A NOTE ON THE COMPUTATION OF THE EULER-KRONECKER CONSTANTS FOR CYCLOTOMIC FIELDS

ALESSANDRO LANGUASCO

Abstract. The goal of this note is to introduce an alternative method to compute the Euler-Kronecker constants for cyclotomic fields and to compare it with other two different ways of computing the same quantity. The new algorithm requires the values of the generalised gamma functions Γ_1, also known as ${}_2\Gamma$, at some rational arguments in $(0,1)$. Using such method we were able to get $EK_{964477901} = -0.182374\ldots$, thus giving an independent confirmation of Theorem 4 of [7], $EK^+_{964477901} = 10.402223\ldots$, and to compute the values of EK_q and EK_q^+ for every odd prime $q \le 100000$. We also computed the value of EK_q for some larger prime number q but with no success in finding another negative value. Moreover, as a by-product, we will also provide more data on the generalised Euler constants in arithmetic progressions. The programs used to performed the computations here described and the numerical results obtained are available at the following web address: `http://www.math.unipd.it/~languasc/EK-comput.html`.

1. introduction

The goal of this note is to introduce an alternative method to compute the Euler-Kronecker constants for cyclotomic fields and to compare it with other two different ways of computing the same quantity. Moreover, as a by-product, we will also provide more data on the generalised Euler constants in arithmetic progressions.

The definition of the Euler-Kronecker constant for number fields is given in section 1.3 of Ford-Luca-Moree [7], see eq. (1.14) there, but we are here just interested in the special case of cyclotomic fields. In this situation we can use eq. (2.6) of [7]: if q is an odd prime then we define the *Euler-Kronecker constant for the cyclotomic field* $\mathbb{Q}(\zeta_q)$ as

$$EK_q := \gamma + \sum_{\chi \ne \chi_0} \frac{L'(1,\chi)}{L(1,\chi)}, \tag{1}$$

where ζ_q is a primitive q-root of unity, γ is the Euler constant, χ are the non-trivial Dirichlet characters mod q and χ_0 is the trivial Dirichlet character mod q. In [7] the quantity EK_q is denoted as γ_q but this conflicts with notations used in literature.

As we will see later, other quantities related to EK_q are the *generalised Euler constants in arithmetic progressions*, sometimes also called *Stieltjes constants in arithmetic progressions*, denoted as $\gamma_k(a,q)$, $k \in \mathbb{N}$, $q \ge 1$, $1 \le a \le q$, which are defined by

$$\begin{aligned}\gamma_k(a,q) &:= \lim_{N\to+\infty}\Big(\sum_{\substack{0<m\le N\\ m\equiv a \bmod q}} \frac{(\log m)^k}{m} - \frac{(\log N)^{k+1}}{q(k+1)}\Big)\\ &= -\frac{1}{q}\Big(\frac{(\log q)^{k+1}}{k+1} + \sum_{n=0}^{k}\binom{k}{n}(\log q)^{k-n}\psi_n\big(\frac{a}{q}\big)\Big),\end{aligned} \tag{2}$$

2010 *Mathematics Subject Classification.* Primary 11-04; Secondary 11Y60.

Key words and phrases. Euler-Kronecker constants, generalised Euler constants in arithmetic progressions, digamma functions.

see eq. (1.3)-(1.4) and (7.3) of Dilcher [5], where

$$\psi_n(z) := -\gamma_n - \frac{(\log z)^n}{z} - \sum_{m=1}^{+\infty}\Big(\frac{(\log(m+z))^n}{m+z} - \frac{(\log m)^n}{m}\Big) \tag{3}$$

for $n \in \mathbb{N}$ and $z \in \mathbb{C} \setminus \{0, -1, -2, \dots\}$, $\psi_n(1) = -\gamma_n$, and the *generalised Euler constants* γ_n are defined as

$$\gamma_n := \lim_{N\to+\infty}\Big(\sum_{j=1}^{N}\frac{(\log j)^n}{j} - \frac{(\log N)^{n+1}}{n+1}\Big) = \sum_{m=1}^{+\infty}\Big(\frac{(\log m)^n}{m} - \frac{(\log(m+1))^{n+1} - (\log m)^{n+1}}{n+1}\Big), \tag{4}$$

by, *e.g.*, eq. (3)-(4) of Bohman-Fröberg [1]. Remark that $\gamma_0 = \gamma = 0.577215664901\dots$, the Euler-Mascheroni constant.

The quantities in (2) and, as we will see in sections 2-3 below, the one in (1), are hence connected with the values of ψ_n, $n \geq 1$, which is the logarithmic derivative of Γ_n, a *generalised Gamma* function, see Deninger [4], Dilcher [6] and Katayama [10], whose definition for $n = 1$ is given in section 3.2. In some sense we can say that the ψ_n-functions, $n \geq 1$, are the analogue of the usual *digamma* function. In the following we will denote as ψ the standard digamma function Γ'/Γ; we also remark that it can be represented as the function ψ_0 defined in (3).

The theoretical part about the proofs of the identities that we will use in sections 2-3 is classical; a key tool is the functional equation for the Dirichlet L-functions. Such proofs can be found in Cohen's books [2]-[3], for instance. Other useful references are also the papers of Dilcher [5, 6] and Deninger [4].

2. Ford-Luca-Moree's method

Recall that q is an odd prime. If we do not distinguish between Dirichlet characters' parities, we can use use eq. (6.1) and (7.4) of Dilcher [5], as in Ford-Luca-Moree, see eq. (3.2) in [7]. In fact eq. (6.1) of [5] gives

$$L'(1,\chi) = -\sum_{a=1}^{q-1}\chi(a)\gamma_1(a,q),$$

where $\gamma_1(a,q)$ is defined in (2) which, for $k = 1$, becomes

$$\gamma_1(a,q) = -\frac{1}{q}\Big(\frac{1}{2}(\log q)^2 + \log q\,\psi\big(\frac{a}{q}\big) + \psi_1\big(\frac{a}{q}\big)\Big),$$

for any $q \geq 1$ and $1 \leq a \leq q$, where ψ, ψ_1 are defined in (3). Again using (3), we define

$$T(x) := \gamma_1 + \psi_1(x) = -\frac{\log x}{x} - \sum_{m=1}^{+\infty}\Big(\frac{\log(x+m)}{x+m} - \frac{\log m}{m}\Big), \tag{5}$$

and, specialising (4), we also have

$$\gamma_1 = \lim_{N\to+\infty}\Big(\sum_{j=1}^{N}\frac{\log j}{j} - \frac{(\log N)^2}{2}\Big) = -0.0728158454835\dots$$

To compute γ_1 and other similar constants with a very large precision, see section 5.3 below.

Recalling now eq. (3.1) of [7], *i.e.*,

$$L(1,\chi) = -\frac{1}{q}\sum_{a=1}^{q-1}\chi(a)\,\psi\big(\frac{a}{q}\big), \tag{6}$$

by the orthogonality of Dirichlet's characters and (6), we obtain eq. (3.2) of [7], *i.e.*

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

where $T(x)$ is defined in (5) (pay attention to the change of sign in (5) with respect to eq. (3.2) of [7]). Summarising, we finally get

$$\frac{L'(1,\chi)}{L(1,\chi)} = -\log q + \frac{1}{qL(1,\chi)}\sum_{a=1}^{q-1}\chi(a)\,T\Big(\frac{a}{q}\Big) = -\log q - \frac{\sum_{a=1}^{q-1}\chi(a)\,T(a/q)}{\sum_{a=1}^{q-1}\chi(a)\,\psi(a/q)}. \tag{7}$$

Formula (7), which was the one used in the paper by Ford-Luca-Moree [7], let us see that we can compute EK_q via (1) using the values of $\psi(a/q)$ and $T(a/q)$, which is connected to $\psi_1(a/q)$ via (5), together with the values of the non-trivial Dirichlet characters mod q.

From a computational point of view it is clear that in (7) we first have to evaluate $T(a/q)$ and $\psi(a/q)$ for every $1 \le a \le q-1$. For the $\psi(a/q)$-values we can rely on the PARI/Gp function `psi` while, for the $T(a/q)$-values we can use the summing function `sumnum`. We'll see more on these computations in section 4.

3. An alternative method: distinguishing Dirichlet characters' parities

3.1. $\chi \ne \chi_0$ is a primitive odd Dirichlet character. Recall that q is an odd prime, let $\chi \ne \chi_0$ be a primitive odd Dirichlet character mod q and let $\tau(\chi) := \sum_{a=1}^{q}\chi(a)\,e(a/q)$, $e(x) := \exp(2\pi i x)$, be the Gauss sum associated with χ. The functional equation for $L(s,\chi)$, see, *e.g.*, the proof of Theorem 3.5 of Gun-Murty-Rath [9], gives

$$L(s,\chi) = \frac{1}{\pi i}\Big(\frac{2\pi}{q}\Big)^{s}\Gamma(1-s)\frac{\tau(\chi)}{\sqrt{q}}\cos\Big(\frac{\pi s}{2}\Big)L(1-s,\overline{\chi})$$

and hence

$$\frac{L'(s,\chi)}{L(s,\chi)} = \log\Big(\frac{2\pi}{q}\Big) - \frac{\Gamma'(1-s)}{\Gamma(1-s)} - \frac{\pi}{2}\tan\Big(\frac{\pi s}{2}\Big) - \frac{L'(1-s,\overline{\chi})}{L(1-s,\overline{\chi})},$$

which, evaluated at $s = 0$, gives

$$\frac{L'(0,\chi)}{L(0,\chi)} = \log\Big(\frac{2\pi}{q}\Big) + \gamma - \frac{L'(1,\overline{\chi})}{L(1,\overline{\chi})}.$$

By the Lerch identity about values of the Hurwitz zeta-function, see, *e.g.*, either eq. (3.1) of Gun-Murty-Rath [9] or Proposition 10.3.5 of Cohen [3], and the orthogonality of Dirichlet characters, we get

$$\begin{aligned} L'(0,\chi) &= -\log q\sum_{a=1}^{q-1}\chi(a)\Big(\frac{1}{2}-\frac{a}{q}\Big) + \sum_{a=1}^{q-1}\chi(a)\log\Big(\Gamma\Big(\frac{a}{q}\Big)\Big) \\ &= \frac{\log q}{q}\sum_{a=1}^{q-1}a\chi(a) + \sum_{a=1}^{q-1}\chi(a)\log\Big(\Gamma\Big(\frac{a}{q}\Big)\Big) \\ &= -(\log q)L(0,\chi) + \sum_{a=1}^{q-1}\chi(a)\log\Big(\Gamma\Big(\frac{a}{q}\Big)\Big), \end{aligned}$$

since, see Corollary 10.3.2 of Cohen [3], we have $L(0,\chi) = -(\sum_{a=1}^{q-1} a\chi(a))/q$. Summarising we obtain

$$\begin{aligned}\frac{L'(1,\chi)}{L(1,\chi)} &= \gamma + \log(2\pi) - \frac{1}{L(0,\overline{\chi})}\sum_{a=1}^{q-1}\overline{\chi}(a)\log\Bigl(\Gamma\bigl(\frac{a}{q}\bigr)\Bigr)\\ &= \gamma + \log(2\pi) + q\frac{\sum_{a=1}^{q-1}\overline{\chi}(a)\log\bigl(\Gamma(a/q)\bigr)}{\sum_{a=1}^{q-1} a\overline{\chi}(a)}. \qquad (8)\end{aligned}$$

From a computational point of view, in (8) we need to compute the $\log\bigl(\Gamma(a/q)\bigr)$-values instead of the $\psi(a/q)$ ones as in (7); to do so we can rely on internal PARI/Gp functions. In the next paragraph we will see that we have to reuse such values for the even Dirichlet characters case.

3.2. $\chi \neq \chi_0$ **is a primitive even Dirichlet character.** Recall that q is an odd prime. Assume now that $\chi \neq \chi_0$ is a primitive even Dirichlet character mod q. We follow Deninger's notation in [4] by calling $R(x) = \log(\Gamma_1(x))$, $x > 0$. By eq. (3.5)-(3.6) of [4] we have

$$L'(1,\chi) = (\gamma + \log(2\pi))L(1,\chi) + \frac{\tau(\chi)}{q}\sum_{a=1}^{q-1}\overline{\chi}(a)\, R\bigl(\frac{a}{q}\bigr),$$

where, see eq. (2.3.2) of [4], the R-function is defined for every $x > 0$ by

$$R(x) := -\zeta''(0) - S(x), \qquad (9)$$

$$S(x) := 2\gamma_1 x + (\log x)^2 + \sum_{m=1}^{+\infty}\Bigl(\bigl(\log(x+m)\bigr)^2 - (\log m)^2 - 2x\frac{\log m}{m}\Bigr) \qquad (10)$$

and $-\zeta''(0) = \bigl((\log(2\pi))^2 + \frac{\pi^2}{12} - \gamma^2 - 2\gamma_1\bigr)/2 = 2.006356455908\ldots$ Comparing (9)-(10) with (5), we see that $\psi_1(x) = R'(x)/2$; please pay attention to the different definition of γ_1 on page 174 of Deninger's paper. Remark also that $R(1) = -\zeta''(0)$ and $S(1) = 0$. By the orthogonality of the Dirichlet characters, we immediately get $\sum_{a=1}^{q-1}\overline{\chi}(a)\, R\bigl(\frac{a}{q}\bigr) = -\sum_{a=1}^{q-1}\overline{\chi}(a)\, S\bigl(\frac{a}{q}\bigr)$.

For $L(1,\chi)$, we use formula (2) of Proposition 10.3.5 of Cohen [3] and the parity of χ to get

$$L(1,\chi) = 2\frac{\tau(\chi)}{q}\sum_{a=1}^{q-1}\overline{\chi}(a)\log\Bigl(\Gamma\bigl(\frac{a}{q}\bigr)\Bigr),$$

since $W(\chi) = \tau(\chi)/q^{1/2}$ for even Dirichlet characters, see Definition 2.2.25 of Cohen [2]. Summarising, if χ is an even Dirichlet character mod q, we finally get

$$\begin{aligned}\frac{L'(1,\chi)}{L(1,\chi)} &= \gamma + \log(2\pi) - \frac{\tau(\chi)}{qL(1,\chi)}\sum_{a=1}^{q-1}\overline{\chi}(a)\, S\bigl(\frac{a}{q}\bigr)\\ &= \gamma + \log(2\pi) - \frac{1}{2}\frac{\sum_{a=1}^{q-1}\overline{\chi}(a)\, S(a/q)}{\sum_{a=1}^{q-1}\overline{\chi}(a)\log\bigl(\Gamma(a/q)\bigr)}. \qquad (11)\end{aligned}$$

We remark that in (11) we have to perform the computation of the S-function but here we can reuse the $\log\bigl(\Gamma(a/q)\bigr)$-values, $1 \le a \le q-1$, already computed in eq. (8). For the $S(a/q)$-values we can use the PARI/Gp summing function `sumnum`.

We finally remark that the computation in this section reveals that the Euler-Kronecker constant EK_q^+ for the maximal real subfield of $\mathbb{Q}(\zeta_q)$ is directly connected with the S-function values at

a/q since, according to eq. (10) of Moree [11] and (11), we have

$$EK_q^+ := \gamma + \sum_{\substack{\chi \neq \chi_0 \\ \chi(-1)=1}} \frac{L'(1,\chi)}{L(1,\chi)} = \frac{q-1}{2}\gamma + \frac{q-3}{2}\log(2\pi) - \frac{1}{2} \sum_{\substack{\chi \neq \chi_0 \\ \chi(-1)=1}} \frac{\sum_{a=1}^{q-1} \overline{\chi}(a)\, S(a/q)}{\sum_{a=1}^{q-1} \overline{\chi}(a) \log\big(\Gamma(a/q)\big)}$$

and hence it seems that in this case the relevant information is encoded in the S-function. By (8) it is then trivial to get that

$$EK_q - EK_q^+ = \sum_{\substack{\chi \\ \chi(-1)=-1}} \frac{L'(1,\chi)}{L(1,\chi)} = \frac{q-1}{2}\Big(\gamma + \log(2\pi)\Big) + q \sum_{\substack{\chi \\ \chi(-1)=-1}} \frac{\sum_{a=1}^{q-1} \overline{\chi}(a) \log\big(\Gamma(a/q)\big)}{\sum_{a=1}^{q-1} a\overline{\chi}(a)}.$$

4. About the computations of EK_q: comparing methods, results and running times

First of all we notice that PARI/Gp, v. 2.11.1, has the ability to generate the Dirichlet L-functions (and many other L-functions) and hence the computation of EK_q can be performed using (1) with few instructions of the gp scripting language. This computation has a linear cost in the number of calls of the `lfun` function of PARI/Gp and, at least for $271 \le q \le 20011$, is, on our Macbook laptop, slower than both the approaches we are about to describe below.

Comparing (8) and (11) with (7), we see that in both cases we can rely on internal PARI/Gp functions to compute either the $\log\big(\Gamma(a/q)\big)$-values or the $\psi(a/q)$-values, $1 \le a \le q-1$, and finally we have to evaluate the T and S series respectively involved. Remarking that all these functions have a pole in 0 and that we will take q very large, it is also relevant to know their order of magnitude for $x \to 0^+$; it is easy to verify that $\log\big(\Gamma(x)\big) \sim \log(1/x)$, $S(x) \sim (\log x)^2$, $\psi(x) \sim -1/x$ and $T(x) \sim \log(1/x)/x$. Hence for $x \to 0^+$, we can expect that $\log\big(\Gamma(x)\big)$ and $S(x)$ will be exponentially smaller than $\psi(x)$ and $T(x)$. Another difference is that, for the odd Dirichlet characters, eq. (8) before does not involve the estimation of an infinite series. So it seems reasonable to compare the following two approaches:

a) use the T-series formulae and the $\psi(a/q)$-values for computing both the odd and the even Dirichlet characters cases like in [7];
b) use the S-series formulae for the even case and the finite sum of the $a\overline{\chi}(a)$-values for the odd one; remark that in both cases we have to evaluate a sum of the $\log\big(\Gamma(a/q)\big)$-values.

This way we can double check the computation performed in [7] not only because we are developing a different implementation of the same formulae, but also because we can approach the problem in an alternative way. In the computation we will use the PARI/Gp scripting language to exploit its ability in accurately evaluate the infinite sums involved in the definition of the T and S series previously described via the predefined summing function `sumnum`.

To check the correctness of the practical computations it is possible to use the following formulae; recalling $\gamma = 0.577215664901\ldots$ and $\zeta''(0) = -2.006356455908\ldots$, we have that

$$\sum_{a=1}^{q-1} \log\Big(\Gamma\big(\frac{a}{q}\big)\Big) = \frac{1}{2}\Big((q-1)\log(2\pi) - \log q\Big), \tag{12}$$

$$\sum_{a=1}^{q-1} S\big(\frac{a}{q}\big) = -\zeta''(0)(q-1) - \log q \log(2\pi) - \frac{(\log q)^2}{2}, \tag{13}$$

$$\sum_{a=1}^{q-1} \psi\big(\frac{a}{q}\big) = -\gamma(q-1) - q\log q, \tag{14}$$

$$\sum_{a=1}^{q-1} T\Big(\frac{a}{q}\Big) = \frac{q}{2}(\log q)^2 + \gamma q \log q. \tag{15}$$

Formula (12) follows from Gauß' multiplication formula, see, *e.g*, section 12.15 of Whittaker-Watson [13], formula (13) is an immediate consequence of Theorem 2.5 of Deninger [4] and formulae (14)-(15) follow respectively from equations (7.9)-(7.10) of Dilcher [5].

We also remark that approaches a)-b) trivially require a quadratic number of products to perform the computations in (7)-(8) and (11), but this can be improved by using the Discrete Fourier Transform (DFT) and the following argument. Focusing on (7)-(8) and (11), we remark that, since q is prime, it is enough to get g, a primitive root of q, and χ_1, the Dirichlet character mod q given by $\chi_1(g) = e^{2\pi i/(q-1)}$, to see that the set of the non-trivial characters mod q is $\{\chi_1^j : j = 1, 2, \dots, q-2\}$. Hence, if, for every $k \in \{0, \dots, q-2\}$, we denote $g^k \equiv a_k \in \{1, \dots, q-1\}$, every summation in (7)-(8) and (11) is either of the type $\sum_{k=1}^{q-1} e^{2\pi ijk/(q-1)} f(a_k/q)$ or $\sum_{k=1}^{q-1} e^{-2\pi ijk/(q-1)} f(a_k/q)$, where $j \in \{1, \dots, q-2\}$ and f is a suitable function. As a consequence, such quantities are the DFT, or its inverse transformation, of the sequence $\{f(a_k/q) : k = 0, \dots, q-2\}$. This was used in [7] to speed-up the computation of these quantities via the use of DFT-dedicated software libraries. Unfortunately in the scripting language of PARI/Gp the DFT-functions work only if $q = 2^\ell + 1$, for some $\ell \in \mathbb{N}$. So we had to trivially perform these summations and hence, in practice, this part is the most time consuming one in both the approaches a) and b) since it has a quadratic cost in q; this is the reason why for $q > 2011$ the direct approach using the `lfun` function of PARI/Gp becomes faster than the others (trivially performing the sums over $a = 1, \dots, q-1$).

Being aware of such limitations, we performed the computation of EK_q with these three approaches for every q prime, $q \le 300$, on a Macbook Air laptop ("Early 2015", 8Gb RAM, 1.6 Ghz Intel Core i5, two cores) using a precision of 30 decimal digits, see Table 1 of section 6. The results coincide up the desired precision and are coherent with the data of Table 1 on page 1472 of [7]. Moreover it seems that the version which uses the T-function is a bit faster than the one with the S-function probably because the internal sequence involved in its summation has a simpler form with respect to the one involved in the sum defining S, although the general term of such series have roughly the same decay order. In fact the computation of the values of Table 1 needed 1 minute and 7 seconds using the T-function, 1 minute and 18 seconds using the S-function and 1 minute and 23 seconds via the direct approach. We also computed the values of EK_q for $q = 1009, 2003, 3001, 4001, 5003, 6007, 7001, 8009, 9001, 10007, 20011, 30011$, as you can see in Table 2 of section 6. These numbers were chosen to heuristically evaluate how the computational cost depends on the size of q. In this case, in the fifth column of Table 2 we also reported the running time of the direct approach, *i.e.* using (1), the third and fourth columns are respectively the running times of the approaches a) and b), while the sixth one indicates the used precision. For these values of q it became clear that the computation time spent in performing the sums having the Dirichlet characters values as coefficients was the longest one. This means that inserting a DFT-algorithm in the approaches a) and b) is fundamental to further improve their performances.

Hence for larger values of q we used the gp2c compiler tool to obtain suitable C programs to perform the precomputations of the needed T, ψ, S and $\log(\Gamma)$-values. Then we passed such values to other C programs which used the `fftw` [8] library to perform the final stage. In this final stage the performances were extremely good in the sense that such a part was thousands-times faster than the same one trivially performed. This way we computed the values of EK_q for $q = 40009, 42611, 50021, 60013, 70001, 80021, 90001, 100003, 305741, 1000003$, 4178771, 6766811, 10000019, 28227761, with double, long double and quadruple precisions, see Table 3. Some of these q-values were chosen for their dimension, while others because their measures using the "greedy sequence of prime offsets", `http://oeis.org/A135311`, are larger than 1.2 so that they are good candidates to have a negative Euler-Kronecker constant, see sect. 1.4 of [7] or sect. 2-3 of [11]. Such computations were performed with and Dell OptiPlex-3050, equipped with an Intel i5-7500 processor, 3.40GHz, four cores, 8 GB of RAM and running Ubuntu 18.04. We remark that the quadruple precision computation performances are affected from a lack of hardware support of the `FLOAT128` type of the C programming language.

After having evaluated the running times of the previous examples, we decided to provide the scattered plots, see Figures 1-2, of the normalised values of EK_q and EK_q^+ (both in long double precision) for every odd prime $q \leq 10^5$ thus doubling the known range of such data, see [7]. Such computation required the use of the cluster of the Mathematical Department of the University of Padova; for a description of the used cluster see `http://www.math.unipd.it/~languasc/EKcomput/Description-Cluster-Math-Unipd.pdf`. The minimal value of $EK_q/\log q$, $3 \leq q \leq 10^5$, q prime, is $0.23449\ldots$ and it is attained at $q = 42611$, as expected.

For even larger values of q the precomputations, if performed on a single desktop computer, would require too much time; hence we parallelised them on the cluster previously mentioned. This way we were able to obtain an independent confirmation of Theorem 4 of [7] getting $EK_{964477901} = -0.18237472563711916085\ldots$, $EK^+_{964477901} = 10.40222338242826353694\ldots$ with the S-function and $EK_{964477901} = -0.18237469280744579234\ldots$ with the T-function, since we computed them using the quadruple precision at the final stage. To do so we first split the computation, with a precision of 38 decimal digits, of the needed values of T, ψ, S and $\log(\Gamma)$ in 97 subintervals I_j of size 10^7 each; the computation time required for each I_j was on average about about 2100 minutes on the cluster for the slower function involved (the S-function); finally the final DFT-step needed about 65 minutes (long double precision) or 522 minutes (quadruple precision) of computation time using the FFTW library [8] on an Intel(R) Xeon(R) CPU E5-2650 v3 @ 2.30GHz, with 160 GB of RAM.

Further improvements on our programs were then performed to lower the RAM and disk occupation and to use a dedicated FFTW interface, called `guru64`, for being able to work on transforms whose length is $\geq \lceil (2^{32}-1)/2 \rceil$ (the `int` bound for the C programming language). This let us to evaluate the Euler-Kronecker constants for larger "good" candidates q (in the sense that their measures using the greedy sequence of prime offsets were larger than 1.2). This way, after having used the cluster to get the values of T, ψ, S and $\log(\Gamma)$, in about 90 minutes of computation time on the same machine mentioned before we got that $EK_{2918643191} = 0.302789\ldots$ and $EK^+_{2918643191} = 12.573983\ldots$ using the long double precision. In this case it seems that double precision version performed using the T-function is much less stable than the one with the S-function probably because of the fact that $T(x)$ and $\psi(x)$ are, for $x \to 0^+$, respectively much larger that $S(x)$ and $\log(\Gamma(x))$.

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/EK-comput.html`.

5. On the generalised Euler constants in arithmetic progressions $\gamma_k(a, q)$

Recall that q is an odd prime. In the case we are using the T-series, we have to precompute their values at a/q and the ones of ψ at the same arguments. Hence, as a by-product we can also obtain the values of the generalised Euler constants $\gamma_0(a, q)$ and $\gamma_1(a, q)$ as you can see in paragraphs 5.1-5.2. In practice this is obtained by activating an optional flag in the main gp script. The case about $\gamma_k(a,q)$, $k \geq 2$, is described in paragraph 5.3.

5.1. **Generalised Euler constants** $\gamma_0(a,q)$**.** For $\gamma_0(a,q)$ with $1 \leq a \leq q-1$, q odd prime, by (2) we have

$$\gamma_0(a,q) = -\frac{1}{q}\Big(\log q + \psi(\frac{a}{q})\Big).$$

Recalling $\psi(1) = -\gamma$, we also have $\gamma_0(q,q) = (\gamma - \log q)/q$.

5.2. **Generalised Euler constants** $\gamma_1(a,q)$**.** For $\gamma_1(a,q)$ with $1 \leq a \leq q-1$, q odd prime, we can use eq. (2) and (5). This way we get

$$\gamma_1(a,q) = -\frac{1}{q}\Big(\frac{(\log q)^2}{2} + (\log q)\psi(\frac{a}{q}) + \psi_1(\frac{a}{q})\Big) = \frac{1}{q}\ \Big(\gamma_1 - \frac{(\log q)^2}{2} - (\log q)\psi(\frac{a}{q}) - T(\frac{a}{q})\Big).$$

Moreover, since $\psi(1) = -\gamma$ and $T(1) = 0$, we also have

$$\gamma_1(q,q) = \frac{1}{q}\Big(\gamma_1 + \gamma \log q - \frac{(\log q)^2}{2}\Big).$$

Using the formulae in the previous two paragraphs we computed $\gamma_0(a,q)$ and $\gamma_1(a,q)$ with q prime, $3 \leq q \leq 100$, $1 \leq a \leq q$, in about 9 seconds of computation time with a precision of 30 digits. Such results are listed at the bottom of the gp-script file that can be downloaded here: `http://www.math.unipd.it/~languasc/EK-comput.html`.

5.3. **The general case** $\gamma_k(a,q)$**,** $k \geq 2$**.** The general case $\gamma_k(a,q)$, $k \in \mathbb{N}$, $k \geq 2$, $q \geq 1$, $1 \leq a \leq q$, do not follow from the data already computed for the Euler-Kronecker constants since we need information about the values of $\psi_n(x)$, for every $2 \leq n \leq k$. Such a direct computation of both $\psi_n(a/q)$ and γ_n can be easily performed via eq. (2)-(3) using the PARI/Gp summing function `sumnum` paying attention to submit a sufficiently fast convergent sum. For example, to compute γ_n, $n \in \mathbb{N}$, we used the formulae

$$\gamma_n = \sum_{m=1}^{+\infty}\Big(\frac{(\log m)^n}{m} - \frac{1}{n+1}\sum_{j=0}^{n}\binom{n+1}{j}(\log m)^j(\log(1+1/m))^{n+1-j}\Big) \tag{16}$$

and

$$\gamma_n = \sum_{m=1}^{+\infty}(\log m)^n\Big(\frac{1}{m} - \log\big(1+\frac{1}{m}\big)\Big) - \frac{1}{n+1}\sum_{j=0}^{n-1}\binom{n+1}{j}(\log m)^j(\log(1+1/m))^{n+1-j}\Big), \tag{17}$$

which both easily follow from (4). We get, in less than 15 seconds of time and with a precision of at least 40 digits, the results in Table 4 of section 6; to be sure about the correctness of such results we computed them twice using the formulae (16)-(17) and then we compared their results. Such values are in agreement with the data on page 282 of Bohman-Fröberg [1] for $n = 0, \dots, 20$. For larger n's the formulae in (16)-(17) seem to be not good enough to get precise results via the `sumnum` function with this precision level.

To compute $\psi_n(a/q)$ and, as a consequence, $\gamma_k(a,q)$, we can proceed in a similar way as we did for $T(a/q)$ and $\gamma_1(a,q)$, see the program file here `http://www.math.unipd.it/~languasc/EK-comput.html`. At the bottom of such program file you can find a large list (too long to be included here) of computed values of $\gamma_k(a,q)$ for $1 \leq k \leq 20$, $1 \leq q \leq 9$, $1 \leq a \leq q$, with a precision of 20 digits. In about 1 minute and 33 seconds of computation time we replicated Dilcher's computations since the values we got are in agreement with the data on pages S21-S24 of [5].

Acknowledgements. I wish to thank Karim Belabas and Bill Allombert for a couple of key suggestions about `libpari` and `gp2c` and Luca Righi for his help in developing the quadruple precision versions of the fft-programs, in designing the parallelised precomputations and in organising the computational runs on the cluster of the Math. Dept. of the University of Padova.

References

[1] J. Bohman and C.-E. Fröberg, *The Stieltjes Function–Definition and Properties*, Math. Comp. **51** (1988), 281–289.

[2] H. Cohen, *Number Theory. Volume I: Tools and Diophantine Equations*, Graduate Texts in Mathematics, vol. 239, Springer, 2007.

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

[4] C. Deninger, *On the analogue of the formula of Chowla and Selberg for real quadratic fields*, J. Reine Angew. Math. **351** (1984), 171–191.

[5] K. Dilcher, *Generalized Euler constants for arithmetical progressions*, Math. Comp. **59** (1992), 259–282.

[6] K. Dilcher, *On generalized gamma functions related to the Laurent coefficients of the Riemann zeta function*, Aequationes Math. **48** (1994), 55–85.

[7] K. Ford, F. Luca, and 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.

[8] M. Frigo and 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`.

[9] S. Gun, M. R. Murty, and P. Rath, *Transcendental nature of special values of L-functions*, Canad. J. Math. **63** (2011), 136–152.

[10] K. Katayama, *Class number formulas, Kronecker's limit formulas, Chowla–Selberg formulas and the generalized gamma functions*, J. Number Theory **133** (2013), 2092–2120.

[11] 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.

[12] The PARI Group, *PARI/GP version* `2.11.1`, Bordeaux, 2018. Available from `http://pari.math.u-bordeaux.fr/`.

[13] E. T. Whittaker and G. N. Watson, *A course of modern analysis*, Cambridge U. P., 1996.

Alessandro Languasco, Università di Padova, Dipartimento di Matematica, "Tullio Levi-Civita", Via Trieste 63, 35121 Padova, Italy. *e-mail*: alessandro.languasco@unipd.it

6. Tables

q	EK_q	q	EK_q
3	0.945497280871680703239749994158 . . .	131	2.836826341588379098602857973221 . . .
5	1.720624212513404761695728788865 . . .	137	4.937000226143684686919629997111 . . .
7	2.087594074717330132815424719571 . . .	139	5.889168633998671867263837303369 . . .
11	2.415425904283267830342879635831 . . .	149	5.983424777695159814502427857391 . . .
13	2.610757737417650196997761088571 . . .	151	5.042016113528721799145194610221 . . .
17	3.581976044097577659271788129191 . . .	157	7.408022065722227293508452013901 . . .
19	4.790409415714283325907039364581 . . .	163	5.929664822887206787554999138441 . . .
23	2.611289176188200925507391649641 . . .	167	8.033001752688724704675833578021 . . .
29	3.093731705994268723162751798191 . . .	173	3.384347536532061903442977988971 . . .
31	4.314442925267475097707574410421 . . .	179	3.862361325499030081121261302821 . . .
37	4.304938189957602017985579264171 . . .	181	5.141118487768481358101366642571 . . .
41	3.971521627921332160282570400141 . . .	191	4.692869902014226640035524348121 . . .
43	4.378627505746950494137750623361 . . .	193	5.163422196739154833200782627201 . . .
47	4.799394258907416134527584299881 . . .	197	7.551487158966406478864851293721 . . .
53	4.337736858597092318696960823071 . . .	199	6.473665136093207386994974597781 . . .
59	5.433516345385003980776344381931 . . .	211	7.736135784245861625328105875851 . . .
61	5.071085190576516195958050981131 . . .	223	7.817779717859913674713367348511 . . .
67	5.292139306628962608734284618311 . . .	227	8.080531569512962186970711937571 . . .
71	5.255258192818946167720131286371 . . .	229	7.162986320580995467457781150581 . . .
73	4.066949090447495292016488156251 . . .	233	3.119483544851275413031152952581 . . .
79	4.998276318170680107894313929451 . . .	239	3.999110172078332495126322979191 . . .
83	3.033136113436074187164038191051 . . .	241	6.037525214010342150657092509351 . . .
89	4.164090798889832768808411103721 . . .	251	5.043137085023473510428111190221 . . .
97	4.891240740403896668307514688571 . . .	257	8.169913912327413916702251552271 . . .
101	5.297012891509669718878600327391 . . .	263	7.303436247368154354143480774061 . . .
103	5.144339551252088221133305032201 . . .	269	6.260348316665771027352527557121 . . .
107	5.458274209970245034216802454531 . . .	271	5.977178048548033042237739059761 . . .
109	6.906638146264236532194698377041 . . .	277	4.592808177140778951647770816611 . . .
113	4.021730382578030675783180066171 . . .	281	4.664964323662114575052208526231 . . .
127	5.088599124153334494232156362401 . . .	283	7.150285797410682514092252311881 . . .
		293	3.384381521219539786584682592381 . . .

Table 1. Values of EK_q for every odd prime up to 300 with a precision of 30 digits; computed with PARI/Gp, v. 2.11.1.

q	EK_q	time T-version	time S-version	time direct version	digits
1009	8.442135151849299275860694672741 . . .	11 sec.	13 sec.	23 sec.	30
2003	5.793421369079363328038498216251 . . .	30 sec.	36 sec.	1 min. 0 sec.	30
3001	8.647465136968386938802345350921 . . .	54 sec.	1 min. 6 sec.	1 min. 51 sec.	30
4001	7.003435546203143994356851768431 . . .	1 min. 28 sec.	1 min. 44 sec.	2 min. 48 sec.	30
5003	5.549293004581614227736879540411 . . .	2 min. 16 sec.	2 min. 37 sec.	3 min. 59 sec.	30
6007	8.311610121998483816562903440381 . . .	3 min. 7 sec.	3 min. 35 sec.	5 min. 30 sec.	30
7001	8.505277876100877139316878038471 . . .	4 min. 15 sec.	4 min. 38 sec.	6 min. 43 sec	30
8009	11.686846391549357535345086996011 . . .	5 min. 48 sec.	6 min. 16 sec.	8 min. 22 sec.	30
9001	10.109478431838340935822503580211 . . .	6 min. 32 sec.	7 min. 35 sec.	9 min. 49 sec.	30
10007	12.664612004560692327538935678311 . . .	7 min. 49 sec.	9 min. 15 sec.	12 min. 21 sec.	30
20011	10.799680311299920518643040289911 . . .	33 min. 31 sec.	35 min. 38 sec.	35 min. 7 sec.	30
30011	10.333079972124024225513606225511 . . .	74 min. 44 sec.	79 min. 18 sec.	65 min. 19 sec.	30

Table 2. Few other values of EK_q with a precision of 30 digits; computed with PARI/Gp, v. 2.11.1.

q	EK_q	q	EK_q
40009	13.14688498189670340252 ...	305741	1.65052277946237473601 ...
42611	2.49968848052266852388 ...	1000003	17.37997023827717073547 ...
50021	9.91050713790501287268 ...	4178771	0.92285541557399964022 ...
60013	12.81035968578751809187 ...	6766811	1.60404528321177759204 ...
70001	12.57276479037455044850 ...	10000019	17.08794469564282064569 ...
80021	14.18563290071312273227 ...	28227761	2.36156224708899177985 ...
90001	11.81942408990480618779 ...	964477901	−0.18237472563711916085 ...
100003	15.16607400737259891798 ...	2918643191	0.302789 ...

TABLE 3. Few other values of EK_q; computed with PARI/Gp, v. 2.11.1. and fftw, v. 3.3.8., with long double or quadruple precision.

n	γ_n
0	0.5772156649015328606065120900824024310 4 ...
1	−0.07281584548367672486058637587490131913 ...
2	−0.00969036319287231848453038603521252934 ...
3	0.00205383442030334586616004654275338428 ...
4	0.00232537006546730005746817017752606800 ...
5	0.00079332381730106270175333487744444483 ...
6	−0.00023876934543019960987242184190800 42 ...
7	−0.00052728956705775104607409750547885828 ...
8	−0.00035212335380303950960205216500120874 ...
9	−0.00003439477441808804817791462379822739 ...
10	0.00020533281490906479468372228923706530 ...
11	0.00027018443954390352667290208206795567 ...
12	0.00016727291210514019335350154334118344 ...
13	−0.00002746380660376015886000760369335518 ...
14	−0.00020920926205929994583713969734458 49 ...
15	−0.00028346865532024144664293447499712697 ...
16	−0.00019969685830896977470778456320324032 ...
17	0.00002627703710991833669946659763051013 ...
18	0.00030736840814925282659275475194862564 ...
19	0.00050360545304735562905559643771716003 ...
20	0.00046634356151155944940059482443355052 ...
21	0.00010443776975600011581079567436772049 ...
22	−0.00054159958220399770165519617317410552 ...
23	−0.00124396209040824577929974159953716582 ...
24	−0.00158851127890356156190619661152111582 ...
25	−0.00107459195273848882472429198735317305 ...
26	0.00065680351863715443150477300335621524 ...
27	0.00347783691361853820900735957425881154 ...
28	0.00640006853170062945810722822194586366 ...
29	0.00737115177047223913441240242355940215 ...
30	0.00355772885557316094791353774890840261 ...

TABLE 4. Computation of the generalised Euler constants γ_n, $0 \le n \le 30$, with a precision of at least 40 digits; computed with PARI/Gp, v. 2.11.1.

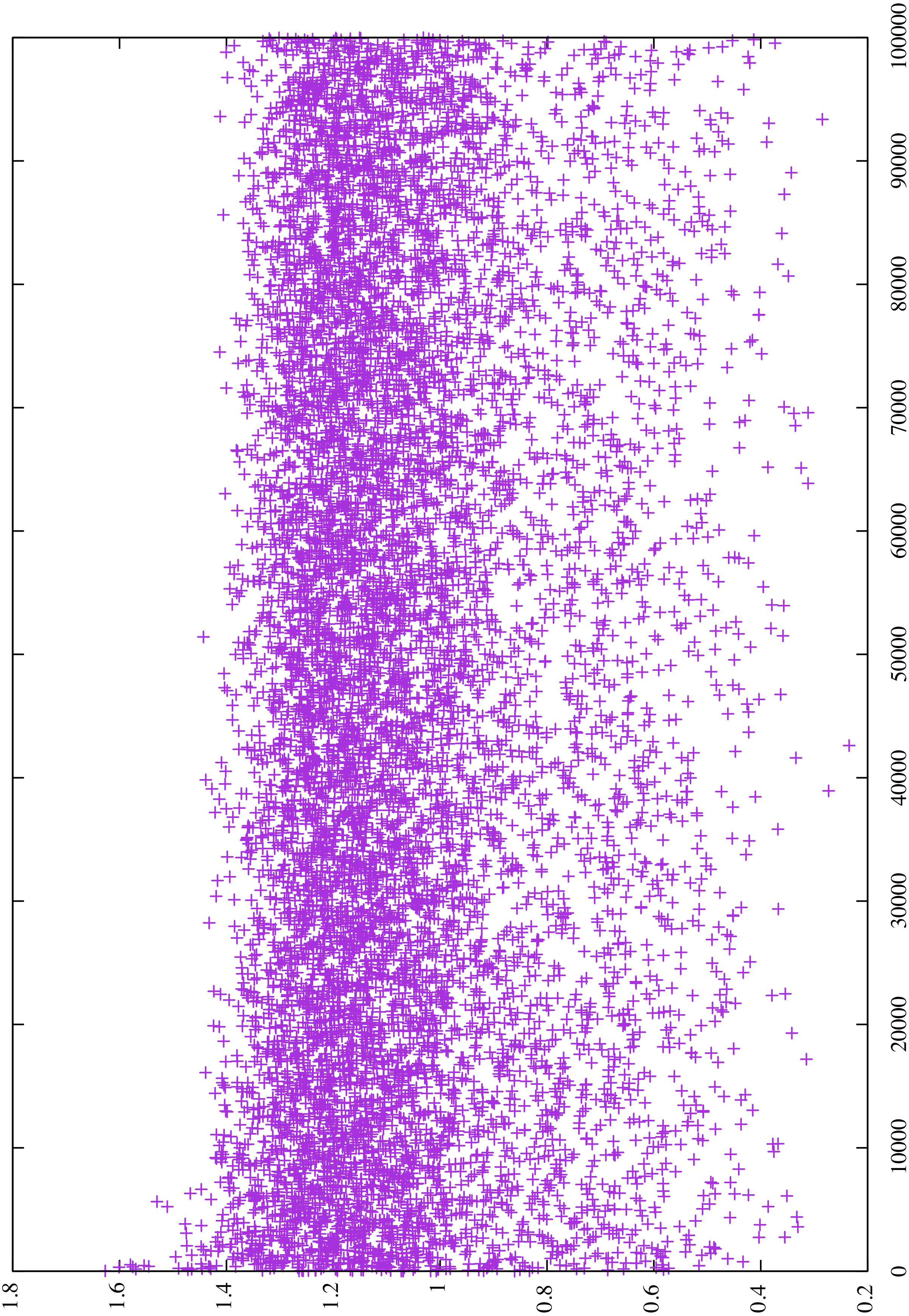

FIGURE 1. The values of $EK_q/\log q$, q prime, $3 \leq q \leq 100000$, plotted using GNUPLOT, v.5.2, patchlevel 2.

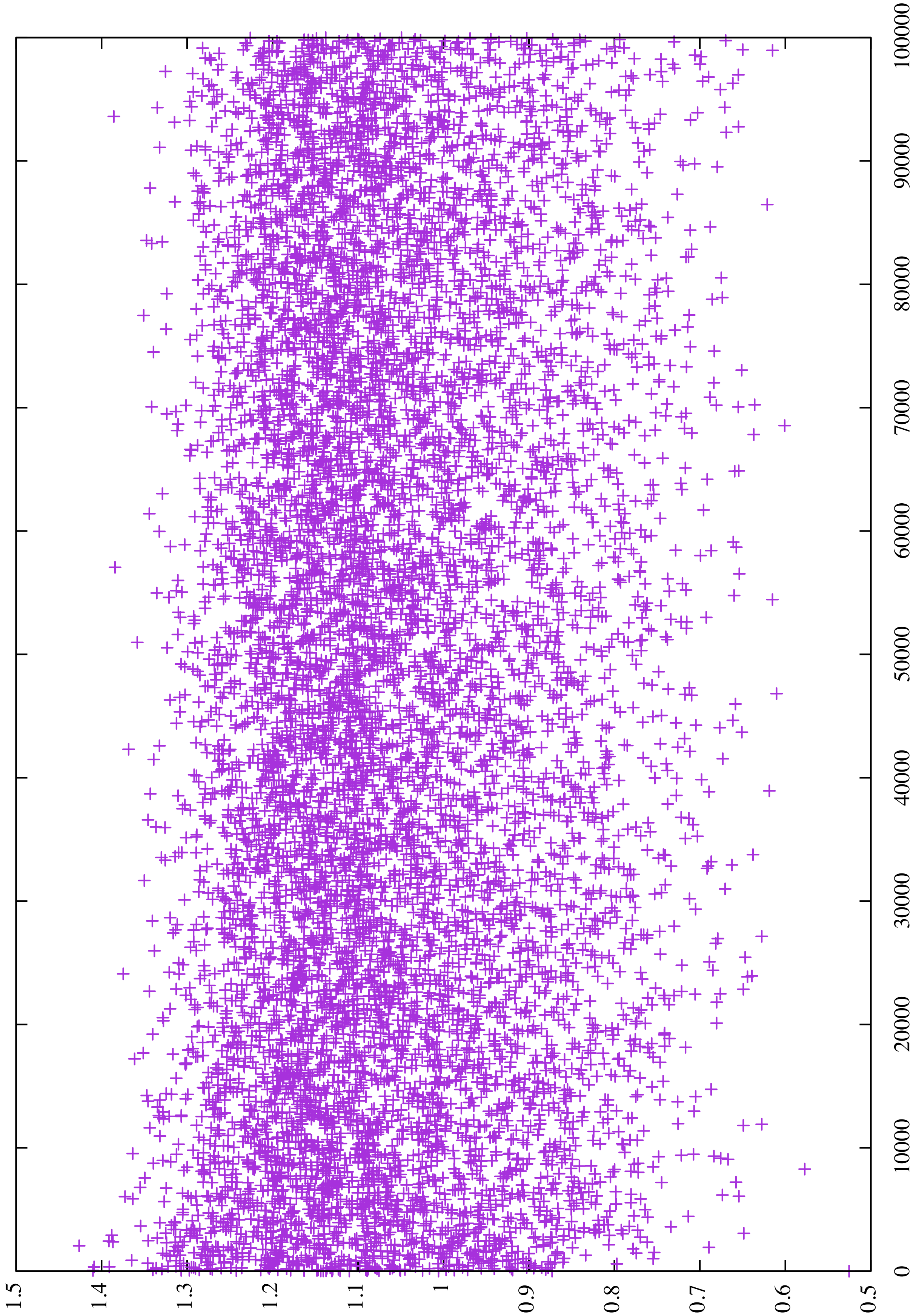

FIGURE 2. The values of $EK_q^+/\log q$, q prime, $3 \le q \le 100000$, plotted using GNUPLOT, v.5.2, patchlevel 2.