Say It Again in Plain Algebra

The growing use of computer algebra systems has driven mathematicians to find ways of simplifying the horrendously complex expressions number crunchers sometimes spit out

LAWYERS AND GOVERNMENT AGENCIES are well known for writing in a variant of English known as gobbledygook: long, tortuous, impersonal sentences that mask a simple—and sometimes vacuous—statement of fact or opinion. It is an easy habit to fall into and a surpassingly difficult one to break.

Oddly enough, mathematicians run a similar risk when they solve problems in algebra. It is easy to wind up writing down a convoluted algebraic expression for what, in fact, is a very simple number. In particular, numbers such as $\sqrt{5 + 2\sqrt{6}}$ that are written with roots within roots—what mathematicians call nested radicals—frequently turn out to represent much simpler expressions. For instance, $\sqrt{5 + 2\sqrt{6}}$ is just a fancy way of saying $\sqrt{2} + \sqrt{3}$. Similarly, $\sqrt[3]{\sqrt{5} + 2} - \sqrt[3]{\sqrt{5} - 2}$ is a grotesquely complicated way of writing the number 1. The problem is, it's nearly impossible to tell the difference between a nested radical that's hiding something simple and one that is honestly complicated.

Mathematicians have long sought some way of reducing algebraic expressions to their simplest, least nested form. The growing use of computer algebra systems has made the search more pressing than ever. It might seem that a computer algebra system, which deals with mathematical formulas in an exact, formal manner, would simplify algebraic expression as a matter of course. Not so. A typical system "knows," for example, that $\sqrt{2}^2 = 2$ and $\sqrt{2} \times \sqrt{3} = \sqrt{6}$, and it can simplify the expression $(x + 1)^2 - x^2$ down to $2x + 1$ (which puts it ahead of your average high school graduate). But it can only do these things because there are algorithms for doing them—an explicit set of instructions that guide the computer from input to output. In the case of nested radicals, no such algorithm was known.

Until now.

Susan Landau, a computer scientist at Wesleyan University (now at the University of Massachusetts at Amherst), has found a way to cut through the mathematical gobbledygook of algebraic expressions. Based on a key observation which she at first considered "much too nice to be true," Landau proved a theorem about algebraic number systems, which led directly to an algorithm for denesting radicals.

Earlier researchers had found algorithms that work in special cases. In 1984, Allan Borodin at the University of Toronto, Ron Fagin at the IBM Almaden Research Center, John Hopcroft at Cornell University, and Martin Tompa at the University of Washington found an efficient way of denesting certain expressions involving square roots, and Richard Zippel at Cornell University gave other conditions that make denesting possible. However, Landau's algorithm is the only one that works in general.

Landau's theorem says that a radical expression can be denested if and only if the denesting occurs within a structure known as the splitting field of the original expression. Consequently, the algorithm needs only to search for denestings within the splitting field. "It's the natural place for an algebraist to go," Landau explains. Even so, Landau looked at lots of examples before attempting to prove the theorem. "I didn't think I would get this theorem; I thought I would get something much less good."

"I didn't think I would get this theorem; I thought I would get something much less good."

—Susan Landau

The splitting field is a familiar concept in abstract algebra. It is built in two steps.

Algebra: A Hotbed of Radicalism

How is it possible to create a complicated algebraic expression without intentionally setting out to do so? After all, if nested radicals that disguise simple numbers arise only in the fiendish imagination of mathematicians, then computer algebra systems would have little to be concerned over. But, in fact, such radicals are easy to come by, and computer algebra systems, if left to themselves, could be the biggest producers of unnecessarily complicated numbers.

One way that nested radicals arise is through routine application of general formulas to find roots of polynomials. For example, the number $\sqrt{5 + 2\sqrt{6}}$ can arise by solving the polynomial equation $x^4 - 10x^2 + 1 = 0$: applying the familiar quadratic formula gives $x^2 = 5 + 2\sqrt{6}$, and the nested radical comes by taking the square root.

Computer algebra systems not only know the quadratic formula, they also know a formula for solving cubic equations. It turns out that one root of any cubic polynomial having the form $x^3 + px - q$ can be written as $\sqrt[3]{\sqrt{(p/3)^3 + (q/2)^2} + (q/2)} - \sqrt[3]{\sqrt{(p/3)^3 + (q/2)^2} - (q/2)}$. Applying this mindlessly to the polynomial $x^3 + 3x - 4$ produces the root $\sqrt[3]{\sqrt{5} + 2} - \sqrt[3]{\sqrt{5} - 2}$. But 1 is an obvious root of this polynomial, so the two must be equal (since this particular polynomial has only one real root). (In the main story, an extra square root was added for dramatic effect.)

Not many people know it, but there is also a formula for solving fourth-degree polynomials. However, at that point the well runs dry: there is no general formula for solving fifth- or higher degree polynomials. Some polynomials—$3x^5 - 15x + 5$ is an example—cannot be solved at all, meaning that it is impossible to express any of their roots using radicals, no matter how deeply nested you allow them to be. (It is, of course, always possible to find numerical approximations to the roots of a polynomial, but that's a different matter.)

Galois theory and much of group theory were developed in the early 19th century to understand why some polynomials can't be solved. Both theories by now have outgrown their origin—group theory is a staple not only in mathematics, but in theoretical physics and chemistry as well—but both have their "roots" in an age-old fascination with algebraic equations. ■ **B.A.C.**

First, you find the simplest polynomial (with integer coefficients) that has your radical expression as a root. (A root of a polynomial is a number that makes the polynomial equal to 0.) For example, the simplest polynomial for $\sqrt[3]{2}$ is $x^3 - 2$. Second, you extend the set of rational numbers by tossing in all the roots of this polynomial, not just the original root, and then taking all possible sums and products, so that the extension is closed under addition and multiplication; this extension is called the splitting field because the polynomial "splits" into linear factors. For example, the splitting field of $\sqrt[3]{2}$ is formed not just by tossing $\sqrt[3]{2}$ in with the rational numbers, but by including $\sqrt{-3}$ as well, since $\sqrt[3]{2}(1 \pm \sqrt{-3})/2$ are the other two roots of the polynomial $x^3 - 2$.

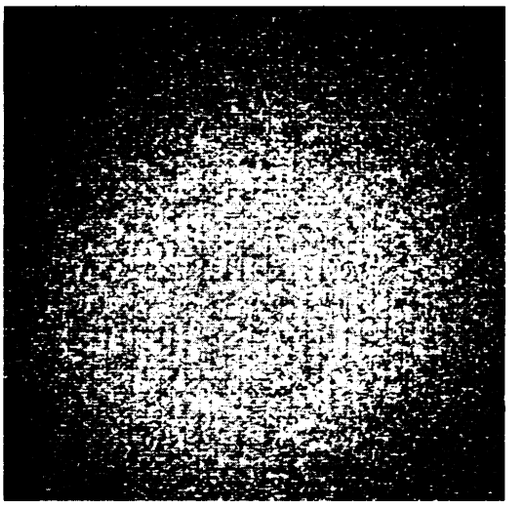
Landau's algorithm requires the construction of a second algebraic structure, called the Galois group, associated to the splitting field. This is still familiar territory to algebraists—Galois theory was created for the purpose of understanding how and when the roots of a polynomial can be written using radicals. Unlike the field, which has infinitely many members, the Galois group is finite, and this makes the computer very happy.

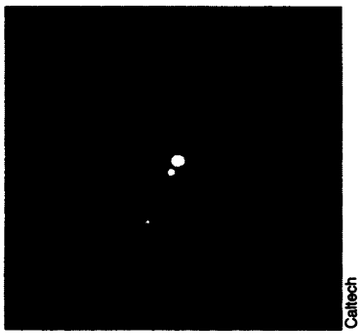
The algorithm searches the Galois group for a sequence of nested subgroups satisfying certain technical conditions. (The subgroups sit inside each other like tightly fitting Russian dolls.) Once the shortest such sequence is found, the algorithm translates the subgroup nesting into a nested radical expression for the original number. The theory behind the algorithm guarantees that this translation produces the least nested version of the number.

Although it solves the problem, Landau's algorithm is not necessarily the last word in denesting radicals. One drawback is that the algorithm requires a potentially huge amount of computation—the splitting field and its associated Galois group can be extremely large. Inserting an extra radical sign in the initial expression can more than double the algorithm's work load.

It may be that inefficiency is the price to be paid for an all-purpose denesting algorithm. However, at the same time, Landau notes that "what is theoretically slow may be practically fast, and vice versa." In any event, it's nice to know that something can be done with those awkward algebraic expressions. Now if only the linguists could come up with some way of untangling bureaucratic officialese. ■ **BARRY A. CIPRA**

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





Caltech

Cleaning up the atmosphere's act. *The image on the left, a very fast exposure of the bright star Sigma Herculi taken through the 5-meter Hale telescope, shows how the atmosphere breaks up a single shaft of starlight into a myriad of speckles. The image on the right shows the Caltech computer's reconstruction of the Sigma Herculi.*

Computer-Age Stargazing

When astronomers use the 5-meter Hale telescope at the Palomar Observatory to look at a star—in this case, the binary system Sigma Herculis—what they actually see is the image on the left: a shimmering, boiling blur caused by the incessant motion of the atmosphere. What they would like to see is the image on the right: a pair of crisp, well-defined spots blurred only by the unavoidable diffraction of light being focused by the huge mirror.

Now they can. Palomar director Gerry Neugebauer and seven other California Institute of Technology astronomers have developed and implemented two techniques that virtually eliminate the distorting effects of the atmosphere, thus allowing this telescope or any other telescope to approach its theoretical maximum resolution. The improvement in this case is a factor of 20, from roughly 1 arc second to about 50 milliarc seconds.

Both techniques rely on the fact that the granulated mess on the left contains precisely as much information about the source as the original starlight did—just scrambled. The trick is to use massive computer processing to unscramble it.

In Non-Redundant Masking, the method used to make the Sigma Herculis image, the Palomar team places an opaque screen pierced with five to seven tiny holes at the prime focus of the telescope. They then take the separate beams of starlight coming through the screen and recombine them into an undistorted image using mathematical algorithms developed for radio observations at the Very Large Array near Socorro, New Mexico.

In the Fully Filled Aperture technique, the group simply turns the computer loose on the whole blur, with nothing interrupting the light. The computer then recovers the image using a refinement of a long-established method known as speckle interferometry.

Neither method is easy. In both, the computational demands are horrendous, requiring the full parallel processing power of Caltech's 512-node "hypercube" supercomputer. Moreover, the techniques are cumbersome for the observers, demanding lots of 5- to 30-millisecond exposures to keep the motion of the atmosphere from smearing the image irretrievably. And they are limited to reconstructing relatively bright objects—in the case of nonredundant masking, no fainter than what can be seen with the naked eye.

Nonetheless, these techniques do show just how far ground-based astronomers have come: leaving aside such issues as faint-object sensitivity and wavelength coverage, where orbital observatories still have a decided advantage, the angular resolution demonstrated here is about as good as that expected from the Hubble Space Telescope. ■ **M. MITCHELL WALDROP**