Future Prospects for Computer-Assisted Mathematics

David H. Bailey* and Jonathan M. Borwein†

December 1, 2005

Abstract

The recent rise of "computer-assisted" and "experimental" mathematics raises intriguing questions as to the future role of computation in mathematics. These results also draw into question the traditional distinctions that have been drawn between formal proof and computationally-assisted proof. This article explores these questions in the context of the growing consensus among computer technologists that Moore's Law is likely to continue unabated for quite some time into the future, producing hardware and software much more powerful than what is available today.

1 Introduction

Recent years have seen the flowering of what is often termed "computer-assisted" or "experimental" mathematics, namely the utilization of modern computer technology as an active tool in mathematical research. In particular, a combination of commercial software (notably *Mathematica* and *Maple*), online tools and custom-written computer programs are being used to test conjectures, discover new identities, perform symbolic manipulations, plot data and even conduct formal proofs.

With regards to computer technology, Moore's Law (the observation that computer technology doubles in aggregate power and capacity every 18 months or so) continues unabated, having defied numerous predictions that it will soon end (most recently the 100 nanometer "barrier"). At the present time, industry experts, including Gordon Moore himself, predict that it will continue for at least ten more years [1]. In fact, if some of the promising research in nanotechnology (the science of constructing devices and materials

*Lawrence Berkeley National Laboratory, Berkeley, CA 94720. dhbailey@lbl.gov. This work supported by the Director, Office of Computational and Technology Research, Division of Mathematical, Information, and Computational Sciences of the U.S. Department of Energy, under contract number DE-AC02-05CH11231.

†Canada Research Chair, Faculty of Computer Science, Dalhousie University, Halifax, NS, B3H 2W5. jmborwein@cs.dal.ca. This work supported in part by NSERC and the Canada Research Chair Programme.

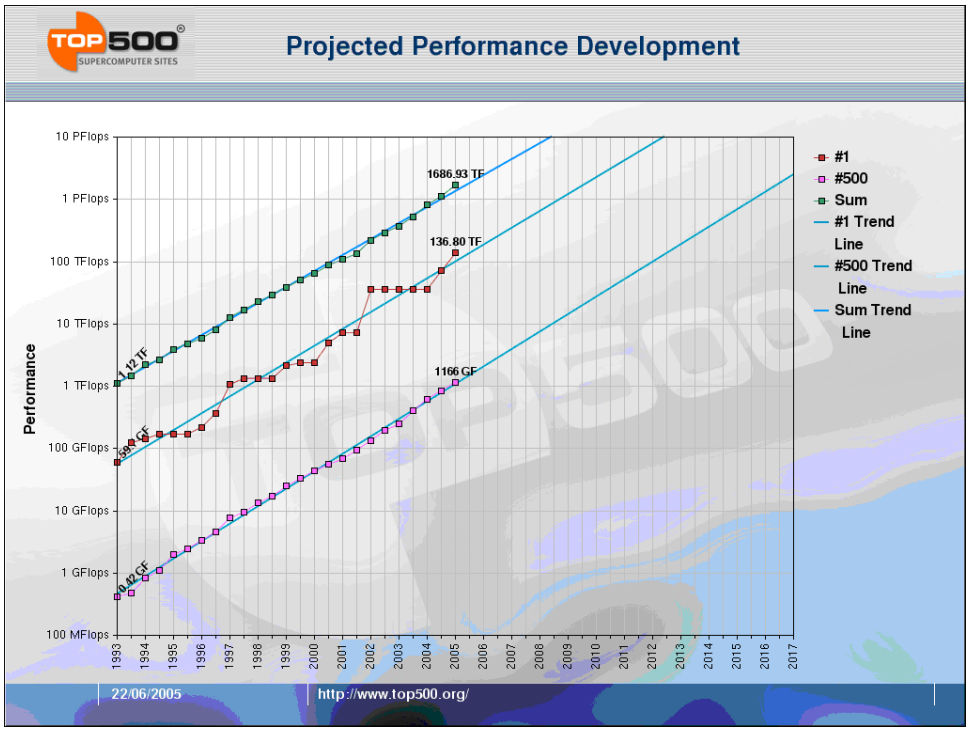

Figure 1: High-end computer performance projections.

at the molecular level) comes to full flower, Moore's Law may well continue for many more years [18, pg 7–30]. For example, carbon nanotube-based memory devices, which promise to store ten times as much data as conventional devices, may be commercially available as early as 2006 [21]. In another development, researchers at Hewlett-Packard have fabricated "nano-imprint crossbar" devices with half-pitch feature spacing of only 17 nanometers, and plan to reduce this figure by a factor of four within two years [17][16].

If these projections are even partially realized, future computer systems will be many thousands of times more powerful than those being used today. See Figure 1, which plots history and future projections of high-end computer system performance, based on data from the Top500 list (a twice-yearly updated ranking of the world's most powerful computer systems). In this figure, 1 Gflop/s, 1 Tflop/s and 1 Pflop/s denote one billion, one trillion, and one quadrillion (10^{15}), respectively, floating-point operations per second.

These developments have led some mathematicians to wonder whether computers one day will be smarter at math than we are. But in many respects, they already are—*Mathematica* and *Maple* today routinely perform integrals and other types of manipulations that are well beyond what humans can reasonably do. What's more, they normally perform such manipulations without the errors that humans are prone to make.

In this article, we will briefly summarize a few examples of computational mathematics in action, and then consider some of the questions that these developments present to future mathematical research. Some additional examples may be found in [2][11].

2 A New Formula for Pi

Perhaps the best-known result to emerge from experimental mathematics so far is the discovery of a new formula for π (now known as the "BBP" formula) by a computer program in 1996 [5][11, pg 118–125]:

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

This formula has the remarkable property that it permits one to directly calculate binary or hexadecimal digits of π beginning at some position n, without needing to calculate any of the preceding digits. It was discovered by Peter Borwein (brother of Jonathan) and Simon Plouffe, using a computer program written by Bailey that implements Helaman Ferguson's "PSLQ" integer relation algorithm, using high-precision arithmetic (200 digits in this case). An integer relation algorithm is one that, given n real numbers $(x_1, x_2, \cdots, x_n)$, finds n integers $(a_1, a_2, \cdots, a_n)$, if they exist, such that $a_1 x_1 + a_2 x_2 + \cdots + a_n x_n = 0$ to within available numeric precision.

In a surprising development, it was recently found that the existence of the BBP formula has an intriguing connection to the question of whether or not the binary digits of π are "normal" (statistically random in a certain sense). In particular, it was found that the question of normality of π reduces to the question of whether a certain iteration is a uniform random number generator in the unit interval [8][9][11, pg 148–156]. This line of research is still being actively investigated.

Numerous other formulas and related results of this general type have now been discovered, using this same methodology, namely a combination of integer relation algorithms and high-precision numerical computations. Here is just a brief sample:

$$\begin{aligned} \pi^2 &= \frac{2}{27} \sum_{k=0}^{\infty} \frac{1}{729^k} \left[\frac{243}{(12k+1)^2} - \frac{405}{(12k+2)^2} - \frac{81}{(12k+4)^2} - \frac{27}{(12k+5)^2} \right. \\ &\quad \left. - \frac{72}{(12k+6)^2} - \frac{9}{(12k+7)^2} - \frac{9}{(12k+8)^2} - \frac{5}{(12k+10)^2} + \frac{1}{(12k+11)^2} \right] \\ \zeta(8) &= \frac{36 \cdot 64}{1373} \left[\sum_{k=1}^{\infty} \frac{1}{k^8 \binom{2k}{k}} + \frac{9}{4} \sum_{k=1}^{\infty} \frac{1}{k^4 \binom{2k}{k}} \sum_{j=1}^{k-1} \frac{1}{j^4} + \frac{3}{2} \sum_{k=1}^{\infty} \frac{1}{k^2 \binom{2k}{k}} \sum_{j=1}^{k-1} \frac{1}{j^6} \right] \end{aligned}$$

$$\frac{25}{2} \log \left[\frac{781}{256} \left(\frac{57 - 5\sqrt{5}}{57 + 5\sqrt{5}} \right)^{\sqrt{5}} \right] = \sum_{k=0}^{\infty} \frac{1}{5^{5k}} \left(\frac{5}{5k+2} + \frac{1}{5k+3} \right).$$

Most of these results have been obtained using very modest computational platforms, typically just a personal computer or workstation, and have involved numeric precision levels of only 100 to 500 digits. However, one such computation required 50,000-digit arithmetic and a 64-CPU parallel computer system to obtain the result [7]. This raises the intriguing question of how many more such formulas will be found when computers many thousands of times more powerful are available.

The computational cost of integer relation detection, using the PSLQ algorithm, grows as roughly n^3, where n is of the number of terms in the relation (in particular, the number of iterations required grows at this rate). When we further take into account the additional numeric precision required to resolve relations when n is large, we obtain a cost that grows as roughly $n^4 \log n$. Today we can usually find integer relations involving 100 terms, typically requiring 5,000-digit precision or so, in a few hours' run time on a personal computer or workstation. Given the $n^4 \log n$ scaling of the computational cost, extending the reach of these methods to say $n = 1000$ seems truly formidable. Yet such power will be available, even in a personal computer, by roughly the year 2020, even assuming no improvement in the underlying integer relation algorithms. If we assume that faster algorithms will be found (which seems very likely), then we may well be able to routinely solve such problems much sooner. Employing highly parallel computer technology will further accelerate this timetable.

3 High-Precision Computations

As we mentioned in the previous section, high-precision numerical computations are a key part of the process of discovering new mathematical identities using integer relation methods. But these same high-precision computations can also be used to verify a relation once it has been discovered by any means, whether by experimental computations or conjectural reasoning.

For example, recently some new techniques have been found to evaluate the numerical value of integrals to very high accuracy (hundreds or thousands of digits), even, in many cases, for functions with singularities [12, pg 306–314]. In one application of these techniques, the present authors established that the relation

$$\frac{24}{7\sqrt{7}} \int_{\pi/3}^{\pi/2} \log \left| \frac{\tan t + \sqrt{7}}{\tan t - \sqrt{7}} \right| dt \stackrel{?}{=}$$
$$\sum_{n=0}^{\infty} \left[\frac{1}{(7n+1)^s} + \frac{1}{(7n+2)^s} - \frac{1}{(7n+3)^s} + \frac{1}{(7n+4)^s} - \frac{1}{(7n+5)^s} - \frac{1}{(7n+6)^s} \right],$$

which arises in quantum physics, holds to 20,000-digit accuracy (note that the integral has a nasty singularity at $t = \tan^{-1} \sqrt{7}$). This calculation required 46 minutes on 1024 CPUs of an Apple-based parallel computer [3]. The question mark is used here because no formal proof is yet known.

These very high-precision numerical confirmations raise an interesting question: Which would you rather trust, a mathematical theorem that is the final result of a long, difficult paper, fully understood only by a handful of people worldwide, or a formula that has been verified to 20,000-digit accuracy by a computer?

Clearly some caution must be exercised in this regard, since examples of high-precision "frauds," namely relations that appear to hold to unusually high accuracy but which are not precisely equal, are known in the mathematical literature [12, pg 11-15]. One

particularly perplexing example is the following [4]:

$$\int_0^\infty \cos(2x) \prod_{n=1}^{\infty} \cos\left(\frac{x}{n}\right) dx \;=$$
$$0.392699081698724154807830422909937860524645434187231595926812285162...$$

One is tempted to conclude from this numerical value that the integral in question is equal to $\pi/8$. But a careful comparison with a high-precision value of $\pi/8$, namely,

$$0.392699081698724154807830422909937860524646174921888227621868074038...$$

discloses that the two differ after the 42nd decimal place! However, such instances are highly exceptional, and only a handful of rather contrived examples are known that hold beyond 100 digits or so.

4 A Computer-Assisted Proof of Kepler's Conjecture

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 rigorous proof. German mathematician David Hilbert included the Kepler conjecture in his famous list of unsolved problems in 1900.

In 1994, Thomas Hales, now at the University of Pittsburgh, proposed a five-step program that would result in a proof. In 1998, Hales announced that the program was now complete [15]. 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 are here: `http://www.math.pitt.edu/~thales/kepler98`.

The journal *Annals of Mathematics* has decided to publish Hales' paper, but with a cautionary note, because, as they explain, although a team of referees is "99% certain" that the computer-assisted proof is sound, they have not been able to verify every detail [22]. One wonders if every other article in this journal has implicitly been certified to be correct with more than 99% certainty! In an attempt to resolve the remaining uncertainty, Hales has decided to embark on a project to construct a computer-based formal proof and anticipates completion by 2012. However, even this is not likely to quell the controversy among some traditional mathematicians regarding computer-based proofs.

5 Probabilistic Primes versus Provable Primes

Today whenever one uses a credit card to make a purchase over the Internet, it is quite likely that at some point in the process, one's Internet browser constructs a pair of large prime numbers. Through the years numerous efficient computational algorithms have been found to provably certify that an integer is prime, culminating with the recent discovery by three Indian mathematicians of a "polynomial time" scheme [12, pg 3023].

However, the most widely used scheme in practice is the Monier-Rabin probabilistic primality test [13][12, pg 300–303]. This test specifies that a certain numerical computation be repeated in several trials, with a certain auxiliary parameter chosen pseudorandomly in each trial. If the integer passes one trial, it is prime with probability at least 3/4, so that m trials increase this probability to $1 - 1/4^m$. In fact, for large test integers n, the probability is even closer to unity. For instance, if n has 500 bits, then this probability is greater than $1 - 1/4^{28m}$. Thus a 500-bit integer that passes this test even once is prime with prohibitively safe odds—the chance of a false declaration of primality is less than one part in Avogadro's number (6×10^{23}). If it passes the test four times, then the chance of false declaration of primality is less than one part in a googol (10^{100}).

It is worth pointing out that such tiny probabilities are many times more remote than the chance that an undetected hardware error occurs during the calculation, not to mention the possibility of a computer program bug. Given these realities, what is the point of distinguishing between a "provable" primality test (performed on a computer) and a probabilistic primality test (also performed on a computer)?

6 Validity Checks for Large Computations

In several recent mathematical computations, computer runs of many hours were required, sometimes on highly parallel computers. Given the many possible sources of error in such calculations (computer program, processor, memory, network, disk, compiler, operating system, etc.), one can rightly ask why anyone would ever have confidence in the results.

In fact, these computations typically employ very rigorous validity checks. For example, Richard Crandall, Ernst Mayer and Jason Papadoupoulos recently determined that the Fermat number $F_{24} = 2^{2^{24}} + 1$ is composite. This calculation employed a "wavefront" scheme, where a faster computer system computed a chain of squares modulo F_{24}, such as $3^{2^{1000000}} \bmod F_{24}$, $3^{2^{2000000}} \bmod F_{24}$, $3^{2^{3000000}} \bmod F_{24}$, $\cdots$. Then each of a set of slower computers started with one of these intermediate values, squared it 1,000,000 times modulo F_{24}, and checked to see if the result (a 16-million-bit integer) precisely reproduced the next value in the chain. If it did, then this is very strong evidence that both computations were correct. If not, then the process was repeated [14, page 187].

Along this line, Yasumada Kanada of the University of Tokyo recently computed the first trillion hexadecimal digits of π (and also the first trillion decimal digits). To validate his results, he first computed the first trillion hexadecimal (base-16) digits of π, using two different formulas. Both of these computations precisely agreed to over one trillion digits. Then he used a variant of the BBP formula for π to independently calculate 24 hexadecimal digits beginning at position one trillion. The result precisely agreed with the two earlier calculations. Needless to say, it is exceedingly unlikely that independent computations, employing a completely different technique, would each produce the same 24-long hexadecimal digit sequence, unless each is in fact correct (the probability of such an accidental agreement, in a heuristic sense, is $16^{-24} \approx 1.26 \times 10^{-29}$).

In a similar way in 1999, Colin Percival, then an undergraduate student at Simon

Fraser University, harnessed a worldwide network of personal computers to calculate a segment of binary digits of π beginning at the quadrillionth (i.e., 10^{15}-th) place, using a variant of the BBP formula for π. Such calculations can be checked by computing two closely overlapping sections of digits, and verifying that the overlapped digits precisely agree (although Percival's actual scheme was somewhat different).

Again, this raises the question: Which would you rather trust, a mathematical theorem that is the final result of a long, difficult paper, fully understood only by a handful of people worldwide, or a computational result that has been confirmed by multiple independent, exacting validity checks?

7 Mathematical Knowledge Management

The global **MKM Interest Group** founded in 2005 writes:

> *Mathematical Knowledge Management is an exciting new field in the intersection of mathematics and computer science. We need efficient, new techniques—based on sophisticated formal mathematics and software technology—for taking fruit of the enormous knowledge available in current mathematical sources and for organizing mathematical knowledge in a new way. On the other side, due its very nature, the realm of mathematical information looks as the best candidate for testing innovative theoretical and technological solutions for content-based systems, interoperability, management of machine understandable information, and the Semantic Web.* (`http://www.mkm-ig.org/index.html`)

Twenty-five years ago, the best theorem-proving systems could return a proof of the *Cantor diagonal theorem*, when fed a careful diet of axioms. Today, as suggested by Hale's project, we stand on the edge of an extraordinary ability to both discover and confirm formal mathematics. The French system COQ, has apparently been used to provide formal proofs of the *Prime number theorem* and the *Four color theorem*—two of the luminous highlights of pure mathematics [20, 19].

Similarly, early work by Herbert Simons and others on computational scientific-theory construction is also showing rapid development, as instanced by the health of software such as *Graffiti*, which can conjecture and often dispose of graph-theoretic results using databases of graph structure.

While many of the issues in mathematical knowledge management are shared by all disciplines, the need for reliable mathematical optical character recognition (OCR) stands out separately. There are still no reliable, scalable methods of identifying mathematics in documents. Some of the most interesting current projects arise in Japan, where the spatial demands of the language recognition are closer to those of mathematics. Nonetheless, it is reasonable to assume that within five to ten years mathematical OCR will be generally accessible. This will further facilitate the computerization of the entire mathematical research process. At that point its future relies more on commercial than technical issues.

8 What We Cannot Do

Even given such examples, it should not be presumed that amassing huge amounts of processing power can solve all mathematical problems, even those that are amenable to computational analysis.

For one thing, it is quite likely that some mathematical computing problems are in a class that fundamentally cannot be solved except by schemes that scale exponentially in cost with the size of the problem (assuming the widely believed equivalence of P and NP, a conjecture from the field of theoretical computer science). For such problems, once they are sufficiently large in size, no amount of advanced computing technology or parallel processing is likely to solve them.

Along this line, consider Clement Lam's 1991 proof of the nonexistence of a finite projective plane of order ten [11, pg 4]. This involved a search for a configuration of n^2+n+1 points and equally many lines. Lam's computer program required thousands of hours of run time on a Cray computer system. Lam estimates that the next case ($n = 18$) susceptible to his methods would take millions of years on any conceivable architecture.

Some mathematical computations are "naturally parallel" and thus readily suited for implementation on highly parallel computer systems. However, these tend not to be problems of central interest in mathematics. The majority of interesting large-scale calculations in computational mathematics, like their counterparts in computational physics and chemistry, require significant communication between independent computing nodes. Parallel implementations of such calculations require considerable programming effort, in addition to a well-designed parallel system with a strong interconnection network.

More importantly, such calculations are subject to fundamental limits of parallel computing, especially *Amdahl's Law*, namely the observation that the serial portion of a computer program will dominate the run time unless it constitutes an extremely small portion of the program. Along this line, it appears that integer relation detection is one example of a demanding computation that possesses only limited concurrency. Variants of PSLQ that are suitable for parallel processing are known, but even in the best schemes, concurrency is limited to less than $n/2$, where n is the length of the test vector [6]. Thus, while PSLQ variants have been successfully implemented on systems with up to 64 CPUs (for problems where n was well over 100), much more highly parallel implementations, for reasonably-sized n, will be a challenge.

9 What Does the Future Hold?

In spite of the difficulties mentioned in the previous section, it seems clear that computation is destined to assume a much more important role in future mathematical research than at the present time. For one thing, present-day mathematical software is greatly improved over what was available just a few years ago, and as a result many mathematicians are just now beginning to be fully skilled and experienced in using these tools. For this reason alone, we believe that we will see much greater usage of computational math facilities in the future.

It also seems inevitable that mathematicians will need to rethink the distinction between human-proven results and computer-proven results. As we have seen, it is increasingly difficult to distinguish between a purely human proof, and a proof that has utilized computational resources for at least part of the overall process of discovery and validation. We may even need to reconsider the distinction between computer-proven results and experimentally discovered results that have been confirmed with very strong numerical evidence. Such distinctions will only fade further as computer-based mathematical tools become more widely utilized.

As we mentioned in the introduction, experts are now confident that Moore's Law, which to date has sustained 40 years of exponential growth, will continue for at least another ten years, and, given some of the recent interesting developments in nanotechnology, it may continue much longer. Even more interesting is the possibility of quantum computing, which is based on quantum superposition, an eerie effects of quantum theory. Quantum computation, if fully realized, will dramatically accelerate certain classes of mathematical computation. At present, only a few small demonstrations have been done (such as the factoring of $15 = 3 \cdot 5$), but scientists worldwide are exploring ways to extend the reach of these demonstrations to significantly larger achievements.

If even some of these predictions of future computing technology are realized, future mathematical computing facilities will be thousands or even millions of times more powerful than they are today. We may well see the day when virtually every mathematical question can be explored on the computer, using much more comprehensive and powerful software and hardware than any that is available today. We can only dimly imagine such a future at the present time. But perhaps it is not too early to try. Certainly there are many philosophical implications [10].

References

[1] Elinor Mills Abreu, "Gordon Moore Sees Another Decade for Moore's Law," 11 Feb 2003, available at http://www.cnn.com/2003/TECH/biztech/02/11/moore.law.reut.

[2] David H. Bailey and Jonathan M. Borwein, "Experimental Mathematics: Examples, Methods and Implications," *Notices of the American Mathematical Society*, May 2005, available at http://crd.lbl.gov/~dhbailey/dhbpapers/ams-expmath.pdf.

[3] David H. Bailey and Jonathan M. Borwein, "Highly Parallel, High-Precision Numerical Integration," manuscript, 2005, available at http://crd.lbl.gov/~dhbailey/dhbpapers/quadparallel.pdf.

[4] David H. Bailey, Jonathan M. Borwein, Vishal Kapoor and Eric Weisstein, "Ten Problems in Experimental Mathematics," American Mathematical Monthly, to appear, 2005, available at http://crd.lbl.gov/~dhbailey/dhbpapers/tenproblems.pdf.

[5] David H. Bailey, Peter B. Borwein and Simon Plouffe, "On The Rapid Computation of Various Polylogarithmic Constants," *Mathematics of Computation*, vol. 66 (1997), pg 903–913.

[6] David H. Bailey and David J. Broadhurst, "Parallel Integer Relation Detection: Techniques and Applications," *Mathematics of Computation*, vol. 70, no. 236 (Oct. 2000), pg 1719–1736.

[7] David H. Bailey and David J. Broadhurst, "A Seventeenth-Order Polylogarithm Ladder," 1999, available at http://crd.lbl.gov/~dhbailey/dhbpapers/ladder.pdf.

[8] David H. Bailey and Richard E. Crandall, "On the Random Character of Fundamental Constant Expansions," *Experimental Mathematics*, vol. 10 (2001), pg 175–190.

[9] David H. Bailey and Richard E. Crandall, "Random Generators and Normal Numbers," *Experimental Mathematics*, vol. 11 (2003), pg 527–546.

[10] J. M. Borwein, "Implications of Experimental Mathematics for the Philosophy of Mathematics," chapter to appear in Bonnie Gold, editor, *Current Issues in the Philosophy of Mathematics From the Viewpoint of Mathematicians and Teachers of Mathematics*, 2006. [D-drive Preprint 280].

[11] Jonathan Borwein and David Bailey, *Mathematics by Experiment: Plausible Reasoning in the 21st Century*, A K Peters, Natick, MA, 2004. A condensed version of this book and the companion book (next in list) is available at http://www.experimentalmath.info.

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

[13] Richard Crandall and Carl Pomerance, *Prime Numbers: A Computational Perspective*, Springer-Verlag, Heidelberg, Germany, 2001.

[14] Richard E. Crandall and Jason S. Papadopoulos, "On the Implementation of AKS-Class Primality Tests," 2003, available at `http://developer.apple.com/hardware/ve/acgresearch.html`.

[15] Thomas C. Hales, "A Proof of the Kepler Conjecture," *Annals of Mathematics*, to appear.

[16] "Who Needs Transistors? HP Scientists Create New Computing Breakthrough at Molecular Scale," press release, 31 Jan 2005, available at `http://www.hp.com/hpinfo/newsroom/press/2005/050201a.html`; "HP Reveals Groundbreaking Design for Future Nano-electronic Circuits," press release, 9 Jun 2005, available at `http://www.hp.com/hpinfo/newsroom/press/2005/050609a.html`. Some information above is from a presentation by Stan Williams, Director of Quantum Science Research at HP, Santa Cruz, CA, 25 Oct 2005.

[17] Philip J. Kuekes, Gregory S. Snider and R. Stanley Williams, "Crossbar Nanocomputers," *Scientific American*, Nov. 2005, pg 72–80.

[18] Ray Kurzweil, *The Singularity Is Near*, Viking, New York, 2005.

[19] Dana Mackenzie, "What in the Name of Euclid Is Going on Here?" *Science*, vol. 307, March 4, 2005, pg 1402–1403.

[20] "Proof and Beauty", *Economist*, March 31, 2005.

[21] Gary Stix, "Nanotubes in the Clean Room," *Scientific American*, Feb. 2005, pg 82–85.

[22] George Szpiro, "Does the Proof Stack Up?" *Nature*, vol. 424 (2003), 3 Jul, 2003, pg 12–13.