Integer Relation Detection and Lattice Reduction

David H. Bailey[1]

To appear in *Computing in Science and Engineering*

1. Introduction

Let $x = (x_1, x_2, \cdots, x_n)$ be a vector of real or complex 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 a practical computational scheme that can recover the vector of integers a_i, if it exists, or can produce bounds within which no integer relation exists. As we shall see, integer relation algorithms have a variety of interesting applications, including the recognition of a numeric constant in terms of the mathematical formula that it satisfies.

The problem of finding integer relations is not new. It was first studied by Euclid, whose Euclidean algorithm solves this problem in the case $n = 2$. The generalization of this problem for $n > 2$ was attempted by Euler, Jacobi, Poincaré, Minkowski, Perron, Brun, Bernstein, among others. The first integer relation algorithm with the required properties mentioned above was discovered in 1977 by Ferguson and Forcade [18].

There is a close connection between integer lattice reduction and integer relation detection. Indeed, one common solution to the integer relation problem is to apply the Lenstra-Lenstra-Lovasz (LLL) lattice reduction algorithm. However, there are some difficulties with this approach, notably the somewhat arbitrary selection of a required multiplier — if it is too small, or too large, the LLL solution will not be the desired integer relation. These difficulties were addressed in the "HJLS" algorithm [19], which is based on the LLL algorithm. Unfortunately, the HJLS algorithm suffers from numerical instability, and it fails as a result in many cases of practical interest.

[1]Author's address: Lawrence Berkeley Laboratory, MS 50B-2239, Berkeley, CA 94720, USA, `dhbailey@lbl.gov`. This work was supported by the Director, Office of Computational and Technology Research, Division of Mathematical, Information, and Computational Sciences of the U.S. Department of Energy, under contract number DE-AC03-76SF00098.

2. The PSLQ Algorithm

At the present time, the most effective algorithm for integer relation detection is Ferguson's recently discovered "PSLQ" algorithm [17]. In addition to possessing good numerical stability, PSLQ is guaranteed to find a relation in a polynomially bounded number of iterations. The name "PSLQ" derives from its usage of a partial sum of squares vector and a LQ (lower-diagonal-orthogonal) matrix factorization. A simple statement of the PSLQ algorithm, equivalent to the original formulation, is as follows: Let x be the n-long input real vector, and let nint denote the nearest integer function. Select $\gamma \geq \sqrt{4/3}$. Then perform the following operations:

Initialize:

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

2. Compute the n-long vector s as $s_k := \sqrt{\sum_{j=k}^{n} x_j^2}$, and set y to the x vector, normalized by s_1.

3. Compute the initial $n \times (n-1)$ matrix H as $H_{ij} = 0$ if $i < j$, $H_{jj} := s_{j+1}/s_j$, and $H_{ij} := -y_i y_j/(s_j s_{j+1})$ if $i > j$.

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

Iterate until an entry of y is within a reasonable tolerance of zero, or precision has been exhausted:

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

2. Exchange the entries of y indexed m and $m+1$, the corresponding rows of A and H, and the corresponding columns of B.

3. Remove the corner on H diagonal: If $m \leq n-2$ then set $t_0 := \sqrt{H_{mm}^2 + H_{m,m+1}^2}$, $t_1 := H_{mm}/t_0$ and $t_2 := H_{m,m+1}/t_0$; for $i := m$ to n: set $t_3 := H_{im}$, $t_4 := H_{i,m+1}$, $H_{im} := t_1 t_3 + t_2 t_4$ and $H_{i,m+1} := -t_2 t_3 + t_1 t_4$; endfor; endif.

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

5. Norm bound: Compute $M := 1/\max_j |H_{jj}|$. Then there can exist no relation vector whose Euclidean norm is less than M.

Upon completion, the desired relation is found in the column of B corresponding to the zero entry of y. Some efficient "multi-level" implementations of PSLQ, as well as a variant of PSLQ that is well-suited for highly parallel computer systems, are given in [5].

It should be emphasized that for almost all applications of an integer relation algorithm such as PSLQ, very high precision arithmetic must be used. Only a very small class of relations can be recovered reliably with the 64-bit IEEE floating-point arithmetic that is available on current computer systems. In general, if one wishes to recover a relation of length n, with coefficients of maximum size d digits, then the input vector x must be specified to at least nd digits, and one must employ floating-point arithmetic accurate to at least nd digits. The software products Maple and Mathematica include multiple precision arithmetic facilities. One may also use any of several freeware multiprecision software packages [1, 2, 15].

3. Finding Algebraic Relations Using PSLQ

One application of PSLQ in the field of mathematical number theory is to determine whether or not a given constant α, 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. If a relation is found for x, then this relation vector is precisely the set of integer coefficients of a polynomial satisfied by α.

One of the first results of this sort was the identification of the constant $B_3 = 3.54409035955\cdots$ [1]. B_3 is the third bifurcation point of the logistic map $x_{k+1} = rx_k(1 - x_k)$, which exhibits period doubling shortly before the onset of chaos. To be precise, B_3 is the smallest value of the parameter r such that successive iterates x_k exhibit eight-way periodicity instead of four-way periodicity. Computations using a predecessor algorithm to PSLQ found that B_3 is a root the polynomial

$$\begin{aligned} 0 &= 4913 + 2108t^2 - 604t^3 - 977t^4 + 8t^5 + 44t^6 + 392t^7 - 193t^8 - 40t^9 \\ &\quad + 48t^{10} - 12t^{11} + t^{12} \end{aligned}$$

Recently, $B_4 = 3.564407268705\cdots$, the fourth bifurcation point of the logistic map, was identified using PSLQ by British physicist David Braodhurst [5]. Some conjectural reasoning had suggested that B_4 might satisfy a 240-degree polynomial, and some further analysis had suggested that the constant $\alpha = -B_4(B_4 - 2)$ might satisfy a 120-degree polynomial. In order to test this hypothesis, Broadhurst applied a PSLQ program to the 121-long vector $(1, \alpha, \alpha^2, \cdots, \alpha^{120})$. Indeed, a relation was found, although 10,000 digit arithmetic was required. The recovered integer coefficients descend monotonically from $257^{30} \approx 1.986 \times 10^{72}$ to one.

4. A New Formula for Pi

Through the centuries mathematicians have assumed that there is no shortcut to computing just the n-th digit of π. Thus, it came as no small surprise when such an algorithm was recently discovered [4]. In particular, this simple scheme allows one to compute the n-th hexadecimal (or binary) digit of π without computing any of the first $n - 1$ digits, without using multiple-precision arithmetic software, and at the expense of very little computer memory. The one millionth hex digit of π can be computed in this manner on a current-generation personal computer in only about 60 seconds run time.

This scheme is based on the following new formula, which was discovered in 1996 using PSLQ:

$$\pi = \sum_{k=0}^{\infty} \frac{1}{16^k} \left[\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6} \right]$$

Similar base-2 formulas are given in [4, 14] for some other mathematical constants. In [13] some base-3 formulas were obtained, including the identity

$$\pi^2 = \frac{2}{27}\sum_{k=0}^{\infty}\frac{1}{729^k}\left[\frac{243}{(12k+1)^2} - \frac{405}{(12k+2)^2} - \frac{81}{(12k+4)^2} - \frac{27}{(12k+5)^2} - \frac{72}{(12k+6)^2} - \frac{9}{(12k+7)^2} - \frac{9}{(12k+8)^2} - \frac{5}{(12k+10)^2} + \frac{1}{(12k+11)^2}\right]$$

5. Identification of Multiple Sum Constants

A large number of results were recently found using PSLQ in the course of research on multiple sums, such as those shown in Table 1. After computing the numerical values of these constants, a PSLQ program was used to determine if a given constant satisfied an identity of a conjectured form. These efforts produced numerous empirical evaluations and suggested general results [3]. Eventually, elegant proofs were found for many of these specific and general results [6, 7]. Three examples of PSLQ results that were subsequently proven are given in Table 1. In the table, $\zeta(t) = \sum_{j=1}^{\infty} j^{-t}$ is the Riemann zeta function, and $\mathrm{Li}_n(x) = \sum_{j=1}^{\infty} x^j j^{-n}$ denotes the polylogarithm function.

$$\begin{aligned}
\sum_{k=1}^{\infty}\left(1+\tfrac{1}{2}+\cdots+\tfrac{1}{k}\right)^2 (k+1)^{-4} &= \tfrac{37}{22680}\pi^6 - \zeta^2(3) \\
\sum_{k=1}^{\infty}\left(1+\tfrac{1}{2}+\cdots+\tfrac{1}{k}\right)^3 (k+1)^{-6} &= \zeta^3(3) + \tfrac{197}{24}\zeta(9) + \tfrac{1}{2}\pi^2\zeta(7) \\
&\quad - \tfrac{11}{120}\pi^4\zeta(5) - \tfrac{37}{7560}\pi^6\zeta(3) \\
\sum_{k=1}^{\infty}\left(1-\tfrac{1}{2}+\cdots+(-1)^{k+1}\tfrac{1}{k}\right)^2 (k+1)^{-3} &= 4\,\mathrm{Li}_5(\tfrac{1}{2}) - \tfrac{1}{30}\ln^5(2) - \tfrac{17}{32}\zeta(5) \\
&\quad - \tfrac{11}{720}\pi^4\ln(2) + \tfrac{7}{4}\zeta(3)\ln^2(2) + \tfrac{1}{18}\pi^2\ln^3(2) - \tfrac{1}{8}\pi^2\zeta(3)
\end{aligned}$$

Table 1: Some multiple sum identities found by PSLQ

In another application to mathematical number theory, PSLQ has been used to investigate sums of the form

$$S(k) := \sum_{n>0}\frac{1}{n^k\binom{2n}{n}}$$

For small k, these constants satisfy simple identities, such as $S(4) = 17\pi^4/3240$. Thus researchers have sought generalizations of these formulas for $k > 4$. As a result of PSLQ computations, the constants $\{S(k)|k = 5\ldots 20\}$ have been evaluated in terms of multiple zeta values [8], which are defined by

$$\zeta(s_1, s_2, \cdots, s_r) = \sum_{k_1>k_2>\cdots>k_r>0}\frac{1}{k_1^{s_1}\,k_2^{s_2}\cdots k_r^{s_r}}$$

and multiple Clausen values [10] of the form

$$M(a,b) := \sum_{n_1>n_2>\ldots>n_b>0}\frac{\sin(n_1\pi/3)}{n_1^a}\prod_{j=1}^{b}\frac{1}{n_j}$$

A sample evaluation is

$$\begin{aligned} S(9) &= \pi\left[2M(7,1)+\frac{8}{3}M(5,3)+\frac{8}{9}\zeta(2)M(5,1)\right]-\frac{13921}{216}\zeta(9) \\ &+\frac{6211}{486}\zeta(7)\zeta(2)+\frac{8101}{648}\zeta(6)\zeta(3)+\frac{331}{18}\zeta(5)\zeta(4)-\frac{8}{9}\zeta^3(3) \end{aligned}$$

The evaluation of $S(20)$ is an integer relation problem with $n = 118$, requiring 5000 digit arithmetic. The full solution is given in [5].

6. Connections to Quantum Field Theory

In a surprising recent development, Broadhurst has found, using PSLQ, that there is an intimate connection between these multiple sums and constants resulting from evaluation of Feynman diagrams in quantum field theory [11, 12]. In particular, the renormalization procedure (which removes infinities from the perturbation expansion) involves multiple zeta values. Broadhurst used PSLQ to find formulas and identities involving these constants. As before, a fruitful theory emerged, including a large number of both specific and general results [8, 9].

More generally, one may define *Euler sums* by [8]

$$\zeta\left(\begin{array}{cccc} s_1, & s_2 & \cdots & s_r \\ \sigma_1, & \sigma_2 & \cdots & \sigma_r \end{array}\right) := \sum_{k_1>k_2>\cdots>k_r>0} \frac{\sigma_1^{k_1}}{k_1^{s_1}}\frac{\sigma_2^{k_2}}{k_2^{s_2}}\cdots\frac{\sigma_r^{k_r}}{k_r^{s_r}}$$

where $\sigma_j = \pm 1$ are signs and $s_j > 0$ are integers. When all the signs are positive, one has a multiple zeta value. Constants with alternating signs appear in problems such as computation of the magnetic moment of the electron.

Broadhurst had conjectured that the dimension of the space of Euler sums with weight $w := \sum_j s_j$ is the Fibonacci number $F_{w+1} = F_w + F_{w-1}$, with $F_1 = F_2 = 1$. Complete reductions of all Euler sums to a basis of size F_{w+1} have now been obtained with PSLQ at weights $w \le 9$. At weights $w = 10$ and $w = 11$ the conjecture has been stringently tested by application of PSLQ in more than 600 cases. At weight $w = 11$ such tests involve solving integer relations of size $n = F_{12} + 1 = 145$ [5].

Some recent quantum field theory results are even more remarkable. Broadhurst has now shown [13], using PSLQ, that in each of ten cases with unit or zero mass, the finite part the scalar 3-loop tetrahedral vacuum Feynman diagram reduces to 4-letter "words" that represent iterated integrals in an alphabet of 7 "letters" comprising the one-forms $\Omega := dx/x$ and $\omega_k := dx/(\lambda^{-k} - x)$, where $\lambda := (1+\sqrt{-3})/2$ is the primitive sixth root of unity, and k runs from 0 to 5. A 4-letter word is a 4-dimensional iterated integral, such as

$$U := \zeta(\Omega^2\omega_3\omega_0) = \int_0^1 \frac{dx_1}{x_1}\int_0^{x_1}\frac{dx_2}{x_2}\int_0^{x_2}\frac{dx_3}{(-1-x_3)}\int_0^{x_3}\frac{dx_4}{(1-x_4)} = \sum_{j>k>0}\frac{(-1)^{j+k}}{j^3k}$$

There are 7^4 four-letter words. Only two of these are primitive terms occurring in the 3-loop Feynman diagrams: U, above, and

$$V := \mathrm{Real}[\zeta(\Omega^2\omega_3\omega_1)] = \sum_{j>k>0}\frac{(-1)^j\cos(2\pi k/3)}{j^3k}.$$

The remaining terms in the diagrams reduce to products of constants found in Feynman diagrams with fewer loops. These ten cases as shown in Figure 1. In these diagrams, dots indicate particles with nonzero rest mass. The formulas that have been found, using PSLQ, for the corresponding constants are given in Table 2. The constant $C = \sum_{k>0} \sin(\pi k/3)/k^2$.

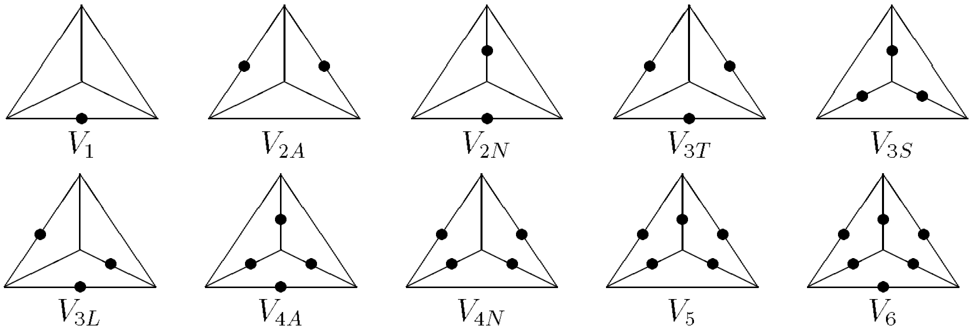

Figure 1: The ten tetrahedral cases

V_1	$=$	$6\zeta(3) + 3\zeta(4)$
V_{2A}	$=$	$6\zeta(3) - 5\zeta(4)$
V_{2N}	$=$	$6\zeta(3) - \frac{13}{2}\zeta(4) - 8U$
V_{3T}	$=$	$6\zeta(3) - 9\zeta(4)$
V_{3S}	$=$	$6\zeta(3) - \frac{11}{2}\zeta(4) - 4C^2$
V_{3L}	$=$	$6\zeta(3) - \frac{15}{4}\zeta(4) - 6C^2$
V_{4A}	$=$	$6\zeta(3) - \frac{77}{12}\zeta(4) - 6C^2$
V_{4N}	$=$	$6\zeta(3) - 14\zeta(4) - 16U$
V_5	$=$	$6\zeta(3) - \frac{469}{27}\zeta(4) + \frac{8}{3}C^2 - 16V$
V_6	$=$	$6\zeta(3) - 13\zeta(4) - 8U - 4C^2$

Table 2: Formulas found by PSLQ for the ten cases of Figure 1

7. Conclusion

For many years, researchers have dreamed of a facility that permits one to recognize a numeric constant in terms of the mathematical formula that it satisfies. With the advent of efficient integer relation detection algorithms, that time has arrived. Using these algorithms, researchers have discovered numerous new facts of mathematics and physics, and these discoveries have in turn led to valuable new insights. This process, which is often termed "experimental mathematics", namely the utilization of modern computer technology in the discovery of new mathematical principles, is expected to play a much wider role in both pure and applied mathematics during the next century.

References

[1] David H. Bailey, "Multiprecision Translation and Execution of Fortran Programs", *ACM Transactions on Mathematical Software*, vol. 19, no. 3, 1993, pp. 288–319.

[2] David H. Bailey, "A Fortran-90 Based Multiprecision System", *ACM Transactions on Mathematical Software*, vol. 21, no. 4, 1995, pg. 379-387. Available from `http://www.nersc.gov/~dhbailey`.

[3] David H. Bailey, Jonathan M. Borwein and Roland Girgensohn, "Experimental Evaluation of Euler Sums", *Experimental Mathematics*, vol. 4, no. 1, 1994, pp. 17–30.

[4] David H. Bailey, Peter B. Borwein and Simon Plouffe, "On The Rapid Computation of Various Polylogarithmic Constants", *Mathematics of Computation*, vol. 66, no. 218, 1997, pp. 903–913.

[5] David H. Bailey and David Broadhurst, "Parallel Integer Relation Detection: Techniques and Applications". Available from `http://www.nersc.gov/~dhbailey`.

[6] David Borwein and Jonathan M. Borwein, "On An Intriguing Integral and Some Series Related to $\zeta(4)$", *Proceedings of the American Mathematical Society*, vol. 123, 1995, pp. 111-118.

[7] David Borwein, Jonathan M. Borwein and Roland Girgensohn, "Explicit Evaluation of Euler Sums", *Proceedings of the Edinburgh Mathematical Society*, vol. 38, 1995, pp. 277–294.

[8] Jonathan M. Borwein, David M. Bradley and David J. Broadhurst, "Evaluations of k-fold Euler/Zagier Sums: A Compendium of Results for Arbitrary k", *Electronic Journal of Combinatorics*, vol. 4, no. 2, 1997, #R5.

[9] Jonathan M. Borwein, David M. Bradley, David J. Broadhurst and Petr Lisonek, "Combinatorial Aspects of Multiple Zeta Values", *Electronic Journal of Combinatorics*, vol. 5, no. 1, 1998, #R38.

[10] Jonathan M. Borwein, David J. Broadhurst and Joel Kamnitzer, "Central Binomial Sums and Multiple Clausen Values (with Connections to Zeta Values)". Available from `http://www.cecm.sfu.ca/preprints`.

[11] David J. Broadhurst, John A. Gracey and Dirk Kreimer, "Beyond the Triangle and Uniqueness Relations: Non-zeta Counterterms at Large N from Positive Knots", *Zeitschrift für Physik*, vol. C75, 1997, pp. 559–574.

[12] David J. Broadhurst and Dirk Kreimer, "Association of Multiple Zeta Values with Positive Knots via Feynman Diagrams up to 9 Loops", *Physics Letters*, vol. B383, 1997, pp. 403–412.

[13] David J. Broadhurst, "Massive 3-loop Feynman Diagrams Reducible to SC* Primitives of Algebras of the Sixth Root of Unity", preprint, March 1998, to appear in *European Physical Journal C*. Available from `http://xxx.lanl.gov/abs/hep-th/9803091` .

[14] David J. Broadhurst, "Polylogarithmic Ladders, Hypergeometric Series and the Ten Millionth Digits of $\zeta(3)$ and $\zeta(5)$', preprint, March 1998. Available from `http://xxx.lanl.gov/abs/math/9803067`.

[15] Sid Chatterjee and Herman Harjono, "MPFUN++: A Multiple Precision Floating Point Computation Package in C++", University of North Carolina, Sept. 1998. Available from `http://www.cs.unc.edu/Research/HARPOON/mpfun++`.

[16] Henri Cohen, *A Course in Computational Algebraic Number Theory*, Springer-Verlag, 1991.

[17] Helaman R. P. Ferguson, David H. Bailey and Stephen Arno, "Analysis of PSLQ, An Integer Relation Finding Algorithm", *Mathematics of Computation*, vol. 68, 1999, pg. 351-369.

[18] Helaman R. P. Ferguson and Rodney W. Forcade, "Generalization of the Euclidean Algorithm for Real Numbers to All Dimensions Higher Than Two", *Bulletin of the American Mathematical Society*, vol. 1, 1979, pp. 912–914.

[19] J. Hastad, B. Just, J. C. Lagarias and C. P. Schnorr, "Polynomial Time Algorithms for Finding Integer Relations Among Real Numbers", *SIAM Journal of Computing*, vol. 18, 1989, pp. 859–881.

[20] A. K. Lenstra, H. W. Lenstra, Jr. and L. Lovasz, "Factoring Polynomials with Rational Coefficients", Mathematische Annalen, vol. 261, 1982, pg. 515-534.