Proof checking software: A paradigm-changing development for mathematical research

- Lean: Initially written by Leonardo de Moura at Microsoft Research; first released in 2017; improved, community-supported version released in 2021.
- Isabelle: Originally written in 1986 by Lawrence Paulson; the current version is from 2024. This was used by Thomas Hales to certify his proof of the Kepler Conjecture in 2014.
- HOL Light: Originally written by John Harrison, drawing on work by several others dating back to the 1980s.

These tools permit one to rigorously confirm all steps of a formal mathematical proof, thus greatly facilitating collaborative mathematical research.

This talk:
David H. Bailey, "Computational mathematics: From pariah to paradigm," available at:
https://www.davidhbailey.com/dhbtalks/dhb-ams-2025.pdf

Terence Tao on AI and proof checkers in mathematical research

"Now you can really collaborate with hundreds of people that you've never met before. And you don't need to trust them, because they upload code and the Lean compiler verifies it. You can do much larger-scale mathematics than we do normally. When I formalized our most recent results with what is called the Polynomial Freiman-Ruzsa (PFR) conjecture, [I was working with] more than 20 people. We had broken up the proof in lots of little steps, and each person contributed a proof to one of these little steps. And I didn't need to check line by line that the contributions were correct. I just needed to sort of manage the whole thing and make sure everything was going in the right direction. It was a different way of doing mathematics, a more modern way."

- C. Drosser, "AI will become mathematicians' 'co-pilot'," *Scientific American*, 8 Jun 2024, `www.scientificamerican.com/article/ai-will-become-mathematicians-co-pilot/`
- M. Wong, "We're entering uncharted territory for math," *Atlantic*, 4 Oct 2024, `https://www.theatlantic.com/technology/archive/2024/10/terence-tao-ai-interview/680153/`

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" and "multipair PSLQ" algorithms of mathematician-sculptor Helaman Ferguson are among the most widely used integer relation algorithms.

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 in the algorithm steps.

1. 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), 351–369.
2. D. H. Bailey and D. J. Broadhurst, "Parallel integer relation detection: Techniques and applications," *Mathematics of Computation*, vol. 70, no. 236 (Oct 2000), 1719–1736.

How to compute binary digits of $\log 2$ at an arbitrary starting position

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 $\mathrm{frac}(2^d \log 2)$, where frac denotes fractional part. Thus we can write:

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

where we have inserted $\bmod n$ since were are only interested in the fractional part when divided by n. The numerator $2^{d-n} \bmod n$ can be calculated very rapidly using the binary algorithm for exponentiation. This can be done using quad-precision arithmetic.

Is there a similar formula and computational scheme for π? None was known in 1996.

The BBP formula for π

In 1996, a PSLQ-like computer 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, without needing to computing any of the preceding digits.

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

Other BBP-type formulas, mostly 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.

Using PSLQ to find the minimal polynomial of an algebraic number

Example: The following constant is suspected to be a degree-30 algebraic number:

$\alpha = 1.23268891306144344533147286961125564706898882454793057605763468477 8\ldots$

What is its minimal polynomial?

Method: Compute the vector $(1, \alpha, \alpha^2, \cdots, \alpha^{30})$ to at least 250-digit arithmetic, then input this vector to PSLQ.

Result:

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

The Poisson potential function

The Poisson potential function appears in mathematical physics and also in practical applications such as sharpening iPhone images. A simple 2-D instance is:

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

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

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

where $\beta(x,y)$ is an algebraic number.

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

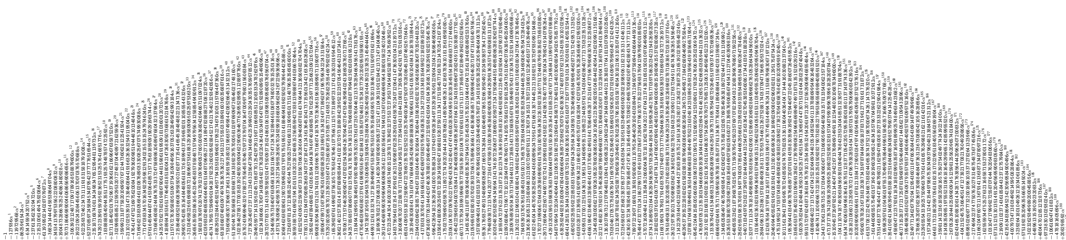
Samples of minimal polynomials found by PSLQ

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	$1 - 216\alpha + 860\alpha^2 - 744\alpha^3 + 454\alpha^4 - 744\alpha^5 + 860\alpha^6 - 216\alpha^7 + \alpha^8$

What is the relationship between the denominator s and the degree of the polynomial?

Also, does the palindromic property for even s above extend to larger cases?

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 floating-point arithmetic and 34 CPU-hours run time.

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

These computations confirmed Kimberley's formula (see next slide) for $(x, y) = (1/s, 1/s)$, for most s up to 52 and also for $s = 60$ and $s = 64$.

Kimberley's formula for the degree of the minimal polynomial

Based on preliminary computational 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 $s = p_1^{e_1} p_2^{e_2} \cdots p_r^{e_r}$,

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

Further research ultimately led to a proof of Kimberley's formula in 2016.

Much more extensive computations found a tentative modification of Kimberley's formula for the more general case $(x, y) = (p/s, q/s)$ for integers $1 \le p < q < s/2$.

- D. H. Bailey, J. M. Borwein, J. Kimberley and W. Ladd, "Computer discovery and analysis of large Poisson polynomials," *Experimental Mathematics*, 27 Aug 2016, vol. 26, 349–363.

December 2024: Results for the Poisson ψ function

The 2013 study also briefly mentioned the closely related function

$$\psi_2(x,y) = \frac{1}{\pi^2} \sum_{m,n \text{ even}} \frac{\cos(\pi m x)\cos(\pi n y)}{m^2+n^2}.$$

As with $\phi_2(x,y)$, the authors found that when x and y are rational, then

$$\psi_2(x,y) = \frac{1}{\pi} \cdot \log \beta(x,y),$$

for algebraic $\beta(x,y)$.

A handful of results were given in the 2013 study, but progress has been stymied by an error in the formulas derived in that study for fast numerical evaluation. This has now been corrected, permitting large-scale numerical explorations.

However, the computations and analysis for $\psi_2(x,y)$ are many times more challenging than with $\phi_2(x,y)$. Some individual cases have required up to 160,000-digit floating-point arithmetic and over 200 CPU-hours run time.

Degree-36 minimal polynomial found for the case $(x, y) = (1/13, 1/13)$

$$
\begin{aligned}
&+1\,\alpha \\
&-102008900\,\alpha^{1} \\
&+3386359201083610\,\alpha^{2} \\
&-4576743060352240027036\,\alpha^{3} \\
&+2358487814308768868232551144847\,\alpha^{4} \\
&-4018085951546127674633453040190691 4\,\alpha^{5} \\
&+3226393199644244340609969690827653454664 92\,\alpha^{6} \\
&-1289351965036787056558584361620156261860934499 26\,\alpha^{7} \\
&+2543661517206950398252099472523978556653522475994 0543\,\alpha^{8} \\
&-1835635719561759818191190195010167655727243690089160673300\,\alpha^{9} \\
&+1295242928423847804911870979061052075344554600559285562745051 1\,\alpha^{10} \\
&-5064396407665154813418840619774597239924756002045577036668076918794\,\alpha^{11} \\
&+1193514740202119424326796180993793510186386708001603921356167265630727 11\,\alpha^{12} \\
&-2244708253496400477104375428540337510408970027526179623030216898818137766158\,\alpha^{13} \\
&-8707860049061337830962269810752798082555475127186562670214864792068274557045949 2\,\alpha^{14} \\
&-1406389080545060285674447683929282293450220435935950931746315677313494777509752153540\,\alpha^{15} \\
&-1138575802554489425677758018772417062354823189468556911582399954138129596503436139030107 66\,\alpha^{16} \\
&-4434711394555846777074046621767781359205585851315181808393626242673174104041251410550289108570\,\alpha^{17} \\
&-8042918502598934232626067127854569308476504829683851787818042565917754813340313503762861939149 5\,\alpha^{18} \\
&-7760924826076994351341045848120769124383934262753787967089934458610635066219704944333213767061 4362\,\alpha^{19} \\
&-9848697296028693505859742747515875272579777838980028076986754643210935843092324178814413511757394 0304\,\alpha^{20} \\
&-1969165091085612807263480595261734799867643994728228645350075151313147370138979643245990858532945372205916\,\alpha^{21} \\
&+9309918787385892745969832005705814375571916352095853488486481765483651912350877087306889144827987050001\,\alpha^{22} \\
&-1066645601082853770577922620932336387880850553390339802716450308583105788248755953683857891159730107675556\,\alpha^{23} \\
&+3541772939676430611417629892952843731851794553110271554764233110846188885153863684384031589356947520154800 23\,\alpha^{24} \\
&+1160699604655502745585716747716388874852245125268657789065785705020328510306870444459268172813586654114359 64\,\alpha^{25} \\
&+8813248335271366720387889938638771316889960935147767428737296965184748883073959063862357217702136814803447 89\,\alpha^{26} \\
&+1667073800369010345168880977008798389289258301449119121624216476871337035450251555949126852167017096872 0238\,\alpha^{27} \\
&-7047103822160612268005331243634945797364875821873084144157723975480184702717917697991110329045401091 22\,\alpha^{28} \\
&+1904835257940803431486370558681639277544588932269608909710967729758295491903807063932456206459754\,\alpha^{29} \\
&+6858013750514713218193500096025262909884261395675712560592001216243402544443202236983867768 9\,\alpha^{30} \\
&-2428521221981343663201504374289792787846829153650245387685427372653242794182945316036\,\alpha^{31} \\
&-1509371819989613719407627286923309655042276667477193799678416335252986311 77\,\alpha^{32} \\
&-9677101261822310858677775011018349995290070484887565356379435 16\,\alpha^{33} \\
&-2876217408417712578461604560530446919799847382399 7\,\alpha^{34} \\
&+3462628969701716790046955098 6\,\alpha^{35} \\
&+1\,\alpha^{36}
\end{aligned}
$$

Initial results for the Poisson ψ function

From computations so far, it appears that Kimberley's rule also holds for polynomials associated with $\psi_2(p/s, q/s)$, except that for even s, the polynomial degree is usually half the degree of the corresponding $\phi_2(p/s, q/s)$ polynomial.

None of these results would be possible without the emergence of very powerful 21st-century computer hardware and software.

As computer technology continues to advance, what new types of mathematical results will be discovered? Will an AI make the next set of discoveries?

This talk:
David H. Bailey, "Computational mathematics: From pariah to paradigm," available at:
https://www.davidhbailey.com/dhbtalks/dhb-ams-2025.pdf