CHALLENGES IN MATHEMATICAL COMPUTING

Almost all interesting mathematical algorithmic questions relate to NP-hard questions. Such computation is prone to explode exponentially. The authors anticipate the greatest benefit will come from mathematical platforms that allow for computer-assisted insight generation, not from solutions of grand-challenge problems.

Some say that pure mathematicians invented digital computers and then proceeded to ignore them for the better part of half a century. In the past two decades, this situation has changed with a vengeance.

Major *symbolic mathematics* and *computer algebra* packages (see the sidebar), most notably Maple and Mathematica, have reached a remarkable degree of sophistication over the last 15 years. (We should also allude to counterparts such as Axiom, Macsyma, Reduce, MuPad; Matlab; and other more specialized packages such as GAP, Magma, or Cayley [for group theoretic computation], Pari [for number theory], KnotPlot [for knot theory], SnapPea [for hyperbolic 3-manifolds], and SPlus [for statistics].) This sophistication has relied on a confluence of algorithmic breakthroughs, dramatically increased processor power, almost limitless storage capacity, and, most recently, network communication, excellent online databases, and Web-distributed (often Java-based) computational tools. Examples include the mathematics front end to the Los Alamos Preprint ArXiv (http://front.math.ucdavis.edu), mathematical reviews on the Web (http://e-math.ams.org/mathscinet), Neil Sloane's encyclopedia of integer sequences (www.research.att.com/personal/njas/sequences/eisonline.html), our own inverse symbolic calculator (www.cecm.sfu.ca/projects/ISC/ISCmain.html), and integer relation finders (www.cecm.sfu.ca/projects/IntegerRelations).

The relatively seamless integration of all these components arguably represents the key challenge for 21st-century computational mathematics. It's hard to think of mathematical problems where a dramatic increase in computational speed and scale would enable a presently intractable line of research. It's easy to give examples where it would not—consider Clement W.H. Lam's 1991 proof (www.cecm.sfu.ca/organics/papers/lam/index.html) of the nonexistence of a finite projective plane of order 10 (a hunt for a configuration of $n^2 + n^1 + 1$ points and lines). It involved thousands of hours of computation. Lam's estimate is that the next case ($n = 18$) susceptible to his methods would take millions of years on any conceivable architecture. Although a certain class of mathematical enquiries is susceptible to massively parallel Web-based computation (for example, discovering Mersenne primes of the form $2^n - 1$), these tend not to be problems central to mathematics.

1521-9615/01/$10.00 © 2001 IEEE

JONATHAN M. BORWEIN AND PETER B. BORWEIN
Simon Fraser University

Computational excursions in contemporary mathematics

Many researchers have made significant inroads into some rather difficult—previously viewed as intractable—problems such as exact integration of elementary functions. Some of the most important mathematical algorithms of the 20th century include

- the fast Fourier transform,
- lattice basis reduction methods and related integer relation algorithms,
- the Risch algorithm for indefinite integration,
- the Gröbner basis computation for solving algebraic equations, and
- the Wilf/Zeilberger algorithms for hypergeometric summation and integration, which rigorously prove very large classes of identities.

All these are—or soon will be—centrally incorporated into symbolic mathematics or computer algebra packages. In fact, the first two were counted among the 10 algorithms with "the greatest influence on the development and practice of science and engineering in the 20th century."[1] Of course, many of the others, such as the sorting algorithms, are fundamental to the needs of contemporary mathematics.

Such packages can now substantially deal with large parts of the standard mathematics curriculum—and can out-perform most of our undergraduates to boot. They provide extraordinary opportunities for research that most mathematicians are only beginning to appreciate and digest. They also provide access to sophisticated mathematics to a very broad cross-section of scientists and engineers.

The emergence of such packages—and their integration into mathematical parole—represents the most significant part of a paradigm shift in how mathematics is done. Certainly these packages have already become a central research tool in many subareas of mathematics, both from an exploratory and a formal point of view—it is acceptable now to see a line in a proof that begins "by a large calculation in Maple, we see …" The first objective of symbolic algebra packages was to do as much exact mathematics as possible. A second, increasingly important objective is to do it very fast and to deal in an arbitrary-precision environment with the more standard algorithms of mathematical analysis. Roughly, users would like to be able to incorporate the usual methods of numerical analysis into an exact environment or at least into an arbitrary-precision environment.

The problems are obvious and hard. For example, how do we do arbitrary precision numerical quadrature? When do we switch methods with precision required or with different analytic properties of the integrand? How do we deal with branch cuts of analytic functions? How do we deal consistently with log? More ambitiously, how do we do a similar analysis for differential equations? Ultimately, can we certify that a given numeric or symbolic computation is indeed a proof or even just correct? The goal is to marry the algorithms of analysis with symbolic and exact computation and to do this with as little loss of speed as possible. Sometimes this means we must first go back and speed up the core algebraic calculations.

Within this context, a number of very interesting problems concerning the visualization of mathematics arise. How do we actually "see" what we are doing? Some say that Cartesian graphing was the most important invention of the last millennium. Certainly it changed how we think about mathematics—the subsequent development of differential calculus rested on it. More subtle and complicated graphics, like those of fractals, enable a previously impossible kind of exploration. There are many issues to work out at the interface of mathematics, pedagogy, and even psychology that are important to get right. An instructive example is the growing reliance of numerical analysts on graphic representation of large sparse matrices—the pictures show structure while numerical output is little help. (An example is JavaView [www-sfb288.math.tu-berlin.de/vgp/javaview/index.html] for 3D geometry on the Web.)

Some Significant Mathematical Packages

Axiom: www.axiomtek.com
Derive: www.derive.com
KnotPlot: www.pims.math.ca/knotplot
Macsyma: http://wombat.doc.ic.ac.uk/foldoc/foldoc.cgi? MACSYMA
Maple: www.maplesoft.com
Mathematica: www.wolfram.com
Matlab: www.mathworks.com
MuPad: www.mupad.com
Pari: www.cs.sunysb.edu/~algorith/implement/pari/implement.shtml
Reduce: www.uni-koeln.de/REDUCE
SnapPea: www.ptf.com/ptf/products/UNIX/current/0465.0.html
SPlus: www.insightful.com

The twin successes of the symbolic algebra packages have been their mathematical generality and ease of use. These packages deal most successfully with algebraic problems whereas many (perhaps most) serious applications require analytic objects such as definite integrals, series, and differential equations. All the elementary notions of analysis, such as continuity and differentiability, need precise computational meaning. The first challenge to meeting this need involves mathematical algorithmic developments to allow the handling of a variety of these only partially handled problems—including the analysis of functions given by programs. Many of these relate to the difficult mathematical problems involved in automatic simplification of complicated analytic formulae and recognition of when two very different such expressions represent the same object. There is also an intrinsic need to mix numeric and symbolic (exact and inexact) methods. Human mathematicians often criticize programs for making dumb errors, but often these errors (such as oversimplifying expressions, leaving out hypotheses, or dividing by zero) are precisely how we start when we do it ourselves. As Jacques Hadamard noted almost a century ago, "The object of mathematical rigor is to sanction and legitimize the conquests of intuition."

The Reimann hypothesis

The question that a pure mathematician is most likely to sell his soul to solve is the so-called Riemann hypothesis, first described in 1859. The bounty on its solution now exceeds $1 million. At the Clay Mathematics Institute's Web site (www.claymath.org/prize_problems/rules.htm), the problem is described in the following form:

> Some numbers have the special property that they cannot be expressed as the product of two smaller numbers, e.g. 2, 3, 5, 7, etc. Such numbers are called prime numbers, and they play an important role, both in pure mathematics and its applications. The distribution of such prime numbers among all natural numbers does not follow any regular pattern, however the German mathematician G.F.B. Riemann (1826-1866) observed that the frequency of prime numbers is closely related to the behavior of an elaborate function $\zeta(s)$ called the Riemann Zeta function. The Riemann hypothesis asserts that all interesting solutions of the equation $\zeta(s) = 0$ lie on a straight line. This has been checked for the first 1,500,000,000 solutions. A proof that it is true for every interesting solution would shed light on many of the mysteries surrounding the distribution of prime numbers.

A little more precisely, the Riemann hypothesis is usually formulated as

> All the zeros in the right half of the complex plane of the analytic continuation of
>
> $$\zeta(s) := \sum_{n=0}^{\infty} \frac{1}{n^s}$$
>
> lie on the vertical line $\Re(s) = 1/2$.

(One of the most famous results in elementary mathematics is Euler's evaluation of $\zeta(2) = \pi^2/6$.)

Without doubt this is one of the "grand challenge" problems of mathematics and for good reason. Large tracts of mathematics fall into place if the Riemann hypothesis is true: while the proof methods may be tremendously significant, the truth of the Riemann hypothesis is central—its falseness would be disquieting. Most mathematicians believe the Riemann hypothesis to be true, although there are notable dissenters. John Littlewood, one of the last century's great analytic number theorists, has hypothesized its falseness.[2] Of course, finding just one nontrivial zero off the line $\Re(s) = 1/2$, should it exist, is worth $1 million, and this might provide additional motivation to extend this particular mountain's climb. (Perhaps the prize is only for a proof, not a disproof—certainly a proof is more interesting.) The fact that more than the first billion zeros are known, by computation, to satisfy the Riemann hypothesis might be considered "strong numerical evidence." However, it is far from overwhelming—there are subtle phenomena in this branch of mathematics that only manifest themselves far outside present computer range.

One reason to extend such computations—which are neither easy nor obvious and rely on some fairly subtle mathematics—is the hope that someone will uncover delicate phenomena that give insight for a proof. Greatly more ambitious is the possibility that, in the very long run, it will be possible to machine-generate a proof—even for problems as difficult as this one.

P vs. *NP*

Of the seven $1 million "Millennium Prize" problems on the Clay Web site, the one that is most germane to this discussion is the so-called $P \neq NP$ problem. Again, from the site:

It is Saturday evening and you arrive at a big party. Feeling shy, you wonder whether you already know anyone in the room. Your host proposes that you must certainly know Rose, the lady in the corner next to the dessert tray. In a fraction of a second you are able to cast a glance and verify that your host is correct. However, in the absence of such a suggestion, you are obliged to make a tour of the whole room, checking out each person one by one, to see if there is anyone you recognize. This is an example of the general phenomenon that generating a solution to a problem often takes far longer than verifying that a given solution is correct. Similarly, if someone tells you that the number 13,717,421 can be written as the product of two smaller numbers, you might not know whether to believe him, but if he tells you that it can be factored as $3{,}607 \times 3{,}803$, you can easily check that it is true using a hand calculator. One of the outstanding problems in logic and computer science is determining whether questions exist whose answer can be quickly checked (for example by computer), but which require a much longer time to solve from scratch (without knowing the answer). There certainly seem to be many such questions. But so far no one has proved that any of them really does require a long time to solve; it may be that we simply have not yet discovered how to solve them quickly. Stephen Cook formulated the *P* versus *NP* problem in 1971.

Although in many instances you could question the practical distinction between polynomial and nonpolynomial algorithms, this problem is central to our current understanding of computing. Roughly, it conjectures that many of the problems we currently find computationally difficult must perforce be that way. It is a question about methods, not about actual computations, but it underlies many of the challenge problems we can imagine posing. A question that requests us to "compute such and such a sized incidence of this or that phenomena" always risks having the answer "it's just not possible" because $P \neq NP$.

Two specific challenges

With the caveat that although factoring is difficult, it is not generally assumed to be in the class of *NP*-hard problems, let us offer two challenges that are far-fetched but not inconceivable goals for the next few decades.

Design an algorithm that can reliably factor a random thousand-digit integer

Even with a huge effort, current algorithms get stuck at about 150 digits. (See www.rsasecurity.com/rsalabs/challenges/factoring/index.html for a list of current factoring challenges.) And there is a \$100,000 cash prize offered for any reliable 10-million-digit prime (www.mersenne.org/prime.htm).

Primality checking is currently easier than factoring, and there are some very fast and powerful probabilistic primality tests—much faster than those providing certificates. Given that any computation has potential errors due to subtle (or even not-so-subtle) programming bugs, compiler errors, software errors, or undetected hardware integrity errors, it may be pointless to distinguish between these two types of primality tests. Many would take their chances with a $(1 - 10^{-100})$ probability statistic over a proof any day.

Learn how to factor large numbers, and most current security systems are crackable.

These questions are intimately related to the Riemann hypothesis, although not obviously so to the nonafficionado. They are also critical to issues of Internet security—learn how to factor large numbers, and most current security systems are crackable.

Find the minima in the merit factor problem up to size 100

There are many old problems that lend themselves to extensive numerical exploration. For example, in signal processing there is the *merit factor problem*, which is due to Marcel Golay with closely related versions due to Littlewood and Paul Erdös. Its pedigree is long, but not as long as the Riemann hypothesis (see http://athene.nat.uni-magdeburg.de/~mertens for recent records and references).

We can formulate it as follows. Suppose $(a_0 := 1, a_1, \ldots, a_n)$ is a sequence of length $n + 1$, where each a_i is either 1 or –1. If

$$c_k = \sum_{j=0}^{n-k} a_j a_{j+k} \tag{1}$$

then the problem is, for each fixed n, to minimize:

$$\sum_{k=-n}^{n} c_k^2 \ . \tag{2}$$

We can find exact minima up to about $n = 50$. The search space of sequences at size 50 is 2^{50}, which is about today's limit for a very large-scale calculation. In fact, the records use a branch-and-bound algorithm that more or less grows like 1.8^n. This is marginally better than the naive 2^n of a completely exhaustive search, but it is still painfully exponential.

The next best hope is radically different computers, perhaps quantum computers.

The best hope for a solution is better algorithms. The problem is widely acknowledged as a very hard problem in combinatorial optimization, but it isn't known to be in one of the recognized hard classes like NP. The next best hope is radically different computers, perhaps quantum computers. And there is always a remote chance that analysis will lead to a mathematical solution.

A concrete example

Let's examine some of the mathematical challenges in a specific problem Donald Knuth recently proposed. He asked solvers to evaluate the following sum:[3]

$$\sum_{k=1}^{\infty}\left(\frac{k^k}{k!e^k}-\frac{1}{\sqrt{2\pi k}}\right). \tag{3}$$

We answer Knuth's question in the following steps.

1. A very rapid Maple computation yielded –0.08406950872765600... as the first 16 digits of the sum.
2. The inverse symbolic calculator has a "smart lookup" feature—alternatively, we could use a sufficiently robust integer relation finder—that replied that this was probably $-2/3-\zeta(1/2)/\sqrt{2\pi}$.
3. Checking this to 50 digits provided ample experimental confirmation. Thus, within minutes we knew the answer.
4. So why did these numerical and symbolic numbers match? A clue was provided by the surprising speed with which Maple computed the slowly convergent infinite sum. The package clearly knew something the user did not. Peering under the covers revealed that it used the *LambertW* function, W, which is the inverse of $w = z \exp(z)$. (A search for "Lambert W function" on MathSciNet provided nine references—all since 1997 when the function appears named for the first time in Maple and Mathematica.)
5. The presence of $\zeta(1/2)$ and standard Euler-MacLaurin techniques, using Stirling's formula (as might be anticipated from the

$$\sum_{k=1}^{\infty}\left(\frac{1}{\sqrt{2\pi k}}-\frac{1}{\sqrt{2}}\frac{(1/2)_{k-1}}{(k-1)!}\right)=\frac{\zeta(1/2)}{\sqrt{2\pi}} \tag{4}$$

where the binomial coefficients are those in the series for

$$\frac{1}{\sqrt{2-2z}} .$$

Now Equation 4 is a formula Maple can prove.
6. However, we still need to show

$$\sum_{k=1}^{\infty}\left(\frac{k^k}{k!e^k}-\frac{1}{\sqrt{2}}\frac{(1/2)_{k-1}}{(k-1)!}\right)=-\frac{2}{3} . \tag{5}$$

7. Guided by the presence of W and its series

$$\sum_{k=1}^{\infty}\frac{(-k)^{k-1}z^k}{k!} ,$$

an appeal to Abel's limit theorem lets us deduce the need to evaluate

$$\lim_{z\to 1}\left(\frac{d}{dz}W(-z/e)+\frac{1}{\sqrt{2-2z}}\right)=\frac{2}{3} . \tag{6}$$

Again, Maple can establish Equation 6.

Of course, this all took a fair amount of human mediation and insight.

In 1996, discussing the philosophy and practice of experimental mathematics, we wrote[4]

As mathematics has continued to grow there has been a recognition that the age of the mathe-

> matical generalist is long over. What has not been so readily acknowledged is just how specialized mathematics has become. As we have already observed, subfields of mathematics have become more and more isolated from each other. At some level, this isolation is inherent but it is imperative that communications between fields should be left as wide open as possible. As fields mature, speciation occurs. The communication of sophisticated proofs will never transcend all boundaries since many boundaries mark true conceptual difficulties. But experimental mathematics, centering on the use of computers in mathematics, would seem to provide a common ground for the transmission of many insights.

This common ground continues to increase and extends throughout the sciences and engineering.

The corresponding need is to retain the robustness and unusually long-livedness of the rigorous mathematical literature. Doron Zeilberger's proposed *Abstract of the Future* challenges this in many ways: "We show in a certain precise sense that the Goldbach conjecture (where every even number is the sum of two primes) is true with probability larger than 0.99999 and that its complete truth could be determined with a budget of 10 billion."[4]

He goes on to suggest that only the Riemann hypothesis merits paying really big bucks for certainty. Relatedly, Greg Chaitin argued that we should introduce the Riemann hypothesis as an axiom: "I believe that elementary number theory and the rest of mathematics should be pursued more in the spirit of experimental science, and that you should be willing to adopt new principles. I believe that Euclid's statement that an axiom is a self-evident truth is a big mistake. The Schrödinger equation certainly isn't a self-evident truth! And the Riemann hypothesis isn't self-evident either, but it's very useful. A physicist would say that there is ample experimental evidence for the Riemann hypothesis and would go ahead and take it as a working assumption."[4]

How do we reconcile these somewhat combative challenges with the inarguable power of the deductive method? How do we continue to produce rigorous mathematics when more research will be performed in large computational environments where we might or might not be able to determine what the system has done or why? This is often described as "relying on proof by 'Von Neumann says'."

At another level we see the core challenge for mathematical computing to be the construction of workspaces that largely or completely automate the diverse steps illustrated in Knuth's and similar problems.

Acknowledgment

The National Sciences and Engineering Research Council and the Network of Centres of Excellence Program supported this research.

References

1. *Computing in Science & Eng.*, vol. 2, no. 1, Jan./Feb. 2000.
2. J.E. Littlewood, "Some Problems in Real and Complex Analysis," *Heath Math. Monographs*, Lexington, Mass., 1968.
3. "Problem 10832," *Am. Math. Monthly*, Nov. 2000.
4. J.M. Borwein et al., "Making Sense of Experimental Mathematics," *Math. Intelligencer*, vol. 18, no. 4, Fall 1996, pp. 12–18.

Jonathan P. Borwein is Shrum Professor of Mathematics at Simon Fraser University and director of the Centre for Experimental and Constructive Mathematics. His interests span pure (analysis), applied (optimization), and computational (complexity and numerical analysis) mathematics. He received his DPhil from Oxford University. He is president of the Canadian Mathematical Society, chairman of the National Science Library NRC-CISTI's Advisory Board, and a fellow of the Royal Society of Canada. Contact him at the Centre for Experimental and Constructive Mathematics, Simon Fraser University, Burnaby, British Columbia V5A 1S6, Canada; jborwein@cecm.sfu.ca.

Peter B. Borwein is a professor of mathematics at Simon Fraser University. His research interests include diophantine and computational number theory, classical analysis, and symbolic computation. He received his PhD from the University of British Columbia. Contact him at the Centre for Experimental and Constructive Mathematics, Simon Fraser University, Burnaby, British Columbia V5A 1S6, Canada; pborwein@cecm.sfu.ca; www.cecm.sfu.ca/~pborwein.