

Opportunities and Challenges in Experimental Mathematics

David H. Bailey and Jonathan M. Borwein

12 October 2014

“Experimental mathematics” has emerged in the past 25 years or so to become a competing paradigm for research in the mathematical sciences. An exciting workshop entitled “Challenges in 21st Century Experimental Mathematical Computation” was held at the Institute for Computational and Experimental Research in Mathematics (ICERM), July 21-25, 2014, which explored emerging challenges of experimental mathematics in the rapidly changing era of modern computer technology. This report summarizes the workshop findings; information about the research presentations can be found at <http://icerm.brown.edu/tw14-5-cemc/>

While several more precise definitions have been offered for “experimental mathematics,” we used the informal one given in the book *The Computer as Crucible*:

Experimental mathematics is the use of a computer to run computations -- sometimes no more than trial-and-error tests -- to look for patterns, to identify particular numbers and sequences, to gather evidence in support of specific mathematical assertions that may themselves arise by computational means, including search.

“Experimental mathematics” is distinguished from “computational mathematics” and “numerical mathematics” in that the latter two generally encompass methods for applied mathematics, whereas “experimental mathematics” refers to advancing the state of the art in mathematical research *per se*.

While the overall approach and philosophy of experimental mathematics has not changed greatly in the past 25 years, its techniques, scale and sociology have changed dramatically. The field has benefited immensely from Moore’s Law and other advances in computer technology, but the increase in speed brought by algorithmic progress has often exceeded Moore’s law, notably in areas such as linear programming, linear system solving and integer factorization.

Software available to experimental mathematicians has also advanced impressively. Not only are commercial products such as *Maple*, *Mathematica* and *Matlab* advanced from earlier eras, but many new “freeware” packages are being used, including the open-source *Sage*, numerous high-precision computation packages and an impressive array of software tools and visualization facilities.

With all these tools and facilities, many new results have been published, ranging from new formulas for mathematical constants such as π , $\log(2)$ and $\zeta(3)$, to computer-verified proofs of the Kepler conjecture. Whereas it was once considered atypical or even improper to mention computations in a published paper, now it is commonplace. Several journals, such as *Experimental Mathematics* and *Mathematics of Computation* are devoted almost exclusively to mathematical research involving computations.

Yet many challenges remain as researchers press the envelope in mathematical computing. Among the most critical issues are the following:

Adapting codes to new platforms. The emergence of powerful, advanced-architecture platforms, particularly those incorporating features such as highly parallel, multi-core or many-core designs, present daunting challenges to researchers, who must now adapt their codes to these new architectural innovations, or else risk being left behind in the scientific computing world.

Ensuring reliability and reproducibility. Reproducibility means, for example, ensuring that the results of floating-point computations are numerically reproducible, and that the results of a symbolic computation are reliable (complications may arise when comparing two different expressions to determine whether they are mathematically equivalent). Many users implicitly trust the results of these tools, losing sight of the fact that they are far from infallible. Stronger interactions with the cousin discipline of formal proof systems (as was used by Thomas Hales to complete in 2014 a multi-year computer-verified proof of the Kepler conjecture on stacking spheres, should be one of the approaches to increased reliability, but huge efficiency issues have to be addressed.

Managing the exploding scale of data. The size of datasets used in the field has increased at least as fast as Moore's Law. *Algorithmic progress* is thus necessary, for example in tools aiding in the quest for structure in large numerical or symbolic datasets.

Large-scale software maintenance. The rapidly increasing size of many of the software tools used in the fields now means that mathematicians must confront the challenge of large-scale software maintenance. This includes the discipline, unfamiliar to many research mathematicians, of strict version control, collaborative protocols for checking out and updating software, validation tests, issues of worldwide distribution and support, and persistence of the code base.

Changing sociological and community issues. Numerous recently published results are the result of Internet-based collaborations, with research ideas, computer code and working manuscripts circulating around the globe multiple times in a single day. One example is the PolyMath Project, whereby a loosely-knit Internet-based team of mathematicians has addressed and, in several cases, “solved” or improved solution of interesting unsolved mathematical problems. Further progress will require improved tools and platforms for such collaborations, as well as an international “clearing house” to collect, validate and coordinate such activities.

Education. Computer-based tools are also being introduced into mathematical education, permitting students to see mathematical concepts emerge from hands-on experimentation, thus attracting to the field a cadre of 21st century computer-savvy students. While this is not the first time that technology has promised to reinvent mathematical education, it is clear that much additional thought is needed on how to best incorporate computation in education.

Other issues. The workshop discussion highlighted the fact that much of the published work to date in experimental mathematics has focused on a few fields that are particularly

amenable to computational exploration -- finite group theory, combinatorics and graph theory, number theory, evaluation of series and integrals, etc. How can we expand the scope of questions that have been examined with these methodologies, not just to other areas of mathematics but to other fields as well?

All of this also raises the question of how all this research work can be paid for. Unlike the case in the 'hard sciences', the majority of published mathematical research (pure and applied) has been completed without direct research funding, done by academic mathematicians or others as they have time alongside teaching or other formal duties. But some of the work described above, particularly that which involves substantial software development and maintenance, cannot be done so informally. Nor does a royalty model work, as it has for traditional publications, since the development costs are too great and the academic reward too small.

Thus it is clear that the field of experimental mathematics needs to work more vigorously with governmental funding agencies to find ways to provide this funding. Perhaps this may be done more easily if projects can be pursued in collaboration with researchers in other disciplines, particularly in fields such as computer science that have typically been somewhat more generously funded.

The full report is available as:

D. H. Bailey, J. M. Borwein, U. Martin, B. Salvy and M. Taufer, "Opportunities and Challenges in 21st Century Experimental Mathematical Computation," 26 Aug 2014, available at <http://www.davidhbailey.com/dhbpapers/icerm-2014.pdf>.

David H. Bailey is a retired senior scientist at the Lawrence Berkeley National Laboratory and a Research Fellow at the University of California, Davis. Jonathan M. Borwein is Laureate Professor in the School of Mathematical and Physical Sciences at the University of Newcastle and director of the university's Priority Research Centre in Computer Assisted Research Mathematics and its Applications (CARMA).