step by a factor of at least 4. A similar asymptotic estimate \( \Theta (q(n + \log q)) \) holds also using
(18) in the applications but in this case we cannot use the decimation in frequency strategy, see

\[
S(x) + S(1 - x) = (\log(1 - x))^2 + 2 \sum_{\ell=1}^{\infty} \frac{\mathcal{L}(2\ell)}{\ell} (1 - x)^{2\ell} \quad \left( \frac{1}{2} < x < 1 \right).
\]

(45)
again [18]; hence in practice such an algorithm has a total cost which is about four times larger than the one which uses the S-function. Anyway, we will need such a T-function implementation for being able to double check the results.

5.5. Actual implementation of the S and T formulae. Using the C programming language, we implemented the formulae of Proposition [1] since they are the ones needed for the applications of Section 4. The summation is performed combining the “pairwise summation” [9] algorithm with Kahan’s [14] 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. To write here a practical computation time, we remark that for 

\[ q = 50,040,955,631 \]

such an implementation computed \( S(a/q) + S(1 - a/q) \) for every \( a = 1, \ldots, (q - 1)/2 \) with a precision of 128 bits in about nine hours using a single computing core of an HP machine equipped with 4 x Eight-Core Intel(R) Xeon(R) CPU E5-4640 0 @ 2.40GHz, and 256GB of RAM. For comparison, in this case the expected running time of the implementation used in [18] would be about 3475 days on the same machine mentioned before (about 9350 times slower).

Clearly this huge improvement let us evaluate the quantities described in Section 4 for some really large prime numbers and also to extend their knowledge for every odd prime up to \( 10^7 \).

Moreover, to be able to double check the results obtained with the S-function, we analogously implemented the formulae of Corollary [4].

5.6. FFT implementation and computational results. To implement the FFT method we used the FFTW [7] package which is also able to handle very large cases via its gnu64 interface. Moreover, to be able to store the large arrays of data we produce to initialise the input sequences involved in the FFT and their outputs, we used the mmap UNIX system call to map such arrays on the hard disk instead of storing them on the RAM during the execution of the C-programs. Thus we were able to enlarge the range of possible computations we can perform far beyond the size of the available RAM memory. But that was not enough to handle the large case we would have liked to evaluate: \( q = 50,040,955,631 = 2 \cdot 5 \cdot 50040955631 + 1 \). We have chosen this prime number because its evaluation using the function \( v(q) \), defined in the next paragraph, see (51), is “large” enough to let us think that it might be a good candidate to have \( \overline{6}_q < 0 \left( v(50040955631) = 1.2194 \ldots \right) \); please see Section 6 for more about \( v(q) \) and its link with the negativity of \( \overline{6}_q \). Moreover, by analysing the prime factor structure of the known examples for which \( \overline{6}_q \) is negative, namely \( q = 964477901, 910934831, 9854964401, \) see [6] and [18], we see that such primes \( q \) have all a “large” prime in the factorisation of \( q - 1 \): 

\[
\begin{align*}
964477901 &= 2 \cdot 5 \cdot 9 \cdot 6447799 + 1, \\
910934831 &= 2 \cdot 5 \cdot 91093483 + 1, \\
9854964401 &= 2^2 \cdot 5^2 \cdot 197 \cdot 125063 + 1.
\end{align*}
\]

These two motivations are hence a strong suggestion about the negativity of \( \overline{6}_{50040955631} \), even if they are not sufficient to be certain of this.

The presence of a “large” prime factor in the factorisation of \( q - 1 \) leads in fact to another problem in using the so-called plan-generation step of the FFTW package. The plan-generation step of FFTW is a procedure in which FFTW self-decides how to combine several FFT algorithms to obtain their best combination to solve the particular instance of the problem the user is interested in. This procedure also depends on the prime factorisation of the length of the transform \( N \): in our case \( N = q - 1 \), or \( N = (q - 1)/2 \). When \( N \) has at least one “large” prime factor, as in our case, the plan-generation step might be very demanding in term of memory usage (RAM). To overcome this, we have then to insert the use of mmap in the body of the FFTW code to be able to divert the memory usage of the plan-generation step from the RAM to the hard disk. Clearly this increases the actual computation time but, at the same time, let us handle much larger cases,
since, essentially, it is much easier, and cheaper, to retrieve large hard disks than a large quantity of RAM.

In this way we obtained a program that needed at most 128GB of RAM at runtime and we used it to perform the computation for the case of \( q = 50040955631 \) on the University of Padova Strategic Research Infrastructure “CAPRI” (Intel(R) Xeon(R) Gold 6130 CPU @ 2.10GHz, with 256 cores and equipped with 6TB of RAM). The total hard disk usage was about 8 TB, the time needed for one computing core to generate the \( S \)-values with a precision of 128 bits was about six hours and 6 minutes, the plan-generation step required about four hours and 5 minutes and the actual FFT transforms about 2 days and half (for \( S \)). The total computation time was about two weeks; we recall that such computation times are affected, as above remarked, from the fact that we were using a slower memory device (the hard disk is used instead of RAM). We got that \( \xi_{500409555631} = -0.16595399 \ldots \) and \( \xi^+_{500409555631} = 13.89764738 \ldots \) thus getting another occurrence of a negative Euler-Kronecker constant.

The computation of \( \xi_q, \xi^+_q \) and \( M_q \) for every odd prime \( q \) up to \( 10^7 \) was performed on CAPRI using at most 60 computing nodes and it required about 48 hours of time (the global execution time, obtained by summing the declared computing time on each node, was of 101 days and 6 hours).

In this range we obtained that there are no negative values for both \( \xi_q \) and \( \xi^+_q \) and that
\[
\frac{17}{20} \log \log q < M_q < \frac{5}{4} \log \log q
\]
for every prime \( 1531 < q \leq 10^7; M_{1531} = 2.5048094 \ldots, M_{1531}/\log \log(1531) = 1.257133 \ldots \) Moreover the lower bound in (46) holds true for \( q > 13 \).

The programs used and the results here described are collected at the following address http://www.math.unipd.it/~languasc/Scomp-appl.html.

5.7. FFT accuracy estimate. According to Schatzman [24 § 3.4, p. 1159-1160], the root mean square relative error in the FFT is bounded by
\[
\Delta = \Delta(N, \varepsilon) := 0.6\varepsilon(\log_2 N)^{1/2},
\]
where \( \varepsilon \) is the machine epsilon and \( N \) is the length of the transform. According to the IEEE 754-2008 specification, we can set \( \varepsilon = 2^{-64} \) for the long double precision of the C programming language. So for the largest case we are considering, \( q = 50040955631, N = (q - 1)/2 \), we get that \( \Delta < 1.92 \cdot 10^{-19} \). To evaluate the euclidean norm of the error we have then to multiply \( \Delta \) and the euclidean norms of the sequences (listed in sections 4.2-4.3 of [18]):
\[
x_k := 2\frac{a_k}{q} - 1, \quad y_k := \log \Gamma\left(\frac{a_k}{q}\right) + \log \Gamma\left(1 - \frac{a_k}{q}\right) - \log \pi,
\]
\[
z_k := \log \Gamma\left(\frac{a_k}{q}\right) - \log \Gamma\left(1 - \frac{a_k}{q}\right), \quad w_k := S\left(\frac{a_k}{q}\right) - S\left(1 - \frac{a_k}{q}\right),
\]
where \( a_k = g^k \mod q, \langle q \rangle = \mathbb{Z}_q^*, k = 0, \ldots, N - 1 \). A straightforward computation gives
\[
\|x_k\|_2 = \frac{(q - 1)(q - 2)}{6q} = 91324.47246 \ldots
\]
Hence, recalling that \( \| \cdot \|_\infty \leq \| \cdot \|_2 \), for this sequence we can estimate that the maximal error in its FFT-computation is bounded by \( 1.75 \cdot 10^{-14} \) (long double precision case). Unfortunately, no closed formulas for the euclidean norms of the other involved sequences are known but, using \( \| \cdot \|_\infty \leq \| \cdot \|_2 \leq \sqrt{N} \cdot \| \cdot \|_\infty \) and the formulae
\[
\|y_k\|_\infty = - \log \sin(\pi/q) = 23.49137 \ldots,
\]
\[ \|z_k\|_\infty = 2 \log \Gamma \left( \frac{1}{q} \right) - \log \left( \frac{\pi}{\sin(\pi/q)} \right) = 24.63610 \ldots, \]
\[ \|w_k\|_\infty = S \left( \frac{1}{q} \right) + S \left( 1 - \frac{1}{q} \right) = 606.93779 \ldots, \]
that can be obtained using straightforward computations, we have that the errors in their FFT-computations are all \( < 1.85 \cdot 10^{-11} \).

We also estimated in practice the accuracy in the actual computations using the FFTW software library by evaluating at run-time the quantity \( \mathcal{E}_j(w_k) := \| \mathcal{F}^{-1}(\mathcal{F}(w_k)) - w_k \|_j, \) \( j \in \{2, \infty\} \), \( \mathcal{F}(\cdot) \) is the Fast Fourier Transform and \( \mathcal{F}^{-1}(\cdot) \) is its inverse transform. We focused our attention on \( w_k \) since, between the sequences mentioned before, it has the largest norms and hence the worst error estimates. Theoretically we have that \( \mathcal{E}_j(w_k) = 0 \); moreover, assuming that the root mean square relative error in the FFT is bounded by \( \Delta > 0 \), it is easy to obtain

\[ \mathcal{E}_2(w_k) < \Delta(2 + \Delta)\|w_k\|_2 \quad \text{and} \quad \mathcal{E}_\infty(w_k) < \Delta(2 + \Delta)\sqrt{N}\|w_k\|_\infty. \quad (48) \]

For \( q = 50040955631, N = (q - 1)/2 \) and \( \varepsilon = 2^{-64} \) in (47), we get \( \Delta(2 + \Delta) < 3.83 \cdot 10^{-19} \) and from (48) we obtain

\[ \mathcal{E}_2(w_k) < 3.70 \cdot 10^{-11} \quad \text{and} \quad \mathcal{E}_\infty(w_k) < 3.70 \cdot 10^{-11}, \quad (49) \]

where the first estimate suffers from the lack of theoretical information about \( \|w_k\|_2 \). Moreover, the actual computations using FFTW for this case gave that \( \|w_k\|_2 = 1099611.166707 \ldots, \)

\[ \frac{\mathcal{E}_2(w_k)}{\|w_k\|_2} < 6.01 \cdot 10^{-19}, \quad \mathcal{E}_2(w_k) < 4.21 \cdot 10^{-13} \quad \text{and} \quad \mathcal{E}_\infty(w_k) < 2.23 \cdot 10^{-16} \quad (50) \]

that are in agreement with (49). It is worth notice that the last two computed estimates in (50) are much better than the corresponding theoretical ones in (49). We finally remark that the computed estimates for the analogous quantities involving \( x_k, y_k, z_k \) are smaller than the ones for \( w_k \) described before.

Summarising, we can conclude that at least ten decimal digits of our final results are correct. If necessary, more accurate results can be obtained using the quadruple precision (128 bits), which let us choose \( \varepsilon = 2^{-113} \in (47) \), at the cost of a much slower execution.

We finally recall that the well-known weakness of the FFT-algorithms is the memory occupation and, as another example of this, we remark that for \( q = 50040955631 \) about 12 TB of hard disk space are required (see also §5.6) to perform the needed FFTW-library computations (long double precision case) used to obtain the data described in this paragraph.

### 6. Figures

We give here some comments about Figures 2-6 and discuss the role of the \( v(q) \)-function.

Referring to Section 4.5 of [18], we recall the definition of \( \mathcal{B} \), the “greedy sequence of prime offsets”, [http://oeis.org/A135311]. We define \( \mathcal{B} \) using induction, by \( b(1) = 0 \in \mathcal{B} \) and \( b(n) \in \mathcal{B} \) if it is the smallest integer exceeding \( b(n - 1) \) such that for every prime \( r \) the set \( \{ b(i) \text{ mod } r : 1 \leq i \leq n \} \) has at most \( r - 1 \) elements. Let now

\[ m(\mathcal{A}) := \sum_{i=1}^{s} \frac{1}{a_i}, \]

where \( \mathcal{A} \) is an admissible set, i.e., \( \mathcal{A} = \{ a_1, \ldots, a_s \}, a_i \in \mathbb{N}, a_i \geq 1, \) such that does not exist a prime \( p \) such that \( p \mid n \prod_{i=1}^{s} (a_i n + 1) \) for every \( n \geq 1. \) Thanks to Theorem 2 of Moree [21], if the prime \( k \)-tuples conjecture holds and if \( \mathcal{A} \) is an admissible set, then \( G_q < (2 - m(\mathcal{A}) + o(1)) \log q \)
for \( x/(\log x)^{|\mathcal{A}| - 1} \) primes \( q \leq x \). Moreover, by Theorem 6 of Moree \([21]\), assuming both the Elliott-Halberstam and the prime \( k \)-tuples conjectures, if \( \mathcal{A} \) is an admissible set then 

\[
\mathcal{A}_q = (1 - m(\mathcal{A}) + o(1)) \log q \quad \text{for} \quad x/(\log x)^{|\mathcal{A}| - 1} \text{ primes } q \leq x.
\]

We recall that the greedy sequence of prime offsets \( \mathcal{B} \) has the property that any finite subsequence is an admissible set. With a PARI/GP script we computed the first 2089 elements of \( \mathcal{B} \) since for \( \mathcal{C} := \{ b(2), \ldots, b(2089) \} \) we get \( m(\mathcal{C}) > 2 \).

So, if we are looking for negative values of \( \mathcal{A}_q \), it seems to be a good criterion to evaluate \( \mathcal{A}_q \) for a prime number \( q \) such that \( b q + 1 \) is prime for many elements \( b \in \mathcal{C} \) (clearly it is better to start with the smaller available \( b \)'s). To be able to measure this fact, we define

\[
v(q) := \sum_{2 \leq i \leq 2089; \ b(i) \in \mathcal{C}} \frac{1}{b(i)} \quad \text{for } b(i) q + 1 \text{ is prime}
\]

and we use such a function to classify \( \mathcal{A}_q \) in the following way. In the scatter plots of Figures 2–3 we classified the normalised values of \( \mathcal{A}_q \) and \( \mathcal{A}_q^+ \) according to \( v(q) \). Orange points are the more frequent ones (72.88% of the total number) and satisfy \( v(q) \leq 0.25 \); green points satisfy \( 0.25 < v(q) \leq 1/2 \) (18.29%); blue points satisfy \( 1/2 < v(q) \leq 0.75 \) (5.98%); black points satisfy \( 0.75 < v(q) \leq 1 \) (2.77%); red points satisfy \( v(q) > 1 \) (0.08%). The behaviour of \( \mathcal{A}_q \) is the expected one since the red strip essentially corresponds with its minimal values, while the minima of \( \mathcal{A}_q^+ \) seem to be less related to \( v(q) \); we plan to investigate this phenomenon in the future. In Figure 1 we also insert two histograms about the distribution of the normalised values of \( \mathcal{A}_q \) and \( \mathcal{A}_q^+ \).

In Figures 4–6 we present the scatter plots on \( M_q \) and \( M_q' := M_q / \log \log q \). All the plots were obtained using GNUPLLOT, v.5.2, patchlevel 8, with the exceptions of the histograms that were obtained using Python 3.9 (matplotlib v.3.3.3).

![Histograms](image.png)

**Figure 1.** The histograms for \( \mathcal{A}_q / \log q \) and \( \mathcal{A}_q^+ / \log q \), \( q \) prime, \( 3 \leq q \leq 10^7 \). In both the number of intervals is 100, the number of odd primes up to \( 10^7 \) is \( \mathcal{P} = 664578 \) and the mass is \( M = I \cdot \mathcal{P} \), where \( I \) is the interval length. For the first we have \( I = 0.017230 \ldots \), the mean is \( \mu = 1.043452 \ldots \) and the standard deviation is \( \sigma = 0.185900 \ldots \) For the second we have \( I = 0.010892 \ldots \), the mean is \( \mu = 1.043347 \ldots \) and the standard deviation is \( \sigma = 0.125566 \ldots \).
The values of $\Theta_q/\log q$, $q$ prime, $3 \leq q \leq 10^7$, classified using $v(q)$. The minimal value is $0.060532 \ldots$ and it is attained at $q = 4178771$; the maximal value is $1.626934 \ldots$ and it is attained at $q = 19$. 

**Figure 2.**
A FAST ALGORITHM TO COMPUTE RAMANUJAN-DEINGER’S GAMMA-FUNCTION

\[ \delta_q^+ / \log q \] with \( v(q) \leq 0.25 \)

\[ \delta_q^+ / \log q \] with \( 0.25 < v(q) \leq 1/2 \)

\[ \delta_q^+ / \log q \] with \( 1/2 < v(q) \leq 0.75 \)

\[ \delta_q^+ / \log q \] with \( 0.75 < v(q) \leq 1 \)

\[ \delta_q^+ / \log q \] with \( v(q) > 1 \)

\[ \delta_q^+ / \log q \]

**Figure 3.** The values of \( \delta_q^+ / \log q \), \( q \) prime, \( 3 \leq q \leq 10^7 \), classified using \( v(q) \). The minimal value is 0.436031 . . . and it is attained at \( q = 5483977 \); the maximal value is 1.426263 . . . and it is attained at \( q = 2053 \).
The values of \( M_q, q \) prime, \( 3 \leq q \leq 2000 \). The minimal value is 0.3682816... and it is attained at \( q = 3 \). The lines represent the functions \( c \cdot \log \log q \), with \( c = 17/20 \) and \( c = 5/4 \). \( M_{1531} = 2.5048094... \)

The values of \( M_q, q \) prime, \( 2000 < q \leq 10^7 \). The maximal value is 3.2466918... and it is attained at \( q = 8430391 \). The lines represent the functions \( c \cdot \log \log q \), with \( c = 17/20 \) and \( c = 5/4 \).

The values of \( M_q, q \) prime, \( 3 \leq q \leq 10^7 \). The minimal value is 0.7392305... and it is attained at \( q = 13 \); the maximal value is 3.9158971... and it is attained at \( q = 3 \) (not represented in the plot); the maximal value for every \( q > 13 \) is 1.204704... and it is attained at \( q = 1645093 \). The lines represent the constant functions \( c = 17/20 \) and \( c = 5/4 \).
A FAST ALGORITHM TO COMPUTE RAMANUJAN-DENINGER’S GAMMA-FUNCTION

REFERENCES

[1] B. C. Berndt, Ramanujan’s notebooks, Part I, Springer, 1985.
[2] H. Cohen, Number Theory. Volume II: Analytic and Modern Tools, Graduate Texts in Mathematics, vol. 240, Springer, 2007.
[3] C. Deninger, On the analogue of the formula of Chowla and Selberg for real quadratic fields, J. Reine Angew. Math. 351 (1984), 171–191.
[4] K. Dilcher, Generalized Euler constants for arithmetical progressions, Math. Comp. 59 (1992), 259–282.
[5] K. Dilcher, On generalized gamma functions related to the Laurent coefficients of the Riemann zeta function, Aequationes Math. 48 (1994), 55–85.
[6] K. Ford, F. Luca, P. Moree, Values of the Euler ϕ-function not divisible by a given odd prime, and the distribution of Euler-Kronecker constants for cyclotomic fields, Math. Comp. 83 (2014), 1447–1476.
[7] M. Frigo, S. G. Johnson, The Design and Implementation of FFTW3, Proceedings of the IEEE 93, 216–231 (2005). The C library is available at http://www.fftw.org.
[8] L. Gordon, A stochastic approach to the Gamma function, Amer. Math. Monthly 101 (1994), 858–865.
[9] N. J. Higham, The accuracy of floating point summation, SIAM Journal on Scientific Computing 14 (1993), 783–799.
[10] Y. Ihara, The Euler-Kronecker invariants in various families of global fields, in V. Ginzburg, ed., Algebraic Geometry and Number Theory: In Honor of Vladimir Drinfeld’s 50th Birthday, Progress in Mathematics 850, Birkhäuser Boston, Cambridge, MA, 2006, 407–451.
[11] Y. Ihara, On “M-functions” closely related to the distribution of L’/L-values, Publ. Res. Inst. Math. Sci. 44 (2008), 893–954.
[12] Y. Ihara, V. K. Murty, M. Shimura, On the logarithmic derivatives of Dirichlet L-functions at s = 1, Acta Arith. 137 (2009), 253–276.
[13] F. Johansson, I. V. Blagouchine, Computing Stieltjes constants using complex integration, Math. Comp. 88 (2019), 1829–1850.
[14] W. Kahan, Further remarks on reducing truncation errors, Communications of the ACM 8 (1965), page 40.
[15] J. C. Lagarias, Euler’s constant: Euler’s work and modern developments, Bull. Amer. Math. Soc. 50 (2013), 527–628.
[16] Y. Lamzouri, The distribution of Euler-Kronecker constants of quadratic fields, J. Math. Anal. Appl. 432 (2015), 632–653.
[17] Y. Lamzouri, A. Languasco, Small values of L’/L(1, χ), Arxiv (2020), http://arxiv.org/abs/2005.10714, to appear in Experimental Mathematics.
[18] A. Languasco, Efficient computation of the Euler-Kronecker constants for prime cyclotomic fields, Research in Number Theory 7 (2021), Paper n. 2, 1–22.
[19] A. Languasco, Numerical verification of Littlewood’s bounds for |L(1, χ)|, Journal of Number Theory 223 (2021), 12–34.
[20] 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.
[21] P. Moree, Irregular Behaviour of Class Numbers and Euler-Kronecker Constants of Cyclotomic Fields: The Log Log Log Devil at Play, Irregularities in the Distribution of Prime Numbers. From the Era of Helmut Maier’s Matrix Method and Beyond (J. Pintz and M.Th. Rassias, eds.), Springer, 2018, pp. 143–163.
[22] H. L. Montgomery, R. C. Vaughan, Multiplicative Number Theory I: Classical Theory, Cambridge University Press, 2007.
[23] The PARI Group, PARI/GP version 2.13.1, Bordeaux, 2021. Available from http://pari.math.u-bordeaux.fr/
[24] J.C. Schatzman, Accuracy of the Discrete Fourier Transform and the Fast Fourier Transform, SIAM J. Sci. Comput. 17 (1996), 1150–1166.

Alessandro Languasco, Luca Righi
Università di Padova, Dipartimento di Matematica “Tullio Levi-Civita”,
Via Trieste 63, 35121 Padova, Italy.
e-mail: alessandro.languasco@unipd.it
e-mail: righi@math.unipd.it