Challenges in Mathematical Computing

Jonathan M. Borwein and Peter B. Borwein*

February 19, 2001

ABSTRACT. Almost all interesting mathematical algorithmic questions relate to NP-hard questions and such computation is prone to explode exponentially. More space, more speed and processors, and even say massive parallelism will have an impact but it will be largely at a 'micro not macro' level. We anticipate the greatest benefit accruing from mathematical platforms that allow for highly computer assisted insight generation (more 'aha's' per cycle), not from solution of grand challenge problems.

1 Mathematics Embraces Computing

It is often said 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 started to change with a vengeance.

Major *symbolic mathematics* or *computer algebra* packages, most notably Maple and Mathematica, have over the last fifteen years reached a remarkable degree of sophistication. We should also allude to counterparts such as Axiom, Macsyma, Reduce, MuPad and Derive and to many other more specialized packages such as GAP, Magma or Cayley (for group theoretic computation), Pari (for number theory), KnotPlot (for knot theory) SnapPea

*Centre for Experimental & Constructive Mathematics, Simon Fraser University, Burnaby, British Columbia V5A 1S6, Canada. Email: `jborwein@cecm.sfu.ca,pborwein@cecm.sfu.ca`. Research supported by NSERC and by the Network of Centres of Excellence Program.

(for hyperbolic 3-manifolds) and SPlus (for statistics), and many more. 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 data bases and web-distributed (often Java-based) computational tools. We mention: the mathematics front end to the Los Alamos Preprint ArXiv (`front.math.ucdavis.edu/`), Mathematical Reviews on the Web (`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`) which infers symbolic structure from numerical input, and an Integer Relation Finder (`www.cecm.sfu.ca/projects/IntegerRelations/`).

The relatively seamless *integration* of all these components arguably represents *the* challenge for 21st Century computational mathematics. By contrast, it is hard to think of mathematical problems where a dramatic increase in speed and scale of computation would make possible a presently intractable line of research. It is easy to give examples where it would not. Thus, consider Lam's 1991 proof (`www.cecm.sfu.ca/organics/papers/lam/index.html`) of the nonexistence of a finite projective plane of order 10.[1] It involved thousands of hours of CRAY and other computation. Lam's estimate is that the next case ($n = 18$) susceptible to his methods would take millions of years on any conceivable architecture. While a certain class of mathematical enquiries is susceptible to massively parallel, even web based 'embarrassingly parallel', computation[2] these tend, however interesting, not to to be problems central to mathematics.

2 Computational Excursions in Contemporary Mathematics

Rather difficult problems, previously viewed as intractable, such as exact integration of elementary functions have been significantly attacked. A number of the most important mathematical algorithms of the twentieth century are (i) the Fast Fourier Transform, (ii) Lattice Basis Reduction methods and

[1] A hunt for a configuration of $n^2 + n^1 + 1$ points and lines.

[2] For example, discovering Mersenne primes: those of the form $2^n - 1$.

related Integer Relation algorithms, (iii) the Risch algorithm for indefinite integration, (iv) Gröbner basis computation for solving algebraic equations, and (v) the Wilf/Zeilberger Algorithms for 'hypergeometric' summation and integration that rigorously prove very large classes of identities. All these are, or soon will be, centrally incorporated in such packages.[3]

Such packages, and powerful more numerical relatives such as MatLab, 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 allow access to sophisticated mathematics to a very broad cross-section of scientists and engineers.

There is a coherent argument that the emergence of such packages, and their integration into mathematical parole, represents the most significant part in a paradigm shift in how mathematics is done; and certainly they have already become a central research tool in many subareas of mathematics both from an exploratory and from 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, one 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 does one do arbitrary precision numerical quadrature? When does one switch methods with precision required or with different analytic properties of the integrand? How does one deal with branch cuts of analytic functions? How does one deal consistently with log? (Even this isn't completely worked out.) More ambitiously how does one do a similar analysis for differential equations? 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. And ultimately, can we provide any 'certificates' that a given numeric or

[3]The first two were among the ten algorithms with "the greatest influence on the development and practice of science and engineering in the 20th century" described in the previous volume of this journal. Of course many of the others, such as sorting algorithms, are fundamental to the needs of contemporary mathematics.

symbolic computation is indeed a proof or even just correct?

Within this context a number of very interesting problems concerning the visualization of mathematics arise. How does one actually "see" what one is doing. It has been argued that Cartesian graphing was the most important invention of the last millennium. Certainly it changed how we thought about mathematics – the subsequent development of differential calculus rested on it. More subtle and complicated graphics, like those of fractals, allow for a kind of exploration that was previously impossible. There are many issues to be worked out here that live at the interface of mathematics, pedagogy and even psychology but are very timely to get right. (Think of how one visualizes the human genome and its patterns – which is after all just a particular several billion digit number base four.) An instructive example is afforded by the growing reliance of numerical analysts on graphic representation of large sparse matrices – the pictures show structure, numerical measurements very little.[4]

The great success of the symbolic algebra packages has been their mathematical generality and ease of use. These packages deal most successfully with algebraic problems while many (perhaps most) serious applications require analytic objects such as definite integrals, series and differential equations. All the elementary notions of analysis, like continuity and differentiability have to be given precise computational meaning. The first challenge 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 over simplifying expressions, leaving out hypotheses or 'dividing by zero') are precisely how one begins oneself. As Hadamard noted almost a century ago:

> "The object of mathematical rigor is to sanction and legitimize the conquests of intuition, and there was never any other object for it."

[4] A nice example is JavaView (`www-sfb288.math.tu-berlin.de/vgp/javaview/index.html`) for doing 3D Geometry on the web.

3 Challenge Problems for Computational Pure Mathematics

1. The question that a pure mathematician might trade his soul with the devil to solve is most likely the so called "Riemann–Hypothesis" of 1859. The bounty on its solution now exceeds $\$1,000,000$ – the amount offered by the *Millennium Prize* of the Clay Mathematics Institute (`www.claymath.org/prize_problems/rules.htm`).

At the Clay Institute website 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 very 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* $\Re(s) = \frac{1}{2}$.

We observe in passing that one of the most famous results in elementary mathematics is Bernoulli'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. Unlike problems such as Fermat's last problem (now theorem) which may prove to be an isolated mountain peak, even if the proof methods are tremendously significant, [5] the truth of the Riemann Hypothesis is central – its falseness would be disquieting. Most mathematicians believe the Riemann Hypothesis is true though there have been notable dissenters. Littlewood, one of the great analytic number theorists of last century is in print hypothesizing its falseness [6] . Of course, finding just one zero off the line $\Re(s) = \frac{1}{2}$,[7] should it exist, is worth a million dollars (although perhaps the prize is only for a proof not a disproof – certainly a proof is more interesting) and this may provide additional motivation to extend the climb of this particular mountain. The fact that more than the first billion zeros are known, by computation, to satisfy the Riemann hypothesis might be considered "strong numerical evidence" as it is the article by Enrico Bombieri that accompanies the prize citation. But it is far from overwhelming – there are subtle phenomena in this branch of mathematics that only manifest themselves far outside of 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 one 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 clearly as difficult as this one.

2. Of the seven million-dollar Millennium Prize problems, the one that is most germane to this discussion is the so called *$P \neq NP$ problem.* Again, we quote from the discussion on the Clay website:

> "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,

[5] A much deeper community understanding of modular and elliptic functions may also pay dividends.

[6] J.E. Littlewood, "Some Problems in Real and Complex Analysis," Heath Mathematical Monographs, 1968.

[7] And off the real line where there are 'trivial' zeros at negative even integers.

> 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 3607 times 3803 then 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 one may question the practical distinction between polynomial and non polynomial algorithms, this problem really is central to our current understanding of computing. Roughly it conjectures that many of the problems we currently find computationally difficult must per force be be that way. It is a question about methods, not about actual computations, but it underlies many of the challenge problems one can imagine posing. A question that requests one to "compute such and such a sized incidence of this or that phenomena" always risks having the answer "it's just not possible" because $P \neq NP$.

4 Two Specific Challenges

With the 'NP' caveat,[8] let us offer two challenges that are let us offer two challenges that are far fetched but not inconceivable goals for the next few decades.

[8] Though factoring is difficult it is not generally assumed to be in the class of NP-hard problems.

3. *Design an algorithm that can reliably factor a random thousand digit integer.*

Current algorithms even with a huge effort get stuck at about 150 digits. Details lie at `www.rsasecurity.com/rsalabs/challenges/factoring/index.html` where the current factoring challenges are listed. Again, in the cash prize game there is also a $\$100,000$ offered for any honest $10,000,000$ 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: (i) subtle (or even not-so-subtle) programming bugs, (ii) compiler errors, (iii) other software errors, (iv) and undetected hardware integrity errors, it seems increasingly 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.

The above questions are intimately related to the Riemann Hypothesis, though not obviously so to the non afficionado. They are also critical to issues of internet security. Learn how to factor large numbers and most current security systems are crackable.

There are many old plum problems that lend themselves to extensive numerical exploration. To name but one other: a problem that arose originally in signal processing called the *Merit Factor problem* that is due in large part to Marcel Golay with closely related versions to Littlewood and Erdős. It has a long pedigree though certainly not as long as the Riemann hypothesis. Recent references and records can be found at (`itp.nat.uni-magdeburg.de/ mertens/`.)

It can be formulated as follows. Suppose $(a_0 := 1, a_1, ..., a_n)$ is a sequence of length $n+1$ where each a_i is either 1 or -1. If

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

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

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

Minima have be found up to about about $n = 50$. The search space of sequences at size 50 is 2^{50} which is about todays limit of a very very large scale calculation. In fact the records use a branch and bound algorithm which grows more or less like 1.8^n. This is marginally better than the naive 2^n of a completely exhaustive search but is still painfully exponential.

4. *Find the minima in the merit factor problem up to size 100.*

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.

5 A Concrete Example

In this section we illustrate some of the mathematical challenges with a specific problem, proposed in the *American Mathematical Monthly* (November, 2000).

10832. *Donald E. Knuth, Stanford University, Stanford, CA.* Evaluate

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

1. A very rapid Maple computation yielded $-0.08406950872765600\ldots$ as the first 16 digits of the sum.
2. The Inverse Symbolic Calculator has a 'smart lookup' feature[9] that replied that this was probably $-\frac{2}{3}-\zeta(\frac{1}{2})/\sqrt{2\pi}$.
3. Ample experimental confirmation was provided by checking this to 50 digits. Thus within minutes we *knew* the answer.
4. As to why? 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

[9] Alternatively, a sufficiently robust integer relation finder could be used.

revealed that it was using the *LambertW* function, W, which is the inverse of $w = z\exp(z)$.[10]

5. The presence of $\zeta(1/2)$ and standard Euler-MacLaurin techniques, using Stirling's formula (as might be anticipated from the question), led to
$$\sum_{k=1}^{\infty}\left(\frac{1}{\sqrt{2\pi k}} - \frac{1}{\sqrt{2}}\frac{(\frac{1}{2})_{k-1}}{(k-1)!}\right) = \frac{\zeta(\frac{1}{2})}{\sqrt{2\pi}}, \tag{1}$$
where the binomial coefficients in (1) are those of $\frac{1}{\sqrt{2-2\,z}}$. Now (1) is a formula Maple can 'prove'.

6. It remains to show
$$\sum_{k=1}^{\infty}\left(\frac{k^k}{k!\;e^k} - \frac{1}{\sqrt{2}}\frac{(\frac{1}{2})_{k-1}}{(k-1)!}\right) = -\frac{2}{3}. \tag{2}$$

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 one deduce the need to evaluate
$$\lim_{z\to 1}\left(\frac{d}{dz}\mathrm{W}(-\frac{z}{e}) + \frac{1}{\sqrt{2-2\,z}}\right) = \frac{2}{3}. \tag{3}$$
Again Maple happily does know (3).

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

6 Conclusion

In 1996, discussing the philosophy and practice of Experimental Mathematics, we wrote:[11]

[10] A search for 'Lambert W function' on MathSciNet provided 9 references – all since 1997 when the function appears named for the first time in Maple and Mathematica.

[11] J.M. Borwein, P.B. Borwein, R. Girgensohn and S. Parnes, "Making Sense of Experimental Mathematics," Mathematical Intelligencer, 18, Number 4 (Fall 1996), 12-18. The quotes from Zeilberger and Chaitin are also cited therein.

> "As mathematics has continued to grow there has been a recognition that the age of the mathematical generalist is long over. What has not been so readily acknowledged is just how specialized mathematics has become. As we have already observed, sub-fields 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* (1993) challenges this in many ways.

> "We show in a certain precise sense that the Goldbach conjecture[12] is true with probability larger than 0.99999 and that its complete truth could be determined with a budget of 10 billion."

He goes on to suggest that only the Riemann hypothesis merits paying really big bucks for certainty. Relatedly, Greg Chaitin (1994) 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[13]. 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

[12] Every even number is the sum of two primes.

[13] There is no evidence that Euclid ever made such a statement. However, the statement does have an undeniable emotional appeal.

> there is ample experimental evidence for the Riemann hypothesis and would go ahead and take it as a working assumption."

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 and more research will be performed in large computational environments where one may or not be able to determine what the system has done or why?[14]

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

[14]This has often been described as "relying on proof by 'Von Neumann says'."