Experimental Mathematics in the Society of the Future

David H. Bailey[*] Jonathan Borwein[†]

January 11, 2015

Abstract

Computer-based tools for mathematics are changing how mathematics is researched, taught and communicated to society. Future technology trends point to ever-more powerful tools in the future. Computation in mathematics is thus giving rise to a new mode of mathematical research, where algorithms, datasets and public databases are as significant as the resulting theorems, and even the definition of what constitutes secure mathematical knowledge is seen in a new light.

1 Introduction

Like most other fields of scientific research, both pure and applied mathematics have been significantly affected in recent years by the introduction of modern computer technology. Just like their peers in other fields, mathematicians are using the computer as a "laboratory" to perform exploratory experiments and test conjectures, in a methodology that has been termed "experimental mathematics." The adjective "experimental" here is entirely appropriate, because, in a fundamental sense, there is little difference between a mathematician using a computer to explore the mathematical universe and an astronomer using a large telescope facility to explore the physical universe.

By *experimental mathematics* we mean the following *computationally-assisted* approach to mathematical research [11]:

1. *Gaining insight* and intuition;
2. *Visualizing* mathematical principles;
3. *Discovering* new relationships;
4. *Testing* and especially *falsifying* conjectures;
5. *Exploring* a possible result to see if it merits formal proof;

[*] Lawrence Berkeley National Laboratory (retired), Berkeley, CA 94720, and University of California, Davis, Davis, CA 95616. Email: `david@davidhbailey.com`.

[†] Centre for Computer Assisted Research Mathematics and its Applications (CARMA), University of Newcastle, Callaghan, NSW 2308, Australia. Email: `jon.borwein@gmail.com`.

6. *Suggesting* approaches for formal proof;
7. *Replacing* lengthy and error-prone hand derivations;
8. *Confirming* analytically derived results.

With regards to 6, we have often found that computer-based tools are useful to tentatively confirm preliminary lemmas; then we can proceed fairly safely to see where they lead. If, at the end of the day, this line of reasoning has not led to anything of significance, at least we have not expended large amounts of time attempting to formally prove these lemmas. And if computer tests falsify a conjecture, then no need to waste any time at all seeking a formal proof.

While computation is proving very useful in the exploration phase, computers are also being employed to produce formal proofs of mathematical results. One notable recent success, just concluded, is Thomas Hales' computer-based formal proof of the Kepler conjecture [15], a topic that we will revisit in Section 4.4.

2 Experimental methods in education

While some still resist, it is clear that computational tools are the wave of the future for mathematics instruction, certainly not replacing the instructor or hand computations and algebraic manipulations, but, instead, permitting mathematical principles to be taught with less pain and greater understanding.

To mention just one simple example, many of us recall using algebraic substitutions to rotate a geometric figure given by a formula. Nowadays such rotations can be performed easily using computer graphics tools. Indeed, modern computer technology places the cart before the horse — rather than using advanced algebra and calculus as tools to graph functions, instead we can now use computer-based graphics tools to learn principles of algebra and calculus.

As another simple example, we are taught in elementary calculus that a definite integral can be seen as the area under a curve, in particular the limit found by adding the areas of rectangles under a curve, subdividing the interval into finer and finer parts. While a few very simple examples of this sort can be done by hand algebra, it is arguably more instructive for students to let a computer program do the hard work. For example, by employing a simple trapezoidal approximation to evaluate the integral $\int_0^1 \mathrm{d}x/(1+x^2)$, with 10,000 subdivisions, one obtains the numerical result $0.78539815\ldots$, which is accurate enough for the Inverse Symbolic Calculator 2.0, available at `http://isc.carma.newcastle.edu.au` to identify as likely to be $\pi/4$ (it is, of course, equal to $\pi/4$).

In Section 4, we will present a few somewhat more sophisticated examples of experimental mathematics in action. Here we present two that require only a very modest background, and are exemplars of how computation can be incorporated into education even at the high school level.

2.1 A number theory example

Many high school students learn that the sum of the first n integers is $n(n+1)/2$. Indeed, Gauss is reputed to have discovered this formula by himself in elementary school. What about sums of higher powers? The simple *Mathematica* command `Sum[k^5,{k,1,n}]` returns the formula

$$1^5 + 2^5 + \cdots + n^5 = \frac{1}{12} n^2 (2n^2 + 2n - 1)(n+1)^2. \tag{1}$$

Note that by typing the command `F5[n_]:=n^2(2n^2+2n-1)(n+1)^2; Simplify[F5[m]-F5[m-1]]`, one can symbolically determine that the difference between formula (1) evaluated at an integer n and at $n-1$ is n^5, which, since the formula is clearly valid for $n = 1$, constitutes a proof by induction that formula (1) is valid for all positive integers n.

In a similar vein, one can use the computer to explore sums of even higher powers. For example, using either *Maple* or *Mathematica*, one obtains the formula

$$\begin{aligned}\sum_{k=1}^{n} k^{10} = {} & \frac{1}{66} n(2n+1)(n+1)(n^2+n-1) \\ & \cdot (3n^6 + 9n^5 + 2n^4 - 11n^3 + 3n^2 + 10n - 5),\end{aligned} \tag{2}$$

so that, for example,

$$\sum_{k=1}^{10{,}000} k^{10} = 9095909924242414242424342424241924242425000. \tag{3}$$

Note the curious pattern of 42 repeated numerous times (except for the central 3) in the center of this number. What is the explanation? By using *Maple* or *Mathematica* to expand formula (2), one obtains

$$\sum_{k=1}^{n} k^{10} = \frac{1}{66} \left(6n^{11} + 33n^{10} + 55n^9 - 66n^7 + 66n^5 - 33n^3 + 5n\right). \tag{4}$$

Note that the fourth and fifth terms are -66 and 66, respectively, which, when divided by 66, are -1 and 1. Also note that without the 1/66, the sum (3) above would be:

$$66 \sum_{k=1}^{10{,}000} k^{10} = 600330054999999340000006599999967000000050000. \tag{5}$$

In this form, the correspondence between (4) and (5) is clear — by examining (5), one can literally read off the coefficients of (4) term by term (remembering that 999 . . . is a key for a negative coefficient). When we evaluate the leading three leading terms of (4) for $n = 10,000$, it gives an integer that, when divided by 66, gives a decimal value that terminates in 4242424242 . . ., which is the source of the 42s above. The central 3 in (3) is produced by the term $66n^5$ in (4), which, when divided by 66, is just n^5, adding one to the decimal digit 2 that is normally in this position.

2.2 An anomaly in computing pi

Gregory's series, discovered in the 17th century, is arguably the most elementary infinite series formula for π, although it converges rather slowly. It can be simply derived by simply noting that

$$\begin{aligned}\frac{\pi}{4} = \int_0^1 \frac{\mathrm{d}x}{1+x^2} &= \int_0^1 \left(1 - x^2 + x^4 - x^6 + x^8 - \cdots\right) \mathrm{d}x \\ &= 1 - 1/3 + 1/5 - 1/7 + 1/9 - 1/11 + \cdots \qquad (6)\end{aligned}$$

In 1988, a colleague noted that Gregory's series, when evaluated to 5,000,000 terms by computer, gives a value that differs strangely from the true value of π. Here is the truncated Gregory value and the true value of π:

```
3.14159245358979323846464338327950278419716939938730582097494182230...
3.14159265358979323846264338327950288419716939937510582097494459230...
        2            -2            10          -122          2770
```

The series value differs, as one might expect from a series truncated to 5,000,000 terms, in the seventh decimal place—a "4" where there should be a "6" (namely an error of 2). But the next 13 digits are correct! Then, following another erroneous digit, the sequence is once again correct for an additional 12 digits. In fact, of the first 46 digits, only four differ from the corresponding decimal digits of π. Further, the "error" digits appear to occur in positions that have a period of 14, as shown above. Why?

A great place to start is by enlisting the help of an excellent online resource for students and research mathematicians alike: Neil Sloane's Online Encyclopedia of Integer Sequences, available at `http://www.oeis.org`. This tool has no difficulty recognizing the sequence above, namely $(2, -2, 10, -122, 2770 \cdots)$, as "Euler numbers," which are coefficients E_{2k} in Taylor's series for the secant function:

$$\sec x = \sum_{k=0}^{\infty} \frac{(-1)^k E_{2k} x^{2k}}{(2k)!}. \qquad (7)$$

Indeed, this discovery, made originally through the print version of the integer sequence recognition tool more than 25 years ago, led to a formal proof that the Euler numbers are indeed the "errors" here [11, p. 50–52].

3 Experimental methods in applied math

By many measures, the record of the field of modern high-performance, applied mathematical computation is one of remarkable success. Accelerated by relentless advances of Moore's law, this technology has enabled researchers in many fields to perform computations that would have been unthinkable in earlier times. Indeed, computation is rapidly becoming a third mode of scientific discovery, after theory and laboratory work.

The progress in performance over the past few decades is truly remarkable, arguably without peer in the history of modern science and technology. For example, in the November 2014 edition of the Top 500 list of the world's most powerful supercomputers (see Figure 1), the best

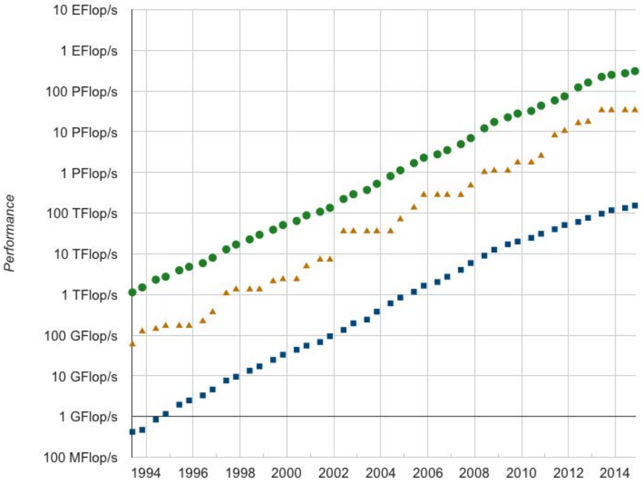


Figure 1: Performance of the Top 500 computers: Red = #1 system; orange = #500 system; blue = sum of #1 through #500.

system performs at over 30 Pflop/s (i.e., 30 "petaflops" or 30 quadrillion floating-point operations per second), a level that exceeds the sum of the top 500 performance figures approximately ten years earlier [20]. Note also that a 2014-era Apple MacPro workstation, which features approximately 7 Tflop/s (i.e., 7 "teraflops" or 7 trillion floating-point operations per second) peak performance, is roughly on a par with the #1 system of the Top 500 list from 15 years earlier (assuming that the MacPro's Linpack performance is at least 15% of its peak performance).

Just as importantly, advances in algorithms and parallel implementation techniques have, in many cases, outstripped the advance from raw hardware advances alone. To mention but a single well-known example, the fast Fourier transform ("FFT") algorithm reduces the number of operations required to evaluate the "discrete Fourier transform," a very important and very widely employed computation (used, for example, to process signals in cell phones), from $8n^2$ arithmetic operations to just $5n \log_2 n$, where n is the total size of the dataset. For large n, the savings are enormous. For example, when n is one billion, the FFT algorithm is more than six million times more efficient.

Remarkable as these developments are, there is no indication that progress is slowing down. Moore's Law, the informal rule in the semiconductor industry that the number of transistors on a chip roughly doubles every 18 months or so, continues apace, yielding a broad range of computer-based devices that continue to advance in aggregate power (even if power management concerns have limited increases in clock rate). And researchers continue to develop new and improved numerical algorithms, which, when coupled with improved hardware, will further improve the power of future systems. Thus we can look forward with some confidence to scientific computer systems in the 2025 year time frame that are roughly 100 times more power than systems available in 2015. Indeed, one's smartphone in the year 2025 may be comparable in power to the world's most powerful supercomputer in, say, 2005 or so.

In this section, we present a few relatively accessible examples of the experimental paradigm in action in applied mathematics. The intent here is certainly not to give an encyclopedic review of modern high-performance applied mathematical computing, but instead to illustrate, by a few examples (most of which require only modest computational resources), the truly exploratory nature of this work.

3.1 Gravitational boosting or "slingshot magic"

One interesting space-age example is the unexpected discovery of *gravitational boosting* by Michael Minovitch, who at the time (1961) was a student working on a summer project at the Jet Propulsion Laboratory in Pasadena, California. Minovitch found that *Hohmann transfer ellipses* were not, as then believed, the minimum-energy way to reach the outer planets. Instead, he discovered, by a combination of clever analytical derivations and heavy-duty computational experiments on IBM 7090 computers (which were the world's most powerful systems at the time), that spacecraft orbits which pass close by other planets could gain a "slingshot effect" substantial boost in speed, compensated by an extremely small change in the orbital velocity of the planet, on their way to a distant location [18]. Some of his earlier computation was not supported enthusiastically by NASA. As Minovitz later wrote,

> Prior to the innovation of gravity-propelled trajectories, it was taken for granted that the rocket engine, operating on the well-known reaction principle of Newton's Third Law of Motion, represented the basic, and for all practical purposes, the only means for propelling an interplanetary space vehicle through the Solar System.[1]

Without such a boost from Jupiter, Saturn, and Uranus, the Voyager mission would have taken more than 30 years to reach Neptune; instead, Voyager reached Neptune in only ten years. Indeed, without gravitational boosting, we would still be waiting!

A very similar type of "slingshot magic," deduced from computational simulations, was much more recently employed by the European Union's Rosetta spacecraft, which, in November 2014, orbited and then deployed a probe to land on a comet many millions of kilometers away. The spacecraft utilized gravity-boost swing-bys around the earth in 2005, 2007 and 2009, and around Mars in 2007. The spacecraft's final approach was very carefully orchestrated, starting with triangular-shaped paths and ending with an elliptical orbit tightly circling the comet, which has only a very

[1] There are differing accounts of how this principle was discovered; we rely on the first-person account at http://www.gravityassist.com/IAF1/IAF1.pdf. Additional information on "slingshot magic" is given at http://www.gravityassist.com/ and http://www2.jpl.nasa.gov/basics/grav/primer.php.

feeble gravitational field. An animation of the Rosetta craft's path is available at `http://www.esa.int/spaceinvideos/Videos/2014/01/Rosetta_s_orbit_around_the_comet`.

Along this line, in December 2014 researchers at Princeton University and the University of Milan announced the discovery, aided by substantial computational simulations, of a new may to achieve Mars orbit, known as "ballistic capture." The idea of ballistic capture is instead of sending the spacecraft to where Mars will be in its orbit, as is done in missions to date, the spacecraft is instead sent to a spot somewhat ahead of the planet. As Mars slowly approaches the craft, it "snags" it into orbit about the planet. In this way most of the large rocket burn to slow the craft is avoided [14].

3.2 Iterative methods for protein conformation

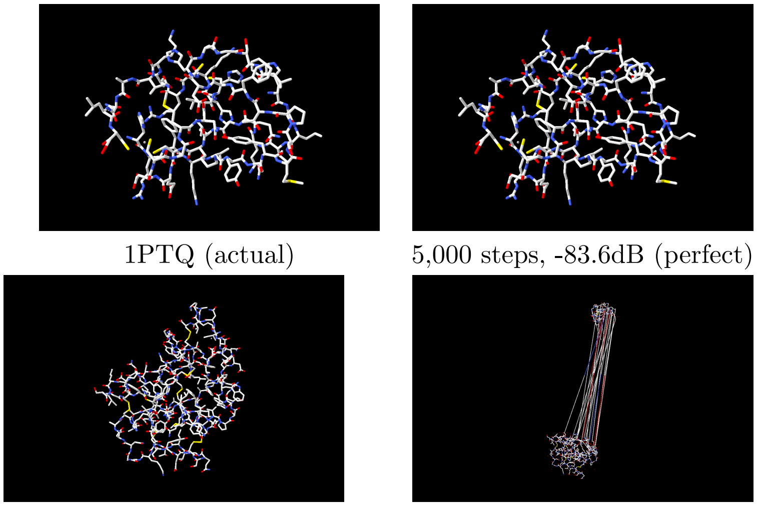
The *method of alternating projections* (MAP), is a computational technique most often used in optimization applications. While a full mathematical treatment would require an excursion into Hilbert spaces and the like, the concept is fairly simple, to find a point in the intersect of several sets to iterate the following process: first "project" a point in a multidimensional space to its closest projection in on each of the sets (a process entirely analogous to the elementary geometry task of finding the closet point on a line to a point outside the line), and average these estimates. The *Douglas-Rachford method* (DR) "reflects," after projection using one of several reflection operations and then averages with the prior step. When the sets are convex, convergence is understood. In general, the sets are not convex, and yet the DR method often works amazing well (with non-convex sets) — see [12].

We illustrate the DR technique with the non-convex problem of reconstruction of protein structure (conformation) using only the short distances below about six Angstroms between atoms that can be measured by nondestructive magnetic resonance imaging (MRI) techniques (interatomic distances below 6Å typically constitute less than 8% of the total distances between atoms in a protein).

Average (maximum) errors from five replications with reflection methods of six proteins taken from a standard database.

Protein	# Atoms	Rel. Error (dB)	RMSE	Max Error
1PTQ	404	-83.6 (-83.7)	0.0200 (0.0219)	0.0802 (0.0923)
1HOE	581	-72.7 (-69.3)	0.191 (0.257)	2.88 (5.49)
1LFB	641	-47.6 (-45.3)	3.24 (3.53)	21.7 (24.0)
1PHT	988	-60.5 (-58.1)	1.03 (1.18)	12.7 (13.8)
1POA	1067	-49.3 (-48.1)	34.1 (34.3)	81.9 (87.6)
1AX8	1074	-46.7 (-43.5)	9.69 (10.36)	58.6 (62.6)

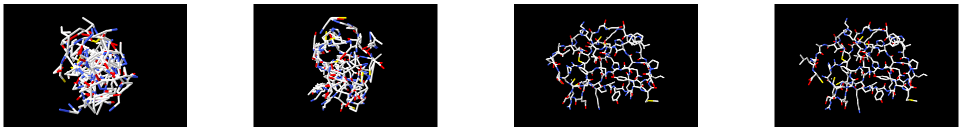
What do the reconstructions look like? We turn to graphic information for 1PTQ and 1POA, in Figure 2. These were respectively our initially most and least successful cases.



1PTQ (actual) 5,000 steps, -83.6dB (perfect)

1POA (actual) 5,000 steps, -49.3dB (mainly good!)

Figure 2: IPTQ and IPOA reconstructions.



500 steps, -25 dB. 1,000 steps, -30 dB. 2,000 steps, -51 dB. 5,000 steps, -84 dB.

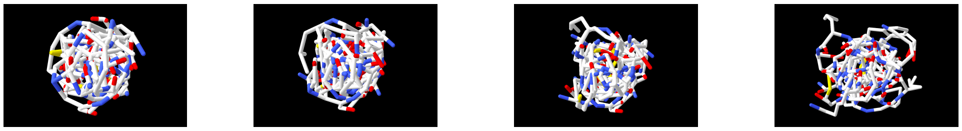
Figure 3: Decibel error by iterations for 1PTQ using DR.

Note that the failure (and large mean or max error) is caused by a very few very large spurious distances. The remainder is near perfect.

While traditional numerical measures (relative error in decibels, root mean square error, and maximum error) of success held some information, graphics-based tools have been dramatically more helpful. It is visually obvious that this method has successfully reconstructed the protein whereas the MAP reconstruction method, shown below, has not. This difference is not evident if one compares the two methods in terms of decibel measurement (beloved of engineers). After 1000 steps or so, using the DR method, the protein shape is becoming apparent. After 2000 steps only minor detail is being fixed.

As shown in Figures 3 and 4, decibel measurement really does not discriminate this from the failure of the MAP method below which after 5000 steps has made less progress than DR after 1000. In Figures 3 and 4, we show the radical visual difference in the behavior of reflection and projection methods on IPTQ.

The first 3,000 steps of the 1PTQ reconstruction are available as a movie at `http://carma.newcastle.edu.au/DRmethods/1PTQ.html`.



500 steps, -22 dB. 1,000 steps, -24 dB. 2,000 steps, -25 dB. 5,000 steps, -28 dB.

Figure 4: Decibel error by iterations for 1PTQ using MAP.

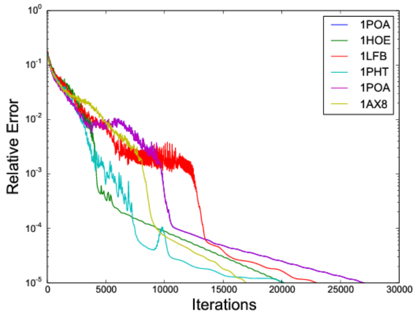
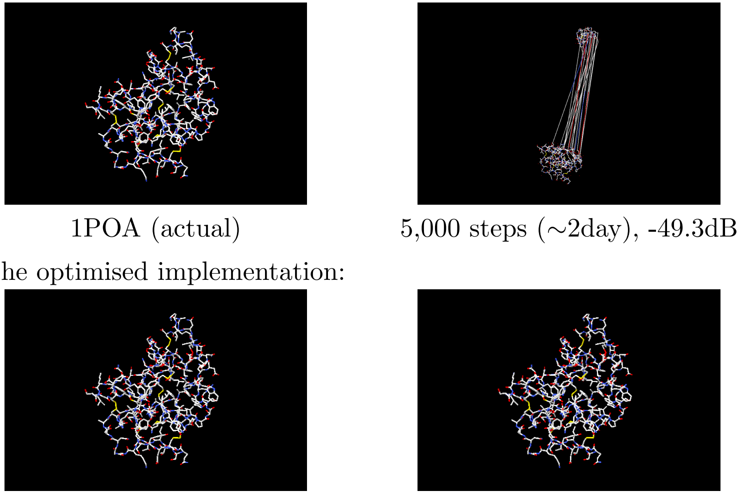


Figure 5: Relative error by iterations (vertical axis logarithmic).

The “un-tuned” implementation (from previous image):



1POA (actual) 5,000 steps (∼2day), -49.3dB

The optimised implementation:

1POA (actual) 28,500 steps (∼1day), -100dB (perfect!)

Figure 6: 1POA conformation before and after tuning.

A More Robust Stopping Criterion. An optimized implementation suggested by the images above gave a ten-fold speed-up. This allowed for the experiment whose results are shown in Figure 5 to be performed. For less than $5,000$ iterations, the error exhibits non-monotone oscillatory behavior. It then decreases sharply. Beyond this point progress is slower. This suggested that perhaps early termination was to blame, so we explored terminating when the error dropped below -100dB.

Similar results were observed for all the other test proteins. Nonetheless, MAP works very well for optical aberration correction (it was used to “fix” the Hubble telescope), and the method is now built in to software for some amateur telescopes.

4 New methods in mathematical research

Both of the present authors recall the time, earlier in their careers, when prominent mathematicians dismissed computation as of no relevance to mathematical research. “Real mathematicians don’t compute” was the by-word. But times have changed. Nowadays it is not at all unusual for mathematicians, particularly relatively young mathematicians, to utilize computer-based tools to explore the mathematical universe, test conjectures and carry out difficult algebraic manipulations — see the list given in the Introduction (Section 1).

In present-day mathematical research, the most widely used tools for experimental mathematics are the following:

- *Symbolic computing.* Symbolic computing, most often done using commercial packages such as *Maple* and *Mathematica*, is a mainstay of modern mathematical research, and is increasingly utilized in classroom instruction as well. Present-day symbolic computing tools are vastly improved over what was available even 10 years ago.

- *High-precision arithmetic.* Most work in scientific or engineering computing relies on either 32-bit IEEE floating-point arithmetic (roughly seven decimal digit precision) or 64-bit IEEE floating-point arithmetic (roughly 16 decimal digit precision). But in experimental mathematics, studies often require very high precision—hundreds or thousands of digits. Fortunately software to perform such computations is widely available either in “freeware” or as a built-in feature of commercial packages such as *Maple* and *Mathematica*.

- *Integer relation detection.* Given a vector of real or complex numbers x_i, an *integer relation algorithm* attempts to find a nontrivial set of integers a_i such that $a_1 x_1 + a_2 x_2 + \cdots + a_n x_n = 0$. One common application of such an algorithm is to find new identities involving computed numeric constants. For example, suppose one suspects that some mathematical object x_1 (e.g., a definite integral, infinite series, etc.) might be given as a sum of terms $x_2, x_3, \ldots, x_n$, with unknown rational coefficients. One can then compute $x_1, x_2, \cdots, x_n$ to high precision (typically several hundred digits), and then apply an integer relation algorithm. If the computation produces an integer relation, then solving it for x_1 produces an experimental identity for the original integral, which then can be proven using conventional methods. The most commonly employed integer relation algorithm is the "PSLQ" algorithm of mathematician-sculptor Helaman Ferguson [11, 230–234]. In 2000, integer relation methods were named one of the top ten algorithms of the twentieth century by *Computing in Science and Engineering.*

4.1 The BBP formula for pi

As noted above, mathematicians have been fascinated by the mathematical constant $\pi = 3.1415926535\ldots$ since antiquity. Archimedes, in the third century BCE, was the first to provide a systematic scheme for calculating π, based on a sequence of inscribed and circumscribed polygons. The Chinese mathematician Tsu Ch'ung Chi, in roughly 480, computed seven correct digits; in 1665, Isaac Newton published 16 digits, but confessed "I am ashamed to tell you how many figures I carried these computations, having no other business at the time." These efforts culminated in the computation of π to 707 digits by William Shanks in 1873; alas, only the first 527 were correct. In the computer era, new techniques (such as FFT-based multiplication) and transistorized hardware resulted in vastly larger calculations — π was calculated to millions, then billions, and, as of October 2011, ten trillion decimal digits [4].

One motivation for such computations has been to see if the decimal expansion (or expansion in other number bases) of π repeats, which would suggest that π is a simple ratio of natural numbers. But in 1761, Lambert proved that π is irrational, and in 1882, Lindemann proved that π is transcendental, meaning that it is not the root of any algebraic polynomial with integer coefficients. Nonetheless, numerous questions still remain, notably the question of whether or not π is a *normal number* in a given number base. This will be discussed further in Section 4.2.

Thus it was with some interest researchers in 1996 announced the discovery of a new formula for π, together with a rather simple scheme for computing binary or hexadecimal digits of π, beginning at an arbitrary starting position, without needing to compute any of the preceding digits. The scheme is based on the following formula, now known as the "BBP formula" for π [9]:

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

For our discussion here, perhaps the most relevant point is that *this formula was discovered using a high-precision computation (200 digits) together with the PSLQ algorithm.* Indeed, it was one of the earliest successes of what is now known as the experimental approach to mathematical research [4]. The proof of this formula (now known as the "BBP" formula for π) is a relatively simple exercise in calculus. It is perhaps puzzling that it had not been discovered centuries before. But then no one was looking for such a formula.

The scheme to compute digits of π beginning at an arbitrary starting point is remarkably simple, but will not be given here; see [4] for details.

This and similar algorithms have been implemented to compute hexadecimal digits of π beginning at stratospherically high positions. On March 14 (Pi Day), 2013, Ed Karrels of Santa Clara University announced the computation of 26 base-16 digits of π beginning at position one quadrillion [4]. His result: `8353CB3F7F0C9ACCFA9AA215F2`. Here hexadecimal digits `A,B,C,D,E,F` denote 10, 11, 12, 13, 14, 15, respectively in base-16.

High-precision computations and PSLQ programs have been used to found numerous other "BBP-type" formulas, which permit arbitrary-digit calculation, for numerous other well-known mathematical constants. See [8] for some additional examples.

Certainly there is no need for computing π or other constants to millions or billions of digits in practical scientific or engineering work. There are certain scientific calculations that require intermediate calculations to be performed to higher than standard 16-digit precision (typically 32 or 64 digits may be required) [2], and certain computations in the field of experimental mathematics have required as high as 50,000 digits [7], but we are not aware of any "practical" applications beyond this level.

Computations of digits of π are, however, excellent tests of computer integrity—if even a single error occurs during a large computation, almost certainly the final result will be badly in error, disagreeing with a check calculation done with a different algorithm. For example, in 1986, a pair of π-calculating programs detected some obscure hardware problems in one of the original Cray-2 supercomputers. Also, some early research into efficient implementations of the fast Fourier transform on modern computer architectures had their origins in efforts to accelerate computations of π [11, p. 115].

4.2 Normal numbers

As we noted in the previous section, a long-standing unanswered question of pure mathematics (with some potential real-world applications) is whether or not π or other well-known mathematical constants are *normal.* A normal number (say in base ten) is a number whose decimal digit expansion satisfies the property that each of the ten digits $(0, 1, 2, \cdots, 9)$ appears, in the limit, 1/10 of the time; every pair of digits, such as "27" or "83," appears, in the limit, 1/100 of the time, and so on. Similar definitions apply for being normal base 2 or in other bases. Such questions have intrigued mathematicians for ages. But to date, no one has been able to prove (or disprove) any of these assertions — not for any well-known mathematical constant, not for any number base. For example, it is likely true that *every* irrational root of an integer polynomial (e.g., $\sqrt{2}$, $\sqrt[3]{10}$, $(1 + \sqrt{5})/2$, etc.) is normal to every integer base, but there are no proofs, even in the simplest cases.

Fortunately, modern computer technology has provided some new tools to deal with this age-old problem. One fruitful approach is to display the digits of π or other constants graphically, cast as a random walk [1]. For example, Figure 7 shows a walk based on one million base-4 pseudorandom digits, where at each step the graph moves one unit east, north, west or south, depending on the whether the pseudorandom base-4 digit at that position is $0, 1, 2$ or 3. The color indicates the path followed by the walk—shifted up the spectrum (red-orange-yellow-green-cyan-blue-purple-red) following a hue-saturation-value (HSV) model.

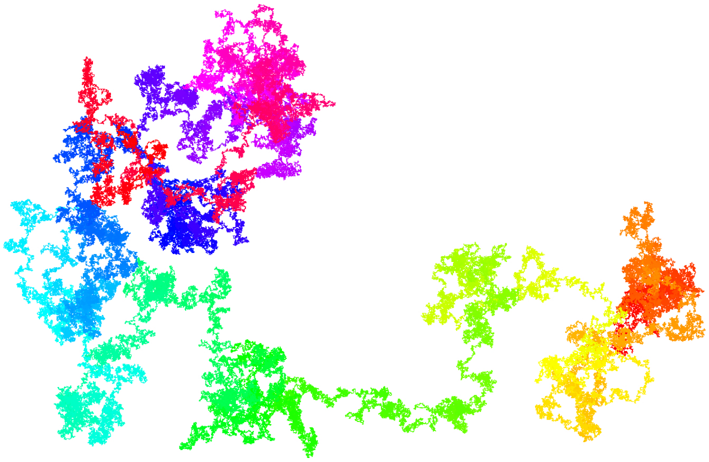


Figure 7: A uniform pseudorandom walk.

Figure 8 shows a walk on the first two billion base-4 digits of π. A hugely more detailed figure (based on 100 billion base-4 digits) is available to explore online at `http://gigapan.org/gigapans/106803`.

Although no rigorous inference regarding the normality of π can be drawn from these figures, it is plausible that π is 4-normal (and thus 2-normal), since the overall appearance of its graph is similar to that of the graph of the pseudorandomly generated base-4 digits.

This same tool can be employed to study the digits of Stoneham's constant, namely

$$\alpha_{2,3} = \frac{1}{2} + \frac{1}{3 \cdot 2^3} + \frac{1}{3^2 \cdot 2^{3^2}} + \cdots \; = \; 1.043478260869564531\ldots. \qquad (9)$$

This constant is one of the few that is provably normal base 2 (meaning that its binary expansion satisfies the normality property). What's more, it is provably *not* normal base 6, so that it is an explicit example of the fact that normality in one number base does not imply normality in another base [5]. For other number bases, including base 3, its normality is not yet known either way. Perhaps additional computer studies will clarify.

Figures 9 and 10 show walks generated from the base-4 and base-6 digit expansions, respectively, of $\alpha_{2,3}$. The base-4 digits are graphed using the same scheme mentioned above, with each step moving east, north, west or south according to whether the digit is $0, 1, 2$ or 3. Similarly, the base-6 graph is generated by moving at an angle of 0, 60, 120, 180, 240, or 300 degrees, respectively, according to whether the digit is 0, 1, 2, 3, 4 or 5.

From these three figures it is clear that while the base-4 graphs appear to be plausibly random (since they are similar in overall structure to Figures 7 and 8), the base-6 walk is vastly different, mostly a horizontal

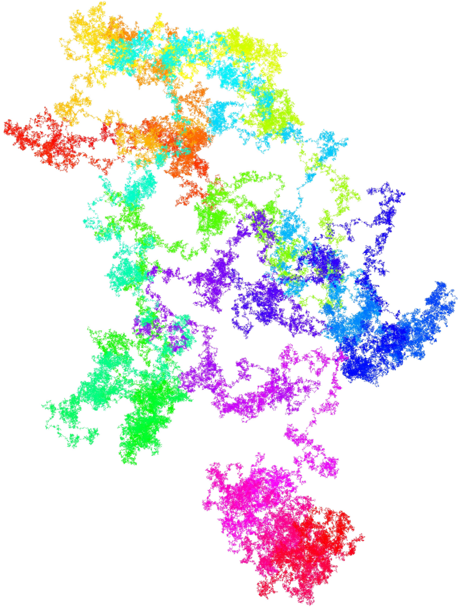


Figure 8: A walk on the first two billion base-4 digits of π.

line. Indeed, we discovered the fact that $\alpha_{2,3}$ fails to be 6-normal by a similar empirical analysis of the base-6 digits—there are long stretches of zeroes in the base-6 expansion [5].

4.3 Ising integrals

High-precision computation and the PSLQ algorithm have been invaluable in analyses of definite integrals that arise in mathematical physics settings. For instance, consider this family of n-dimensional integrals [6]:

$$C_n = \frac{4}{n!}\int_0^\infty \cdots \int_0^\infty \frac{\mathrm{d}u_1 \mathrm{d}u_2, \cdots \mathrm{d}u_n}{\left(\sum_{j=1}^n (u_j + 1/u_j)\right)^2}\,. \tag{10}$$

Direct numerical computation of these integrals from (10) is very difficult, but it can be shown that

$$C_n = \frac{2^n}{n!}\int_0^\infty pK_0^n(p)\,\mathrm{d}p, \tag{11}$$

where K_0 is the *modified Bessel function*, and in this form these integrals can be numerically computed. Indeed, it is in this form that C_n arises in quantum field theory, as we subsequently learned from David Broadhurst. Upon calculating these values, we quickly discovered that the C_n rapidly approach a limiting value. For example,

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

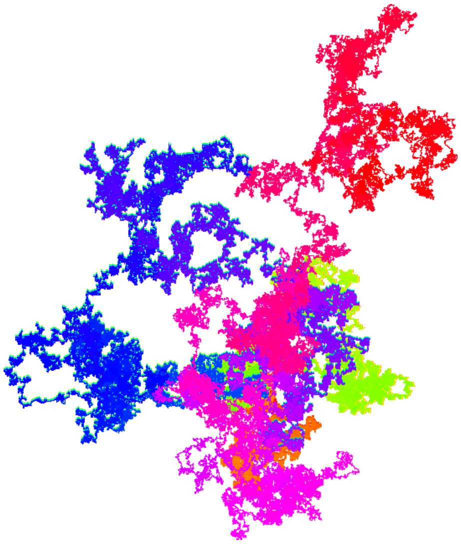


Figure 9: A walk on the provably normal base-4 digits of Stoneham's constant.

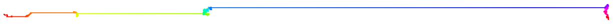


Figure 10: A walk on the *nonnormal* base-6 digits of Stoneham's constant.

What is this number? When one copies the first 30 or 40 digits into the online Inverse Symbolic Calculator (ISC) at `http://carma-lx1.newcastle.edu.au:8087`, it quickly returns the result

$$\lim_{n\to\infty} C_n = 2e^{-2\gamma}, \tag{12}$$

where γ denotes Euler's constant $= 0.5772156649\ldots$. This fact was rigorously proven shortly after discovery. Numerous other results for related families of integrals have been discovered by computer in this fashion [6].

4.4 Formal verification of proof

In 1611, Kepler described the stacking of equal-sized spheres into the familiar arrangement we see for oranges in the grocery store. He asserted that this packing is the tightest possible. This assertion is now known as the Kepler conjecture, and has persisted for centuries without proof.

In 1994, Thomas Hales, now at the University of Pittsburgh, proposed a five-step program that would result in a proof: (a) treat maps that only have triangular faces; (b) show that the face-centered cubic and hexagonal-close packings are local maxima in the strong sense that they have a higher score than any Delaunay star with the same graph; (c) treat maps that contain only triangular and quadrilateral faces (except the pentagonal prism); (d) treat maps that contain something other than a triangular or quadrilateral face; and (e) treat pentagonal prisms.

In 1998, Hales announced that the program was now complete, with Samuel Ferguson (son of mathematician-sculptor Helaman Ferguson, who discovered the PSLQ algorithm) completing the crucial fifth step. This project involved extensive computation, using an interval arithmetic package, a graph generator, and *Mathematica*. The computer files containing the source code and computational results occupy more than three Gbytes of disk space. Additional details, including papers, are available at `http://www.math.pitt.edu/~thales/kepler98`. For a mixture of reasons—some more defensible than others—the *Annals of Mathematics* initially decided to publish Hales' paper with a cautionary note, but this disclaimer was deleted before final publication.

Hales subsequently embarked on a multi-year program to certify the proof by means of computer-based formal methods, a project he has named the "Flyspeck" project. This was completed in July 2014 [13]. As these computer-based formal proof techniques become better understood, we can envision a large number of mathematical results eventually being confirmed by computer, but this will take decades.

5 Why should we trust computations?

There are many possible sources of errors in large computations of the type discussed above:

- The underlying mathematical formulas and algorithms used might conceivably be in error, or the formulas may have been incorrectly transcribed.
- The computer programs implementing these algorithms, which typically employ sophisticated algorithms such as matrix operations or the fast Fourier transform (FFT), may contain bugs.
- Inadequate numeric precision may have been employed, invalidating some key steps of the algorithm.
- Erroneous programming constructs may have been employed to control parallel processing. Such errors are typically very hard to detect and rectify, since in many cases they cannot easily be replicated.
- Hardware errors may have occurred during the run, rendering all subsequent computation invalid. This was a factor in the 1986 computation of π, as noted above in Section 4.1.
- Quantum-mechanical errors, induced by stray subatomic particles, may have corrupted the results in storage registers.

So why should anyone believe any results of such calculations? The answer is that such calculations are double-checked, either by an independent calculation done using some other algorithm, or by rigorous internal checks. For instance, Kanada's 2002 computation of π to 1.3 trillion decimal digits involved first computing slightly over one trillion hexadecimal (base-16) digits, using formulas found by one of the present authors

(Jonathan Borwein) and Peter Borwein. Kanada found that the 20 hex digits of π beginning at position $10^{12} + 1$ are `B4466E8D21 5388C4E014`.

Kanada then calculated these same 20 hexadecimal digits directly, using the "BBP" formula and algorithm, mentioned above in Section 4.1. The result of the BBP calculation was `B4466E8D21 5388C4E014`. Needless to say, in spite of the many potential sources of error in both computations, each of which required many hours of computation on a supercomputer, the final results dramatically agree, thus confirming in a convincing albeit heuristic sense that both results are almost certainly correct. Although one cannot rigorously assign a "probability" to this event, note that the probability that two 20-long random hexadecimal digit strings perfectly agree is one in $16^{20} \approx 1.2089 \times 10^{24}$.

This raises the following question: *Which is more securely established*, the assertion that the hexadecimal digits of π in positions $10^{12}+1$ through $10^{12}+20$ are `B4466E8D21 5388C4E014`, or the final result of some very difficult work of mathematics that required hundreds or thousands of pages, that relied on many results quoted from other sources, and that (as is frequently the case) has been read in detail by only only a relative handful of mathematicians besides the author? In our opinion, computation often trumps cerebration.

6 What will the future bring?

We have discussed numerous examples of the "experimental" paradigm of mathematics in action, both for "pure" and "applied" applications. It is clear that these experimental-computational methods are rapidly becoming central to all mathematical research, and also to mathematical education, where students can now see, hands-on, many of the principles that heretofore were only abstractions. What's more, the rapidly improving quality of mathematical software, when combined with the inexorable forward march of Moore's Law in hardware technology, means that ever-more-powerful software tools will be available in the future. Indeed, almost certainly one day we will look back on the present epoch with puzzlement, wondering how anyone ever got any serious done with such primitive tools as we use today!

It is not just *Maple* and *Mathematica* that are improving. In Section 2.2 we mentioned Neil Sloane's Online Encyclopedia of Integer Sequences (`http://www.oeis.org`) and the Inverse Symbolic Calculator (`http://isc.carma.newcastle.edu.au`. A few other valuable online resources include the Digital Library of Mathematical Functions (`http://dlmf.nist.gov`), a large compendium of formulas for special functions, LAPACK (`http://www.netlib.org/lapack`), a large library of highly optimized linear algebra programs, and SAGE (`http://www.sagemath.org`), an open-source mathematical computing package. We can certainly expect substantial improvements in these tools as well.

Of related interest is online collaborative efforts in mathematics, notably the PolyMath activity, which joins together a large team of mathematicians in online, computer-based collaborations to explore conjectures and prove theorems. One notable success was a dramatic lowering of the bound on prime gaps, completed in September 2014 [19].

But aside from steadily improving mathematical tools and online facilities, can we expect anything fundamentally different?

One possibility is a wide-ranging "intelligent assistant" for mathematics [10]. Some readers may recall the 2011 televised match, on the "Jeopardy!" quiz show in North America, between IBM's "Watson" computer and the two most accomplished champions of the show (Ken Jennings and Brad Rutter). This was the culmination of a multi-year project by researchers at the IBM Yorktown Heights research center, who employed state-of-the-art artificially intelligent and machine learning software that could first "understand" the clues (which are often quirky and involve puns) and then quickly produce the correct response. It has been reported that that the project cost IBM more than one billion U.S. dollars. While in their first match, the humans were competitive, in the second match Watson creamed its human opponents, and easily won the $1 million prize (which IBM donated to charity) [17]. IBM is now adapting the Watson technology for medical applications, among other things.

We can thus envision an enhanced Watson-class intelligent system for mathematics. Such a system would not only incorporate powerful software to "understand" and respond to natural-language queries, but it would also acquire and absorb the entire existing corpus of published mathematics for its database. One way or another, it is clear that future advances in mathematics will be intertwined with advances in artificial intelligence. Along this line, Eric Horvitz, Managing Director of the Microsoft Research Lab in Redmond, Washington, has launched a project to track the advance of artificial intelligence over the next 100 years, with progress reports every five years [21].

So will computers ever completely replace human mathematicians? Probably not, according to the 2014 Breakthrough Prize in Mathematics recipient Terence Tao — even 100 years from now, much of mathematics will continue to be done with humans working in partnership with computers [3].

But, as we mentioned above, more is at stake than merely accelerating the pace at which mathematical researchers, teachers and students can do their work. The very notion of what constitutes *secure mathematical knowledge* may be changing, before our eyes. It will be interesting in any event. We look forward to what the future will bring.

References

[1] F. J. Aragon Artacho, D. H. Bailey, J. M. Borwein and P. B. Borwein, "Walking on real numbers," *Mathematical Intelligencer*, vol. 35 (2013), 42–60.

[2] D. H. Bailey, R. Barrio and J. M. Borwein, "High-precision computation: Mathematical physics and dynamics," *Appl. Math. and Comp.*, vol. 218 (2012), 10106–10121.

[3] D. H. Bailey and J. M. Borwein, "Big bucks for big breakthroughs: Prize recipients give three million dollar maths talks," Nov. 11, 2014, `http://experimentalmath.info/blog/2014/11/breakthrough-prize-recipients-give-math-seminar-talks`.

[4] D. H. Bailey and J. M. Borwein, "Pi day is upon us again, and we still do not know if pi is normal," *American Mathematical Monthly*, March 2014, pg. 191–206.

[5] D. H. Bailey and J. M. Borwein, "Nonnormality of Stoneham constants," *Ramanujan Journal*, vol. 29 (2012), pg. 409–422; DOI 10.1007/s11139-012-9417-3.

[6] 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.

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

[8] D. H. Bailey, J. M. Borwein, A. Mattingly and G. Wightwick, "The Computation of Previously Inaccessible Digits of π^2 and Catalan's Constant," *Notices of the American Mathematical Society*, vol. 60 (2013), no. 7, 844-854.

[9] D. H. Bailey, P. B. Borwein and S. Plouffe, "On the Rapid Computation of Various Polylogarithmic Constants," *Mathematics of Computation*, vol. 66, no. 218 (Apr 1997), 903–913.

[10] J. M. Borwein, "The future of mathematics: 1965 to 2065," *MAA Centenary Volume*. In press (2015), `http://www.carma.newcastle.edu.au/jon/future.pdf`.

[11] J. M. Borwein and D. H. Bailey, *Mathematics by Experiment: Plausible Reasoning in the 21st Century*, A K Peters, second edition, 2008.

[12] J.M. Borwein and M.K. Tam, "Reflection methods for inverse problems with applications to protein conformation determination." Springer volume on *Generalized Nash Equilibrium Problems, Bilevel programming and MPEC* . In Press, August 2014.

[13] "Announcement of completion," Aug. 2014, `https://code.google.com/p/flyspeck/wiki/AnnouncingCompletion`.

[14] A. Hadhazy, "A new way to reach Mars safely, anytime and on the cheap," *Scientific American*, Dec 22, 2014, `http://www.scientificamerican.com/article/a-new-way-to-reach-mars-safely-anytime-and-on-the-cheap`.

[15] T. C. Hales, "Formal Proof," *Notices of the American Mathematical Society*, vol. 55, no. 11 (Dec. 2008), 1370–1380.

[16] G. Hanna and M. de Villiers (Eds.), *On Proof and Proving in Mathematics Education*, the 19th ICMI Study, New ICMI Study Series, 15, Springer-Verlag, 2012.

[17] J. Markoff, "Computer wins of 'Jeopardy!': Trivial it's not," *New York Times*, Feb. 16, 2011, `http://www.nytimes.com/2011/02/17/science/17jeopardy-watson.html`.

[18] M. Minovitch, "A method for determining interplanetary free-fall reconnaissance trajectories," *JPL Tech. Memo TM-312-130*, (23 Aug 1961), 38–44.

[19] D. H. J. Polymath, "The 'bounded gaps between primes' Polymath project – a retrospective," Sep. 30, 2014, `http://arxiv.org/abs/1409.8361`.

[20] "The Top 500 list," available at `http://www.top500.org`.

[21] Jia You, "A 100-year study of artificial intelligence? Microsoft Researchs Eric Horvitz explains," *Science*, Jan. 9, 2015, `http://news.sciencemag.org/people-events/2015/01/100-year-study-artificial-intelligence-microsoft-research-s-eric-horvitz`.