# Experimental Evaluation of Euler Sums

David H. Bailey, Jonathan M. Borwein and Roland Girgensohn

RNR Technical Report RNR-93-014
June 6, 1994

**Abstract**

In response to a letter from Goldbach, Euler considered sums of the form

$$\sum_{k=1}^{\infty} \left(1 + \frac{1}{2^m} + \cdots + \frac{1}{k^m}\right) (k+1)^{-n}$$

for positive integers $m$ and $n$. Euler was able to give explicit values for certain of these sums in terms of the Riemann zeta function. In a recent companion paper, Euler's results were extended to a significantly larger class of sums of this type, including sums with alternating signs.

This research was facilitated by numerical computations using an algorithm that can determine, with high confidence, whether or not a particular numerical value can be expressed as a rational linear combination of several given constants. The present paper presents the numerical techniques used in these computations and lists many of the experimental results that have been obtained.

D. H. Bailey: NAS Applied Research Branch, NASA Ames Research Center, Moffett Field, CA 94035-1000, USA; dbailey@nas.nasa.gov.

J. M. Borwein: Department of Mathematics and Statistics, Simon Fraser University, Burnaby, BC V5A 1S6, Canada; jborwein@cecm.sfu.ca. Research supported by NSERC and the Shrum Endowment at Simon Fraser University.

R. Girgensohn: Department of Mathematics and Statistics, Simon Fraser University, Burnaby, BC V5A 1S6, Canada; girgen@cecm.sfu.ca. Research supported by a DFG fellowship.

## 1. Introduction

In response to a letter from Goldbach, Euler considered sums of the form

$$\sum_{k=1}^{\infty} \left(1 + \frac{1}{2^m} + \cdots + \frac{1}{k^m}\right) (k+1)^{-n}.$$

Euler was able to give explicit values for certain of these sums in terms of the Riemann zeta function. For example, Euler found an explicit formula for the case $m = 1$, $n \geq 2$. Little has been done on this problem in the intervening years (see [5] for some references).

In April 1993, Enrico Au-Yeung, an undergraduate at the University of Waterloo, brought to the attention of one of us the curious fact that

$$\begin{aligned} \sum_{k=1}^{\infty} \left(1 + \frac{1}{2} + \cdots + \frac{1}{k}\right)^2 k^{-2} &= 4.59987\cdots \\ &\approx \frac{17}{4}\zeta(4) = \frac{17\pi^4}{360} \end{aligned}$$

based on a computation to 500,000 terms. This author's reaction was to compute the value of this constant to a higher level of precision in order to dispel this conjecture. Surprisingly, a computation to 30 and later to 100 decimal digits still affirmed it. (Unknown to us at that time, De Doelder had proved a related result in 1991 [11] from which the above identity follows.)

Intrigued by this empirical result, we computed numerical values for several of these and similar sums, which we have termed Euler sums. We then analyzed these values by a technique we will present below that permits one to determine, with a high level of confidence, whether a numerical value can be expressed as a rational linear combination of several given constants. These efforts produced even more empirical evaluations, suggesting broad patterns and general conjectures. Ultimately proofs were found for many of these experimental results.

We will consider the following classes of Euler sums:

$$\begin{aligned} s_{\mathrm{h}}(m,n) &= \sum_{k=1}^{\infty} \left(1 + \frac{1}{2} + \cdots + \frac{1}{k}\right)^m (k+1)^{-n} \qquad m \geq 1,\ n \geq 2, \\ s_{\mathrm{a}}(m,n) &= \sum_{k=1}^{\infty} \left(1 - \frac{1}{2} + \cdots + \frac{(-1)^{k+1}}{k}\right)^m (k+1)^{-n} \quad m \geq 1,\ n \geq 2, \end{aligned}$$

$$
\begin{aligned}
a_{\mathrm{h}}(m,n) &= \sum_{k=1}^{\infty}\left(1+\frac{1}{2}+\cdots+\frac{1}{k}\right)^{m}(-1)^{k+1}(k+1)^{-n} && m\ge 1,\ n\ge 1,\\
a_{\mathrm{a}}(m,n) &= \sum_{k=1}^{\infty}\left(1-\frac{1}{2}+\cdots+\frac{(-1)^{k+1}}{k}\right)^{m}(-1)^{k+1}(k+1)^{-n} && m\ge 1,\ n\ge 1,\\
\sigma_{\mathrm{h}}(m,n) &= \sum_{k=1}^{\infty}\left(1+\frac{1}{2^m}+\cdots+\frac{1}{k^m}\right)(k+1)^{-n} && m\ge 1,\ n\ge 2,\\
\sigma_{\mathrm{a}}(m,n) &= \sum_{k=1}^{\infty}\left(1-\frac{1}{2^m}+\cdots+\frac{(-1)^{k+1}}{k^m}\right)(k+1)^{-n} && m\ge 1,\ n\ge 2,\\
\alpha_{\mathrm{h}}(m,n) &= \sum_{k=1}^{\infty}\left(1+\frac{1}{2^m}+\cdots+\frac{1}{k^m}\right)(-1)^{k+1}(k+1)^{-n} && m\ge 1,\ n\ge 1,\\
\alpha_{\mathrm{a}}(m,n) &= \sum_{k=1}^{\infty}\left(1-\frac{1}{2^m}+\cdots+\frac{(-1)^{k+1}}{k^m}\right)(-1)^{k+1}(k+1)^{-n} && m\ge 1,\ n\ge 1.
\end{aligned}
$$

Explicit evaluations of some of the constants in these classes are presented with proofs in [6] and [7]. Table 1 contains a summary of these results (these include some facts already known to Euler). Here $\zeta(t)=\sum_{k=1}^{\infty}k^{-t}$ is the Riemann zeta function. Results for alternating sums are also given in [7].

Variants of the sums defined above can be evaluated by using these results. Note for example that for all $m\ge 1$, $n\ge 2$ one can write

$$
\begin{aligned}
\sum_{k=1}^{\infty}\left(1+\frac{1}{2^m}+\cdots+\frac{1}{k^m}\right)k^{-n} &= \sum_{k=0}^{\infty}\left(1+\frac{1}{2^m}+\cdots+\frac{1}{(k+1)^m}\right)(k+1)^{-n}\\
&= \sum_{k=1}^{\infty}\left(1+\frac{1}{2^m}+\cdots+\frac{1}{k^m}\right)(k+1)^{-n}+\sum_{k=1}^{\infty}k^{-m-n}\\
&= \sigma_{\mathrm{h}}(m,n)+\zeta(m+n).
\end{aligned}
$$

Similarly, let $h_k=1+1/2+\cdots+1/k$, and define $h_0=0$. Then for all $n\ge 2$ one can write

$$
\begin{aligned}
\sum_{k=1}^{\infty}\left(1+\frac{1}{2}+\cdots+\frac{1}{k}\right)^{2}k^{-n} &= \sum_{k=0}^{\infty}h_{k+1}^{2}(k+1)^{-n}\\
&= \sum_{k=0}^{\infty}\left(h_k+\frac{1}{k+1}\right)^{2}(k+1)^{-n}\\
&= \sum_{k=0}^{\infty}h_k^{2}(k+1)^{-n}\ +\ 2\sum_{k=0}^{\infty}h_k(k+1)^{-n-1}\\
&\quad +\ \sum_{k=0}^{\infty}(k+1)^{-n-2}\\
&= s_{\mathrm{h}}(2,n)\ +\ 2s_{\mathrm{h}}(1,n+1)\ +\zeta(n+2).
\end{aligned}
$$

$$
\begin{aligned}
s_{\rm h}(2,2) &= \frac{3}{2}\zeta(4)+\frac{1}{2}\zeta^2(2) \;=\; \frac{11\pi^4}{360},\\
s_{\rm h}(2,4) &= \frac{2}{3}\zeta(6)-\frac{1}{3}\zeta(2)\zeta(4)+\frac{1}{3}\zeta^3(2)-\zeta^2(3) \;=\; \frac{37\pi^6}{22680}-\zeta^2(3),\\
\sigma_{\rm h}(2,2) &= \frac{1}{2}\zeta^2(2)-\frac{1}{2}\zeta(4) \;=\; \frac{\pi^4}{120},\\
\sigma_{\rm h}(2,4) &= -6\zeta(6)+\frac{8}{3}\zeta(2)\zeta(4)+\zeta^2(3) \;=\; \zeta^2(3)-\frac{4\pi^6}{2835},\\
s_{\rm h}(1,n) &= \sigma_{\rm h}(1,n) \;=\; \frac{n\zeta(n+1)}{2}-\frac{1}{2}\sum_{k=1}^{n-2}\zeta(n-k)\zeta(k+1),\\
s_{\rm h}(2,n) &= \frac{n(n+1)}{3}\zeta(n+2)+\zeta(2)\zeta(n)-\frac{n}{2}\sum_{k=0}^{n-2}\zeta(n-k)\zeta(k+2)\\
&\quad +\frac{1}{3}\sum_{k=2}^{n-2}\zeta(n-k)\sum_{j=1}^{k-1}\zeta(j+1)\zeta(k+1-j) \;+\; \sigma_{\rm h}(2,n),\\
\sigma_{\rm h}(2,2n-1) &= -\frac{2n^2+n+1}{2}\zeta(2n+1) \;+\; \zeta(2)\zeta(2n-1)\\
&\quad +\sum_{k=1}^{n-1}2k\zeta(2k+1)\zeta(2n-2k),\\
s_{\rm h}(2,2n-1) &= \frac{2n^2-7n-3}{6}\zeta(2n+1) \;+\; \zeta(2)\zeta(2n-1)\\
&\quad -\frac{1}{2}\sum_{k=1}^{n-2}(2k-1)\zeta(2n-1-2k)\zeta(2k+2)\\
&\quad +\frac{1}{3}\sum_{k=1}^{n-2}\zeta(2k+1)\sum_{j=1}^{n-2-k}\zeta(2j+1)\zeta(2n-1-2k-2j),
\end{aligned}
$$

for $m+n$ odd:

$$
\begin{aligned}
\sigma_{\rm h}(m,n) &= \frac{1}{2}\left[\binom{m+n}{m}-1\right]\zeta(m+n)+\zeta(m)\zeta(n)\\
&\quad -\sum_{j=1}^{m+n}\left[\binom{2j-2}{m-1}+\binom{2j-2}{n-1}\right]\zeta(2j-1)\zeta(m+n-2j+1)\\
&\quad \text{if } m \text{ is odd},\\
\sigma_{\rm h}(m,n) &= -\frac{1}{2}\left[\binom{m+n}{m}+1\right]\zeta(m+n)\\
&\quad +\sum_{j=1}^{m+n}\left[\binom{2j-2}{m-1}+\binom{2j-2}{n-1}\right]\zeta(2j-1)\zeta(m+n-2j+1)\\
&\quad \text{if } m \text{ is even}.
\end{aligned}
$$

Table 1: Explicit Evaluations of Euler Sums

## 2. Numerical Techniques

It is not easy to naïvely compute numerical values of any of these Euler sums to high precision. Straightforward evaluation using the defining formulas, to some upper limit feasible on present-day computers, yields only about eight digits accuracy. Because the integer relation detection algorithm described in section 4 requires much higher precision to obtain reliable results, more advanced techniques must be employed.

We present here a method that is reasonably straightforward and generally applicable to all Euler sums discussed in this paper. This scheme involves the compound application of the Euler-Maclaurin summation formula (see [1, p. 806], [2, p. 289] and [16, p. 108]), which can be stated as follows. Suppose $f(t)$ has at least $2p+2$ continuous derivatives on $(a,b)$. Let $D$ be the differentiation operator, let $B_k$ denote the $k$-th Bernoulli number, and let $B_k(\cdot)$ denote the $k$-th Bernoulli polynomial. Then

$$\begin{aligned}\sum_{j=a}^{b} f(j) &= \int_a^b f(t)\,dt \;+\; \frac{1}{2}[f(a)+f(b)] \\ &\quad + \sum_{j=1}^{p} \frac{B_{2j}}{(2j)!}[D^{2j-1}f(b) - D^{2j-1}f(a)] \;+\; R_p(a,b). \end{aligned} \tag{1}$$

where the remainder $R_p(a,b)$ is given [2, p. 289] by

$$R_p(a,b) \;=\; \frac{-1}{(2p+2)!}\int_a^b B_{2p+2}(t-[t])D^{2p+2}f(t)\,dt.$$

We will start by demonstrating a method for computing $s_{\rm h}(m,n)$. Let $h(k)=\sum_{j=1}^{k} 1/j$ and $f(t)=1/t$. By the Euler-Maclaurin summation formula,

$$h(k) \;=\; \ln k \;+\; \frac{1}{2} \;+\; \frac{1}{2k} \;+\; \sum_{j=1}^{p} \frac{B_{2j}}{2j\,k^{2j}} \;-\; \sum_{j=1}^{p} \frac{B_{2j}}{2j} \;+\; R_p(1,k).$$

Since $|B_{2k}(t)| \le |B_{2k}|$ for all $k$ and for $|t| \le 1$ (see [1, p. 805]), it follows that the remainder $R_p(1,k)$ has a well-defined limit $R_p(1,\infty)$ as $k$ approaches infinity. Now since Euler's constant $\gamma = \lim_{k\to\infty}[h(k) - \ln k]$, it follows that

$$h(k) \;=\; \gamma \;+\; \ln k \;+\; \frac{1}{2k} \;+\; \sum_{j=1}^{p} \frac{B_{2j}}{2j\,k^{2j}} \;-\; R_p(k,\infty). \tag{2}$$

We have

$$\begin{aligned} |R_p(k,\infty)| &= \left| \int_k^\infty B_{2p+2}(t-[t])t^{-2p-3}\,dt \right| \\ &\le \frac{|B_{2p+2}|}{(2p+2)k^{2p+2}}, \end{aligned}$$

so that the remainder in (2) is no greater than the first term omitted in the summation. We can then write, for example,

$$\begin{aligned} h(k) &= \gamma + \ln k + \frac{1}{2k} - \frac{1}{12k^2} + \frac{1}{120k^4} - \frac{1}{252k^6} + \frac{1}{240k^8} \\ &\quad - \frac{1}{132k^{10}} + \frac{691}{32760k^{12}} - \frac{1}{12k^{14}} + \frac{3617}{8160k^{16}} + O(k^{-18}). \end{aligned} \tag{3}$$

We will use $\bar{h}(k)$ to denote this particular approximation (i.e., (3) without the error term). It is an unfortunate fact that $\bar{h}$ cannot be extended to a valid infinite series. The difficulty is that for any fixed $k$, the Bernoulli coefficients eventually become very large and the series diverges. On the other hand, it is clear that for any fixed number of terms, approximations such as $\bar{h}$ become ever more accurate as $k$ increases to infinity.

Now let us consider the sum

$$s_{\rm h}(m,n) = \sum_{k=1}^\infty \left(1+\frac{1}{2}+\cdots+\frac{1}{k}\right)^m (k+1)^{-n}.$$

Let $c$ be a large integer, and let $g(t) = \bar{h}^m(t)(t+1)^{-n}$. Applying the Euler-Maclaurin summation formula (1) again, we can write

$$\begin{aligned} s_{\rm h}(m,n) &= \sum_{k=1}^c \left(1+\frac{1}{2}+\cdots+\frac{1}{k}\right)^m (k+1)^{-n} \\ &\quad + \sum_{k=c+1}^\infty \left(1+\frac{1}{2}+\cdots+\frac{1}{k}\right)^m (k+1)^{-n} \\ &= \sum_{k=1}^c h^m(k)(k+1)^{-n} + \int_{c+1}^\infty g(t)\,dt + \frac{1}{2}g(c+1) \\ &\quad - \sum_{k=1}^9 \frac{B_{2k}}{(2k)!} D^{2k-1}g(c+1) + O(c^{-18}). \end{aligned} \tag{4}$$

This formula suggests the following computational scheme. First, explicitly evaluate the sum $\sum_{k=1}^c h^m(k)(k+1)^{-n}$ for $c = 10^8$, using a numeric working precision of 150 digits.

Secondly, perform the symbolic integration and differentiation steps indicated in formula (4). Finally, evaluate the resulting expression, again using a working precision of 150 digits. The final result should be equal to $s_{\rm h}(m,n)$ to approximately 135 significant digits.

The difficulty and cost of performing the symbolic integration and differentiation operations indicated in (4) can be greatly reduced by approximating $g(t)$ as follows: first, expand $\bar{h}^m(t)$, the numerator of $g(t)$, into a sum of individual terms; next, write $(1+t)^{-n}$ as $t^{-n}(1+1/t)^{-n}$; next, expand $(1+1/t)^{-n}$ using the binomial theorem to 18 terms; next, multiply together the resulting numerator and denominator expressions; finally, omit all terms whose exponent of $1/t$ is greater than 18. The result is a linear sum of terms of the form $t^{-p}\ln^q(t)$ for modest-sized integers $p$ and $q$.

Of course, even more accurate results can be obtained by utilizing more terms in the Euler-Maclaurin expansions, although the cost of the required symbolic manipulation correspondingly increases. Determining the optimal balance between the numeric and symbolic calculations, and determining the number of Euler-Maclaurin terms (at both steps) required for various levels of precision, are interesting problems in their own right. However, we found that only minor tuning of the above scheme, based on simple timing and accuracy experiments, sufficed for the cases we studied.

For Euler sums with alternating signs, the scheme is a bit more complicated. Consider the sum $s_{\rm a}(m,n)$. In this case, we can write

$$\begin{aligned}
s_{\rm a}(m,n) &= \sum_{k=1}^{\infty}\left(1-\frac{1}{2}+\cdots+\frac{(-1)^{k+1}}{k}\right)^m (k+1)^{-n} \\
&= \sum_{k=1}^{\infty}\left(\sum_{j=1}^{k}\frac{1}{2j-1}-\sum_{j=1}^{k-1}\frac{1}{2j}\right)^m\frac{1}{(2k)^n} \\
&\quad+\sum_{k=1}^{\infty}\left(\sum_{j=1}^{k}\frac{1}{2j-1}-\sum_{j=1}^{k}\frac{1}{2j}\right)^m\frac{1}{(2k+1)^n} \\
&= \sum_{k=1}^{\infty}\left(r_k+\frac{1}{2k}\right)^m\frac{1}{(2k)^n}+\sum_{k=1}^{\infty}\frac{r_k^m}{(2k+1)^n}
\end{aligned}$$

where

$$r_k = \sum_{j=1}^{k}\frac{1}{2j-1}-\sum_{j=1}^{k}\frac{1}{2j} = \sum_{j=1}^{k}\frac{1}{2j(2j-1)}.$$

The Euler-Maclaurin formula (1) can then be applied first to obtain a highly accurate approximation to $r_k$, and then to evaluate the two remaining outer summations.

Another approach for alternating Euler sums is to apply the Boole summation formula (see [9]), which deals specifically with alternating sums.

### 3. Software and Hardware Technology

We have performed many computations of this type. The integration and differentiation operations required in (4) can be handled using a symbolic mathematics package, such as Maple [10] or Mathematica [20]. The explicit summation of the first $c$ terms, as indicated in (4), could be performed by utilizing the multiple precision facility in the Maple or Mathematica packages. However, it was found that the MPFUN multiple precision package and translator developed by one of us [3] was significantly faster for this purpose.

Whatever software is used, this explicit summation is a very expensive operation. For example, the evaluation of $s_{\rm h}(3,4)$ to $10^8$ terms, using the MPFUN package with 150-digit precision arithmetic, requires 20 hours on a "Crimson" workstation manufactured by Silicon Graphics, Inc. Thus while such runs can be made, clearly this is pressing the limits of current workstation technology.

The MPFUN multiple precision software is available on vector supercomputers, such as those manufactured by Cray Research, Inc. However, for the modest precision levels typical of these problems (i.e., 100 to 200 decimal digits), the resulting vector lengths are too short to yield the high performance these systems are capable of. On the other hand, multiple precision calculations of this type are well suited for RISC processors, because they are well behaved in cache memory systems.

These considerations suggest that, in principle, a highly parallel computer based on RISC processors could be effectively employed for computing these explicit sums. However, at first glance these computations appear not to possess any significant opportunity for parallelism, since evidently both the inner and outer sums must be simultaneously accumulated.

Fortunately, it is possible to compute these explicit sums efficiently on parallel comput-

ers, as in the following example of the $s_{\rm h}(m,n)$ constants:

**Algorithm 1.** *(Parallel Summation) Let $P$ be the number of processor nodes of the parallel system, and let $p$ be the processor node index, $1 \le p \le P$. Assume that the number of terms to be accumulated is $N = KP$. Let $r$ denote a $K$-long multiple precision array separately available in each node. Let $t$ and $q$ denote $P$-long multiple precision arrays stored such that node $p$ holds $t_p$ and $q_p$. Initialize by setting $i := 0$, $Q := 0$ and $S := 0$. Then perform the following steps on each node, using multiple precision arithmetic where appropriate:*

1. *Set $i := i + 1$ and $t_p := 0$.*

2. *For $j := 1$ to $K$: set $r_j := 1/[j + (p-1)K + (i-1)N]$ and $t_p := t_p + r_j$; endfor. Set $r_{K+1} := 1/[1 + pK + (i-1)N]$.*

3. *In node 1, set $q_1 := Q$; in other nodes $p$ set $q_p := t_{p-1}$ and $t_p := t_p + t_{p-1}$. Each node $p > 1$ must receive the value of $t_{p-1}$ from node $p-1$ before updating $q_p$ or $t_p$.*

4. *Set $Q := t_P$ (from node $P$) and $t_p := 0$.*

5. *For $j := 1$ to $K$: set $q_p := q_p + r_j$ and $t_p := t_p + q_p^m r_{j+1}^n$; endfor.*

6. *Compute the global sum $T$ of all $t_p$, and set $S := S + T$.*

Suppose one wishes to accumulate approximately $10^8$ terms, as in many of our computations. One complication is that when $N = 10^8$, the amount of memory required for the $r$ array on each node might not be available on some highly parallel computers. This difficulty can be remedied by setting $N = 10^6$ and then iterating this procedure 100 times. Upon completion of all iterations, $S$ is the required explicit sum to $10^8$ terms.

This algorithm has been implemented on an Intel Paragon parallel computer at NASA Ames Research Center, using the MPFUN multiple precision software. Using 128 nodes and 150-digit precision arithmetic, with $m = 3$, $n = 4$ and $N = 2^{20}$, performing 100 iterations of Algorithm 1 (i.e., $c = 100N = 104,857,600$) requires only 971 seconds, or

9.7 seconds per iteration. This is 110 times faster than the per-iteration timing on a single node of the Paragon, using a straightforward serial algorithm, and 40 times faster than on one processor of a Cray Y-MP, using the MPFUN package tuned for the Cray. It may be possible, by reorganizing the computation, to achieve higher performance on the Cray system; thus caution should be exercised when interpreting this last figure. But these results nonetheless confirm that Algorithm 1, running on a highly parallel RISC supercomputer, is a highly efficient and cost-effective solution to the problem of computing the explicit sums required in (4).

## 4. Integer Relation Detection Algorithms

Let $x = (x_1, x_2, \cdots, x_n)$ be a vector of real numbers. $x$ is said to possess an integer relation if there exist integers $a_i$ not all zero such that $a_1 x_1 + a_2 x_2 + \cdots + a_n x_n = 0$. By an integer relation algorithm, we mean an algorithm that is guaranteed (provided the computer implementation has sufficient numeric precision) to recover the vector of integers $a_i$, if it exists, or to produce bounds within which no integer relation can exist.

The problem of finding integer relations among a set of real numbers was first studied by Euclid, who gave an iterative algorithm which, when applied to two real numbers, either terminates, yielding an exact relation, or produces an infinite sequence of approximate relations. The generalization of this problem for $n > 2$ has been attempted by Euler, Jacobi, Poincare, Minkowski, Perron, Brun, Bernstein, among others. However, none of their algorithms has been proven to work for $n > 3$, and numerous counterexamples have been found.

The first integer relation algorithm with the desired properties mentioned above was discovered by Ferguson and Forcade in 1977 [13]. In the intervening years a number of other integer relation algorithms have been discovered, including a variant of the original algorithm [14], the "LLL" algorithm [18], the "HJLS" algorithm [15] (which is based on the LLL algorithm), and the "PSOS" [4] algorithm.

Recently a new algorithm, known as the "PSLQ" algorithm, was developed by Ferguson and one of us. It appears to combine some of the best features separately possessed by

previous algorithms, including fast run times, numerical stability, numerical efficiency (i.e. successfully recovering a relation when the input is known to only limited precision), and a guaranteed completion in a polynomially bounded number of iterations. We present here a simplified but equivalent version of PSLQ. The proof of the PSLQ algorithm and notes for efficient implementations are given in [12].

**Algorithm 2.** *(PSLQ) Let $x$ be the $n$-long input real vector, and let* nint *denote the nearest integer function (for exact half-integer values, define* nint *to be the integer with greater absolute value). Let $\gamma := \sqrt{4/3}$. Then perform the following:*

*Initialize:*

1. *Set the $n \times n$ matrices $A$ and $B$ to the identity.*

2. *For $k := 1$ to $n$: compute $s_k := \sqrt{\sum_{j=k}^{n} x_j^2}$; endfor. For $k := 1$ to $n$: $y_k := x_k/s_1$; $s_k := s_k/s_1$; endfor.*

3. *Compute the $n \times (n-1)$ matrix $H$ as follows:*

   *For $i := 1$ to $n$: for $j := i+1$ to $n-1$: set $H_{ij} := 0$; endfor; if $i \leq n-1$ then set $H_{ii} := s_{i+1}/s_i$; for $j := 1$ to $i-1$: set $H_{ij} := -y_i y_j/(s_j s_{j+1})$; endfor; endfor.*

4. *Perform full reduction on $H$, simultaneously updating $y$, $A$ and $B$:*

   *For $i := 2$ to $n$: for $j := i-1$ to $1$ step $-1$: $t := \mathrm{nint}(H_{ij}/H_{jj})$; $y_j := y_j + t y_i$; for $k := 1$ to $j$: $H_{ik} := H_{ik} - t H_{jk}$; endfor; for $k := 1$ to $n$: $A_{ik} := A_{ik} - t A_{jk}$, $B_{kj} := B_{kj} + t B_{ki}$; endfor; endfor; endfor.*

*Repeat until precision is exhausted or a relation has been detected:*

1. *Select $m$ such that $\gamma^i |H_{ii}|$ is maximal when $i = m$.*

2. *Perform block reduction on $H$, simultaneously updating $y$, $A$ and $B$:*

   *For $i := m+1$ to $n$: for $j := \min(i-1, m+1)$ to $1$ step $-1$: $t := \mathrm{nint}(H_{ij}/H_{jj})$; $y_j := y_j + t y_i$; for $k := 1$ to $j$: $H_{ik} := H_{ik} - t H_{jk}$; ; endfor; for $k := 1$ to $n$: $A_{ik} := A_{ik} - t A_{jk}$, $B_{kj} := B_{kj} + t B_{ki}$; endfor; endfor; endfor.*

3. *Exchange entries $m$ and $m+1$ of $y$, corresponding rows of $A$ and $H$, and corresponding columns of $B$.*

4. *If $m \leq n-2$ then update $H$ as follows:*

   *Set $t_0 := \sqrt{H_{mm}^2 + H_{m,m+1}^2}$, $t_1 := H_{mm}/t_0$ and $t_2 := H_{m,m+1}/t_0$. Then for $i := m$ to $n$: $t_3 := H_{im}$; $t_4 := H_{i,m+1}$; $H_{im} := t_1 t_3 + t_2 t_4$; $H_{i,m+1} := -t_2 t_3 + t_1 t_4$; endfor.*

5. *Norm bound: Compute $M := 1/\max_j |H_j|$, where $H_j$ denotes the $j$-th row of $H$. Then there can exist no relation vector whose Euclidean norm is less than $M$.*

6. *Termination test: If the largest entry of $A$ exceeds the level of numeric precision used, then precision is exhausted. If the smallest entry of the $y$ vector is less than the detection threshold, a relation has been detected and is given in the corresponding column of $B$.*

With regards to the termination criteria in step 6, it sometimes happens that a relation is missed at the point of potential detection because the $y$ entry is not quite as small as the detection threshold being used (the threshold is typically set to the "epsilon" of the precision level). When this happens, however, one will note that the ratio of the smallest and largest $y$ vector entries is suddenly very small, provided sufficient numeric precision is being used.

The actual probability distribution of this ratio is not known for the PSLQ algorithm. Most likely, however, it is closely approximated by a modest-sized constant times this ratio. This is because the entries of the $y$ vector are related to the iterates of the continued fraction algorithm, which are distributed according to the Kuzmin distribution (see [17], p. 346). In a normal computer run using the PSLQ algorithm, prior to the detection of a relation, this ratio is seldom smaller than $10^{-2}$. Thus if this ratio suddenly decreases to a very small value, such as $10^{-20}$, then almost certainly a relation has been detected — one need only adjust the detection threshold for the algorithm to terminate properly and output the relation. When detection does occur, this ratio may be thought of as a "confidence level" of the detection.

In practice, the PSLQ algorithm is very effective in finding relations. For example, in tests described in [12], relations of degree up to 82, with coefficients of size up to $10^{14}$, were successfully detected. As a general rule, one can expect to detect a relation of degree $n$, with coefficients of size $10^m$, provided that the input vector is known to somewhat greater than $mn$ digit precision, and provided that computations are performed using at least this level of numeric precision.

## 5. Applications of the PSLQ Algorithm

There are a number of applications of integer relation detection algorithms in computational mathematics. One application is to analyze whether or not a given constant $\alpha$, whose value can be computed to high precision, is algebraic of some degree $n$ or less. This can be done by first computing the vector $x = (1, \alpha, \alpha^2, \cdots, \alpha^n)$ to high precision and then applying an integer relation algorithm to the vector $x$. If a relation is found, this integer vector is precisely the set of coefficients of a polynomial satisfied by $\alpha$. Even if a relation is not found, the resulting bound means that $\alpha$ cannot possibly be the root of a polynomial of degree $n$, with coefficients of size less than the established bound. Even negative results of this sort are often of interest.

One of us has performed several computations of this type [4]. These computations have established, for example, that if Euler's constant $\gamma$ satisfies an integer polynomial of degree 50 or less, then the Euclidean norm of the coefficients must exceed $7 \times 10^{17}$. Computations of this sort have also been applied to study a certain conjecture regarding the Riemann zeta function. It is well known [8] that

$$\begin{aligned}
\zeta(2) &= 3 \sum_{k=1}^{\infty} \frac{1}{k^2 \binom{2k}{k}} \\
\zeta(3) &= \frac{5}{2} \sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k^3 \binom{2k}{k}} \\
\zeta(4) &= \frac{36}{17} \sum_{k=1}^{\infty} \frac{1}{k^4 \binom{2k}{k}}
\end{aligned}$$

These results have led some to suggest that

$$Z_5 \;=\; \zeta(5) / \sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k^5 \binom{2k}{k}}$$

might also be a simple rational or algebraic number. Unfortunately, integer relation calculations [3] have established that if $Z_5$ satisfies a polynomial of degree 25 or less, then the Euclidean norm of the coefficients must exceed $2 \times 10^{37}$.

The present application of Euler sum constants is well suited to analysis with integer relation algorithms. We will present but one example of these computations. Consider

$$\begin{aligned} s_{\rm a}(2,3) &= \sum_{k=1}^{\infty} \left(1 - \frac{1}{2} + \cdots + \frac{(-1)^{k+1}}{k}\right)^2 (k+1)^{-3} \\ &= 0.15616693338117691588103590968798819368577670984 0\cdots \end{aligned}$$

Based on experience with other constants, we conjectured that this constant satisfies a relation involving homogeneous combinations of $\zeta(2), \zeta(3), \zeta(4), \zeta(5), \ln(2), {\rm Li}_4(1/2)$ and ${\rm Li}_5(1/2)$, where ${\rm Li}_n(x) = \sum_{k=1}^{\infty} x^k k^{-n}$ denotes the polylogarithm function. The numerical values of these constants, to 50 decimal digits, are as follows:

$$\begin{aligned} \zeta(2) &= 1.64493406684822643647241516664602518921894990120 6\cdots \\ \zeta(3) &= 1.20205690315959428539973816151144999076498629234 0\cdots \\ \zeta(4) &= 1.08232323371113819151600369654116790277475095191 8\cdots \\ \zeta(5) &= 1.03692775514336992633136548645703416805708091950 1\cdots \\ \ln(2) &= 0.69314718055994530941723212145817656807550013436 0\cdots \\ {\rm Li}_4(1/2) &= 0.51747906167389938633075816189886294562237747514 1\cdots \\ {\rm Li}_5(1/2) &= 0.50840057924226870745910884925858994131954112566 4\cdots \end{aligned}$$

The set of terms involving these constants with degree five (see section 7) are as follows: ${\rm Li}_5(1/2)$, ${\rm Li}_4(1/2)\ln(2)$, $\ln^5(2)$, $\zeta(5)$, $\zeta(4)\ln(2)$, $\zeta(3)\ln^2(2)$, $\zeta(2)\ln^3(2)$, $\zeta(2)\zeta(3)$. When $s_{\rm a}(2,3)$ is augmented with this set of terms, all computed to 135 decimal digits accuracy, and the resulting 9-long vector is input to the PSLQ algorithm, it detects the relation

$(480, -1920, 0, 16, 255, 660, -840, -160, 360)$ at iteration 390. Solving this relation for $s_{\rm a}(2,3)$, we obtain the formula

$$\begin{aligned} s_{\rm a}(2,3) &= 4\,{\rm Li}_5(1/2) - \frac{1}{30}\ln^5(2) - \frac{17}{32}\zeta(5) - \frac{11}{8}\zeta(4)\ln(2) + \frac{7}{4}\zeta(3)\ln^2(2) \\ &\quad + \frac{1}{3}\zeta(2)\ln^3(2) - \frac{3}{4}\zeta(2)\zeta(3) \\ &= 4\,{\rm Li}_5(1/2) - \frac{1}{30}\ln^5(2) - \frac{17}{32}\zeta(5) + \frac{11}{720}\pi^4\ln(2) + \frac{7}{4}\zeta(3)\ln^2(2) \\ &\quad + \frac{1}{18}\pi^2\ln^3(2) - \frac{3}{24}\pi^2\zeta(3) \end{aligned}$$

(recall that $\zeta(2n) = (2\pi)^{2n}|B_{2n}|/\,[2(2n)!]$).

When the relation is detected, the minimum and maximum $y$ vector entries are $1.60 \times 10^{-134}$ and $5.98 \times 10^{-29}$, respectively. Thus the confidence level of this detection is on the order of $10^{-105}$, indicating a very reliable detection.

Although 135-digit input values and 150-digit working precision were used by us when this relation was originally detected, the fact that the maximum $y$-vector entry is only $10^{-29}$ at detection implies that such high levels of numeric precision are not required in this case. Indeed, the above relation can be successfully detected using only the 50-digit input values listed above and 50-digit working precision when performing the PSLQ algorithm.

## 6. Experimental Results

Many special cases of the proven results listed in Table 1 were first obtained using the experimental method presented in sections 2 through 4. In addition, we have obtained a number of experimental results for which formal proofs have not yet been found. Tables 2 and 3 list some of these experimental identities.

It should be emphasized that the results in Tables 2 and 3 are not established in any rigorous mathematical sense by these calculations. However, in each case the "confidence level" (see section 3) of these detections is less than $10^{-50}$, and in most cases is in the neighborhood of $10^{-100}$. Note that Table 2, together with the results in [7], gives all $s_{\rm h}(m,n)$ results for $m+n \le 7$ and $m+n = 9$, while Table 3 gives all results for the alternating sums if $m+n \le 5$. Some of these identities can be proved by ad hoc methods,

$$
\begin{aligned}
{}^*s_{\rm h}(3,2) &= \frac{15}{2}\zeta(5)+\zeta(2)\zeta(3)\\
{}^*s_{\rm h}(3,3) &= -\frac{33}{16}\zeta(6)+2\zeta^2(3)\\
s_{\rm h}(3,4) &= \frac{119}{16}\zeta(7)-\frac{33}{4}\zeta(3)\zeta(4)+2\zeta(2)\zeta(5)\\
s_{\rm h}(3,6) &= \frac{197}{24}\zeta(9)-\frac{33}{4}\zeta(4)\zeta(5)-\frac{37}{8}\zeta(3)\zeta(6)+\zeta^3(3)+3\zeta(2)\zeta(7)\\
s_{\rm h}(4,2) &= \frac{859}{24}\zeta(6)+3\zeta^2(3)\\
s_{\rm h}(4,3) &= -\frac{109}{8}\zeta(7)+\frac{37}{2}\zeta(3)\zeta(4)-5\zeta(2)\zeta(5)\\
s_{\rm h}(4,5) &= -\frac{29}{2}\zeta(9)+\frac{37}{2}\zeta(4)\zeta(5)+\frac{33}{4}\zeta(3)\zeta(6)-\frac{8}{3}\zeta^3(3)-7\zeta(2)\zeta(7)\\
s_{\rm h}(5,2) &= \frac{1855}{16}\zeta(7)+33\zeta(3)\zeta(4)+\frac{57}{2}\zeta(2)\zeta(5)\\
s_{\rm h}(5,4) &= \frac{890}{9}\zeta(9)+66\zeta(4)\zeta(5)-\frac{4295}{24}\zeta(3)\zeta(6)-5\zeta^3(3)+\frac{265}{8}\zeta(2)\zeta(7)\\
s_{\rm h}(6,3) &= -\frac{3073}{12}\zeta(9)-243\zeta(4)\zeta(5)+\frac{2097}{4}\zeta(3)\zeta(6)+\frac{67}{3}\zeta^3(3)-\frac{651}{8}\zeta(2)\zeta(7)\\
s_{\rm h}(7,2) &= \frac{134701}{36}\zeta(9)+\frac{15697}{8}\zeta(4)\zeta(5)+\frac{29555}{24}\zeta(3)\zeta(6)+56\zeta^3(3)\\
&\quad +\frac{3287}{4}\zeta(2)\zeta(7)
\end{aligned}
$$

Table 2: Experimentally Detected Identities

based on [19], and we have indicated these with a asterisk.

In many other cases we were not able to obtain a formula for the Euler sum constant explicitly in terms of values of the Riemann zeta, logarithm and polylogarithm functions, but we were able to obtain relations involving two or more Euler sum constants of the same degree (where by "degree" we mean $m+n$, where $m$ and $n$ are the indices of the constant). Some of these relations are shown in Table 4. This is not a complete list; we have obtained numerous other relations of this type. The "confidence level" of each of these relations is smaller than $10^{-25}$. The uniqueness of each of these relations was checked by repeating the run with one fewer constant input to PSLQ (there should be no relation detected when

$$
\begin{aligned}
{}^*s_{\rm a}(2,2) &= 6{\rm Li}_4(1/2) + \frac{1}{4}\ln^4(2) - \frac{29}{8}\zeta(4) + \frac{3}{2}\zeta(2)\ln^2(2) \\
{}^*s_{\rm a}(2,3) &= 4{\rm Li}_5(1/2) - \frac{1}{30}\ln^5(2) - \frac{17}{32}\zeta(5) - \frac{11}{8}\zeta(4)\ln(2) + \frac{7}{4}\zeta(3)\ln^2(2) \\
&\quad + \frac{1}{3}\zeta(2)\ln^3(2) - \frac{3}{4}\zeta(2)\zeta(3) \\
{}^*s_{\rm a}(3,2) &= -24{\rm Li}_5(1/2) + 6\ln(2){\rm Li}_4(1/2) + \frac{9}{20}\ln^5(2) + \frac{659}{32}\zeta(5) - \frac{285}{16}\zeta(4)\ln(2) \\
&\quad + \frac{5}{2}\zeta(2)\ln^3(2) + \frac{1}{2}\zeta(2)\zeta(3) \\
{}^*a_{\rm h}(2,2) &= -2{\rm Li}_4(1/2) - \frac{1}{12}\ln^4(2) + \frac{99}{48}\zeta(4) - \frac{7}{4}\zeta(3)\ln(2) + \frac{1}{2}\zeta(2)\ln^2(2) \\
{}^*a_{\rm h}(2,3) &= -4{\rm Li}_5(1/2) - 4\ln(2){\rm Li}_4(1/2) - \frac{2}{15}\ln^5(2) + \frac{107}{32}\zeta(5) - \frac{7}{4}\zeta(3)\ln^2(2) \\
&\quad + \frac{2}{3}\zeta(2)\ln^3(2) + \frac{3}{8}\zeta(2)\zeta(3) \\
{}^*a_{\rm h}(3,2) &= 6{\rm Li}_5(1/2) + 6\ln(2){\rm Li}_4(1/2) + \frac{1}{5}\ln^5(2) - \frac{33}{8}\zeta(5) + \frac{21}{8}\zeta(3)\ln^2(2) \\
&\quad - \zeta(2)\ln^3(2) - \frac{15}{16}\zeta(2)\zeta(3) \\
{}^*a_{\rm a}(2,2) &= -4{\rm Li}_4(1/2) - \frac{1}{6}\ln^4(2) + \frac{37}{16}\zeta(4) + \frac{7}{4}\zeta(3)\ln(2) - 2\zeta(2)\ln^2(2) \\
{}^*a_{\rm a}(2,3) &= 4\ln(2){\rm Li}_4(1/2) + \frac{1}{6}\ln^5(2) - \frac{79}{32}\zeta(5) + \frac{11}{8}\zeta(4)\ln(2) - 1\zeta(2)\ln^3(2) \\
&\quad + \frac{3}{8}\zeta(2)\zeta(3) \\
{}^*a_{\rm a}(3,2) &= 30{\rm Li}_5(1/2) - \frac{1}{4}\ln^5(2) - \frac{1813}{64}\zeta(5) + \frac{285}{16}\zeta(4)\ln(2) + \frac{21}{8}\zeta(3)\ln^2(2) \\
&\quad - \frac{7}{2}\zeta(2)\ln^3(2) + \frac{3}{4}\zeta(2)\zeta(3)
\end{aligned}
$$

Table 3: Experimentally Detected Identities, Cont.

this is done).

In still other cases we were not successful in finding relations, but we were able to obtain bound results from the PSLQ program that exclude a large class of potential relations among the list of candidate terms. These results do not conclusively prove that there is no such relation, only that if one exists, the Euclidean norm of its coefficients must be larger than a certain bound (assuming, of course, that the algorithm is correctly implemented and the computer works flawlessly). Some of these "negative" results are listed in Tables 5 and 6. In this table, "Norm Bound" is the minimum Euclidean norm of any possible integer relation involving the listed constants. To save space in the table the following abbreviations are used:

$$
\begin{aligned}
H_{11} &= \{\zeta(11),\ \zeta(5)\zeta(6),\ \zeta(4)\zeta(7),\ \zeta(3)\zeta(8),\ \zeta^2(3)\zeta(5),\ \zeta(2)\zeta(9),\ \zeta(2)\zeta^3(3)\}, \\
H_{12} &= \{\zeta(12),\ \zeta(5)\zeta(7),\ \zeta(3)\zeta(9),\ \zeta(3)\zeta(4)\zeta(5),\ \zeta^2(3)\zeta(6),\ \zeta^4(3),\ \zeta(2)\zeta^2(5), \\
&\quad \zeta(2)\zeta(3)\zeta(7)\}, \\
A_6 &= \{\mathrm{Li}_6(1/2),\ \ln(2)\mathrm{Li}_5(1/2),\ \ln^2(2)\mathrm{Li}_4(1/2),\ \ln^6(2),\ \zeta(6),\ \zeta(5)\ln(2), \\
&\quad \zeta(4)\ln^2(2),\ \zeta(3)\ln^3(2),\ \zeta^2(3),\ \zeta(2)\ln^4(2),\ \zeta(2)\zeta(3)\ln(2)\}, \\
A_7 &= \{\mathrm{Li}_7(1/2),\ \ln(2)\mathrm{Li}_6(1/2),\ \ln^2(2)\mathrm{Li}_5(1/2),\ \ln^3(2)\mathrm{Li}_4(1/2), \ln^7(2),\ \zeta(7), \\
&\quad \zeta(6)\ln(2),\ \zeta(5)\ln^2(2),\ \zeta(4)\ln^3(2),\ \zeta(3)\mathrm{Li}_4(1/2),\ \zeta(3)\ln^4(2),\ \zeta(3)\zeta(4), \\
&\quad \zeta^2(3)\ln(2),\ \zeta(2)\mathrm{Li}_5(1/2),\ \zeta(2)\ln^5(2),\ \zeta(2)\zeta(5),\ \zeta(2)\zeta(3)\ln^2(2)\}.
\end{aligned}
$$

One interesting by-product of the bound results in Table 5 is that there are no modest-sized integer relations among homogeneous products of $\zeta(k)$ with degree 12 or less (see section 7), except of course the well-known relations when all $k$ are even integers.

The bound result for $a_{\mathrm{h}}(1,5)$ in Table 6 confirms the observation in [7] that $a_{\mathrm{h}}(1,n)$, which equals $\alpha_{\mathrm{h}}(1,n)$, does not appear to possess an explicit evaluation when $n$ is odd and greater than three. The bound results for $\sigma_{\mathrm{h}}(2,6)$ and $\sigma_{\mathrm{h}}(2,8)$ in Table 6 confirm the observation in [7] that $\sigma_{\mathrm{h}}(2,n)$ does not appear to possess an explicit evaluation for $n$ even and greater than four.

The numerical values of the various Euler sum constants, which were used to obtain the

$$
\begin{aligned}
0 &= 84549 s_{\rm h}(1,7) + 211468 s_{\rm h}(2,6) + 148902 s_{\rm h}(3,5) - 13360 s_{\rm h}(4,4) - 1978 s_{\rm h}(5,3) \\
0 &= -127\zeta(8) + 336\zeta(3)\zeta(5) - 120\zeta(2)\zeta^2(3) - 24 s_{\rm h}(2,6) - 96 s_{\rm h}(3,5) \\
0 &= -2718587 s_{\rm h}(1,8) - 164525664 s_{\rm h}(2,7) - 178042944 s_{\rm h}(3,6) - 88947862 s_{\rm h}(4,5) \\
&\quad +3863940 s_{\rm h}(5,4) + 672100 s_{\rm h}(6,3) \\
0 &= -5138 s_{\rm h}(1,8) - 566656 s_{\rm h}(2,7) - 624016 s_{\rm h}(3,6) - 316988 s_{\rm h}(4,5) + 6480 s_{\rm h}(5,4) \\
&\quad +33605\zeta(3)\zeta(6) \\
0 &= -14269408 s_{\rm h}(1,9) + 2578470 s_{\rm h}(2,8) + 2815376 s_{\rm h}(3,7) + 5814550 s_{\rm h}(4,6) \\
&\quad +6238884 s_{\rm h}(5,5) + 3938912 s_{\rm h}(6,4) + 1122784 s_{\rm h}(7,3) - 1860 s_{\rm h}(8,2) \\
&\quad +63164285\zeta(10) \\
0 &= 321\zeta(10) - 440\zeta^2(5) - 720\zeta(3)\zeta(7) - 80\zeta^2(3)\zeta(4) + 560\zeta(2)\zeta(3)\zeta(5) \\
&\quad -40 s_{\rm h}(2,8) + 160 s_{\rm h}(3,7) \\
0 &= -1691755503 s_{\rm h}(1,10) - 3172589688 s_{\rm h}(2,9) + 837511504 s_{\rm h}(3,8) \\
&\quad -7302717576 s_{\rm h}(4,7) - 13958660016 s_{\rm h}(5,6) - 12910466064 s_{\rm h}(6,5) \\
&\quad -7099332912 s_{\rm h}(7,4) - 1773212688 s_{\rm h}(8,3) + 658360 s_{\rm h}(9,2) \\
&\quad +53491434679\zeta(11) - 21868248971\zeta(2)\zeta(9) \\
0 &= -589\zeta(11) + 322\zeta(5)\zeta(6) + 756\zeta(4)\zeta(7) + 254\zeta(3)\zeta(8) - 336\zeta^2(3)\zeta(5) \\
&\quad -368\zeta(2)\zeta(9) + 80\zeta(2)\zeta^3(3) - 16 s_{\rm h}(3,8) - 48 s_{\rm h}(4,7) \\
0 &= 70663\zeta(12) - 165840\zeta(5)\zeta(7) - 121616\zeta(3)\zeta(9) - 33168\zeta^2(3)\zeta(6) + 5528\zeta^4(3) \\
&\quad +49752\zeta(2)\zeta^2(5) + 99504\zeta(2)\zeta(3)\zeta(7) - 16584 s_{\rm h}(2,10) + 22112 s_{\rm h}(3,9) \\
0 &= 1152 s_{\rm a}(2,4) + 640 s_{\rm a}(3,3) - 7680\ln(2){\rm Li}_5(1/2) + 64\ln^6(2) - 1881\zeta(6) \\
&\quad +7440\zeta(5)\ln(2) - 1680\zeta(4)\ln^2(2) - 1120\zeta(3)\ln^3(2) + 864\zeta(3)\zeta(3) \\
&\quad -640\zeta(2)\ln^4(2) - 432\zeta(2)\zeta(3)\ln(2)
\end{aligned}
$$

Table 4: Experimentally Detected Relations

| Set of Constants | Norm Bound |
|---|---|
| $s_{\rm h}(2,6)$, $\zeta(8)$, $\zeta(3)\zeta(5)$, $\zeta(2)\zeta^2(3)$ | $7.06 \times 10^{41}$ |
| $s_{\rm h}(2,8)$, $\zeta(10)$, $\zeta^2(5)$, $\zeta(3)\zeta(7)$, $\zeta^2(3)\zeta(4)$, $\zeta(2)\zeta(3)\zeta(5)$ | $1.28 \times 10^{26}$ |
| $s_{\rm h}(2,10)$ plus $H_{12}$ | $1.43 \times 10^{15}$ |
| $s_{\rm h}(3,5)$, $\zeta(8)$, $\zeta(3)\zeta(5)$, $\zeta(2)\zeta^2(3)$ | $4.31 \times 10^{41}$ |
| $s_{\rm h}(3,7)$, $\zeta(10)$, $\zeta^2(5)$, $\zeta(3)\zeta(7)$, $\zeta^2(3)\zeta(4)$, $\zeta(2)\zeta(3)\zeta(5)$ | $3.03 \times 10^{26}$ |
| $s_{\rm h}(3,8)$ plus $H_{11}$ | $2.01 \times 10^{17}$ |
| $s_{\rm h}(3,9)$ plus $H_{12}$ | $8.21 \times 10^{14}$ |
| $s_{\rm h}(4,4)$, $\zeta(8)$, $\zeta(3)\zeta(5)$, $\zeta(2)\zeta^2(3)$ | $1.63 \times 10^{40}$ |
| $s_{\rm h}(4,6)$, $\zeta(10)$, $\zeta^2(5)$, $\zeta(3)\zeta(7)$, $\zeta^2(3)\zeta(4)$, $\zeta(2)\zeta(3)\zeta(5)$ | $3.33 \times 10^{24}$ |
| $s_{\rm h}(4,7)$ plus $H_{11}$ | $1.89 \times 10^{17}$ |
| $s_{\rm h}(4,8)$ plus $H_{12}$ | $1.06 \times 10^{15}$ |
| $s_{\rm h}(1,9)$, $s_{\rm h}(2,8)$, $s_{\rm h}(3,7)$, $s_{\rm h}(4,6)$, $s_{\rm h}(5,5)$, $s_{\rm h}(6,4)$, $s_{\rm h}(7,3)$, $s_{\rm h}(8,2)$ | $2.31 \times 10^{16}$ |
| $s_{\rm h}(1,10)$, $s_{\rm h}(2,9)$, $s_{\rm h}(3,8)$, $s_{\rm h}(4,7)$, $s_{\rm h}(5,6)$, $s_{\rm h}(6,5)$, $s_{\rm h}(7,4)$, $s_{\rm h}(8,3)$, $s_{\rm h}(9,2)$ | $1.05 \times 10^{15}$ |
| $s_{\rm h}(1,10)$, $s_{\rm h}(2,9)$, $s_{\rm h}(3,8)$, $s_{\rm h}(4,7)$, $s_{\rm h}(5,6)$, $s_{\rm h}(6,5)$, $s_{\rm h}(7,4)$, $s_{\rm h}(8,3)$, $s_{\rm h}(9,2)$, $\zeta(11)$ | $6.54 \times 10^{13}$ |
| $s_{\rm h}(1,11)$, $s_{\rm h}(2,10)$, $s_{\rm h}(3,9)$, $s_{\rm h}(4,8)$, $s_{\rm h}(5,7)$, $s_{\rm h}(6,6)$, $s_{\rm h}(7,5)$, $s_{\rm h}(8,4)$, $s_{\rm h}(9,3)$, $s_{\rm h}(10,2)$ | $6.77 \times 10^{13}$ |
| $s_{\rm h}(1,11)$, $s_{\rm h}(2,10)$, $s_{\rm h}(3,9)$, $s_{\rm h}(4,8)$, $s_{\rm h}(5,7)$, $s_{\rm h}(6,6)$, $s_{\rm h}(7,5)$, $s_{\rm h}(8,4)$, $s_{\rm h}(9,3)$, $s_{\rm h}(10,2)$, $\zeta(12)$ | $2.67 \times 10^{11}$ |

Table 5: Relation Exclusion Bounds

| Set of Constants | Norm Bound |
|---|---|
| $s_{\rm a}(2,4)$ plus $A_6$ | $6.08 \times 10^{10}$ |
| $s_{\rm a}(2,5)$ plus $A_7$ | $2.63 \times 10^{6}$ |
| $s_{\rm a}(3,3)$ plus $A_6$ | $5.95 \times 10^{10}$ |
| $s_{\rm a}(3,4)$ plus $A_7$ | $4.73 \times 10^{6}$ |
| $s_{\rm a}(2,5)$, $s_{\rm a}(3,4)$, $s_{\rm a}(4,3)$, $s_{\rm a}(5,2)$ plus $A_7$ | $3.16 \times 10^{5}$ |
| $a_{\rm h}(1,5)$ plus $A_6$ | $7.29 \times 10^{10}$ |
| $\sigma_{\rm h}(2,6)$, $\zeta(8)$, $\zeta(3)\zeta(5)$, $\zeta(2)\zeta^2(3)$ | $6.81 \times 10^{41}$ |
| $\sigma_{\rm h}(3,5)$, $\zeta(8)$, $\zeta(3)\zeta(5)$, $\zeta(2)\zeta^2(3)$ | $6.26 \times 10^{41}$ |
| $\sigma_{\rm h}(2,8)$, $\zeta(10)$, $\zeta^2(5)$, $\zeta(3)\zeta(7)$, $\zeta^2(3)\zeta(4)$, $\zeta(2)\zeta(3)\zeta(5)$ | $3.92 \times 10^{26}$ |
| $\sigma_{\rm h}(3,7)$, $\zeta(10)$, $\zeta^2(5)$, $\zeta(3)\zeta(7)$, $\zeta^2(3)\zeta(4)$, $\zeta(2)\zeta(3)\zeta(5)$ | $2.78 \times 10^{24}$ |

Table 6: Relation Exclusion Bounds, Cont.

results listed in Tables 2–6, were computed as described in sections 2 and 3. The explicit sum in formula (4) was computed on the Intel Paragon parallel computer system, using 100 iterations of Algorithm 1 (or its equivalent for alternating sums), i.e. $N = 2^{20}$, $c = 100N = 104,857,600$. The symbolic operations indicated in (4) were performed using the Maple package. The final numerical values were checked by comparing them with the values obtained from using 99 iterations of Algorithm 1 (or equivalent), i.e. $c = 99N = 103,809,024$.

## 7. Conjectures

It is not known whether closed-form evaluations of the type listed in Tables 2 and 3 exist for all of the various classes of Euler sums studied in this paper. It is possible that such formulas always exist and could be uncovered by the techniques described in this paper, if one could only deduce the form of the missing terms. We present this as an open question for further research.

One principle we have observed in this work is that in every case where we have obtained

a relation, this relation has always involved homogeneous terms. By *homogeneous* we mean that the degree of each term involved in the relation is the same integer value, namely $m+n$, where $m$ and $n$ are the indices in the definition of the constant. For these purposes, the degree of $\zeta(k)$ is taken to be $k$, as is the degree of $\mathrm{Li}_k(1/2)$, while that of $\ln(2)$ is taken as one. Although we believe this principle may hold in general, we have no idea how to prove it. We therefore present it as a conjecture. However, since it is important to limit the number of constants input to Algorithm 2 (PSLQ), in order to enhance the possibility of detecting a relation, we have often used this principle to determine the form of the candidate constants.

The modular properties of the sums $\sigma_{\mathrm{h}}(m,n)$ are being investigated by D. Zagier (as he informed us in a private communication). His work provides an alternate, abstract proof that $\sigma_{\mathrm{h}}(m,n)$ evaluates in terms of zeta functions if $m+n$ is odd. This corresponds to our results and still leaves the case $m+n$ even as an open problem.

**Acknowledgments**

The authors wish to thank David Borwein, Peter Borwein, Helaman R. P. Ferguson and Paul O. Frederickson for valuable discussions during the course of this work.

# References


[1] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*, Dover Publications, New York, 1972.

[2] K. E. Atkinson, *An Introduction to Numerical Analysis*, John Wiley, New York, 1989.

[3] D. H. Bailey, "Multiprecision Translation and Execution of Fortran Programs," *ACM Transactions on Mathematical Software*, to appear. This software and documentation may be obtained by sending electronic mail to `mp-request@nas.nasa.gov`.

[4] D. H. Bailey and H. R. P. Ferguson, "Numerical Results on Relations Between Numerical Constants Using a New Algorithm," *Mathematics of Computation*, vol. 53 (October 1989), p. 649 - 656.

[5] B. C. Berndt, *Ramanujan's Notebook*, Part I, Springer Verlag, New York, 1985.

[6] D. Borwein and J. M. Borwein, "On An Intriguing Integral and Some Series Related to $\zeta(4)$," *Proceedings of the American Mathematical Society*, in press.

[7] D. Borwein, J. M. Borwein and R. Girgensohn, "Explicit Evaluation of Euler Sums," to appear in *Proceedings of the Edinburgh Mathematical Society*.

[8] J. M. Borwein and P. B. Borwein, *Pi and the AGM*, John Wiley, New York, 1987.

[9] J. M. Borwein, P. B. Borwein and K. Dilcher, "Pi, Euler Numbers and Asymptotic Expansions," *American Mathematical Monthly*, vol. 96, no. 8 (October 1989), p. 681-687.

[10] B. W. Char, K. O. Geddes, G. H. Gonnet, B. L. Leong, M. B. Monagan, S. M. Watt, *Maple V Language Reference Manual*, Springer-Verlag, New York, 1991.

[11] P. J. De Doelder, "On Some Series Containing $\Psi(x) - \Psi(y)$ and $(\Psi(x) - \Psi(y))^2$ for Certain Values of $x$ and $y$," *Journal of Computational and Applied Mathematics*, vol. 37 (1991), p. 125-141.

[12] H. R. P. Ferguson and D. H. Bailey, "A Polynomial Time, Numerically Stable Integer Relation Algorithm," RNR Technical Report RNR-91-032, NASA Ames Research Center, MS T045-1, Moffett Field, CA 94035-1000.

[13] H. R. P. Ferguson and R. W. Forcade, "Generalization of the Euclidean Algorithm for Real Numbers to All Dimensions Higher Than Two," *Bulletin of the American Mathematical Society*, 1 (1979), p. 912 - 914.

[14] H. R. P. Ferguson, "A Non-Inductive GL(n,Z) Algorithm That Constructs Linear Relations for n Z-Linearly Dependent Real Numbers," *Journal of Algorithms*, Vol. 8 (1987), p. 131 - 145.

[15] J. Hastad, B. Just, J. C. Lagarias and C. P. Schnorr, "Polynomial Time Algorithms for Finding Integer Relations Among Real Numbers," *SIAM Journal on Computing*, vol. 18 (1988), p. 859 - 881.

[16] D. E. Knuth, *The Art of Computer Programming*, vol. 1, Addison Wesley, Menlo Park, 1973.

[17] D. E. Knuth, *The Art of Computer Programming*, vol. 2, Addison Wesley, Menlo Park, 1981.

[18] A. K. Lenstra, H. W. Lenstra and L. Lovasz, "Factoring Polynomials with Rational Coefficients", *Math. Annalen*, vol. 261 (1982), p. 515 - 534.

[19] L. Lewin, *Polylogarithms and Associated Functions*, North Holland, New York, 1981.

[20] S. Wolfram, *Mathematica: A System for Doing Mathematics by Computer*, Addison Wesley, Menlo Park, 1988.