Experimentation and Proof in Mathematics

David Epstein and Silvio Levy

The English word "prove"—as its Old French and Latin ancestors—has two basic meanings: to try or test, and to establish beyond doubt. The first meaning is largely archaic, though it survives in technical expressions (printer's proofs) and adages (the exception proves the rule, the proof of the pudding). That these two meanings could have coexisted for so long may seem strange to us mathematicians today, accustomed as we are to thinking of "proof" as an unambiguous term. But it is in fact quite natural, because the most common way to establish something in everyday life is to examine it, test it, probe it, experiment with it.

Prove: to try or test or to establish beyond doubt

As it turns out, much the same is true in mathematics as well. Most mathematicians spend a lot of time thinking about and analyzing particular examples. This motivates future development of theory and gives one a deeper understanding of existing theory. Gauss declared, and his notebooks attest to it, that his way of arriving at mathematical truths was "through systematic experimentation". It is probably the case that most significant advances in mathematics have arisen from experimentation with examples. For instance, the theory of dynamical systems arose from observations made on the stars and planets and, more generally, from the study of physically motivated differential equations. A nice modern example is the discovery of the tree structure of certain Julia sets by Douady and Hubbard: this was first observed by looking at pictures produced by computers and was then proved by formal arguments.

It is disturbing that such considerations are usually totally excluded from the published record. What one generally gets in print is a daunting logical cliff that only an experienced mountaineer might attempt to scale, and even then only with special equipment. Is this the best thing for the research community? Is it fair to graduate students? Should we give the impression that the best mathematics is some sort of magic conjured out of thin air by extraordinary people when it is actually the result of hard work and of intuition built on the study of many special cases? In our educational institutions, we spend too much time revealing these almost unscalable logical edifices instead of giving others

David Epstein is professor of mathematics at the University of Warwick, Coventry, U.K. His e-mail address is `dbae@maths.warwick.ac.uk`. *Silvio Levy is a research associate at the Geometry Center (University of Minnesota) and a writer, editor, and translator of mathematics and computer science. His e-mail address is* `levy@geom.umn.edu`.

the feeling that they too can participate in the work.

Partly in response to this lack, we founded in 1991 the quarterly *Experimental Mathematics*[1], whose mission is to present mathematics as a living entity, with examples, conjectures, and theory all interacting with and reinforcing each other. Now in its third volume, *Experimental Mathematics* has established a solid reputation as a first-rate research journal, but it definitely has a different flavor from most traditional journals. For one thing, its range is broad: among the fields represented so far are algebraic geometry, cellular automata, combinatorics, differential geometry, dynamical systems, geometric topology, group theory, harmonic analysis, number theory, optimal geometries, several complex variables, singularity theory, and wavelets. For another, each article is individually edited to ensure a high standard of exposition in an effort to reach as many readers as possible. But the main difference reflects the philosophy above: we are interested not only in theorems and proofs but also in the way in which they have been or can be reached.

Note that we do value proofs: experimentally inspired results that can be proved are more desirable than conjectural ones. However, we do publish significant conjectures or explorations in the hope of inspiring other, perhaps better-equipped researchers to carry on the investigation. The objective of *Experimental Mathematics* is to play a role in the discovery of formal proofs, not to displace them.

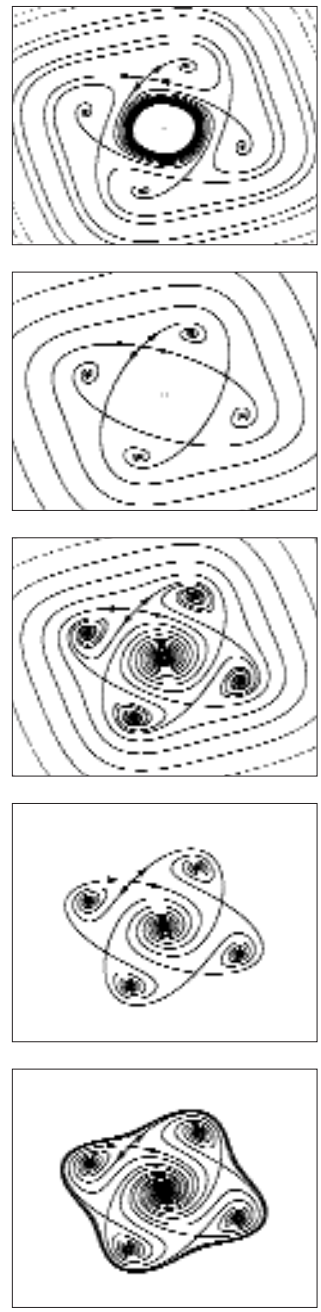
Enter the Computer

As we have argued, experimental mathematics has always been around, so the introduction of the computer into mathematics has made more of a quantitative than a qualitative difference. But in another sense it has been quite revolutionary, affecting profoundly, if subtly, the usually unexamined and unstated basic philosophy of professional mathematicians. Its deepest effect may also be the least frequently noted.

In terms of formal training, most mathematicians are brought up (brainwashed?) to believe in the standard von Neumann–Bernays–Gödel set theory. Even those who never study set theory have no option but to absorb basic dogmas like Zorn's Lemma. Epstein remembers as a young researcher coming across the ideas of constructive mathematics through the works of Bishop. But somehow one imbibed from the atmosphere, "This is not the way to succeed as a mathematician." Only a few strange people, like mathematical logicians, bothered about such questions, and one did not see any of them winning Fields Medals.

With the silent but constant education provided through the use of computers, it is now much easier to appreciate the meaning and significance of Bishop's work [1]. Earlier work of Brouwer and his followers had concentrated more on the destruction of the current philosophical foundation of mathematics and was therefore rejected by the vast majority of mathematicians. Instead of destruction, Bishop showed how to put a large part of analysis into a constructive framework. An interesting and challenging task, currently underway, is to put other parts of mathematics onto an equally solid foundation, and this will no doubt result in new insights and new algorithmic methods.

Nowadays, of course, even the constructive approach is not constructive enough. Algorithms that will finish in $10^{10^{10}}$ years are not as popular as those that give the answer a few seconds after the return key is hit. Everyone is in a hurry, even the contemplative mathematician.



The computer allows experimentation with examples, as shown in the illustration above from *Experimental Mathematics* [9].

The theory of three-dimensional manifolds provides an interesting example. Early on, low-dimensional topology was the source of the many interesting advances in constructive mathematics, largely through the work of Dehn in the 1920s [2]. In the late 1950s Haken created his theory of normal surfaces [3], which has been used to solve some important decidability problems: for example, Hemion and Waldhausen [4] produced an algorithm that takes as its input two knot diagrams and outputs whether the two knots are equivalent. Another recent success

[1] *More information about this journal and sample issues can be obtained over the World Wide Web at* `http://www.geom.umn.edu/locate/expmath` *or by sending e-mail to* `expmath@geom.umn.edu`. *Electronic subscriptions will soon be available. Low priced individual subscriptions are available for members of the AMS.*

for the theory of normal surfaces is Rubinstein's algorithm (improved and simplified by Thompson) to detect whether a finite simplicial complex is homeomorphic to the three-sphere.

Unfortunately, the theory of normal surfaces does not lend itself to practical programs, and this work, though constructive, is usually not feasible. Attempts are currently being made to bring such computations into range, but this seems a long way off, even if the number of simplices is small.

In contrast, Jeff Weeks's program *SnapPea*, largely based on work of Bill Thurston, can quickly compute amazing facts about a three-dimensional manifold, such as its geometric structure. *SnapPea* has had many remarkable successes in helping forward the understanding of three-dimensional manifolds—we are aware of over thirty research articles describing work that used *SnapPea*, in such journals as *Topology*, *Journal of Differential Geometry*, *Proceedings of the AMS*, and, naturally enough, *Experimental Mathematics* [5]. The fact that it has been possible to write such a program is a most remarkable achievement of the constructive approach to mathematics.

Another program that has had a substantial effect on theoretical developments is the *Surface Evolver*, by Ken Brakke [6], which computes surfaces with properties stated by the user. Among other successes, the *Evolver* can be credited with helping provide a counterexample to Lord Kelvin's longstanding conjecture about the least-area regular arrangement of membranes dividing three-space into cells of equal volume [7].

These and many other programs written daily in a variety of languages have shown us in a practical way what constructive mathematics means. We no longer need to think of constructivism in the abstract; instead, we can construct away to our heart's content, at least where we have good algorithms. The intuitionists' trichotomy "True, False, Don't know", so strange to the trained and rigorous mathematician raised on conventional set theory, is completely intuitive and obvious when the answer is to be provided by a machine.

There is a feedback loop: computers in mathematics enhance the importance of the constructivist point of view, and the constructivist point of view increases the use of computers.

Another consequence of computer experimentation we have observed is psychological—the theory achieves concreteness and immediacy. Have you ever had the feeling, while reading some abstract mathematical text, of being totally disconnected from reality? It is different when the theory interacts with examples. Even if the computer is only presenting to you lists of numbers and symbols, the fact that you have connected the theory with a physical object, a computing machine, can make you feel different about it. The effect is greatly heightened if you can induce the computer to draw pictures representing the mathematical objects in question; this is, of course, only possible in certain cases but is extremely rewarding.

Contact with data and examples helps to keep one on solid ground and helps to avoid the crass blunders to which we are nearly all prone when we adopt purely formal methods of reasoning. Multi-channel research is likely to be more successful than one-channel research, and mathematicians would do well to explore as many avenues as possible of mathematical (and nonmathematical) reality related to the concepts that are of interest to them.

But How Do You Know the Computer Gave the Right Answer?

The history of mathematics has witnessed a steady tightening of standards of rigor, in part from experience of previously accepted standards of argument going wrong. Mathematicians should be prepared to maintain their uncompromising support of rigor. Some of our colleagues worry that the computer culture will undermine the concept of proof.

Such worries seem to us misplaced and often stem from unmerited confidence in conventional proofs. Mistakes in the conventional mathematical literature sometimes survive for years without being noticed, a famous example being the original proof of Dehn's Lemma in three-manifold theory. Conversely, sometimes correct proofs are presented and mistakenly rejected. Haken's work on normal surfaces referred to above, although flawless from the formal point of view, took many years to penetrate the barrier of skepticism. It was "obvious" to mathematicians of the mid-1950s, who had just been exposed to the unsolvability of most problems in the theory of presentations of groups, that three-dimensional manifolds could not be treated algorithmically.

At a more humdrum level, almost every published mathematics paper we have read contains errors, and often these errors, although easy to correct once one understands the paper, can and do seriously mislead readers, particularly the inexperienced.

The danger of taking the results of a computer program on trust are not that different from taking a traditional result on trust. How many people who quote the resolution of singularities can vouch for the correctness of the proof? How many who quote the classification of finite simple groups are aware that gaps remain in the proof, even a decade after it was "established", not to mention that some details depend on

computer verifications? And these are two of the most famous successes of postwar mathematics. We are not advocating that one should take results on trust; we are merely pointing out inconsistencies in the approach of some mathematicians who are suspicious of computer proofs.

When an ordinary mathematical theorem is published, we do not require it to have a formal, mechanically verifiable proof. That would be too long and incomprehensible and may, in any case, be unattainable because of the world's limited amount of paper, brain cells, computer memory, whatever. Likewise, it is unreasonable to insist that a result that depends on computer calculations must come with a formal proof of correctness. What seems reasonable is to require a statement of the algorithm used and a publicly available implementation that can be independently checked. By "the algorithm" we mean, roughly, a list of steps, each of which is fairly easy to program correctly, and a proof (in the usual sense of a mathematics journal) that these steps would lead to the claimed performance.

Should we give the impression that the best mathematics is some sort of magic conjured out of thin air by extraordinary people when it is actually the result of hard work and of intuition built on the study of many special cases?

Skeptics will retort that there is no such thing as a nontrivial task that is "fairly easy to program correctly", and they are right. Programs almost invariably fail to do exactly what they are designed to do. Indeed, the authors' understanding of the state of the art in formal program verification is that specifying "what the program is designed to do" in a way that lends itself to automated analysis is itself very hard. Another difficulty is that even if a program is proved correct, the result relies on the correctness of the infrastructure (e.g., the absence of compiler bugs), which is currently impossible to guarantee.

Still, computers are used every day to design bridges, keep track of inventory, and fly space shuttles. We must learn to assign degrees of reliability to mathematical results that depend on computer calculations, as we do (or should) to mathematical results that do not depend on computer calculations, or to experimental results in other sciences, or to everyday operations such as balancing a checkbook. One of the hopes of the editors of *Experimental Mathematics* is to help the establishment of standards for experimental mathematics.

There is no question that claims are made about particular results of computer work without adequate justification, but this does not mean that every computer calculation that has not been formally verified is useless. Theorems survive by the creation of different and independent proofs, by being tested against examples, and, of course, by direct checking of the logic of the proof. This also applies to results that involve computer calculations. A researcher who announces a computer result should start by opening the program to public scrutiny, as we have said. If the result is verified independently by other investigators using different programs, its reliability is established. If not, the level of reliability depends on many factors: whether the code (program) is documented so its logic can be checked by other workers, how many people have done the checking, what language the program is written in, and so on.

(Regarding languages: other things being equal, one should consider more reliable a program written in a language of which well-established public implementations are available. But other things are seldom equal. A short program written in a language that does not satisfy this condition, such as *Mathematica*, may be more reliable than a longer *C* program that does the same job if it avoids the need to reimplement basic routines. Of course, it would certainly be better if details on *Mathematica*'s algorithms and their implementation were more widely known, even if commercial reasons inhibit the manufacturers from complete disclosure. At the other extreme, we have been told by an expert on computer risks that some UK government contracts insist on programs in assembly language because the compilers have not been checked for correctness. Unfortunately only fairly small programs can be checked by machine, and who checks the checkers?)

Sometimes it is much simpler to validate a proof than to discover it. Recently, Nathaniel Thurston [8], following work of David Gabai and Rob Meyerhoff, wrote a program to prove the nonexistence of groups of hyperbolic isometries satisfying certain conditions. The program is complex, and it ran for weeks; but the resulting mass of data can be checked by another, much shorter program, and if this shorter program can be proved correct, then the result is proved, even if the longer program is incorrect. Somewhat in the same spirit, for some problems there are computationally expensive algorithms that always return the right answer, together with a proof, and cheaper algorithms that have not been proved correct but seem to perform well in practice. The *Mathematica* functions PrimeQCertificate (reliable) and PrimeQ (not proved reliable) are examples for primality testing. A researcher might use the careless algorithm for exploratory work and thus arrive at a new result that can then be proved using the careful mode (always modulo the correctness of the implementation).

Note that the algorithms for proof verification that we are talking about are very specialized: the "proofs" they work on consist of data with a certain interpretation, a kind of shorthand, and only constitute proofs because of a background of traditional results that holds things together. (As an illustration of such an encoded proof, here is *Mathematica*'s proof that 1021 is prime:

```
{1021;10;{2;{3;2;{2}};{5;2;{2}};{17;3;{2}}}}:
```

A user of *Mathematica* who does not know number theory probably would not be able to make any sense of this, but a specialist in primality who has never used *Mathematica* may.) Conceivably, computers could also help check the logic of more general, traditional proofs. This is the subject of important research. But most mathematicians who are familiar with computers, including us, are skeptical about this possibility, at least any time soon.

Conclusion

The role of the computer in suggesting conjectures and enriching our understanding of abstract concepts by means of examples and visualization is a healthy and welcome development. There are caveats, but many of them also apply to the traditional procedures that have long been accepted by mathematicians.

We believe that, far from undermining rigor, the use of computers in mathematics research will enhance it in several ways. Firstly, mathematicians who write complicated computer programs soon realize that subjecting lines of code to the usual techniques of mathematical analysis and proof often reveals faults in the programs. Programming can enhance and extend to a larger population an appreciation of why mathematicians regard proof as important. Secondly, the use of computers gives mathematicians another view of mathematical reality and another tool for investigating the correctness of a piece of mathematics through examining examples. Thirdly, and most fundamentally, it will strengthen the trend towards constructivism, helping to recast mathematics on more solid foundations.

References

[1] Errett Bishop and Douglas Bridges, *Constructive analysis*, Springer, New York, 1985. An outgrowth of Bishop's *Foundations of constructive analysis*, 1967.

[2] Max Dehn, *Papers on group theory and topology*, Springer, New York, 1987. Translated and introduced by J. Stillwell.

[3] W. Haken, *Theorie der Normalflächen. Ein Isotopiekriterium für den Kreisknoten*, Acta Math. **105** (1961), 245–375.

[4] Geoffrey Hemion, *The classification of knots and 3-dimensional spaces*, Oxford Univ. Press, Oxford and New York, 1992.

[5] Craig D. Hodgson and Jeffrey R. Weeks, *Symmetries, isometries, and length spectra of closed hyperbolic three-manifolds*, Experimental Math. **3** (1994), 261–276.

[6] Kenneth A. Brakke, *The Surface Evolver*, Experimental Math. **1** (1992), 141–165.

[7] D. Weaire and R. Phelan, *A counter-example to Kelvin's conjecture on minimal surfaces*, Philos. Mag. Lett. **69** (1994), 107–110.

[8] This work will be part of Thurston's Ph.D. thesis. E-mail addresses for the authors named are `njt@math.berkeley.edu` (Thurston), `gabai@math.caltech.edu`, and `Meyerhoff-MT@hermes.bc.edu`.

[9] B. Krauskopf, *The bifurcation set for the 1:4 resonance problem*, Experimental Mathematics **3** (1994), 107–128.