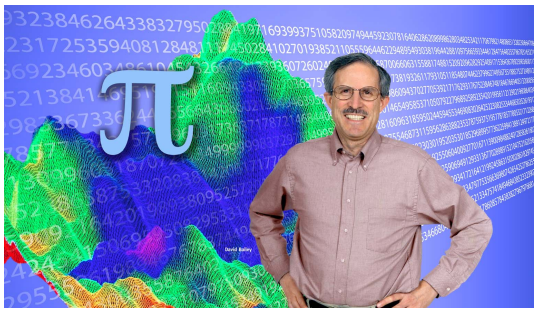
Experimental mathematics and the omega function

David H. Bailey
http://www.davidhbailey.com
Lawrence Berkeley National Laboratory (retired)
University of California, Davis, Department of Computer Science



The PSLQ integer relation algorithm

Let (x_n) be a given vector of real numbers. An integer relation algorithm either finds integers (a_n) such that

$$a_1 x_1 + a_2 x_2 + \cdots + a_n x_n \quad = \quad 0$$

(to within the "epsilon" of the arithmetic being used), or else finds bounds within which no relation can exist.

The "PSLQ" algorithm of mathematician-sculptor Helaman Ferguson is the most widely used integer relation algorithm.

Integer relation detection requires very high precision (at least $n \times d$ digits, where d is the size in digits of the largest a_k), both in the input data and the algorithm itself.

- 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), pg. 351–369.
- D. H. Bailey and D. J. Broadhurst, "Parallel Integer Relation Detection: Techniques and Applications," *Mathematics of Computation*, vol. 70, no. 236 (Oct 2000), pg. 1719–1736.

Ferguson's "Umbilic Torus" sculpture at Stony Brook University

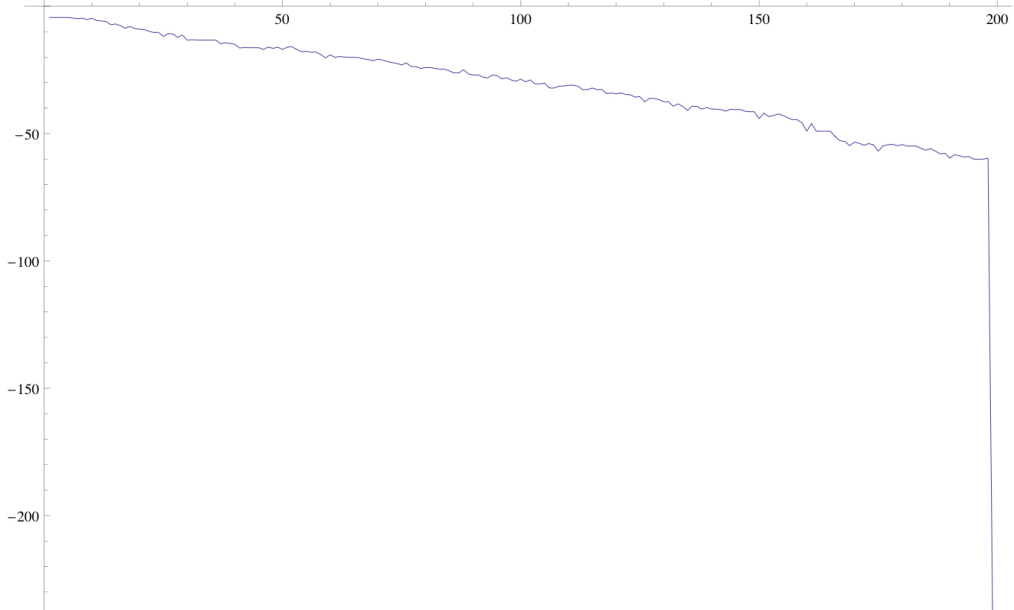
PSLQ, continued

- PSLQ constructs a sequence of integer-valued matrices B_n that reduce the vector $y = x \cdot B_n$, until either the relation is found (as one of the columns of matrix B_n), or else precision is exhausted.
- A relation is detected when the size of smallest entry of the y vector suddenly drops to roughly "epsilon" (i.e. 10^{-p}, where p is the number of digits of precision).
- The size of this drop can be viewed as a "confidence level" that the relation is not a numerical artifact: a drop of 20+ orders of magnitude almost always indicates a real relation.

Efficient variants of PSLQ:

- 2-level and 3-level PSLQ perform almost all iterations with only double precision, updating full-precision arrays as needed. They are hundreds of times faster than the original PSLQ.
- Multi-pair PSLQ dramatically reduces the number of iterations required. It was designed for parallel systems, but runs faster even on 1 CPU.

Decrease of $\log_{10}(\min |y_i|)$ in multipair PSLQ run



The binary algorithm for exponentiation

Problem: What is 3^{17} mod 10?

Algorithm A:
$3^{17} = 3 \times 3 \times 3 \times 3 \times 3 \times 3 \times 3 \times 3 \times 3 \times 3 \times 3 \times 3 \times 3 \times 3 \times 3 \times 3 \times \times 3 = 129140163$,
so answer $= 3$.

Algorithm B (faster): $3^{17} = ((((3^2)^2)^2)^2) \times 3 = 129140163$, so answer $= 3$.

Algorithm C (fastest):
$3^{17} = ((((3^2 \bmod 10)^2 \bmod 10)^2 \bmod 10)^2 \bmod 10) \times 3 \bmod 10 = 3$.

Note that in Algorithm C, we never have to deal with integers larger than $9 \times 9 = 81$.
In general, if reducing mod n, we never have to deal with integers larger than $(n-1)^2$.

1996 result: How to compute arbitrary binary digits of $\log 2$

Consider this well-known formula for $\log 2$:

$$\log 2 = \sum_{n=1}^{\infty} \frac{1}{n2^n} = 0.10110001011100100001011111110111110100011100111101 1\ldots_2$$

Note that the binary digits of $\log 2$ beginning after position d can be written as $\{2^d \log 2\}$, where $\{\cdot\}$ denotes fractional part. Thus we can write:

$$\begin{aligned}\{2^d \log 2\} &= \left\{ \sum_{n=1}^{d} \frac{2^{d-n}}{n} \right\} + \left\{ \sum_{n=d+1}^{\infty} \frac{2^{d-n}}{n} \right\} \\ &= \left\{ \sum_{n=1}^{d} \frac{2^{d-n} \bmod n}{n} \right\} + \left\{ \sum_{n=d+1}^{\infty} \frac{2^{d-n}}{n} \right\},\end{aligned}$$

where we have inserted $\bmod n$ since were are only interested in the fractional part when divided by n. Now note that the numerator $2^{d-n} \bmod n$ can be calculated very rapidly using the binary algorithm for exponentiation, using only standard arithmetic.

Is there a similar formula and scheme for π?

The BBP formula for π

In 1996, a PSLQ program discovered this new formula for π:

$$\pi = \sum_{n=0}^{\infty} \frac{1}{16^k} \left(\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6} \right).$$

This formula permits one to compute binary (or hexadecimal) digits of π beginning at an arbitrary starting position.

This is the first known instance of a computer program discovering a new formula for π.

BBP-type formulas (also discovered using PSLQ) are now known for numerous other mathematical constants.

- D. H. Bailey, P. B. Borwein and S. Plouffe, "On the rapid computation of various polylogarithmic constants," *Mathematics of Computation*, vol. 66 (Apr 1997), 903–913.

Some other BBP-type formulas found using PSLQ

$$\pi^2 = \frac{1}{8}\sum_{k=0}^{\infty}\frac{1}{64^k}\left(\frac{144}{(6k+1)^2} - \frac{216}{(6k+2)^2} - \frac{72}{(6k+3)^2} - \frac{54}{(6k+4)^2} + \frac{9}{(6k+5)^2}\right)$$

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

$$\zeta(3) = \frac{1}{1792}\sum_{k=0}^{\infty}\frac{1}{2^{12k}}\left(\frac{6144}{(24k+1)^3} - \frac{43008}{(24k+2)^3} + \frac{24576}{(24k+3)^3} + \frac{30720}{(24k+4)^3} - \frac{1536}{(24k+5)^3}\right.$$
$$+ \frac{3072}{(24k+6)^3} + \frac{768}{(24k+7)^3} - \frac{3072}{(24k+9)^3} - \frac{2688}{(24k+10)^3} - \frac{192}{(24k+11)^3} - \frac{1536}{(24k+12)^3}$$
$$- \frac{96}{(24k+13)^3} - \frac{672}{(24k+14)^3} - \frac{384}{(24k+15)^3} + \frac{24}{(24k+17)^3} + \frac{48}{(24k+18)^3} - \frac{12}{(24k+19)^3}$$
$$\left. + \frac{120}{(24k+20)^3} + \frac{48}{(24k+21)^3} - \frac{42}{(24k+22)^3} + \frac{3}{(24k+23)^3}\right)$$

- D. H. Bailey, J. M. Borwein, A. Mattingly and G. Wightwick, "The computation of previously inaccessible digits of Pi^2 and Catalan's constant," *Notices of the AMS*, vol. 60 (2013), 844–854.

Normal numbers

Given integer $b \geq 2$, a real number x is b-normal (or "normal base b") if every m-long string of digits appears in the base-b expansion of x with limiting frequency $1/b^m$.

Using measure theory, it can be shown that almost all real numbers are b-normal for a given integer base b (in fact, for all b simultaneously).

These are widely believed to be b-normal, for all integer bases $b \geq 2$ and for all $m \geq 1$:

- $\pi = 3.14159265358979323846\ldots$
- $e = 2.7182818284590452354\ldots$
- $\sqrt{2} = 1.4142135623730950488\ldots$
- $\log(2) = 0.69314718055994530942\ldots$
- *Every* irrational algebraic number (this conjecture is due to Borel).

But there are no proofs of normality for any of these constants. Even the question of whether their binary expansions have an equal number of 0s and 1s is unknown.

Until recently, normality proofs were known only for a few contrived examples such as Champernowne's constant $= 0.12345678910111213141516\ldots$ (which is 10-normal).

BBP formulas and normality

Let $\{\cdot\}$ denote fractional part. Consider the sequence $x_0 = 0$ and

$$x_n = \left\{ 2x_{n-1} + \frac{1}{n} \right\},$$

which is closely connected to the BBP formula for $\log 2$. Richard Crandall and I proved that $\log 2$ is 2-normal if and only if this sequence is equidistributed in the unit interval. Similarly, consider the sequence $x_0 = 0$ and

$$x_n = \left\{ 16x_{n-1} + \frac{120n^2 - 89n + 16}{512n^4 - 1024n^3 + 712n^2 - 206n + 21} \right\},$$

which is closely connected to the BBP formula for π. We proved that π is 16-normal (and hence 2-normal) if and only if this sequence is equidistributed in the unit interval.

- D. H. Bailey and R. E. Crandall, "Random generators and normal numbers," *Experimental Mathematics*, vol. 11, no. 4 (2002), pg. 527–546.
- D. H. Bailey and M. Misiurewicz, "A strong hot spot theorem," *Proceedings of the American Mathematical Society*, vol. 134 (2006), no. 9, pg. 2495–2501.

High-precision numerical integration using the tanh-sinh scheme

Given $f(x)$ defined on $(-1, 1)$, define $g(t) = \tanh(\pi/2 \cdot \sinh t)$. Seting $x = g(t)$ yields

$$\int_{-1}^{1} f(x)\,dx = \int_{-\infty}^{\infty} f(g(t))g'(t)\,dt \;\approx\; h \sum_{j=-N}^{N} w_j f(x_j),$$

where the abscissas x_j and the weights w_j are given by:

$$x_j = g(hj) = \tanh(\pi/2 \cdot \sinh(hj))$$
$$w_j = g'(hj) = \pi/2 \cdot \cosh(hj)/\cosh^2(\pi/2 \cdot \sinh(hj)).$$

Features:

- Halving h (i.e., doubling N) typically *doubles* the number of correct digits.
- The scheme works well even for functions with blow-up singularities at endpoints.
- The cost of computing abscissas and weights increases *linearly* with the number N of subdivisions, whereas it increases *quadratically* for most other schemes.

1. D. H. Bailey, X. S. Li and K. Jeyabalan, "A comparison of three high-precision quadrature schemes," *Experimental Mathematics*, vol. 14 (2005), no. 3, pg. 317–329.
2. H. Takahasi and M. Mori, "Double exponential formulas for numerical integration," *Publications of RIMS*, Kyoto University, vol. 9 (1974), pg. 721–741.

Tanh-sinh quadrature and the Euler-Maclaurin formula

The Euler-Maclaurin formula gives the error in a trapezoidal approximation:

$$\int_a^b f(x)\,\mathrm{d}x = h\sum_{j=0}^{n} f(a+jh) - \frac{h}{2}(f(a)+f(b)) - \sum_{i=1}^{m} \frac{h^{2i}B_{2i}}{(2i)!}\left(D^{2i-1}f(b) - D^{2i-1}f(a)\right) - E_m(h),$$

where $h = (b-a)/n$, B_k are Bernoulli numbers, and

$$|E_m(h)| \le 2(b-a)(h/(2\pi))^{2m+2} \max_{a\le x\le b} |D^{2m+2}f(x)|.$$

Note that when $f(t)$ and all of its derivatives are zero and a and b, then the error $E_m(h)$ of a simple block-function approximation to the integral goes to zero more rapidly than any power of h.

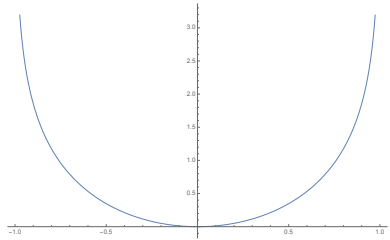
The tanh-sinh substitution $x = g(t)$, where $g(t) = \tanh(\pi/2 \cdot \sinh t)$, converts almost any reasonable integrand into a function that satisfies this condition.

Example of the tanh-sinh transformation

Original function:

$$f(x) = -\log\cos\left(\frac{\pi x}{2}\right)$$

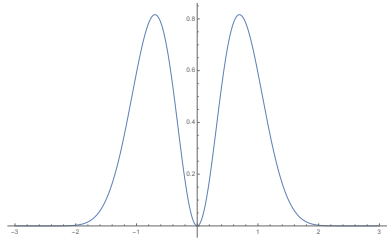
Note singularities at $x = -1$ and $x = 1$.



Transformed function:

$$f(g(t))g'(t) = -\frac{1}{2}\pi\cosh(t)\operatorname{sech}^2\left(\frac{1}{2}\pi\sinh(t)\right) \cdot \log\left(\cos\left(\frac{1}{2}\pi\tanh\left(\frac{1}{2}\pi\sinh(t)\right)\right)\right)$$

The function and all of its derivatives go to zero very rapidly for large negative or positive t.



Ising integrals from mathematical physics

We applied our methods to study three classes of integrals: C_n are connected to quantum field theory, D_n arise in the Ising theory of mathematical physics, while the E_n integrands are derived from D_n:

$$C_n := \frac{4}{n!}\int_0^\infty \cdots \int_0^\infty \frac{1}{\left(\sum_{j=1}^n (u_j + 1/u_j)\right)^2} \frac{\mathrm{d}u_1}{u_1} \cdots \frac{\mathrm{d}u_n}{u_n}$$

$$D_n := \frac{4}{n!}\int_0^\infty \cdots \int_0^\infty \frac{\prod_{i<j}\left(\frac{u_i-u_j}{u_i+u_j}\right)^2}{\left(\sum_{j=1}^n (u_j + 1/u_j)\right)^2} \frac{\mathrm{d}u_1}{u_1} \cdots \frac{\mathrm{d}u_n}{u_n}$$

$$E_n = 2\int_0^1 \cdots \int_0^1 \left(\prod_{1\le j<k\le n} \frac{u_k - u_j}{u_k + u_j}\right)^2 \mathrm{d}t_2\, \mathrm{d}t_3 \cdots \mathrm{d}t_n$$

where in the last line $u_k = t_1 t_2 \cdots t_k$.

- D. H. Bailey, J. M. Borwein and R. E. Crandall, "Integrals of the Ising class," *Journal of Physics A: Mathematical and General*, vol. 39 (2006), pg. 12271–12302.

Limiting value of C_n: What is this number?

Key observation: The C_n integrals can be converted to one- dimensional integrals involving the modified Bessel function $K_0(t)$:

$$C_n \;=\; \frac{2^n}{n!}\int_0^\infty t K_0^n(t)\,\mathrm{d}t$$

High-precision numerical values, computed using this formula and tanh-sinh quadrature, approach a limit. For example:

$$C_{1024} \;=\; 0.63047350337438679612204019271087890435458707 87\ldots$$

What is this number? We copied the first 50 digits into the online Inverse Symbolic Calculator (ISC) at https://isc.carma.newcastle.edu.au. The result was:

$$\lim_{n\to\infty} C_n \;=\; 2e^{-2\gamma}.$$

where γ denotes Euler's constant. This is now proven.

Other Ising integral evaluations found using PSLQ

$$
\begin{aligned}
D_3 &= 8 + 4\pi^2/3 - 27\,\mathrm{L}_{-3}(2) \\
D_4 &= 4\pi^2/9 - 1/6 - 7\zeta(3)/2 \\
E_2 &= 6 - 8\log 2 \\
E_3 &= 10 - 2\pi^2 - 8\log 2 + 32\log^2 2 \\
E_4 &= 22 - 82\zeta(3) - 24\log 2 + 176\log^2 2 - 256(\log^3 2)/3 \\
&\quad +16\pi^2\log 2 - 22\pi^2/3 \\
E_5 &= 42 - 1984\,\mathrm{Li}_4(1/2) + 189\pi^4/10 - 74\zeta(3) - 1272\zeta(3)\log 2 + 40\pi^2\log^2 2 \\
&\quad -62\pi^2/3 + 40(\pi^2\log 2)/3 + 88\log^4 2 + 464\log^2 2 - 40\log 2
\end{aligned}
$$

where ζ is the Riemann zeta function and $\mathrm{Li}_n(x)$ is the polylog function. E_5 remained a "numerical conjecture" for several years, but was proven in March 2014 by Erik Panzer.

Using PSLQ to find the minimal polynomial of an algebraic number

One simple but important application of PSLQ is to recognize a computed numerical value as the root of an integer polynomial of degree m.

Example: The following constant is suspected to be an algebraic number:

$\alpha = 1.232688913061443445331472869611255647068988824547930576057634684778\ldots$

What is its minimal polynomial?

Method: Compute the vector $(1, \alpha, \alpha^2, \cdots, \alpha^m)$ for $m = 30$, then input this vector to a PSLQ program. The output integer vector is precisely the set of coefficients of the minimal polynomial of α.

Answer (using 250-digit arithmetic):

$$\begin{aligned} 0 = {} & 697 - 1440\alpha - 20520\alpha^2 - 98280\alpha^3 - 102060\alpha^4 - 1458\alpha^5 + 80\alpha^6 - 43920\alpha^7 \\ & + 538380\alpha^8 - 336420\alpha^9 + 1215\alpha^{10} - 80\alpha^{12} - 56160\alpha^{13} - 135540\alpha^{14} - 540\alpha^{15} \\ & + 40\alpha^{18} - 7380\alpha^{19} + 135\alpha^{20} - 10\alpha^{24} - 18\alpha^{25} + \alpha^{30} \end{aligned}$$

The Poisson potential function

In 2012, Richard Crandall, while investigating techniques to sharpen images, noted that each pixel was given by a form of the 2-D Poisson potential function:

$$\phi_2(x,y) = \frac{1}{\pi^2} \sum_{m,n \text{ odd}} \frac{\cos(m\pi x)\cos(n\pi y)}{m^2+n^2}$$

In a 2013 study, we numerically discovered, and then proved the intriguing fact that for rational numbers x and y,

$$\phi_2(x,y) = \frac{1}{\pi} \log \alpha$$

where α is *algebraic*, i.e., the root of a some integer polynomial of degree m.

By computing high-precision numerical values of $\phi_2(x,y)$ for various specific rational x and y, and applying a multipair PSLQ program, we were able to produce the explicit minimal polynomials for α in numerous specific cases.

- D. H. Bailey, J. M. Borwein, R. E. Crandall and J. Zucker, "Lattice sums arising from the Poisson equation," *Journal of Physics A: Mathematical and Theoretical*, vol. 46 (2013), 115201.

Samples of minimal polynomials found by PSLQ

s	Minimal polynomial corresponding to $x = y = 1/s$:
5	$1 + 52\alpha - 26\alpha^2 - 12\alpha^3 + \alpha^4$
6	$1 - 28\alpha + 6\alpha^2 - 28\alpha^3 + \alpha^4$
7	$-1 - 196\alpha + 1302\alpha^2 - 14756\alpha^3 + 15673\alpha^4 + 42168\alpha^5 - 111916\alpha^6 + 82264\alpha^7$ $-35231\alpha^8 + 19852\alpha^9 - 2954\alpha^{10} - 308\alpha^{11} + 7\alpha^{12}$
8	$1 - 88\alpha + 92\alpha^2 - 872\alpha^3 + 1990\alpha^4 - 872\alpha^5 + 92\alpha^6 - 88\alpha^7 + \alpha^8$
9	$-1 - 534\alpha + 10923\alpha^2 - 342864\alpha^3 + 2304684\alpha^4 - 7820712\alpha^5 + 13729068\alpha^6$ $-22321584\alpha^7 + 39775986\alpha^8 - 44431044\alpha^9 + 19899882\alpha^{10} + 3546576\alpha^{11}$ $-8458020\alpha^{12} + 4009176\alpha^{13} - 273348\alpha^{14} + 121392\alpha^{15}$ $-11385\alpha^{16} - 342\alpha^{17} + 3\alpha^{18}$
10	$1 - 216\alpha + 860\alpha^2 - 744\alpha^3 + 454\alpha^4 - 744\alpha^5 + 860\alpha^6 - 216\alpha^7 + \alpha^8$

The case $x = y = 1/32$ required 10,000-digit arithmetic and ran for 45 hours.

Other runs, using even higher precision, ultimately failed. HELP!

Kimberley's formula for the degree of the polynomial

Based on our preliminary results, Jason Kimberley of the University of Newcastle, Australia observed that the degree $m(s)$ of the minimal polynomial associated with the case $x = y = 1/s$ appears to be given by the following:

Set $m(2) = 1/2$. Otherwise for primes p congruent to 1 mod 4, set $m(p) = \mathrm{int}^2(p/2)$, where int denotes greatest integer, and for primes p congruent to 3 mod 4, set $m(p) = \mathrm{int}\,(p/2)(\mathrm{int}\,(p/2) + 1)$. Then for any other positive integer s whose prime factorization is $s = p_1^{e_1} p_2^{e_2} \cdots p_r^{e_r}$,

$$m(s) = 4^{r-1} \prod_{i=1}^{r} p_i^{2(e_i-1)}\, m(p_i).$$

Does Kimberley's formula hold for larger s? Why?

What is the true mathematical connection between the pair of rationals (x, y) and the algebraic number α?

Three improvements to the Poisson polynomial computation program

1. MPFUN-MPFR: A new thread-safe multiprecision package.
   - Speedup: 3X
2. A new 3-level multipair PSLQ program.
   - Speedup: 4.2X
3. Parallel implementation on a 16-core system.
   - Speedup: 12.2X

Overall speedup: 156X

Total size of code: 190,000 lines

A fast algorithm to compute the Poisson potential function $\phi_2(x, y)$

Computing $\phi_2(x, y)$ by the definition (see earlier slide) is much too slow, but high-precision values can be obtained by this formula:

$$\phi_2(x, y) = \frac{1}{2\pi} \log \left| \frac{\theta_2(z, q)\theta_4(z, q)}{\theta_1(z, q)\theta_3(z, q)} \right|,$$

where $q = e^{-\pi} = 0.0432139\ldots$, and $z = \frac{\pi}{2}(y + ix)$. The four theta functions can be computed using the following rapidly convergent formulas:

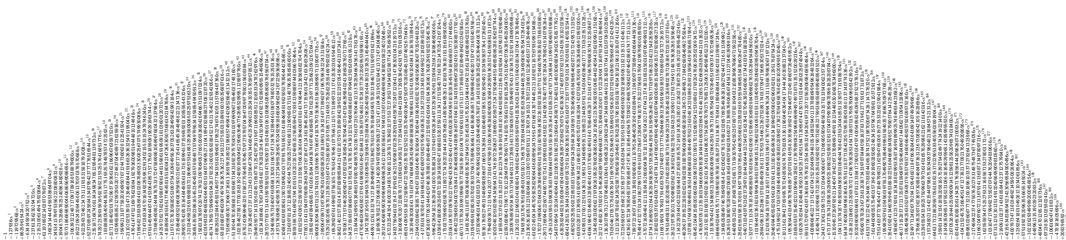
$$\theta_1(z, q) = 2 \sum_{k=1}^{\infty} (-1)^{k-1} q^{(2k-1)^2/4} \sin((2k-1)z),$$

$$\theta_2(z, q) = 2 \sum_{k=1}^{\infty} q^{(2k-1)^2/4} \cos((2k-1)z),$$

$$\theta_3(z, q) = 1 + 2 \sum_{k=1}^{\infty} q^{k^2} \cos(2kz),$$

$$\theta_4(z, q) = 1 + 2 \sum_{k=1}^{\infty} (-1)^k q^{k^2} \cos(2kz).$$

192-degree minimal polynomial found by multipair PSLQ for $x = y = 1/35$



This polynomial has degree 192, with coefficients as large as 10^{85}. This computation required 18,000-digit arithmetic and 34 CPU-hours.

The case $(1/37, 1/37)$ required 51,000-digit arithmetic and 90 CPU-days (5.6 days on a 16-core parallel system).

Kimberley's formula was upheld for $(1/s, 1/s)$, for all s up to 52 (except for $s = 41, 43, 47, 49, 51$, which were too expensive), and also for $s = 60$ and $s = 64$.

Palindromic polynomials

From our results, in the case $(1/s, 1/s)$ where s is even, the resulting polynomial is always palindromic ($a_k = a_{m-k}$). For instance, when $s = 16$,

$$\begin{aligned}
p_{16}(\alpha) = {} & 1 - 1376\alpha^1 - 12560\alpha^2 - 3550496\alpha^3 + 81241720\alpha^4 - 169589984\alpha^5 \\
& + 1334964944\alpha^6 - 24307725984\alpha^7 + 238934926108\alpha^8 - 1043027124704\alpha^9 \\
& + 2328675366384\alpha^{10} - 3219896325280\alpha^{11} + 4238551472456\alpha^{12} \\
& - 10247414430048\alpha^{13} + 28552105805904\alpha^{14} - 55832851687968\alpha^{15} \\
& + 70020268309062\alpha^{16} \\
& - 55832851687968\alpha^{17} + 28552105805904\alpha^{18} - 10247414430048\alpha^{19} \\
& + 4238551472456\alpha^{20} - 3219896325280\alpha^{21} + 2328675366384\alpha^{22} \\
& - 1043027124704\alpha^{23} + 238934926108\alpha^{24} - 24307725984\alpha^{25} + 1334964944\alpha^{26} \\
& - 169589984\alpha^{27} + 81241720\alpha^{28} - 3550496\alpha^{29} - 12560\alpha^{30} - 1376\alpha^{31} + \alpha^{32}
\end{aligned}$$

Nitya Mani, an undergraduate student at Stanford University, observed that if α is a root of a palindromic polynomial such as this, then $\alpha + 1/\alpha$ is a root of a transformed polynomial of half the degree.

Proofs of Kimberley's formula and the palindromic property

- On March 16, DHB presented our results at a seminar at the University of California, Berkeley.
- Following the presentation, Watson Ladd, a graduate student in mathematics, brought to our attention the fact that some of our conjectures should follow from results in the theory of elliptic curves, Gaussian integers and ideals.
- After some effort, Ladd produced proofs of Kimberley's formula and the palindromic property, which proofs were then included in our paper.

- D. H. Bailey, J. M. Borwein, J. Kimberley and W. Ladd, "Computer discovery and analysis of large Poisson polynomials," *Experimental Mathematics*, 27 Aug 2016, `http://www.davidhbailey.com/dhbpapers/poisson-res.pdf`.

Mordell-Tornheim-Witten sums and the omega function

Mordell-Tornheim-Witten sums are defined as:

$$W(r_1, r_2, \ldots, r_n, t) = \sum_{m_1,\ldots,m_n \geq 1} \frac{1}{m_1^{r_1} m_2^{r_2} \cdots m_n^{r_n} (m_1 + m_2 + \cdots + m_n)^t}.$$

They can be continued analytically, as specified by Matsumoto.

Such sums arise in combinatorics, mathematical physics (e.g., Feynmann diagrams and string theory), Lie algebras, number theory and numerous other fields.

We will focus on the special case $r_1 = r_2 = \cdots = r_n = t = s$ for real s (analytically continued as above), namely

$$\omega_{n+1}(s) = \sum_{m_1,m_2,\ldots,m_n \geq 1} \frac{1}{(m_1 m_2 \cdots m_n (m_1 + m_2 + \cdots + m_n))^s}.$$

- D. Romik, "On the number of n-dimensional representations of SU(3), the Bernoulli numbers, and the Witten zeta function," 17 Apr 2015, `https://arxiv.org/pdf/1503.03776v3.pdf`.
- H. Tomkins, "An exploration of multiple zeta functions," Honours Thesis, Dalhousie University, 21 Apr 2016.

The degree-two omega function

The degree-two instance of the omega function is

$$\omega(s) = \sum_{j,k=1}^{\infty} \frac{1}{(jk(j+k))^s}.$$

For positive integer arguments, papers by Subbarao, Sitaramachandrarao, Garoufalidis, Zagier, Huard, Williams and Nan-Yue have shown that for positive integers n,

$$\omega(2n) = \frac{4}{3} \sum_{k=0}^{n} \binom{4n-2k-1}{2n-1} \zeta(2k)\zeta(6n-2k)$$

$$\omega(2n+1) = -4 \sum_{k=0}^{n} \binom{4n-2k-1}{2n} \zeta(2k)\zeta(6n-2k+3).$$

Specific values of omega

$$\omega(1) = 2\zeta(3)$$
$$\omega(2) = \frac{1}{2835}\pi^6$$
$$\omega(3) = -2\pi^2\zeta(7) + 20\zeta(9)$$
$$\omega(4) = \frac{19}{273648375}\pi^{12}$$
$$\omega(5) = -\frac{2}{9}\pi^4\zeta(11) - \frac{70}{3}\pi^2\zeta(13) + 252\zeta(15)$$
$$\omega(6) = \frac{2062}{116937886440375}\pi^{18}$$
$$\omega(7) = -\frac{4}{135}\pi^6\zeta(15) - \frac{56}{15}\pi^4\zeta(17) - 308\pi^2\zeta(19) + 3432\zeta(21)$$
$$\omega(8) = \frac{32293}{7060124893837640625}\pi^{24}$$

Romik's formulas for omega

Dan Romik has shown that

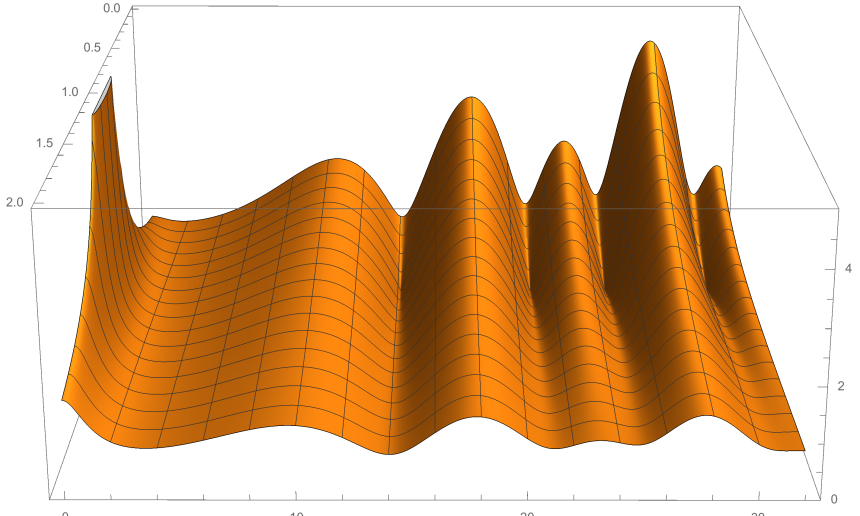
$$\omega(s) = \frac{1}{2\pi\Gamma(s)} \int_{(\alpha)} \Gamma(s+z)\Gamma(-z)\zeta(2s+z)\zeta(s-z)\,\mathrm{d}z,$$

where (α) means the vertical line $\{\alpha + ti, -\infty < t < \infty\}$. This is valid for all complex s satisfying $\max(-\mathrm{Re}(s), 1 - 2\mathrm{Re}(s)) < \alpha < \min(0, \mathrm{Re}(s) - 1)$.

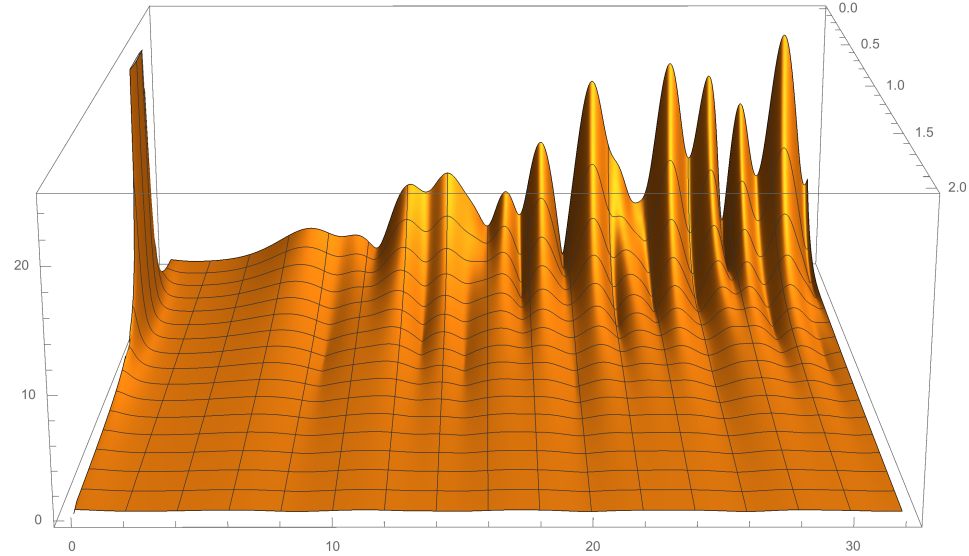
$$\begin{aligned}\omega(s) = \frac{1}{\Gamma(s)}\Bigg(&\Gamma(2s-1)\Gamma(1-s)\zeta(3s-1) \\ &+ \sum_{k=0}^{M-1} \frac{(-1)^k}{k!}\Gamma(s+k)\zeta(2s+k)\zeta(s-k) \\ &+ \frac{1}{2\pi}\int_{(\alpha)} \Gamma(s+z)\Gamma(-z)\zeta(2s+z)\zeta(s-z)\,\mathrm{d}z\Bigg).\end{aligned}$$

This is valid for all complex s satisfying $3/4 - M/2 < \mathrm{Re}(s) < M + 1/2$.

The Riemann zeta function in the complex plane



The omega function



Research questions for omega

- Does omega have any zeroes in the complex plane? Where? How many?
- Does omega have nice analytic formulas for any arguments other than positive integers? Can we discover these formulas experimentally?
- Can we do a similar analysis of the derivative of omega? For higher derivatives?
- How about for higher-degree omega?

The argument principle

Note, by the argument principle, that the number of zeroes of $\omega(s)$ in a pole-free region bounded by a contour C in the complex plane can be computed as:

$$-\frac{1}{2\pi i}\int_{(C)}\frac{\omega'(s)}{\omega(s)}\,\mathrm{d}s$$

Computing the omega function

Key elements of computing $\omega(s)$ for complex s:

- Evaluating $\Gamma(s)$.
- Evaluating $\zeta(s)$.
- Evaluating Romik's integral formula.

To count zeroes in a region using the argument principle, it is also necessary to compute the derivative $\omega'(s)$. This can be done by

$$\omega'(s) \approx \frac{\omega(s+h) - \omega(s)}{h},$$

where $h = 10^{-p/2}$ and p is the working precision level in digits (yielding the derivative accurate to $p/2$ digits).

The reliability of these results strongly depends on performing the computations to convincingly high precision — more than standard 64-bit floating-point arithmetic.

Computing gamma and zeta

The best way to compute $\Gamma(s)$ is to use a formula originally due to Spouge:

$$\Gamma(1+z) \approx (z+a)^{z+1/2} e^{-z-a} \sqrt{2\pi} \left(1 + \sum_{k=1}^{\lceil a \rceil - 1} \frac{c_k(a)}{z+k} \right),$$

where $a =$ number of desired digits, and

$$c_k(a) = \frac{(-1)^{k-1}(a-k)^{k-1/2} e^{-k+a}}{\sqrt{2\pi}(k-1)!}.$$

The best way to compute $\zeta(s)$ is to use a formula due to Richard Crandall:

$$\zeta(s) = \frac{1}{\Gamma(s/2)} \left[\frac{\pi^{s/2}}{s(s-1)} + \sum_{n=1}^{\infty} \left(\frac{\Gamma(s/2, \pi n^2)}{n^s} + \frac{\pi^{s-1/2} \Gamma((1-s)/2, \pi n^2)}{n^{1-s}} \right) \right],$$

where $\Gamma(s,x)$ is the incomplete gamma function:

$$\Gamma(s,x) = \Gamma(s) \left(1 - x^s e^{-x} \sum_{k=0}^{\infty} \frac{x^k}{\Gamma(s+k+1)} \right).$$

Software for high-precision computation

- Mathematica and Maple: Both feature well-integrated arbitrary precision facilities, but are often quite slow.
- Sage: Also includes arbitrary precision facility, but not as many functions.
- gfortran's quad precision: Now includes full support for `real(16)` and `complex(16)` datatypes (128-bit IEEE arithmetic, approximately 33 digits).
- GMP: GNU-supported package for arbitrary precision *integer* computation.
- MPFR: Very efficient package for arbitrary precision *floating-point* computation, built on top of GMP; does not include high-level language facilities.
- DHB's software (each includes high-level language support):
  - QD: Double-double (31 digits) and quad-double (62 digits); C++ and Fortran.
  - ARPREC: Arbitrary precision floating-point package; C++ and Fortran.
  - MPFUN2015: New Fortran arbitrary precision package, based on MPFR; Fortran.

What has been done

These have been implemented, using the MPFUN2015 multiprecision software:

- Routine to compute $\Gamma(s)$ for any complex s.
- Routine to compute $\Gamma(s,x)$ for any complex s and all real x.
- Routine to compute $\zeta(s)$ for any complex s.
- Routine to compute $\omega(s)$ for any complex s, using Romik's integral formulas and the tanh-sinh quadrature algorithm (using MPFUN2015).
- Computation of the contour integral for the rectangular contour $(0,1/8)$ to $(2,1/8)$ to $(2,32)$ to $(0,32)$, and back to $(0,1/8)$, all using 70-digit precision.
- The contour integral calculation has been running 20 days and is still not yet complete!

What remains to be done

- Run times are ridiculously high. Can the program be tuned to run faster? Use different high-precision software? Run on a parallel system?
- Can a much larger range of complex arguments be explored, looking for zeroes, perhaps not necessarily using the argument principle?
- Can we do the same for the derivative of omega or higher-order derivatives? Can we do the same for higher-degree omega?

Any volunteers?

Contact me at: dhbailey@ucdavis.edu

This talk is available at:
http://www.davidhbailey.com/dhbtalks/dhb-ucdavis-omega.pdf