Do Mathematicians Still Do Math?

Yes, but increasingly they are turning to computers to help them with some of their trickier proofs

IN HIS TREATISE ON CELESTIAL MECHANICS, the 19th-century mathematician Pierre-Simon de Laplace found it convenient to shorten some of his proofs to the phrase, "It is easily seen that. . . ." Students of his work came to dread those words. What Laplace so easily saw often took others days if not weeks to reproduce. Even Laplace sometimes had trouble figuring out how he had seen things so easily.

That hasn't stopped mathematicians from mimicking Laplace. But a growing number of research papers offer a new wrinkle as well: mathematicians are increasingly turning to computers to fill in critical steps in their calculations and logical arguments. Some mathematical truths are easily seen, it seems, if you happen to be a VAX 8600.

To be sure, most mathematical proofs are still done the old-fashioned way, with pencil and paper. And for some mathematicians, proof by computer will always be anathema. But for others the computer is a versatile and powerful tool for attacking otherwise intractable problems. Computer proofs have penetrated into areas of number theory, geometry, dynamical systems, functional analysis, combinatorics, and even Laplace's domain of celestial mechanics.

The most famous computer-assisted proof is the 1976 proof of the Four Color Theorem by Kenneth Appel and Wolfgang Haken at the University of Illinois. Last year Clement Lam and colleagues at Concordia University in Montreal completed a computer proof in finite geometry showing that a certain type of projective plane does not exist (*Science*, 16 December 1988, p. 1507). Each proof boiled down to a theoretical analysis followed by a brute-force computer search through thousands of cases, subcases, and sub-subcases—just the sort of thing that machines excel at.

In 1981, Oscar Lanford III at the Institut des Hautes Etudes Scientifiques, Bures-Sur-Yvette, made a quite different use of computer power to prove a result known as the Feigenbaum conjecture. Loosely speaking, the Feigenbaum conjecture concerns the rate at which a process known as period doubling occurs in a large class of dynamical systems. Lanford called on the computer to carry out a set of complicated numerical calculations that proved the conjecture.

Numerical computation such as Lanford's might sound like a natural thing for a computer to do, but making computations precise is not. The problem stems from the fact that computers can represent only a finite, necessarily discrete, subset of real numbers, so that even simple arithmetic falls prey to round-off errors. Fortunately, the proof of the Feigenbaum conjecture rests, in part, on showing that a certain number is less than 1. The computer succeeds in proving this by calculating not the number in question but a slightly larger number, which is obtained by consistently rounding up at each step in the computation; since this larger number turns out to be less than 1, the proof is complete.

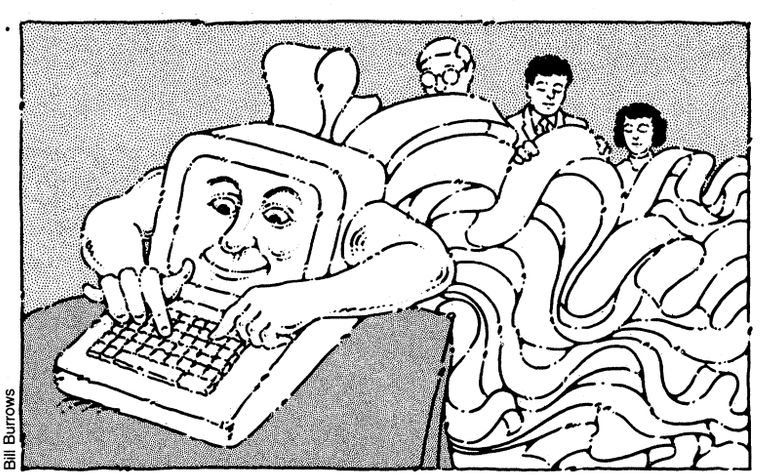
More generally, computers can supply rigorous numerical estimates through an approach called interval arithmetic. Instead of trying to approximate a real number with a single machine value, interval arithmetic works with *two* approximations: one slightly large and one slightly small. Arithmetic is then done on the two bounds. Any time a round-off occurs, the lower bound is rounded down and the upper bound is rounded up. In other words, interval arithmetic keeps track of an interval that is guaranteed to contain the number it claims to represent.

Jean-Pierre Eckmann, a mathematical physicist at the Université de Genève, says that further analysis has made it possible to do some of the Feigenbaum estimates by hand—especially if the hand is holding a pocket calculator—but others continue to require more calculation than a human would care to do. Eckmann and colleagues have also applied interval arithmetic to other problems in functional analysis that depend on extensive rigorous computations.

Case crunching and number (or interval) crunching are not the only ways that computers go about proving things. They also do a lot of symbol crunching. Powerful computer algebra systems such as Mathematica, MACSYMA, Reduce, and Scratchpad allow mathematicians to play with exact algebraic and analytic expressions, which the computer, like a dull but industrious student, treats as meaningless strings of symbols subject to mysterious rules and operations. Much as pocket calculators alleviate the pain of filling out tax forms, algebraic processors take the tedium and worry out of long, complicated derivations.

Kenneth Meyer and Dieter Schmidt at the University of Cincinnati used computer algebra to get new results on the n-body problem in celestial mechanics. The n-body problem is perhaps the simplest mathematical problem that defies exact solution. It concerns the motion of objects, such as stars or planets, that attract each other gravitationally according to an inverse square law. When $n = 2$, such a system does have an exact solution: the elliptical orbits of Kepler. But when more than two objects are flying around, the problem becomes intractable: there are no analytic expressions that describe future positions based on arbitrary initial conditions.

However, certain initial conditions do permit exact solutions. In particular, if $n - 1$ identical "planets" are placed symmetrically around a central "sun" and spun at just the right rate, the circular motion is eternal—what



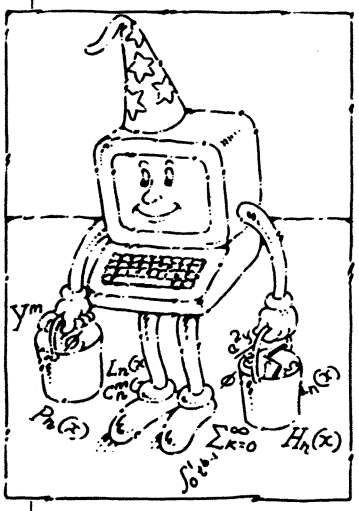
Bill Burrows

The Sorcerer's Apprentice

While some mathematicians are working on computer-assisted proofs, Siemion Fajtlowicz has taken the opposite approach: computer-assisted conjectures. Fajtlowicz, a mathematician at the University of Houston, has developed a computer program, called Graffiti, that generates conjectures by the truckload. The program's output is so prodigious that Fajtlowicz has had to write other programs that sift through Graffiti's offerings and decide which ones are interesting enough to pass along.

Graffiti works in a branch of mathematics called graph theory. A graph in this context is simply a finite collection of points, called vertices, and edges, which connect various pairs of vertices. Represented geometrically, a graph may look like a stick model of an organic molecule or the complicated design of a printed circuit. In fact, chemistry and computer science are two areas where graph theory has found applications.

The basic idea of Graffiti, Fajtlowicz explains, is that it "knows" certain graphs and can evaluate various formulas based on certain fundamentals of graph theory called invariants. Anytime Graffiti runs across a formula that is true of all the graphs in its library, it considers the formula a conjecture. If a conjecture later turns out to be false, Fajtlowicz can tell Graffiti about it by adding a counterexample to the program's repertoire of graphs. Usually a single graph acts as a counterexample to several false conjectures, so Graffiti's library is still fairly small; it now contains about 200 graphs.



Bill Burrows

Invariants of a graph are numbers that don't change if the graph is redrawn or relabeled. They range from geometric data, such as the graph's diameter (determined by finding the shortest path between each pair of vertices), to algebraic information, such as the number of positive eigenvalues of the graph's adjacency matrix. While some are purely mathematical concoctions, other invariants seem to have real physical meaning. One called the Randic index, for instance, is used to predict boiling points of hydrocarbons.

Graffiti's conjectures all say that one invariant is less than or equal to another invariant (or combination of invariants). For instance, Graffiti conjectured (correctly) that the Randic index is never more than the number of vertices; Graffiti also claims that the average distance between vertices is never more than the Randic index, a conjecture that remains to be settled. So far Graffiti boasts a respectable batting average: of the conjectures that have been settled, roughly a third have been proved correct.

Fajtlowicz first ran Graffiti in 1985. Using a library of about 40 graphs, the program generated more than 7000 conjectures. The sheer volume forced Fajtlowicz to introduce programs that would evaluate Graffiti's output. Picking out interesting conjectures, Fajtlowicz says, is itself an important problem. "The main problem is not how to write a program which will decide whether a given conjecture is interesting, but on what basis to make this decision."

One approach is to trace the "genealogy" of invariants—that is, how their definitions are related. From this point of view a conjecture about sibling invariants is less interesting than a conjecture about distant relatives. For instance, an inequality comparing geometric data to algebraic information is more interesting than a comparison of, say, the radius to the diameter of a graph.

Fajtlowicz acknowledges that interesting conjectures might get lost in the shuffle, but says that the alternative is a "flood of dull conjectures." A somewhat futuristic possibility is to hook Graffiti up with an automatic theorem prover—a program that looks for short or obvious proofs—and reject as dull anything the machine can prove for itself. Given that the current list of "interesting" conjectures numbers close to 1000 statements, the future of Graffiti may lie with machines rather than humans. Mathematicians sometimes joke about journal papers that are written for and understood by only a handful of specialists. Could computers wind up doing the same—and will they get the joke? ■ **B.A.C.**

mathematicians call a relative equilibrium. Using a combination of symbol crunching and number crunching, Meyer and Schmidt found a host of new relative equilibria, which arise as deformations of the symmetric, central configuration.

Schmidt wrote a special-purpose algebraic processor, POLYPAK, to carry out the symbolic manipulations on the n-body equations needed to bring them into a form amenable to further analysis—just the sort of computation Laplace might have called "easily seen." Theoretical considerations then proved that the central configuration split, or bifurcated, into new, less symmetric equilibria as the mass of the "sun" increases. The new configurations could then be drawn using ordinary numerical methods.

Can computer proofs be trusted? There is always the risk that a programming bug might cause the system to overlook cases or carry out the wrong calculation, but these traps occur with people-generated proofs too. Proponents therefore believe that computer proofs are reliable, provided that the programs are thoroughly checked and that the proofs are run on reliable software. Says Charles Fefferman of Princeton University: "There are differences of orders of magnitude—a computer proof is much more complicated than a standard proof, so it has to be checked more carefully. But it's the same kind of issue."

The doubts are partly philosophical, partly a matter of taste. Computer proofs call into question the nature of human participation in mathematics, reminiscent of the sound-of-a-falling-tree conundrum: Has a theorem been proved if no one has read the proof? There is also a question of aesthetics: Do computer-assisted proofs meet the criteria of simplicity and elegance that mathematicians hold dear? There is a long-standing tradition in mathematics that short proofs are best. Can a few lines of code that call for 50 million calculations be considered a short proof?

Science fiction scenarios aside (one can imagine a computer saying it has a marvelous proof of Fermat's Last Theorem, which the floppy disk is too small to contain), computers are unlikely to replace human mathematicians anytime soon. Nevertheless, Eckmann has introduced a phrase that may become commonplace. Speaking on the Feigenbaum conjecture at a recent conference at the University of Cincinnati on computer-assisted proofs in analysis, Eckmann pointed to a technical definition and joked, "This is the brain-assisted part of the proof."

■ **BARRY A. CIPRA**

Barry A. Cipra is a mathematician and writer based in Northfield, Minnesota.