The numerical computation of high-order derivatives, with application to Euler sums

David H. Bailey,
Lawrence Berkeley National Laboratory, USA,
dhbailey@lbl.gov
Ross McPhedran,
School of Physics, University of Sydney 2006, Australia,
ross.mcphedran@sydney.edu.au
and Bruno Salvy,
Institut National de Recherche en Sciences et Technologies du Numérique (INRIA),
bruno.salvy@inria.fr

June 6, 2026

Abstract

Euler sums, namely infinite series involving the harmonic function $H_k = 1 + 1/2 + 1/3 + \cdots + 1/k$, have been studied for centuries, as they arise in numerous contexts. In two previous studies by the present authors, methods were presented for the closed-form evaluation of certain classes of Euler sums. This research relied heavily on very high-precision numerical computations, which, when coupled with an integer relation algorithm, often produced interesting closed-form evaluations (although the resulting formulas must still be proved analytically).

In this paper, we present a simple scheme for accelerating one key step of these computations, which involves the computation of high-order, high-precision derivatives. In initial tests, we find this scheme is comparable to commonly-used symbolic methods, and in fact is much faster on difficult, high-degree problems of research interest. As a major plus, it leads to a self-contained numerical approach for the high-precision computation of Euler sums that does not require symbolic processing.

1 Introduction

An *Euler sum* (also termed *Euler-Zagier sum* or merely *harmonic sum*) is an infinite series involving the harmonic function $H_k = H(k) = 1 + 1/2 + 1/3 + \cdots + 1/k$. Such sums arise in mathematical physics, in the study of the Riemann hypothesis and in numerous other contexts. They have been studied since Euler, and more recently in [1, 2, 4, 9, 10, 11, 13, 17, 18]. One notable feature of these sums is that many have surprisingly elegant analytic evaluations, for example (here $\zeta(\cdot)$ is the Riemannn zeta function):

$$\sum_{k=1}^{\infty} \frac{H_k}{k^3} = \frac{5}{4}\zeta(4) = \frac{1}{72}\pi^4$$
$$\sum_{k=1}^{\infty} \frac{H_k}{(k+1)^5} = \frac{1}{4}\left(3\zeta(6) - 2\zeta(3)^2\right) = \frac{1}{1260}\left(\pi^6 - 630\zeta(3)^2\right) \tag{1}$$

In one earlier study [8], we presented results for mixed-denominator Euler sums of the form

$$\sum_{k=1}^{\infty} \frac{H_k^m}{k^{n_0}(k+1)^{n_1}(k+2)^{n_2}\cdots(k+t)^{n_t}}, \tag{2}$$

for nonnegative integers m and (n_i), with $m \geq 1$ and $n_0 + n_1 + \cdots + n_t \geq 2$. In a more recent study, we presented a scheme based on complex residues that leads to closed-form evaluations for a wide class of Euler sums, including the following: For integers $m, n \geq 1, \gcd(m,n) = 1, p \geq 2$, and with $\psi(k,x) = \psi^{(k)}(x)$ denoting the polygamma function,

$$\sum_{k=1}^{\infty} \frac{H_k}{(mk+n)^p} = \frac{(-1)^{p-1}}{2m^p(p-1)!} \left(\psi(p, n/m) - 2\gamma\psi(p-1, n/m) - \sum_{k=0}^{p-1} \binom{p-1}{k} \psi(k, n/m)\psi(p-1-k, n/m) \right).$$

In spite of these results, it is clear that a large class of Euler sums, as well as countless other infinite series of this general type, still remain unexplored. For example, no closed-form evaluations are currently known for these general classes of sums:

$$\sum_{k=1}^{\infty} \frac{H_k^2}{(mk+n)^p}, \quad \sum_{k=1}^{\infty} \frac{(-1)^k H_k}{(mk+n)^p}, \quad \sum_{k=1}^{\infty} \frac{H_{2k}}{(mk+n)^p}, \tag{3}$$

for positive integers $m \geq 3, \gcd(m,n) = 1$ and $p \geq 2$ (except for a few isolated instances).

2 Computing Euler sums to high precision

In our earlier research on Euler sums, the present authors have relied heavily on high-precision numerical computations, which, when coupled with an integer relation algorithm [7], often produce interesting closed-form evaluations. Formula (3) above, among many others, was found in this way. Such experimental discoveries must still be proved analytically, but numerical discovery is an important first step.

Here we briefly review the computational techniques that we have employed in these investigations, condensed and adapted from [8]. The key tool here is the Euler-Maclaurin summation formula [3, pg. 285], which approximates a summation as an integral with high-order corrections (here $f(t)$ is assumed to have $(2s+2)$-th order derivatives on $[a, b]$):

$$\sum_{j=a}^{b} f(j) = \int_a^b f(t)\, \mathrm{d}t \; + \; \frac{1}{2}\left(f(a) + f(b)\right) \; + \; \sum_{j=1}^{s} \frac{B_{2j}\left(D^{2j-1}f(b) - D^{2j-1}f(a)\right)}{(2j)!} \; + \; R_s(a,b), \tag{4}$$

where B_k is the k-th Bernoulli number [12], $D^k f(a)$ is the k-th derivative of $f(t)$ evaluated at $t = a$, and

$$R_s(a,b) = \frac{-1}{(2s+2)!} \int_a^b B_{2s+2}(t - [t]) D^{2s+2} f(t)\, \mathrm{d}t, \tag{5}$$

where $[\cdot]$ denotes greatest integer and $B_k(\cdot)$ is the k-th Bernoulli polynomial [12] (note $B_k = B_k(0)$).

Applying the Euler-Maclaurin summation formula to the harmonic function $H(t) = \sum_{j=1}^{t} 1/j$ yields

$$H(t) = \gamma + \log(t) + \frac{1}{2t} + \sum_{j=1}^{s} \frac{B_{2j}}{2jt^{2j}} + R_s(t), \tag{6}$$

where $\gamma = 0.5772156649\ldots$ is Euler's constant and $|R_s(t)| \leq |B_{2s+2}|/((2s+2)t^{2s+2})$; see [4] for full details. In the computations for the present study, we set $s = 18$, so that $H(t)$ is approximated by

$$\begin{aligned}\hat{H}(t) = \gamma + \log(t) + \frac{1}{2t} - \frac{1}{12t^2} + \frac{1}{120t^4} - \frac{1}{252t^6} + \frac{1}{240t^8} - \frac{1}{132t^{10}} + \frac{691}{32760t^{12}} - \frac{1}{12t^{14}} \\ + \frac{3617}{8160t^{16}} - \frac{43867}{14364t^{18}} + \frac{174611}{6600t^{20}} - \frac{77683}{276t^{22}} + \frac{236364091}{65520t^{24}} - \frac{657931}{12t^{26}} + \frac{3392780147}{3480t^{28}} \\ - \frac{1723168255201}{85932t^{30}} + \frac{7709321041217}{16320t^{32}} - \frac{151628697551}{12t^{34}} + \frac{26315271553053477373}{69090840t^{36}},\end{aligned} \tag{7}$$

which approximates $H(t)$ to within roughly t^{-38} for large t. The expression (7) can be obtained using Wolfram Mathematica with the command `Series[HarmonicNumber[t],{t,Infinity,36}]`.

Consider now, for nonnegative integers m, n, p, q, with $q \geq 2$, the Euler sums (for example)

$$E(m,n,p,q) = \sum_{k=1}^{\infty} \frac{H(k)^m}{(nk+p)^q}. \tag{8}$$

Let $\hat{G}(t) = \hat{H}(t)^m/(nt+p)^q$, where $\hat{H}(t)$ is the complicated approximation given in (7). Using the Euler-Maclaurin summation formula (4) once again, one can write

$$\begin{aligned} E(m,n,p,q) &= \sum_{j=1}^{k} \frac{H(j)^m}{(nj+p)^q} + \sum_{j=k+1}^{\infty} \frac{H(j)^m}{(nj+p)^q} \approx \sum_{j=1}^{k} \frac{H(j)^m}{(nj+p)^q} + \sum_{j=k+1}^{\infty} \hat{G}(j) \\ &\approx \sum_{j=1}^{k} \frac{H(j)^m}{(nj+p)^q} + \int_{k+1}^{\infty} \hat{G}(t)\,\mathrm{d}t + \frac{1}{2}\hat{G}(k+1) - \sum_{j=1}^{s} \frac{B_{2j} D^{2j-1} \hat{G}(k+1)}{(2j)!}, \end{aligned} \tag{9}$$

where $s = 18$, which is accurate to within roughly k^{-38}. For the tests described below, we set $k = 10^8$, so the approximation in the second line of (9) is correct to within roughly 10^{-300}.

Formula (9) leads immediately to a four-step algorithm for high-precision evaluation of Euler sums of the class (8) (note that the outline here is applicable to a wide range of infinite series):

1. Select d (digits desired for final answer), k (number of terms in first summation), and s (number of terms in final summation), selected so that $(2s+2)\log_{10}(k) > d$. Then use arbitrary precision software to explicitly evaluate the sum $S_1 = \sum_{j=1}^{k} H(j)^m/(nj+p)^q$, using at least d-digit precision.

2. Use an arbitrary precision quadrature facility to numerically evaluate the integral $S_2 = \int_{k+1}^{\infty} \hat{G}(t)\,\mathrm{d}t$ to d-digit accuracy. We have employed the exp-sinh algorithm, a variation of the tanh-sinh scheme described in [6, 5].

3. Use arbitrary precision software to evaluate the term $S_3 = \frac{1}{2}\hat{G}(k+1)$ to d-digit accuracy.

4. Use symbolic math software, such as Mathematica or Maple, to symbolically evaluate the high-degree derivatives in $S_4 = \sum_{j=1}^{s} B_{2j} D^{2j-1} \hat{G}(k+1)/(2j)!$, then numerically evaluate these derivatives to at least d-digit accuracy, then multiply by Bernoulli coefficients and sum as shown to produce S_4. The set of even Bernoulli coefficients required in this set can be computed by a scheme described in [5].

The final result is then $S = S_1 + S_2 + S_3 - S_4$. This sketch omits numerous details; also, the precision levels given above may need to be adjusted for certain problems.

In our implementations, we have most often employed software written and maintained by one of us for Steps 1 through 3 [5], in part because in some cases we have not been able to produce reliable fully accurate high-precision values for Step 2 (the integral) using Mathematica. In contrast, our software, which as noted above employs the exp-sinh algorithm, unfailingly produces fully accurate results.

Step 4, however, has required symbolic math software, as noted above, so we have usually employed Mathematica for this step. Although in most cases Mathematica performs the Step 4 evaluations quite rapidly, in some cases, especially for Euler sums involving powers of $H(k)$, it can require up to 10 minutes run time. Also, needing to copy results between Mathematica and our custom software is awkward at best and introduces the possibility of human error.

This raises the question: Is there some purely numerical scheme to evaluate Step 4, one that does not require symbolic processing?

3 High-precision evaluation of high-order derivatives

We describe here a scheme to numerically evaluate high-order derivatives of complicated formulas such as (7) above, which then permits Step 4 in the above algorithm to be rapidly evaluated to high precision, all without any need for symbolic processing. The approach we present here is embarrassingly simple in concept, although at first glance it may appear absurdly impractical.

The basis for this calculation is the following well-known formula for the n-th derivative [15]:

$$D^n[f(x)] = \lim_{h\to 0} \frac{1}{h^n} \sum_{k=0}^{n} (-1)^{n-k} \binom{n}{k} f(x+kh). \tag{10}$$

A d-digit approximation can be written as

$$D^n[f(x)] \approx \frac{1}{h^n} \sum_{k=0}^{n} (-1)^{n-k} \binom{n}{k} f(x+kh), \tag{11}$$

where h is chosen to be less than $1/10^d$. However, lurking behind this formula is what at first appears to be an insuperable numerical difficulty: the function values $f(x+kh)$ must be evaluated, and the computation performed, using at least $(n+1)d$-digit precision, or else the result will be completely inaccurate.

For example, in the application mentioned above, where we wish to evaluate

$$S_4 = \sum_{j=1}^{s} \frac{B_{2j} D^{2j-1} \hat{G}(k+1)}{(2j)!}, \tag{12}$$

note that with $k = 10^8$ and $s = 18$ as above, one must compute derivatives of the complicated $\hat{G}$ function up to order 35. Thus if one requires a result accurate to 300 digits, then one must employ least $2sd$-digit (10,800-digit) precision. In practice even higher precision is required to fully compensate for round-off error (our tests employed $2(s+2)d = 12,000$-digit arithmetic).

4 Performance results

So how does this admittedly absurd-looking scheme perform? Quite well, actually.

For the runs described in this note, we implemented formulas (11) and (12) basically as written above. One economization is possible by noting that many function evaluations are repeated with the same argument. Thus it is best to first compute the set $\{\hat{G}(k+1+jh), 0 \le j \le 2s\}$ to $2(s+2)d$-digit precision, then access this data as needed. This reduces the run time by roughly a factor of ten. Also, extra-high precision need only be employed in (11); modest (d-digit) precision suffices for (12).

Table 1 below shows some performance results using this scheme on a set of eight problems. Each of these series has a known analytic evaluation (items 1–4 are from [9]; items 5–8 are from [8]):

$$\begin{aligned}
1: \sum_{k=1}^{\infty} \frac{1}{(2k+1)^3} &= \frac{1}{8}\left(-8 + 7\zeta(3)\right) \\
2: \sum_{k=1}^{\infty} \frac{1}{(5k+4)^3} &= -\frac{1}{250}\psi(2, 9/5) \\
3: \sum_{k=1}^{\infty} \frac{H_k}{(2k+1)^4} &= \frac{1}{192}\left(-\psi(4,1/2) + 2\gamma\psi(3,1/2) + 2\psi(0,1/2)\psi(3,1/2) + 6\psi(1,1/2)\psi(2,1/2)\right) \\
4: \sum_{k=1}^{\infty} \frac{H_k}{(3k+1)^6} &= \frac{1}{174960}\left(-\psi(6,1/3) + 2\gamma\psi(5,1/3) + 2\psi(0,1/3)\psi(5,1/3) + 10\psi(1,1/3)\psi(4,1/3)\right. \\
&\quad \left. +20\psi(2,1/3)\psi(3,1/3)\right)
\end{aligned} \tag{13}$$

$$5 : \sum_{k=1}^{\infty} \frac{H(k)}{(k+2)^4} = -4 + \zeta(2) + \zeta(3) + \zeta(4) + 2\zeta(5) - \zeta(2)\zeta(3)$$

$$6 : \sum_{k=1}^{\infty} \frac{H(k)^2}{(k+1)^4} = \frac{1}{24}\left(37\zeta(6) - 24\zeta(3)^2\right)$$

$$7 : \sum_{k=1}^{\infty} \frac{H(k)^4}{(k+2)^3} = \frac{1}{8}\big(-120 - 112\zeta(2) - 160\zeta(3) + 124\zeta(4) + 168\zeta(5) + 80\zeta(2)\zeta(3) -66\zeta(6) + 64\zeta(3)^2 - 109\zeta(7) - 40\zeta(2)\zeta(5) + 148\zeta(3)\zeta(4)\big)$$

$$8 : \sum_{k=1}^{\infty} \frac{H(k)^6}{(k+2)^3} = \frac{1}{48}\big(1344 + 3312\zeta(2) + 14832\zeta(3) + 33120\zeta(4) + 20592\zeta(5) + 5184\zeta(2)\zeta(3) -24396\zeta(6) - 3024\zeta(3)^2 - 23580\zeta(7) - 4608\zeta(2)\zeta(5) - 22824\zeta(3)\zeta(4) -65621\zeta(8) - 17640\zeta(2)\zeta(3)^2 + 72432\zeta(3)\zeta(5) + 15480M(2,6) + 12292\zeta(9) -25164\zeta(3)\zeta(6) + 11664\zeta(4)\zeta(5) + 3906\zeta(2)\zeta(7) - 1072\zeta(3)^3\big), \quad (14)$$

where in Problem 8, $M(2,6) = \sum_{k=1}^{\infty} H(k)^2/k^6$. These problems were each addressed by these three software systems:

1. Wolfram Mathematica, with symbolic processing for Step 4.
2. Wolfram Mathematica, with the above numeric scheme for Step 4.
3. MPFUN20-MPFR (employs the GMP and MPFR packages [14] for lower-level operations), with the numeric scheme for Step 4.

	Mathematica (symbolic Step 4)		Mathematica (numeric Step 4)		MPFUN20-MPFR	
Problem	Steps 1–3	Step 4	Steps 1–3	Step 4	Steps 1–3	Step 4
1	326.68	0.00	Same as	0.09	56.71	0.01
2	339.75	0.00	column 2	0.09	57.02	0.01
3	417.65	0.10		0.09	86.17	0.10
4	430.53	0.10		0.09	97.30	0.10
5	421.29	0.10		0.09	86.50	0.10
6	435.65	0.33		0.10	97.90	0.10
7	443.03	6.19		0.15	108.01	0.10
8	460.79	29.95		0.11	118.80	0.10

Table 1: CPU times for test runs in seconds (Apple M5 processor)

In looking at these results, it is immediately evident that the numeric scheme is not prohibitively expensive, as one might expect. Instead, on each platform the CPU time for this step is only a tiny fraction of the total run time, which is dominated by the cost of Step 1. The Mathematica results are instructive, as they permit a direct comparison of the symbolic and numeric schemes for Step 4. Although the numeric method is slower than the symbolic method for the first two relatively easy problems, it is many times faster on the two most demanding problems (problems 7–8). The MPFUN20-MPFR results are illustrative of the performance of this scheme on a very fast custom software platform. Here, the numeric scheme in Step 4 never exceeds 0.10 seconds, which is typically only about 0.1% of the total.

In short, the numeric scheme we have described above for computing high-order derivatives appears to be both efficient and practical, in spite of its daunting precision requirement. It permits computations of Euler sums (and virtually any infinite series) to be performed entirely using a noncommercial arbitrary precision package, without the need of symbolic computation.

References

[1] J. Ablinger, "A computer algebra toolbox for harmonic sums related to particle physics," Johannes Kepler University, 2009, available at https://inspirehep.net/literature/875721.

[2] J. Ablinger, J. Bluemlein and C. Schneider, "Harmonic sums and polylogarithms generated by cyclotomic polynomials," *Journal of Mathematical Physics*, **52** (2011), 102301.

[3] K. E. Atkinson, *An Introduction to Numerical Analysis*, 2nd ed., John Wiley and Sons, New York, 1990.

[4] D.H. Bailey, J.M. Borwein and R. Girgensohn, "Experimental evaluation of Euler sums," *Experimental Mathematics*, **3** (1994), 17–30.

[5] D.H. Bailey, "MPFUN2020: A new thread-safe arbitrary precision package," 2024, https://www.davidhbailey.com/dhbpapers/mpfun2020.pdf.

[6] D.H. Bailey, X.S. Li and K. Jeyabalan, 2005. A comparison of three high-precision quadrature schemes, *Experimental Mathematics*, **14**, 317–329.

[7] D.H. Bailey, "The two-level multipair PSLQ algorithm," 2025, https://www.davidhbailey.com/dhbpapers/pslqm2-alg.pdf.

[8] D.H. Bailey and R.C. McPhedran, "New results for Euler sums," arXiv:2311.06294, Jul 2025, https://arxiv.org/abs/2311.06294.

[9] D.H. Bailey, R.C. McPhedran and B. Salvy, "General formulas for a class of Euler sums," 1 Apr 2026, https://arxiv.org/abs/2604.02384.

[10] D. Borwein, J.M. Borwein and R. Girgenshon, "Explicit evaluation of Euler sums," *Proceedings of the Edinburgh Mathematical Society*, **38** (1995), 277–294.

[11] J. Choi and H.M. Srivastava, "Some summation formulas involving harmonic numbers and generalised harmonic numbers," *Mathematical and Computer Modeling*, **54** (2011), 220–2234.

[12] "Digital library of mathematical functions," National Institute of Standards and Technology, 2015, http://dlmf.nist.gov.

[13] P. Flajolet and B. Salvy, "Euler sums and contour integral representations," *Experimental Mathematics*, vol. 7 (1998), 15–35.

[14] "The GNU MPFR library," http://www.mpfr.org.

[15] "Numerical differentiation," *Wikipedia article*, viewed 2 Jun 2026, https://en.wikipedia.org/wiki/Numerical_differentiation.

[16] "Polygamma function," *Wikipedia article*, viewed 2 Jun 2026, https://en.wikipedia.org/wiki/Polygamma_function.

[17] A. Sofo and J. Choi, "Linear harmonic Euler sums of even weight," *Journal of Mathematical Analysis and Applications*, **553** (2026), 129926.

[18] D-Y. Zheng, "Further summation formulae related to generalized harmonic numbers," *Journal of Mathematical Analysis and Applications*, **335** (2007), 692–706.