**CDMTCS Research Report Series**

# A Computational Mathematics View of Space, Time and Complexity

**David H. Bailey**[1]
**Jonathan M. Borwein**[2]

[1]Lawrence Berkeley National Laboratory, USA
[2]University of Newcastle, Australia

CDMTCS-498
September 2016

Centre for Discrete Mathematics and
Theoretical Computer Science

# A computational mathematics view of space, time and complexity

David H. Bailey* Jonathan M. Borwein†

August 24, 2016

**Abstract**

Modern computational mathematics requires a philosophical perspective largely at odds with that of traditional mathematics, since current computational mathematics (as distinct from computer science) is by its very nature is discrete, not continuous, and tied to the real world in ways that the more theoretical branches of mathematics (and computer science) often are not. Indeed, computational mathematics provides a means to escape the trap feared by John von Neumann when he wrote,

> [T]here is a grave danger that the subject [of mathematics] will develop along the line of least resistance, that the stream so far from its source [in empirical reality] will separate into a multitude of insignificant branches, and that the discipline will become a disorganized mass of details and complexities.

But even a computational approach to mathematics has limits, not the least of which are the uncertainties of errors in hardware, software and algorithms that inevitably are part-and-parcel with computation, although there are ways to limit these uncertainties.

In our chapter, bulwarked by concrete examples, we will try to situate past, present and future mathematical views of space, time, infinity and certainty within a computational context in which, for example, error due to quantum effects begins to compete with traditional sources of logical and numerical inaccuracy. We shall also argue that traditional taxonomies of complexity and completeness are not only outmoded but actually destructive of progress.

## 1 Historical perspective of computational mathematics

In traditional pegagogy, mathematics is taught as a sequence of axioms, definitions, theorems and proofs, entirely similar in style to Euclid's 2300-year-old *Elements*. While we in no way wish to disparage the axiomatic approach (and any serious mathematician surely must master it), there are downsides to an overly pedantic focus on purely formal methods.

To begin with, a purely axiomatic approach is historically dishonest, because the real work of mathematical discovery throughout history has almost always involved numerical and algebraic experimentation, from which insight is gained and hypotheses formulated.

*Lawrence Berkeley National Laboratory (retired), Berkeley, CA 94720, and University of California, Davis, Davis, CA 95616, USA. E-mail: david@davidhbailey.com.

†CARMA, University of Newcastle, Callaghan, NSW 2308, Australia. Borwein passed away on 2 August 2016.

Only later were these hypotheses turned into precisely worded theorems and proofs that we read today in journals and textbooks.

Carl Fredrich Gauss, arguably the 19th century's greatest mathematician, explained that his way of arriving at mathematical truths was through "systematic experimentation" [15]. When just 14 or 15 years old, after computing long tables of prime numbers, he conjectured that the number of primes less than $n$ is, for large $n$, approximately $n/\log n$, which is now known as the prime number theorem [10, pg. 13]. On another occasion, while examining tables of integrals provided by James Stirling, he noticed that the reciprocal of one integral agreed numerically with a limit of the arithmetic-geometric mean iteration. This purely computational observation led Gauss to discover and develop elliptic and modular function theory [10, pg. 13].

Indeed many great mathematicians from Archimedes and Galileo — who apparently said *"All truths are easy to understand once they are discovered; the point is to discover them."* — to Gauss, Poincaré, and Lennard Carleson, have emphasized how much it helps to "know" the answer.

More importantly, the axiomatic approach fails to train mathematicians, pure or applied, in the real work of 21st century mathematical discovery, which, more than in any previous era, involves substantial amounts of computational experimentation. Although some of the older generation still say that "real mathematicians don't compute," the majority of research mathematicians today are fluent and accustomed to using a variety of computational tools, such as the commercial products *Maple* and *Mathematica*, as well as any number of custom-written tools for both symbolic and numeric computing.

This synergy between humans and computers has produced a new set of mathematical results that would not have been possible (or at least not likely) to have been discovered in an earlier era. As a single example, out of many that could be listed, in 1996 the following formula for $\pi$ (known as the "BBP" formula [2]) was discovered by a computer program:

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

This formula has the remarkable property that it permits one to calculate, by means of a surprisingly simple algorithm, binary or base-16 digits of $\pi$ beginning at an arbitrary starting position, without needing to compute any of the digits before the specified position. The numerical computation that led to the discovery of (1) certainly does not constitute a rigorous proof of this identity, but once discovered numerically, the proof turned out to be a relatively simple exercise of calculus [10, pg. 118].

We are hardly the first to observe that discovery is often the more important part of proof. Two millennia ago, Archimedes wrote, in the Introduction to his long-lost and recently reconstituted *Method* manuscript,

> For it is easier to supply the proof when we have previously acquired, by the method, some knowledge of the questions than it is to find it without any previous knowledge.

Similarly, as 2006 Abel Prize winner Lennart Carleson described in his 1966 International Congress of Mathematics speech on his positive resolution of Luzin's 1913 conjecture, only after many years of seeking a counterexample, he finally decided none could exist. He expressed the importance of this confidence as follows [14]:

> The most important aspect in solving a mathematical problem is the conviction of what is the true result. Then it took two or three years using the techniques that had been developed during the past 20 years or so [to prove the result].

## 2 Reliability of computations

Computing in mathematics (or in any other discipline) raises troubling questions about the reliability of computation. After all, there are many possible sources of error:

- The underlying formulas or algorithms used might be incorrectly deduced or incorrectly transcribed from underlying theory.
- Computer programs implementing these formulas and algorithms, which often employ highly sophisticated techniques to accelerate the computation (e.g., FFTs, parallel programming constructs, etc.), are certainly prone to error.
- Mathematical and scientific computing relies heavily on floating-point calculations, and the results of these calculations may be numerically unreliable, especially when performed on a parallel computer system with uncertain order of operations.
- All computers rely on a vast infrastructure of system software and compilers that inevitably contains errors.
- Hardware errors can and do occur, particularly in large-scale, long-running calculations. Even quantum-level errors are now expected to be a factor in future computation.

So why should anyone believe the results of calculations? The answer is that most important calculations are checked with one or more independent calculations done using a different algorithm, software package, precision level, hardware system or all of the above.

For example, floating-point calculations, as mentioned above, often give slightly different results when run on a different computer system or one with a different number of processors. Indeed, ensuring reproducibility of floating-point results is a growing challenge in computing in general and mathematical and scientific computing in particular [5]. But there are relatively simple ways to ensure reproducibility, such as by changing the default rounding mode (to see what difference this makes in the results), or by employing software-based higher precision. Although one cannot guarantee absolute integrity in this way, if two calculations using high-precision arithmetic agree to many digits beyond the level required to discover the phenomena in the first place, this is a pretty strong confirmation that the computation is numerically sound (although see below in Section 3).

As a related example, 21st century mathematicians, continuing a centuries-old tradition, have computed large numbers of digits of constants such as $\pi$, in an effort to explore the long-standing unanswered question of whether and why these digit expansions are statistcially random in a certain sense [1]. As of the present date, the state-of-the-art in this regard is the computation of over 10 *trillion* base-16 digits and 12.1 *trillion* decimal digits of $\pi$, by Alexander Yee and Shigeru Kondo [24]. Their main computation employed the following formula, due to David and Gregory Chudnovsky [10, pg. 108]:

$$\frac{1}{\pi} = 12 \sum_{k=0}^{\infty} \frac{(-1)^k \, (6k)! (13591409 + 545140134k)}{(3k)! \, (k!)^3 \, 640320^{3k+3/2}}. \tag{2}$$

Using this formula, they computed 10,048,832,487,050 base-16 digits of $\pi$. Then, in a separate computation, they directly computed 64 base-16 digits of $\pi$, beginning at position 10,048,832,487,013, using the following variant of the "BBP" formula [10, pg. 124]:

$$\pi = \frac{1}{64}\sum_{k=0}^{\infty}\frac{(-1)^k}{1024^k}\left(\frac{256}{10k+1}+\frac{1}{10k+9}-\frac{64}{10k+3}-\frac{4}{10k+5}-\frac{4}{10k+7}-\frac{32}{4k+1}-\frac{1}{4k+3}\right).$$

In both calculations, the slightest error at any stage almost certainly renders the final results to be utterly and wholly incorrect. Here are the two corresponding sets of results:

`d9ae13df df0c64d9 49bacf10 f55ae963 254699a8 bb24624b d47aea96 8016b052`
`d9ae13df cf0c64d9 49bacf10 f55ae963 254699a8 bb24624b d47aea96 8016b052`

Needless to say, the final results dramatically agree, thus confirming (in a convincing but heuristic sense) that both sets of results are almost certainly correct.

This raises the following question: What is more securely established, the assertion that the base-16 digits of $\pi$ in positions $10,048,832,487,043$ through $10,048,832,487,050$ are `8016b052`, or the final theorem(s) 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) only a relative handful of mathematicians besides the author can or have carefully read in detail? When is computation more reliable than proof?

As another example, when one uses a credit card to purchase an item online, it is quite likely that the underlying software generates two pseudorandom prime numbers. To establish that a generated integer is prime, one could use the provable polynomial-time algorithm recently discovered by three Indian mathematicians [11, pg. 302], but in practice the Monier-Rabin probabilistic primality test is used instead [11, pg. 301].

The Monier-Rabin algorithm is *not* a "provable" test, in the sense of an absolute guarantee of primality. But if the integer has 500 or more bits (as is typical in e-commerce) and passes this test once, then we may argue that the 'probability' that it is not prime is less than $1.6 \times 10^{-24}$. If it passes the test five times, then the probability is much less than $10^{-100}$. Note that such tiny probabilities are inconceivably smaller than the chance that an undetected hardware error glitch or quantum-mechanical fluke occurs during the calculation, not to mention the possibility of a computer program bug or a human mathematical error in deriving, transcribing or applying the test. Given these realities, what is the point of distinguishing between a "provable" primality test (performed either by a human or by a computer) and a probabilistic primality test (performed by a computer)?

Such considerations have led a growing number of mathematicians to now regard computation as on a par with formal reasoning, provided calculations are double-checked by sufficiently rigorous tests. They have also drawn into question the value, from a practical point of view, of pursuing algorithms whose only advantage over other algorithms is that they are "provably correct."

# 3 Finite computations

Computation, by its very definition, is a decidedly *finite* and *discrete* process. There is no place in real-world scientific or mathematical computing for actual infinities, nor can computation deal directly with the uncountably infinite and continuous real line (or 2-D or 3-D space). It is true that symbolic mathematical software can represent infinity, and can,

for example, symbolically evaluate the integral of a function over an infinite interval, but beneath the symbolic processing, the software and the computer it runs on are, of course, discrete, finite entities.

While in some cases the integral of a function over an infinite interval can be computed symbolically, evaluation over some finite interval may well not be possible symbolically and will require numerical computation to obtain a concrete answer. As Stanislaw Ulam once said, "The infinite we shall do right away. The finite may take a little longer." [17]. On the other hand, the space of functions that can be numerically integrated to any desired accuracy is far vaster than the space of functions that can be symbolically integrated in closed form. Similarly, the space of ordinary and partial differential equations (which are a mainstay of applied mathematics) that can be numerically solved to any desired accuracy is far vaster than the space of such equations that can be analytically solved by symbolic processing.

It is true that one cannot abolutely rely on numerical computations. For example, consider the innocent-looking integral [4]

$$\int_0^\infty \cos(2x) \prod_{n=1}^{\infty} \cos\left(\frac{x}{n}\right)\, dx = \tag{3}$$

$$0.392699081698724154807830422909937860524645434187231595926812285162\ldots$$

One might first be tempted to think that this is equal to $\pi/8$, namely

$$\frac{\pi}{8} = 0.392699081698724154807830422909937860524646174921888227621868074038\ldots,$$

but note that while two values agree for 42 digits, they differ beginning in the 43rd digit; they are *not* equal. As it turns out, the integral (3) is merely the first term of a very rapidly convergent series; if two terms are taken, the sum is $\pi/8$ to over 500 digits; if three are taken, the sum is $\pi/8$ to over 8,000 digits.

A related example if the following. Consider the following "identity" [3, 7]:

$$\sum_{n=-\infty}^{\infty} \operatorname{sinc}(n)\operatorname{sinc}(n/3)\operatorname{sinc}(n/5)\operatorname{sinc}(n/7)\cdots\operatorname{sinc}(n/p)$$

$$\stackrel{?}{=} \int_{-\infty}^{\infty} \operatorname{sinc}(x)\operatorname{sinc}(x/3)\operatorname{sinc}(x/5)\operatorname{sinc}(x/7)\cdots\operatorname{sinc}(x/p)\, dx, \tag{4}$$

where sinc $x$ means $(\sin x)/x$. Provably, the following is true: This identity is precisely valid for prime $p$ among the first at least $10^{176}$ primes; but stops holding after some larger prime. Thereafter the sum is less than the integral, but they differ by much less than $10^{-100}$.

In short, numerical coincidence is no guarantee of mathematical certainty. On the other hand, when is a mathematical identity "close enough for government work"? When can one for all practical purposes replace the infinities in the above formulas with some large finite values? Many mathematicians are quite content, say, with 100-digit agreement, to conclude that the apparent result is worth seeking a proof for, or at least understanding why the near-identity holds.

# 4 A finite universe

As mentioned above, much mathematics from calculus forward (with the exception of finite and/or discrete algebraic structures) assumes a continuous, complete, infinite space, so, for

example, we can subdivide the real line *ad infinitum*, the limits of convergent sequences are actual points in the real line, and if a continuous real function is positive for one argument and negative for another, then there is at least one real point between these two arguments for which the function is exactly zero. Since the nineteenth century a careful development of countably infinite limiting processes has been central to mathematical analysis.

But, one can ask, is nature, at the most basic, fundamental level, really like this?

From several lines of research, the answer appears to be "no." For example, the Bekenstein bound, derived from quantum theory by Israeli theoretical physicist Jacob Bekenstein in 1981, places an upper limit on the number of quantum states that can be contained within a volume [8]. For a sphere with mass $m$ kg and radius $R$ meters, the limit is

$$I \leq \frac{2\pi cRm}{\bar{h}\log 2} \approx 2.577 \cdot 10^{43} mR, \tag{5}$$

where $c$ is the speed of light and $\bar{h}$ is Planck's constant. Thus, for example, at most $2.6 \times 10^{42}$ bits of information are sufficient to perfectly recreate any human brain down to the quantum level [9].

But one can also apply the Bekenstein bound to the entire observable universe. If this is done, one obtains approximately $10^{123}$ bits, so that the maximum number of quantum states is some $10^{10^{123}}$ [22, pg. 108]. These reckonings are tightly connected with the "holographic principle," in which the information within a sphere is inevitably limited to the information that can be held on its boundary [16]. Whatever these values are, they are finite — the real universe is not the same as the infinite, continuous space of classical mathematics.

We may then ask how many digits of $\pi$ can be stored in the known universe? Surely it cannot be more than the total number of quantum states, which while enormous is still finite.

Along this line, physicist Max Tegmark, in his recent book *Our Mathematical Universe: My Quest for the Ultimate Nature of Reality* [21], argues that at its most fundamental level, our universe is not merely described by a mathematical structure; it *is* a mathematical structure. He points out that only a mathematical structure, defined by axioms and relations, can qualify for a description of reality that is completely free from human (or even extraterrestrial) baggage. He further proposes two additional hypothesis [21, pg. 267]:

- *Computable Universe Hypothesis*: Our external physical reality is a mathematical structure defined by computable functions.
- *Finite Universe Hypothesis*: Our external physical reality is a finite mathematical structure."

These two hypotheses are explored in Chapter 12 of his book. But whichever of these one is more inclined to accept, both suggest that fundamental physical reality is discrete, not continuous.

# 5 Theoretical versus practical

As we noted above, computational mathematics, if performed with suitably rigorous double-checks, can produce results that are arguably as reliable, in a practical sense, as formal proofs in many cases, or at least can be thought of as another avenue to approach secure mathematical knowledge.

But there is one other advantage of modern computational mathematics: It provides a means to escape the trap feared by John von Neumann when he wrote in 1947,

> As a mathematical discipline travels far from its empirical source, or still more, if it is a second and third generation only indirectly inspired by "reality" it is beset with very grave dangers. It becomes more and more pure aestheticizing, more and more purely *l'art pour l'art*. This need need not be bad if the field is surrounded by correlated subjects which have still closer empirical connections, or if the discipline is under the influence of men with exceptionally well-developed taste. But there is a grave danger that the subject will develop along the line of least resistance, that the stream so far from its source will separate into a multitude of insignificant branches, and that the discipline will become a disorganized mass of details and complexities. In other words, at a great distance form its empirical source, or after much abstract inbreeding, a mathematical subject is in danger of degeneration. [18, pg. 291]

Modern computational mathematics provides a medium of communication between theoretical and applied mathematics, and a route to empirical reality. What we can compute, we can be fairly certain has some tangible existence and some tangible possibility of leading to greater understanding of both the corpus of modern mathematics and also the universe around us.

For example, one can argue that the traditional hierarchy of rational, algebraic, elementary and "advanced" functions was driven by a pre-computer age. Exponentials, logarithms, sines and cosines are all considered "elementary," yet the elliptic integral functions

$$\mathrm{K}(x) := \int_0^1 \frac{\mathrm{d}t}{\sqrt{(1-t^2)(1-x^2t^2)}}$$
$$\mathrm{E}(x) := \int_0^1 \frac{\sqrt{1-x^2t^2}}{\sqrt{1-t^2}}\,\mathrm{d}t$$

are, from a computational point of view, simpler and easier to compute. This is because these functions can be quickly evaluated by quadratically convergent arithmetic-geometric mean calculations, and so for high precision calculations they are faster to compute than the "elementary" functions. Indeed, K admits of a quadratic transformation

$$\mathrm{K}\left(k\right) = \left(1+k_1\right)\mathrm{K}\left(k_1\right), \quad k_1 := \frac{1-\sqrt{1-k^2}}{1+\sqrt{1-k^2}}, \tag{6}$$

as was known already to Landen, Legendre and Gauss. To compute $\mathrm{K}(\pi/6) = 1.699075885\ldots$ to five places requires using (6) only twice and then estimating the resultant integral by $\pi/2$. A third step gives the ten-digit precision shown.

In fact, the elementary functions can be computed from the elliptic functions. For example, the logarithm to high precision can be computed as an approximation of K [12, 10, 11]. Explicitly, [12, Algorithm 7.1] gives for $1/2 < x < 1$ and $n > 3$ that

$$\left|\log\left(x\right) - \mathrm{K}'\left(\frac{1}{10^n}\right) + \mathrm{K}'\left(\frac{x}{10^n}\right)\right| \leq \frac{n}{10^{2(n-1)}}, \tag{7}$$

where $\mathrm{K}'(\mathrm{x}) = \mathrm{K}(\sqrt{1-\mathrm{x}^2})$.

The same can be said of algebraic functions. Even in the case of cubic algebraic numbers, in practice it is usually more efficient to numerically solve the underlying polynomial, using Newton iterations, than to use the closed form for a solution due to Cardano; for higher-degree polynomials it is almost always faster to use numerical methods (and hardly any analytic solutions exist).

Thus, perhaps the traditional taxonomy of functions and their computational complexity needs to be re-thought in the computer age.

Along this line, it is interesting to compare theoretical mathematics with the field of theoretical computer science. On one hand, some remarkable and significant results have been produced in the field. We now know that there are some computations that are simply not possible — for example, we know that it is not possible to infallibly determine, by means of a finite computer program running on a real computer, whether or not another computer program has an infinite loop (this is a result originally due to Turing). It cannot, because such a program cannot perform this diagnosis on itself.

Similarly, the $P = NP$? problem is as intriguing as it is far-reaching. This problem is described as follows by the Clay Mathematics Institute, which has offered a US$1,000,000 prize for its resolution:

> Suppose that you are organizing housing accommodations for a group of four hundred university students. Space is limited and only one hundred of the students will receive places in the dormitory. To complicate matters, the Dean has provided you with a list of pairs of incompatible students, and requested that no pair from this list appear in your final choice. This is an example of what computer scientists call an NP-problem, since it is easy to check if a given choice of one hundred students proposed by a coworker is satisfactory (i.e., no pair taken from your coworker's list also appears on the list from the Dean's office), however the task of generating such a list from scratch seems to be so hard as to be completely impractical. Indeed, the total number of ways of choosing one hundred students from the four hundred applicants is greater than the number of atoms in the known universe!

Some problems now known to be in the category $NP$-complete (and thus presumed to be not solvable in reasonable computer time) include [20]:

- The *traveling salesman problem*. Given a list of cities and distances between each pair, find the shortest route that visits each city exactly once and returns to the start.
- The *knapsack problem*. Given a set of items, each with a weight (or size) and a value, find how many of each item to include in a collection so that the total weight is less than a specified maximum weight, and the total value is as great as possible.
- The *subgraph isomorphism problem*. Given two graphs, determine whether one contains a subgraph that is isomorphic to the other.

However, these lines of research have only had muted impact on the real world of mathematical and scientific computing. The main difficulty is that $NP$-complete considerations are driven by a worst-case analysis of a measure that does not reflect practical usage. After all, providing $O(n)$ bounds on the cost, polynomial or otherwise, does not say much if we do not know the size of the high-order coefficients of the polynomial. At a minimum, one must distinguish between questions of the possibility or impossibility of solving the most

general form of these problems and the practical difficulty of obtaining satisfactory practical solutions to real-world versions of these problems.

For example, while the traveling salesman problem is known to be $NP$-complete, problems of this general type are solved routinely, every day, in both scientific research and private industry. Airlines, for example, use sophisticated computer programs running on highly parallel computer systems to schedule their aircraft, logistics and staff to the ever-changing requirements of their passenger-destination load. This is done using linear or semidefinite programming techniques, which are special cases of mathematical optimization. They are known to have effective algorithms in $P$, while the classical simplex method, which is still heavily used, does not lie in $P$.

Relatedly Goemans and Williamson showed for the $NP$-hard 'min-cut/max-flow' problem that a randomized version of a semidefinite relaxation after dualization — which being a semidefinite program can be solved approximately in polynomial time — has expected performance exceeding 0.87856% of the exact solution [13, §3.3].

Even the navigation software now incorporated into virtually all smartphones performs computations of this type to find a shortest-time route. For that matter, the C, C++, Fortran and Java compilers that are used countless times every day by software programmers routinely employ similar algorithms to schedule instructions for optimal performance of the generated code.

It is important to note that in real-world applications, certainly including airline scheduling, navigation and compilers, it is *not* necessary to find the absolutely optimal solution — a solution that is reasonably close to an optimal solution is entirely satisfactory. In this regards, the theoretical problems addressed, say, in studies of $NP$-complete problems, are of a different class and not really applicable.

Similar considerations apply to other arenas of computer science. In parallel computing, for example, thousands of papers have been written presenting algorithms for the parallel random access (PRAM) model of computing. This model presumes that each of $n$ processors can either perform one arithmetic operation or read from or write to any location in a shared memory. However, scientists performing scientific computations on parallel computer systems have not found this body of research very useful, because the model does not match very well the characteristics of real parallel computers. Instead, virtually all computers at the present and foreseeable future time employ a cache memory system, wherein data access to anything but local registers requires many more clock periods than arithmetic operations; and accesses to data in a remote memory node requires many times more clock periods than accessing data in the local node. For these reasons, today most performance analysis and tuning of algorithms and applications are today targeted to more realistic models, such as the $\log P$ model developed by researchers at the University of California, Berkeley [19] and the roofline model developed by Samuel Williams [23].

A related issue in parallel computing is that for many problems of practical, everyday interest, in mathematical, scientific or even business applications, considerations of massive parallelism are moot, because the problem to be solved simply does not possess sufficient concurrency to justify parallel processing (a consequence of *Amdahl's Law* [6, pg. 348]), particularly when the ever-present software and hardware overhead of coordinating large numbers of parallel processors is considered. It is true that some applications do possess very high levels of concurrency; many such applications are running on large-scale parallel supercomputers operated by universities and government laboratories. But for the vast majority of applications, only modest levels of parallelism can be efficiently exploited; and

for still others, only single-threaded execution makes practical sense.

One final item that should be mentioned is the ineluctable fact that mathematicians, scientists and others who are using computers today in their research often do their work via heavy-duty software layers, such as the mathematical software environments *Maple* and *Mathematica*. In such environments it is often not possible to know what algorithms are really being performed "under the hood." This fact considerably complicates any attempts by the user to employ advanced algorithms.

# 6 Conclusion

We have observed that computational mathematics is a more honest representation of mathematics, in that it does not hide or obscure the experimental processes by which a mathematical hypothesis is discovered. Only later are such discoveries turned into the polished, axiomatic expositions that we read in textbooks.

We have also observed that computational mathematics leads to interesting and important considerations of reliability — how to define and arrive at secure mathematical knowledge, given that computation is inherently subject to errors of many types, ranging from programming errors to submicroscopic glitches rooted in quantum mechanics. Yet we have also seen that suitably chosen validity tests can mitigate these errors, turning questionable calculations into extremely reliable mathematical and scientific assertions. Indeed, at some point, computations are arguably as reliable, if not more so, than formal reasoning.

Additionally, there is considerable evidence that a computational approach to mathematics permits the field to escape the trap, originally highlighted by John von Neumann, of becoming so isolated from the richer empirical world of modern science that it becomes both sterile and irrelevant. And even in computer science, connections to real-world mathematical and scientific computations can help focus research in the field in useful directions and help it avoid similar traps of isolation and irrelevance.

While we have discussed current computer architecture and not advanced quantum computing, we are relatively sure that even such advances will not entirely change the situation.

Finally, an expanded reliance on computation in mathematics does need to guard against a diminution in the reliability of the mathematical corpus. Proofs still need to be developed when possible and heuristics need to be clearly labeled.

## References


[1] David H. Bailey, Jonathan M. Borwein, Cristian S. Calude, Michael J. Dinneen, Monica Dumitrescu and Alex Yee, "An empirical approach to the normality of pi," *Experimental Mathematics*, **21** (2012), 375–384.

[2] David H. Bailey and Peter B. Borwein and Simon Plouffe, "On the rapid computation of various polylogarithmic constants," *Mathematics of Computation*, **66** (1997), 903–913.

[3] David H. Bailey and Jonathan M. Borwein, "Exploratory experimentation and computation," *Notices of the American Mathematical Society*, **58** (Nov 2011), 1410–1419.

[4] David H. Bailey, Jonathan M. Borwein, Vishal Kapoor and Eric Weisstein, "Ten problems in experimental mathematics," *American Mathematical Monthly*, **113** (June 2006), 481–509.

[5] David H. Bailey, Jonathan M. Borwein and Victoria Stodden, "Facilitating reproducibility in scientific computing: Principles and practice," in Harald Atmanspacher and Sabine Maasen, eds, *Reproducibility: Principles, Problems, Practices*, John Wiley and Sons, New York, to appear, 2016.

[6] David H. Bailey, Lin-Wang Wang, Hongzhang Shan, Zhengji Zhao, Juan Meza, Erich Strohmaier and Byounghak Lee, "Tuning an electronic structure code," in David H. Bailey, Robert F. Lucas and Samuel W. Williams, ed., *Performance Tuning of Scientific Applications*, CRC Press, 2011.

[7] Robert Baillie, David Borwein, and Jonathan Borwein, "Some sinc sums and integrals," *American Mathematical Monthly*, **115** (2008), 888–901.

[8] Jacob D. Bekenstein, "Universal upper bound on the entropy-to-energy ratio for bounded systems," *Physical Review D*, **23** (15 Jan 1981), 287–298.

[9] "Bekenstein bound," *Wikipedia*, viewed 8 Jan 2016, `https://en.wikipedia.org/wiki/Bekenstein_bound`.

[10] Jonathan M. Borwein and David H. Bailey, *Mathematics by Experiment: Plausible Reasoning in the 21st Century*, A. K. Peters Ltd., Natick, MA, 2004. Second edition, 2008.

[11] Jonathan M. Borwein, David H. Bailey and Roland Girgensohn, *Experimentation in Mathematics: Computational Paths to Discovery*, A. K. Peters Ltd., Natick, MA, 2004.

[12] Jonathan M. Borwein and Peter B. Borwein, *Pi and the AGM: A Study in Analytic Number Theory and Computational Complexity*, John Wiley, New York, 1987, paperback 1998.

[13] Jonathan M. Borwein and Jon D. Vanderwerff, *Convex Functions: Constructions, Characterizations and Counterexamples*, Cambridge University Press, London, 2010.

[14] "Lennart Axel Edvard Carleson," University of St. Andrews, Scotland, `http://www-groups.dcs.st-and.ac.uk/history/Biographies/Carleson.html`.

[15] David Epstein and Sylvio Levy, "Experimentation and proof in mathematics. Notices of the American Mathematical Society, June, 1995.

[16] "Holographic principle," *Wikipedia*, viewed 8 Jan 2016, `https://en.wikipedia.org/wiki/Holographic_principle`.

[17] Des MacHale, *Comic Sections: Book of Mathematical Jokes, Humour, Wit and Wisdom*, Boole Press, Ltd., New York, 1993.

[18] Morris Kline, *Mathematics: The Loss of Certainty*, Oxford University Press, London, 1980.

[19] David Culler, Richard Karp, David Patterson, Abhijit Sahay, Klaus E. Schauser, Eunice Santos, Ramesh Subramonian and Thorsten von Eicken, "LogP: Towards a realistic model of parallel computation," *Proceedigns of the Fourth ACM SIGPLAN Symmposium on Principles and Practice of Parallel Programming*, May 1993, San Diego, CA.

[20] "P versus NP problem," *Wikipedia*, viewed 10 Jan 2016, `https://en.wikipedia.org/wiki/P_versus_NP_problem`.

[21] Max Tegmark, *Our Mathematical Universe: My Quest for the Ultimate Nature of Reality*, Knopf, New York, 2014.

[22] Alex Vilenkin, *Many Words in One: The Search for Other Universe*, Hill and Wang, New York, 2006.

[23] Samuel W. Williams, "The roofline model," in David H. Bailey, Robert F. Lucas and Samuel W. Williams, ed., *Performance Tuning of Scientific Applications*, CRC Press, 2011.

[24] Alexander J. Yee and Shigeru Kondo, "12.1 trillion digits of pi, and we're out of disk space," `http://www.numberworld.org/misc_runs/pi-12t/`.