Integer Relation Detection

Practical algorithms for integer relation detection have become a staple in the emerging discipline of experimental mathematics—using modern computer technology to explore mathematical questions. After briefly discussing the problem of integer relation detection, the author describes several recent, remarkable applications of these techniques in both mathematics and physics.

For many years, researchers have dreamt of a facility that lets them 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.

Integer relation detection

Let $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ be a vector of real or complex numbers. $\mathbf{x}$ is said to possess an integer relation if there exist integers a_i (not all zero), such that $a_1x_1 + a_2x_2 + \ldots + a_nx_n = 0$. An integer relation algorithm is 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.

The problem of finding integer relations is not new. Euclid (whose Euclidean algorithm solves this problem in the case $n = 2$) first studied it about 300 BC. Leonard Euler, Karl Jacobi, Jules Poincaré, and other well-known 18th, 19th, and 20th century mathematicians pursued solutions for a larger n. The first integer relation algorithm for general n was discovered in 1977 by Helaman Ferguson and Rodney Forcade.[1]

There is a close connection between integer relation detection and integer-lattice reduction. Indeed, one common solution to the integer relation problem is to apply the Lenstra-Lenstra-Lovasz 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 determine the desired integer relation. The HJLS algorithm,[2] which is based on the LLL algorithm, addresses these dif-

1521-9615/00/$10.00 © 2000 IEEE

David H. Bailey
Lawrence Berkeley Laboratory

ficulties, but, unfortunately, it suffers from numerical instability and thus fails in many cases of practical interest.

The PSLQ algorithm

The most effective algorithm for integer relation detection is Ferguson's recently discovered PSLQ algorithm;[3] the name derives from its usage of a partial sum-of-squares vector and an LQ (lower-diagonal-orthogonal) matrix factorization. In addition to possessing good numerical stability, PSLQ finds a relation in a polynomially bounded number of iterations. A simple statement of the PSLQ algorithm, equivalent to the original formulation, is as follows: Let $\mathbf{x}$ 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.

1. Set the $n \times n$ matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ to the identity.
2. Compute the n-long vector $\mathbf{s}$ as
$$\mathbf{s}_k := \sqrt{\sum_{j=k}^{n} \mathbf{x}_j^2}$$
and set $\mathbf{y}$ to the $\mathbf{x}$ vector, normalized by $\mathbf{s}_1$.
3. Compute the initial $n \times (n-1)$ matrix $\boldsymbol{H}$ as $\boldsymbol{H}_{ij} = 0$ if $i < j$, $\boldsymbol{H}_{jj} := \mathbf{s}_{j+1}/\mathbf{s}_j$, and $\boldsymbol{H}_{ij} := -\mathbf{y}_i\mathbf{y}_j/(\mathbf{s}_j\mathbf{s}_{j+1})$ if $i > j$.
4. Reduce $\boldsymbol{H}$: For $i := 2$ to n: for $j := i - 1$ to 1 step −1 : set $t :=$ nint $(\boldsymbol{H}_{ij}/\boldsymbol{H}_{jj})$; and $\mathbf{y}_j := \mathbf{y}_j + t\mathbf{y}_i$; for $k := 1$ to j: set $\boldsymbol{H}_{ik} := \boldsymbol{H}_{ik} - t\boldsymbol{H}_{jk}$; endfor; for $k := 1$ to n: set $\boldsymbol{A}_{ik} := \boldsymbol{A}_{ik} - t\boldsymbol{A}_{jk}$ and $\boldsymbol{B}_{kj} := \boldsymbol{B}_{kj} + t\boldsymbol{B}_{ki}$; endfor; endfor; endfor.

Iterate until an entry of $\mathbf{y}$ is within a reasonable tolerance of 0, or until precision is exhausted:

1. Select m such that $\gamma^i\ |\boldsymbol{H}_{ii}|$ is maximal when $i = m$.
2. Exchange the entries of $\mathbf{y}$ indexed m and $m + 1$, the corresponding rows of $\boldsymbol{A}$ and $\boldsymbol{H}$, and the corresponding columns of $\boldsymbol{B}$.
3. Remove the corner on $\boldsymbol{H}$ diagonal: If $m \leq n - 2$, set
$$t_0 := \sqrt{\boldsymbol{H}_{mm}^2 + \boldsymbol{H}_{m,m+1}^2}, \quad t_1 := \boldsymbol{H}_{mm} / t_0$$
and $t_2 := \boldsymbol{H}_{m,m+1} / t_0$; for $i := m$ to n: set $t_3 := \boldsymbol{H}_{im}$, $t_4 := \boldsymbol{H}_{i,m+1}$, $\boldsymbol{H}_{im} := t_1t_3 + t_2t_4$ and $\boldsymbol{H}_{i,m+1} := -t_2t_3 + t_1t_4$; endfor; endif.
4. Reduce $\boldsymbol{H}$: For $i := m + 1$ to n: for $j := \min(i - 1, m + 1)$ to 1 step −1: set $t :=$ nint $(\boldsymbol{H}_{ij} / \boldsymbol{H}_{jj})$ and $\mathbf{y}_j := \mathbf{y}_j + t\mathbf{y}_i$; for $k := 1$ to j: set $\boldsymbol{H}_{ik} := \boldsymbol{H}_{ik} - t\,\boldsymbol{H}_{jk}$; endfor; for $k := 1$ to n: set $\boldsymbol{A}_{ik} := \boldsymbol{A}_{ik} - t\,\boldsymbol{A}_{jk}$ and $\boldsymbol{B}_{kj} := \boldsymbol{B}_{kj} + t\boldsymbol{B}_{ki}$; endfor; endfor; endfor.
5. Norm bound: Compute $M := 1/\max_j |\boldsymbol{H}_{jj}|$. Then there can exist no relation vector whose Euclidean norm is less than M.

Upon completion, the desired relation is found in column $\boldsymbol{B}$ corresponding to the 0 entry of $\mathbf{y}$. (Some efficient "multilevel" implementations of PSLQ, as well as a variant of PSLQ that is well-suited for highly parallel computer systems, are described in a recent paper.[4])

Almost all applications of an integer relation algorithm such as PSLQ require high-precision arithmetic. The 64-bit IEEE floating-point arithmetic available on current computer systems can reliably recover only a very small class of relations. In general, if we want to recover a relation of length n with coefficients of maximum size d digits, then the input vector $\mathbf{x}$ must be specified to at least nd digits, and we must employ floating-point arithmetic accurate to at least nd digits. Maple and Mathematica include multiple precision arithmetic facilities. There are several freeware multiprecision software packages.[5–7]

Finding algebraic relations using PSLQ

One PSLQ application 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. First compute the vector $\mathbf{x} = (1, \alpha, \alpha^2, \ldots, \alpha^n)$ to high precision, and then apply an integer relation algorithm. If you find a relation for $\mathbf{x}$, then this relation vector is precisely the set of integer coefficients of a polynomial α satisfies.

One of the first results of this sort was the identification of the constant $B_3 = 3.54409035955\ \ldots$. B_3 is the third bifurcation point of the logistic map $z_{k+1} = rz_k(1 - z_k)$, which exhibits period doubling shortly before the onset of chaos.[5] To be precise, B_3 is the smallest value of the parameter r such that successive iterates z_k exhibit eight-way periodicity instead of four-way periodicity. Computations using a predecessor algorithm to PSLQ found that B_3 is a root of the polynomial $0 = 4913 + 2108t^2 - 604t^3 - 977t^4 + 8t^5 + 44t^6 + 392t^7 - 193t^8 - 40t^9 + 48t^{10} - 12t^{11} + t^{12}$.

Recently, British physicist David Braodhurst identified $B_4 = 3.564407268705\ \ldots$, the fourth bifurcation point of the logistic map, using PSLQ.[4] It had been conjectured that B_4 might satisfy a 240-degree polynomial, and further analysis sug-

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

Figure 1. The formula found by PSLQ, which can be used to compute hexadecimal digits of π.

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

Figure 2. A base-3 formula for π^2.

$$\sum_{k=1}^{\infty}\left(1+\frac{1}{2}+\cdots+\frac{1}{k}\right)^2 (k+1)^{-4} = \frac{37}{22{,}680}\pi^6 - \zeta^2(3)$$

$$\sum_{k=1}^{\infty}\left(1+\frac{1}{2}+\cdots+\frac{1}{k}\right)^3 (k+1)^{-6} = \zeta^3(3) + \frac{197}{24}\zeta(9) + \frac{1}{2}\pi^2\zeta(7) - \frac{11}{120}\pi^4\zeta(5) - \frac{37}{7560}\pi^6\zeta(3)$$

$$\sum_{k=1}^{\infty}\left(1-\frac{1}{2}+\cdots+(-1)^{k+1}\frac{1}{k}\right)^2 (k+1)^{-3} = 4\mathrm{Li}_5\left(\frac{1}{2}\right) - \frac{1}{30}\ln^5(2) - \frac{17}{32}\zeta(5) - \frac{11}{720}\pi^4\ln(2) + \frac{7}{4}\zeta(3)\ln^2(2) + \frac{1}{18}\pi^2\ln^3(2) - \frac{1}{8}\pi^2\zeta(3)$$

Figure 3. Some multiple-sum identities found by PSLQ. $\zeta(t) = \sum_{j=1}^{\infty} j^{-t}$ is the Riemann zeta function, and $\mathrm{Li}_n(x) = \sum_{j=1}^{\infty} x^j j^{-n}$ denotes the polylogarithm function.

gested that the constant $\alpha = -B_4(B_4 - 2)$ might satisfy a 120-degree polynomial. To test this hypothesis, Broadhurst applied a PSLQ program to the 121-long vector $(1, \alpha, \alpha^2, \ldots, \alpha^{120})$. Indeed, a relation was found, but the computation required 10,000-digit arithmetic. The recovered integer coefficients descend monotonically from $257^{30} \approx 1.986 \times 10^{72}$ to 1.

A new formula for π

Throughout the centuries, mathematicians have assumed that there is no shortcut to computing just the nth digit of π. Thus, it came as no small surprise when such an algorithm was recently discovered.[8] In particular, this simple scheme lets us compute the nth 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 runtime.

This scheme for computing the nth digit of π is based on a formula that was discovered in 1996 using PSLQ (see Figure 1). Similar base-2 formulas exist for other mathematical constants,[8,9] and Broadhurst recently has obtained some base-3 formulas,[10] including an identity for π^2 (see Figure 2), using PSLQ.

Identifying multiple-sum constants

In the course of researching multiple sums, scientists recently found many results using PSLQ, such as those shown in Figure 3. After computing the numerical values of these constants, a PSLQ program determined if a given constant satisfied an identity of a conjectured form. These efforts produced numerous empirical evaluations and suggested general results[11]—for which scientists eventually found elegant proofs.[12,13]

In another application to mathematical number theory, several researchers have used PSLQ 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, several researchers have evaluated the constants $\{S(k) \mid k = 5 \ldots 20\}$ in terms of multiple zeta values,[14] which are defined by

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

and multiple Clausen values[15] of the form

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

Figure 4 gives a sample evaluation.

The evaluation of $S(20)$ is an integer relation problem with $n = 118$, requiring 5,000-digit arithmetic. (The full solution is given in a recent paper.[4])

$$S(9) = \pi\left[2M(7,1) + \frac{8}{3}M(5,3) + \frac{8}{9}\varsigma(2)M(5,1)\right] - \frac{13{,}921}{216}\varsigma(9) + \frac{6{,}211}{486}\varsigma(7)\varsigma(2) + \frac{8{,}101}{648}\varsigma(6)\varsigma(3) + \frac{331}{18}\varsigma(5)\varsigma(4) - \frac{8}{9}\varsigma^3(3)$$

Figure 4. A sample evaluation of an of an Apery sum constant using PSLQ.

Connections to quantum field theory

In a surprising recent development, Broadhurst found an intimate connection between these multiple sums and constants resulting from evaluating Feynman diagrams in quantum field theory.[16,17] 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.[14,18]

More generally, we can define Euler sums by[14]

$$\varsigma\begin{pmatrix} s_1, & s_2 & \ldots & s_r \\ \sigma_1, & \sigma_2 & \ldots & \sigma_r \end{pmatrix} = \sum_{k_1 > k_2 > \ldots > k_r > 0} \frac{\sigma_1^{k_1}}{k_1^{s_1}} \frac{\sigma_2^{k_2}}{k_2^{s_2}} \ldots \frac{\sigma_r^{k_r}}{k_r^{s_r}}$$

where $\sigma_j = \pm 1$ are signs and $s_j > 0$ are integers. When all the signs are positive, we have a multiple zeta value. Constants with alternating signs appear in problems such as computing an electron's magnetic moment.

Broadhurst conjectured that the dimension of the Euler sum spaces with weight $w = \Sigma s_j$ is the Fibonacci number $F_{w+1} = F_w + F_{w-1}$, with $F_1 = F_2 = 1$. I've obtained complete reductions of all Euler sums to a basis of size F_{w+1} with PSLQ at weights $w \le 9$. At weights $w = 10$ and $w = 11$, Broadhurst stringently tested the conjecture by applying PSLQ in more than 600 cases. At weight $w = 11$, such tests involve solving integer relations of size $n = F_{12} + 1 = 145$.[4]

Some recent quantum field theory results are even more remarkable. Broadhurst has now shown, using PSLQ, that in each of 10 cases with unit or 0 mass, the finite part of the scalar, three-loop, tetrahedral vacuum Feynman diagram reduces to four-letter "words." These words represent iterated integrals in an alphabet of seven "letters" comprising the one-forms $\Omega := dx/x$ and $w_k := dx/(\lambda^{-k} - x)$, where $\lambda := (1 + \sqrt{-3})/2$ is the primitive sixth root of unity, and k runs from 0 to 5.[10]

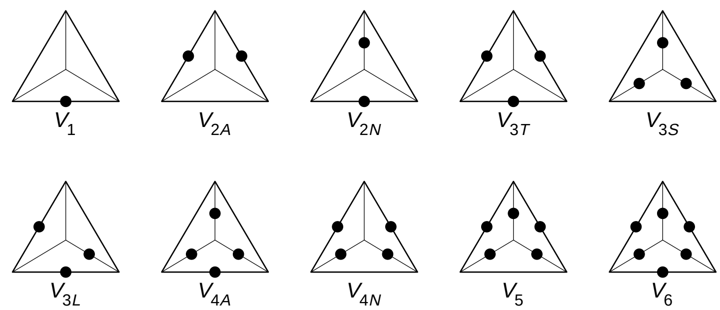

Figure 5. The 10 tetrahedral cases.

Figure 5 shows these 10 cases. In the diagrams, dots indicate particles with nonzero rest mass. Table 1 gives the formulas that Broadhurst found, using PSLQ, for the corresponding constants. In the table, $U = \Sigma_{j>k>0}(-1)^{j+k}/(j^3 k)$; $V = \Sigma_{j>k>0}(-1)^j \cos(2\pi k/3)/(j^3 k)$; and the constant $C = \Sigma_{k>0} \sin(\pi k/3)/k^2$.

Table 1. The formulas found by PSLQ for the 10 cases in Figure 5.

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

Using integer relation algorithms, researchers have discovered numerous new facts of mathematics and physics, and these discoveries have in turn led to valuable new insights. This process, often called "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. In particular, when fast integer relation detection facilities are integrated into widely used mathematical computing environments, such as Mathematica and Maple, we can expect numerous more discoveries of the sort described in this article, and, possibly, new applications that we cannot yet foresee.

Acknowledgments

This work was supported by the Director of the Office of Computational and Technology Research, Division of Mathematical, Information, and Computational Sciences of the US Department of Energy, under contract DE-AC03-76SF00098.

References

1. H.R.P. Ferguson and R.W. Forcade, "Generalization of the Euclidean Algorithm for Real Numbers to All Dimensions Higher Than Two," *Bulletin Am. Mathematical Soc.*, Vol. 1, No. 6, Nov. 1979, pp. 912–914.
2. J. Hastad et al., "Polynomial Time Algorithms for Finding Integer Relations Among Real Numbers," *SIAM J. Computing*, Vol. 18, No. 5, Oct. 1989, pp. 859–881.
3. H.R.P. Ferguson, D.H. Bailey, and S. Arno, "Analysis of PSLQ, An Integer Relation Finding Algorithm," *Mathematics of Computation*, Vol. 68, No. 225, Jan. 1999, pp. 351–369.
4. D.H. Bailey and D. Broadhurst, "Parallel Integer Relation Detection: Techniques and Applications," submitted for publication, 1999; also available at www.nersc.gov/~dhbailey, online paper 34 (current Nov. 1999).
5. D.H. Bailey, "Multiprecision Translation and Execution of Fortran Programs," *ACM Trans. Mathematical Software*, Vol. 19, No. 3, 1993, pp. 288–319.
6. D.H. Bailey, "A Fortran-90 Based Multiprecision System," *ACM Trans. Mathematical Software*, Vol. 21, No. 4, 1995, pp. 379–387.
7. S. Chatterjee and H. Harjono, "MPFUN++: A Multiple Precision Floating Point Computation Package in C++," University of North Carolina, Sept. 1998; www.cs.unc.edu/Research/HARPOON/mpfun++ (current Nov. 1999).
8. D.H. Bailey, P.B. Borwein and S. Plouffe, "On The Rapid Computation of Various Polylogarithmic Constants," *Mathematics of Computation*, Vol. 66, No. 218, 1997, pp. 903–913.
9. D.J. Broadhurst, "Polylogarithmic Ladders, Hypergeometric Series and the Ten Millionth Digits of $\zeta(3)$ and $\zeta(5)$," preprint, Mar. 1998; xxx.lanl.gov/abs/math/9803067 (current Nov. 1999).
10. D.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 J. Computing*; xxx.lanl.gov/abs/hep-th/9803091 (current Nov. 1999).
11. D.H. Bailey, J.M. Borwein, and R. Girgensohn, "Experimental Evaluation of Euler Sums," *Experimental Mathematics*, Vol. 4, No. 1, 1994, pp. 17–30.
12. D. Borwein and J.M. Borwein, "On an Intriguing Integral and Some Series Related to $\zeta(4)$," *Proc. American Mathematical Soc.*, Vol. 123, 1995, pp. 111–118.
13. D. Borwein, J.M. Borwein, and R. Girgensohn, "Explicit Evaluation of Euler Sums," *Proc. Edinburgh Mathematical Soc.*, Vol. 38, 1995, pp. 277–294.
14. J.M. Borwein, D.M. Bradley, and D.J. Broadhurst, "Evaluations of *k*-fold Euler/Zagier Sums: A Compendium of Results for Arbitrary *k*," *Electronic J. Combinatorics*, Vol. 4, No. 2, #R5, 1997.
15. J.M. Borwein, D.J. Broadhurst, and J. Kamnitzer, "Central Binomial Sums and Multiple Clausen Values (with Connections to Zeta Values)," No. 137, 1999; www.cecm.sfu.ca/preprints/1999pp.html (current Nov. 1999).
16. D.J. Broadhurst, J.A. Gracey, and D. 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.
17. D.J. Broadhurst and D. Kreimer, "Association of Multiple Zeta Values with Positive Knots via Feynman Diagrams up to 9 Loops," *Physics Letters*, Vol. B383, 1997, pp. 403–412.
18. J.M. Borwein et al., "Combinatorial Aspects of Multiple Zeta Values," *Electronic J. Combinatorics*, Vol. 5, No. 1, #R38, 1998.

David H. Bailey is the chief technologist of the National Energy Research Scientific Computing Center at the Lawrence Berkeley National Laboratory. His research has included studies in highly parallel computing, numerical algorithms, fast Fourier transforms, multiprecision computation, supercomputer performance, and computational number theory. He received his BS in mathematics from Brigham Young University and his PhD in mathematics from Stanford University. In 1993, he received the Sidney Fernbach Award, which is bestowed by the IEEE Computer Society for contributions to the field of high-performance computing. He has also received the Chauvenet Prize and the Merten Hasse Prize from the Mathematical Association of America and the H. Julian Allen Award from NASA. He just served as the technical papers chair for Supercomputing '99 and as the chair of the 1999 Gordon Bell Prize committee. Contact him at the Lawrence Berkeley Laboratory, MS 50B-2239, Berkeley, CA 94720; dhbailey@lbl.gov.