From Real Numbers to Strings of Zeros

Talks at the joint mathematics meetings of the American Mathematical Society and the Mathematics Association of America, held 11 to 14 January in Phoenix, Arizona, ranged from information-based complexity in numerical analysis to primitive recursive functions in arithmetic. Here is a sample of topics covered.

Computing over the Reals

Real computers do not really deal with real numbers. There is something intrinsically infinite about the real numbers—their unending decimal expansions, for instance—that goes against the grain of finite-state automata. However, mathematicians are finding it useful to imagine abstract machines that store and operate on real numbers—or, more generally, the elements of any number system, called a "ring," that permits addition, subtraction, multiplication, and, sometimes, division. The theory of such "machines" provides a convenient framework for studying problems in numerical analysis, optimization, and other areas of computational mathematics.

Computability and computational complexity are twin questions for any computing machine, actual or imaginary. Roughly speaking, computability concerns whether a given problem can be solved at all; complexity worries about how much work is necessary. Lenore Blum, at the International Computer Science Institute in Berkeley, California, Michael Shub, at the IBM Watson Research Center in Yorktown Heights, New York, and Stephen Smale, at the University of California at Berkeley, have developed a theory of computation for arbitrary ordered rings, with particular attention to the real numbers. (An ordered ring is one that includes a comparison of elements, such as $2 < 3$.)

Computability and complexity theory are usually studied over the ring of integers, reflecting the discrete nature of digital computers. In particular, the bit size (that is, the number of digits) of a number is a standard measure of complexity: as any fourth grader will agree, multiplying four-digit numbers is more work than multiplying two-digit numbers. In complexity over the reals, however, the "cost" of multiplication—or any algebraic operation—is taken to be independent of the size of the numbers involved. This is a natural viewpoint for scientific computation, where the complexity of a problem is better measured by the number of variables involved.

Some problems that are undecidable over the integers turn out to be computable over the reals. The "4-Feasibility problem" is a case in point. The problem is to determine whether or not a polynomial of degree four in any number of variables has a zero. When the problem is posed over the ring of integers—that is, when integer zeros are requested—the problem is undecidable: there is no algorithm that guarantees a yes-or-no answer for the existence of zeros.

Over the reals, however, the 4-Feasibility problem is known to be decidable. It is also in the curious class of "NP problems" over the reals. An NP problem is one whose answer is easy to verify, provided you are lucky enough to guess a solution. "The notion of NP is very peculiar," Blum says. "God gives you an answer, and you check it out."

Recent work by D. Yu. Grigorév in the Soviet Union, which has been followed up by John Canny at the University of California at Berkeley and by James Renegar, of Cornell University, has shown that the 4-Feasibility problem over the reals can be solved in "exponential time": each additional variable increases by a constant factor the amount of computation required to determine if there is a zero. It would be nice to find a "polynomial-time" algorithm: in that case, the amount of computation increases like a power of the number of variables, rather than exponentially (compare, say, n^2 with $2n$ when $n = 100$). However, Blum *et al.* have made this doubtful; they have shown that the 4-Feasibility problem is "NP complete" over the reals. This means, essentially, that any other NP problem over the reals can be rephrased to sound like the 4-Feasibility problem, so that a fast algorithm for solving the 4-Feasibility problem would translate into a fast algorithm for solving any other NP problem over the reals.

There are thousands of NP-complete problems over the integers, including important problems in scheduling and sorting. It would be a stunning advance either way if any of these were shown to have a polynomial-time algorithm, or if any NP problem were shown *not* to have such an algorithm. Blum and her co-workers hope that studying a more general theory of complexity will shed light on the nature of NP problems.

Blum, Shub, and Smale have also shown that some of the familiar objects in dynamical systems involve an element of undecidability over the reals. A Julia set is a geometric region in the complex plane, defined by iterating a function such as a polynomial (for example, $f(z) = z^2 + 1$). For polynomials, values of z that eventually become large are said to be outside the Julia set; the Julia set consists of z's that never get large. There is a clear-cut criterion for what "large" means, so that any point not in the Julia set will eventually identify its status.

Points in the Julia set, however, only indicate their status by *never* getting large. One might hope that there would be some other way of characterizing these points—say by showing that a point belongs to the Julia set if after 100 iterations it still has not gotten large. In some cases this is possible. For instance, the Julia set for the function $f(z) = z^2$ is the unit circle in the complex plane. But Blum and her colleagues have shown that, in general, there is no clear-cut characterization.

Information-Based Complexity

As the lyric says, you can't always get what you want—but if you try sometimes, you just might find you get what you need.

How do you solve a computational problem when the information available is partial, approximate, or hard to come by? Researchers in information-based complexity (IBC) look for general results on the difficulty of solving problems with these features. Such problems are common in numerical analysis, physics, economics, robotic and human vision, signal processing, decision theory, and control theory, says Joseph Traub, a computer scientist at Columbia University. Traub and Edward Packel, of Lake Forest College in Illinois, outlined the current status of IBC at a session on foundations of complexity theory for numerical analysis at the math meetings in Phoenix.

There are two basic types of computational problems, Traub notes: discrete combinatorial problems where the information available is complete, exact, and free; and continuous algebraic or analytic problems where the information is partial, contaminated, and priced. An example of the former is the well-known Traveling Salesman Problem: given a map showing exact distances between a set of cities, find the shortest circuit

that passes through every city.

A typical IBC-type problem is that of numerically integrating a continuous function (that is, computing $\int_0^1 f(x)dx$) when it is possible to know only finitely many values of the function—often only approximately. What is the best information to ask for, and what is the worst error that can occur with the best algorithm? A general theory of information-based complexity gives guidelines for what can and cannot be done, and suggests algorithmic approaches for specific problems.

Many IBC problems, including the integration example, are "linear" in nature: if the problem is split into pieces, the whole problem can be solved by adding up the solutions of the pieces [in symbols, $S(P_1 + P_2) = S(P_1) + S(P_1)$]. If a linear problem can be solved at all, it can be solved by a linear algorithm, Packel notes. [An algorithm is linear if its answer to a sum is the sum of its answers—that is, if $A(P_1 + P_2) = A(P_1) + A(P_2)$. Keep in mind that $A(P)$ is, in general, only an approximation to the true solution $S(P)$.] But does that mean that linear algorithms are best? Not necessarily, Packel says. Although linear algorithms are optimal in a wide class of linear problems, researchers have created examples where nonlinear approaches are better.

Until recently, these examples have been contrivances unlikely to occur in practice. However, in 1986, Arthur Werschulz of Fordham University and Henryk Woźniakowski of the University of Warsaw and Columbia University found a class of counterexamples with real-world overtones. Their result includes a "naturally occurring" problem of inverting a Laplace transform—a familiar task in many engineering applications. (The exact mathematical problem, though, is still purely theoretical.) Werschulz and Woźniakowski found that all linear algorithms for this problem necessarily have infinite error, but that the problem can be solved by nonlinear algorithms.

On the other hand, researchers, including Packel, have found general conditions under which linear algorithms are optimal. "Our faith in linear algorithms for linear problems is alternately strengthened and shaken, leaving us in a state of tantalizing mathematical ambiguity," Packel says.

Progress in Progressions

Few things are simpler than an arithmetic progression: a string of integers, such as 7, 10, 13, 16, in which each number exceeds its predecessor by a constant amount. But simple ideas often lead into surprisingly difficult problems. Ron Graham, of Bell Laboratories, described some recent progress in arithmetic progressions.

Suppose the natural numbers are separated into two sets—for convenience, imagine coloring each number, say, either red or blue. An old theorem of B. L. van der Waerden says that at least one of the sets must contain arithmetic progressions of any length you care to ask for. In fact, van der Waerden's theorem, proved in 1927, holds no matter how many "colors" you use.

A simple question is this: How far do you have to look in order to be sure of finding, say, three terms of the same color that are in arithmetic progression? To put it another way, how many natural numbers can you paint without creating an arithmetic progression of a given length?

It is easy to check that coloring 1, 3, 6, and 8 red, and 2, 4, 5, and 7 blue avoids any three-term progression. But if you include 9, it is impossible to avoid creating a three-term progression. How about four-term progressions? It turns out that the numbers 1–35 cannot be painted red and blue without creating a four-term progression. For five-term progressions, unavoidability begins at 178. For six-term progressions, no one knows.

Van der Waerden's proof provides an upper bound on how far you can paint before creating an arithmetic progression of a given length, but the upper bound is ridiculously large. The bound grows so rapidly as a function of the length of the progression that it falls outside of the normal class of functions that logicians call "primitive recursive." In some axiomatic models of arithmetic, in fact, the only functions are the primitive recursive ones (which include polynomials and exponentials). Van der Waerden's proof uses a technique called double induction, which is responsible for the explosive growth of the bound. Some mathematicians suggested that the actual growth might be non-primitive recursive, which would point to the double induction as an unavoidable step in proving that arithmetic progressions exist.

Saharon Shelah of the Hebrew University in Jerusalem has shown that they were wrong. In an article in the July 1988 issue of the *Journal of the American Mathematical Society*, Shelah shows that van der Waerden's upper bound can indeed be replaced by a primitive recursive function. Shelah's result applies, in fact, to a more general problem, and includes the multi-coloring case of van der Waerden's theorem. The upper bound is still enormous (it is not explicitly written out in Shelah's article), but, Graham says, it is a "fantastic improvement" over what was known.

Zeta Zero Update

The 10^{20}th zero of the Riemann zeta function is $\frac{1}{2} + 1520244011592074726 8.6290299\ldots \sqrt{-1}$. Andrew Odlyzko of Bell Laboratories in Murray Hill, New Jersey, has recently completed a statistical analysis of more than 78 million consecutive zeros on either side of the 10^{20}th. These zeros lie nearly a hundred million times higher than those of previous computations (see *Science* 11 March 1988, p. 1241).

The Riemann zeta function, and especially its zeros (points in the complex plane where the function vanishes), contain a wealth of number-theoretic information. Aside from a well-understood set of "trivial" zeros, all the zeros of the zeta function are known to lie in a thin vertical strip in the complex plane. The famous Riemann Hypothesis asserts that they all lie on a line running up the middle of the strip. The zeros in Odlyzko's study all satisfy the Riemann Hypothesis.

More importantly, Odlyzko's zeros provide evidence favoring another conjecture on the spacing between consecutive zeros. It has been proposed that the distribution of spacings between zeros of the zeta function is similar to that of eigenvalues of random matrices that are studied in many-particle systems in physics. This hypothesis, which implies the Riemann Hypothesis, suggests that the zeta function could be used as a model of quantum chaos.

Brian Conrey, at Oklahoma State University, has taken a different tack on statistics of the zeta function. Writing in the January 1989 issue of the *Bulletin of the American Mathematical Society*, Conrey says that, whatever else happens, at least two-fifths of the zeros of the zeta function lie on the line where they are supposed to be. This improves the previous lower bound of one-third, which was proved in 1973 by Norman Levinson at the Massachusetts Institute of Technology.

To understand what the lower bound means, imagine cutting off the vertical strip at a finite height, and counting the zeros inside the resulting rectangle. There will be only finitely many zeros, so the fraction that lie on the line can be computed. Now increase the height of the rectangle and recompute the fraction. Conrey's result says that, as the rectangle is taken taller and taller, the fraction will eventually exceed two-fifths. ■ **BARRY A. CIPRA**

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