Opportunities and Challenges in 21st Century Experimental Mathematical Computation: ICERM Workshop Report

David H. Bailey* Jonathan M. Borwein† Ursula Martin‡
Bruno Salvy§ Michela Taufer¶

August 28, 2014

1 Executive summary

"Experimental mathematics" has emerged in the past 25 years or so to become a competitive paradigm for research in the mathematical sciences. A workshop held at the Institute for Computational and Experimental Research in Mathematics (ICERM), July 21–25, 2014, explored emerging challenges of experimental mathematics in the rapidly changing era of modern computer technology. This report summarizes the workshop findings.

So what exactly is "experimental mathematics"? While several definitions have been offered (e.g., [8]), perhaps the most succinct definition is given in the Borwein-Devlin 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. [9, p. 1]

Here we should distinguish "experimental mathematics" from "computational mathematics" and "numerical mathematics." While there is no clear delineation, the latter two terms generally encompass computational methods for concrete applied mathematics and engineering applications, whereas "experimental mathematics" usually applies more specifically to computations that

*Lawrence Berkeley National Lab (retired), Berkeley, CA 94720 and University of California, Davis, Dept. of Computer Science, Davis, CA 95616, USA, david@davidhbailey.com.
†CARMA, University of Newcastle, NSW 2303, Australia, jon.borwein@gmail.com
‡Oxford University, Oxford OX1 2JD, UK, Ursula.Martin@cs.ox.ac.uk.
§INRIA and ENS-Lyon, Lyon, 69342, France, bruno.salvy@inria.fr.
¶University of Delaware, Dept. of Computer Science, Newark, DE 19716, taufer@udel.edu.

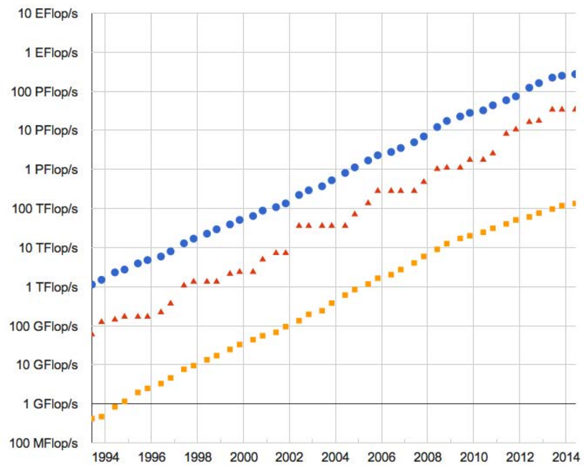

Figure 1: Performance of the Top 500 computers: Red = #1 system; orange = #500 system; blue = sum of #1 through #500.

advance the state of the art in mathematical research. In particular, approximations are usually not the end result, but an intermediate step in the discovery of an exact mathematical object.

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. New algorithms and computer implementations, scarcely dreamed when modern experimental mathematics first arose in the 1980s, are now widely employed. And the field has, of course, benefited immensely from Moore's Law and other advances in computer technology, which have magnified raw computing power by a factor of more than 100,000 in this same time frame — see Figure 1, which tracks the performance of the world's 500 most powerful supercomputers over the past twenty years. Note, strikingly, that the aggregate performance of the 500 systems in the 1994 list was surpassed by the lowest-ranked machine a decade later!

In the same period, the increase in speed brought by algorithmic progress has often exceeded Moore's law. Documented examples include linear programming [22, p. 71], linear system solving [39] and integer factorization [10]. Several areas of symbolic computation that are relevant to experimental mathematics have thus been radically transformed, due to the algorithmic progress in polynomial system solving, polynomial factorization and the handling of linear differential equations.

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 freely available packages are being used [5]. These include the open-source *Sage* [25], numerous high-precision computation packages and an impressive array of software tools. Advanced visualization facilities are also widely utilized to visually explore mathematical structures and theorems, yielding impressive insights not apparent with conventional tools.

With all these tools and facilities, many new results have been published, ranging from new formulas for mathematical constants such as $\pi, \log(n)$ and $\zeta(n)$ [8] to computer-verified proofs of the Kepler conjecture [24]. 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 them are the following:

Porting and 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. Code adaptation will be a major challenge of the next decade or two in the field of scientific computation. The field of experimental mathematics will be dependent upon and must engage with this adaptation.

Ensuring reliability and reproducibility of computed results. Reproducibility means, e.g., to ensure that the results of floating-point computations are numerically reliable and reproducible on other platforms. In the domain of symbolic computation, reliability is a more relevant question than reproducibility, with complications coming from undecidability issues when comparing two different expressions to determine whether they are mathematically equivalent. Along this line, the community has to face the reality that many less sophisticated users implicitly trust the results of these tools, losing sight of the fact that they are far from infallible. In any event, it is clear that we must build even greater reliability into these tools. Stronger interactions with the cousin discipline of *formal proof systems* 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 that users want to handle has increased at least as fast as Moore's Law, in many cases because the data itself is produced by faster, larger computer systems. However, most of the experimental mathematics algorithms now in use, both numeric and symbolic, have a complexity that is more than linear in the size of their input. Thus, doubling simultaneously the power of a computer and the size of its input make it appear slower to the user. Thus *algorithmic progress is necessary* for experimental mathematicians, notably 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. Many recently published results are the result of long-distance, internationally distributed, Internet-based collaborations. It is not uncommon for research ideas, computer code and working manuscripts to circulate around the globe multiple times in a single day. Even more ambitious are efforts such as the *PolyMath Project* [23], whereby a loosely-knit Internet-based team of mathematicians has addressed and, in several cases, "solved" some key unsolved mathematical problems. Such collaborations are qualitatively different than research of past years, which was mostly conducted by individual researchers working in relative isolation from one another. But it is clear that the effective deployment of these efforts will rely on *improved tools and platforms for collaborative mathematics.* There is also arguably a need for some sort of international "clearing house" to collect, validate and coordinate such activities.

Providing evidence-based rationales for experimental mathematics in the classroom. Current computer-based tools are also being introduced into mathematical education, often with very promising results, as students are able to see mathematical concepts emerge from direct, hands-on experimentation. Indeed, computer-based mathematics is already attracting to the field a cadre of 21st century computer-savvy students eager to press forward with these tools. This is not the first time that technology has promised to reinvent mathematical education. But it is clear that the entire process of mathematics education, from elementary to advanced levels, needs to be rethought. At the least, much "experimentation" will be required to see which approaches really work.

2 Emerging techniques in experimental mathematics

While "experimental mathematics" encompasses many techniques and methodologies, the specific objectives that we deal with here are (i) the process of experimenting to discover new mathematical facts and (ii) of proving experimentally discovered facts. Both aspects of experimental mathematics are the target of newly developed or enhanced techniques and methodologies.

2.1 Algorithms

The current increase of computing power parallel to an explosion of the size of data available demand algorithmic progress throughout computer science. In the area of experimental mathematics, several specific issues were raised during the workshop.

Tools for *discovering structure* from a set of floating point expansions, such as the PSLQ algorithm, the LLL algorithm and their variants, typically suffer badly from an explosion of computational cost as the dimension of the search space is increased. Thus more research effort is needed in the design of algorithms for such purposes whose computational cost does not grow so rapidly. Similarly, recovering a conjectured linear differential or recurrence equation from a sequence of coefficients relies on efficient algorithms for Padé-Hermite approximants or, more generally, on structured matrices with polynomial entries, whose complexity is still increasing too much with respect to the order of the target equation. Improvement is clearly needed, as equations of very large orders have been found to occur naturally in various combinatorial problems.

Tools for *proofs* encompass potentially all of computer algebra. The area of identity proving in the hypergeometric context has made tremendous progress since the pioneering work of Zeilberger and the recent years have still led to many advances in terms of scope and efficiency. Computation with systems of polynomial equations has also benefited both from constant implementation effort and from improved algorithms; recent years have seen an important and interesting activity towards specific algorithms for structured polynomial systems. Other recent advances in effective real algebraic geometry should also benefit to applications in experimental mathematics.

Yet another direction where algorithmic development would impact experimental mathematics is in the visualization of large datasets. There, progress is likely to come from the user needs of other, much larger, communities, such as statisticians dealing with web-size data or physicists handling experiments involving terabytes of data.

2.2 Reproducibility

The issue of *reproducibility* has recently come to the fore, not just in experimental mathematics but also in the larger realm of scientific computing, as discussed at length in an earlier ICERM workshop [26], in the First Summer School on Experimental Methodology in computational Science Research [2] and at the website `http://recomputation.org`. While no universally accepted definition of reproducibility exists, it is a sad fact that the field of scientific computing has never incorporated a culture of reproducibility, however one may define the term. In particular, computational scientists typically do not keep detailed records of their research processes, and as a result confusion has reigned when other research teams cannot reproduce a published result, or even when the same research team cannot reproduce its own result. Thus the enterprise of experimental mathematics needs to adopt procedures similar to those that have been or are being adopted in other scientific disciplines.

Numerical reproducibility has emerged as a particularly important issue, since the scale of computations has greatly increased in recent years, particularly with computations performed on many thousands of processors and involving similarly large datasets. Large computations often greatly magnify the level of numeric error, so that numerical difficulties that were once of little import now are large enough to alter the course of the computation or to draw into question the overall validity of the results.

Numerical difficulties now typically come to light when only a minor change is made to the computation, producing final results differing surprisingly from benchmark results. For example, in a recent case reported at the workshop, a computer program processing data from the Large Hadron Collider missed some previously detected collisions and misclassified others, all as a result of a minor change made to the transcendental function library, which change should only affect the least significant bit returned in such operations [14]. Higher-precision arithmetic may be required to ameliorate such numerical problems, or, at the least, much more careful analysis is required.

Given the widespread usage of high-precision arithmetic in experimental mathematics, it is clear that increased attention must be given to the question of whether sufficient numeric precision is employed to produce reliable results. Thus, researchers in the field need to investigate validity checks specifically targeted to determining whether adequate numeric precision is being used (this may vary inside the computation). Such considerations are particularly acute when floating-point computations may have been employed in a computer algebra system without this being known to the end-user.

2.3 Reliability of results

Although reproducibility is an important goal by itself, the ultimate objective of computations in the field of experimental mathematics is mathematical certainty or at least secure mathematical knowledge. In symbolic computation, various representations are available for a single mathematical object: polynomials may be factored or expanded, different kinds of rewriting may be applied to trigonometric expressions, etc., and no representation is universally better than the others and different programs may return different forms for the same answer. Another source of non-reproducibility is the use of randomized algorithms whose execution may vary from one execution to the next. In this context, reliability (mathematical correctness of the result) is more relevant than reproducibility, even though the lack of reproducibility makes testing and debugging software harder.

A fundamental issue in this area is that for important classes of mathematical expressions, recognizing equality is undecidable. This is also behind the difficulty in comparing results obtained by different computer algebra systems, or even by different versions of the same one. Several approaches help mitigate this problem. For specialists of computer algebra, the natural technique when faced with a computation is to define precisely an algebraic structure where equality is decidable (e.g., rational functions over the rationals, or algebraic extensions) and where all the computation will take place. An approach taken by Zeilberger's algorithm and variants [38] is to provide a certificate (an identity between rational functions) together with their results. Currently, the best way to gain confidence in a result is through *formal proofs*. For instance, tools for inequalities exist in the *Flyspeck* project [24]; the recent complete proof of the Feit-Thompson theorem in Coq [20] has led to libraries of formal lemmas for several parts of algebra. Thus formal proofs will become more and more useful here. Additional work is needed to make them faster and they are unlikely to be broadly accessible within the next few years.

Thus for most users, the best advice currently is to learn to distrust the results of computer algebra systems and get used to performing a few sanity checks on their results. Here are a few examples of checks that are now routine in the field:

– Identities involving parameters are often tested numerically at one or more appropriate evaluation points.

– High precision numerical calculations leading to integers, for example using an FFT-convolution algorithm in a multiplication operation, are checked to see that they produce values that are very close to integers.

– When new formulas are found using the PSLQ *integer relation algorithm* [8, 5], it is common practice to track the size of the reduced vector as the algorithm proceeds, and then note the magnitude of the drop in this value when a tentative relation is discovered. If this drop is, say, 50 or more orders of magnitude, then this indicates that the tentative relation is very likely a real mathematical relation (although rigorous proof is still required).

– When the output itself is very large and no other simple check is available, e.g. in the very high precision computation of mathematical constants such as $\pi, e, \log 2$, etc., it is now customary to check the results by a separate independent computation, say using a different algorithm.

The recently initiated *Digital Repository of Mathematical Formulae* (DRMF) [12] project ties specific LaTeX character sequences to well-defined mathematical objects. The DRMF, like the Digital Library of Mathematical Functions (DLMF) [11], is being developed at the National Institute of Standards and Technology (NIST). One workshop participant reported work in the direction of linking formulas in the DLMF to machine-checkable proofs, thus adding another level of validation to the results.

2.4 Standards

Along this line, reliability concerns are an issue even for explorations that involve large public datasets — the possibility that an error has been made in producing the data, or that an error has occurred when accessing the data, cannot be ruled out and must be guarded against. Large datasets that include the specific algorithms used to generate the data are particularly useful in the regard, as they permit one to independently reconstruct the data if a question arises as to the accuracy of some item in the dataset. Neil Sloane's *Online Encyclopedia of*

Integer Sequences [34] is an excellent model for how this can be done effectively over a period of decades.

In most cases however, the community is very far from having adopted a culture of reproducibility and reliability at this level. One step that would greatly help foster such a culture in the field is to establish some standard practices:

- Field-specific standards for reports on computational experiments (and insisting that researchers report hunts that found nothing or tests that failed). Even simple data from mathematical experiments are useful but relatively rarely available. When they exist, they generally occur as typeset tables at the end of mathematical articles. Even more exceptional and laudable is the access to files with actual data, such as the supplementary material with several of M. Kauer's publications[1], or the software that accompanies many of D. Zeilberger's articles;

- Standards for test suites of mathematical software. Such test suites exist for many different areas but with no coherence; more generally, no information is available concerning the test suites that has been employed;

- Standards for granting access to experimental datasets;

- Standards for reporting experiments that involve floating-point arithmetic: e.g., what is the claimed level of numerical reliability (i.e., how many digits), and what tests have been made to back up this claim?

The experimental mathematics community is currently very far from agreeing on and adopting such standards. As first steps, researchers should make an habit of giving access to files related to their published research, even if they are only accessible or readable by a computer for a couple of years. We have to start somewhere, so perhaps some individual research groups can adopt one or more of these practices and then report their experience at future workshops and conferences.

3 Computer systems and software

As mentioned above, the computational power and designs of computer systems used in the field have changed dramatically over the past 25 years. Just as importantly, the range of software available to experimental mathematicians has also advanced impressively, opening new challenges ahead for the field of experimental mathematics.

[1] `http://www.kauers.de/publications.html`

3.1 Computational program, software packages, and libraries

Not only are commercial products such as *Maple*, *Mathematica* and *Matlab* advanced from earlier eras, but many new freely available packages are being used. These include the open-source *Sage*, as well as numerous high-level computational libraries such as SuperLU, LAPACK, PETSc and others. In addition there exists an impressive array of individually-written software tools based upon this collection of libraries. Modern software projects often incorporate a complex combination of many libraries and packages, and this combinatorial explosion can make it more difficult to examine, reproduce, and extend results obtained using them. Higher level environments, like *Sage*, and hierarchical library interfaces are an attempt to control this complexity.

Another important component of present and future experimental mathematics software are packages that support certain types of computation, such as high-precision computation, linear system solution or statistical analysis, or which implement a specific algorithm such as PSLQ or the tanh-sinh quadrature algorithm.

Some specific software packages that have proven to be particularly useful in experimental mathematics include the following:

- Computer algebra packages: *Mathematica*, *Maple*, *Magma* and *Sage* [25].
- Statistics and machine learning: *R* [35].
- Linear algebra: *Matlab*, LAPACK [17], LINPACK [18] and SuperLU (for sparse direct matrix computations) [32].
- High-precision and arbitrary precision libraries: ARPREC [13], GMP [16], MPFR [16] and QD [15].
- Other commonly used libraries: ADCIRC (circulation and transport), PETSc (for systems modeled by partial differential equations) [29], FGb (for Gröbner bases) and LattE (for computations with polytopes).
- Experimental math tools incorporating specific algorithms: Implementations of algorithms such as PSLQ (for integer relation detection) and tanh-sinh quadrature (for high-precision numerical integration) are often included with high-precision arithmetic packages such as ARPREC and QD. Both *Mathematica* and *Maple* include commands for Padé approximants (for finding rational functions approximations) and for integer relation detection (with applications e.g., to finding a polynomial satisfied by a floating-point value), as well as options for adaptive doubly exponential integration. *Maple* also has a command for Padé-Hermite approximants (for finding linear dependencies with polynomial coefficients among power series) with applications to the discovery of linear differential or recurrence equations. There are numerous other examples that users of these packages may or may not be aware of.

Advanced visualization facilities are also widely utilized to visually explore mathematical structures and theorems, yielding impressive insights not apparent with conventional tools. For example, some intriguing results were recently obtained on the normality of real numbers by representing the base-b expansions of various mathematical constants visually as a "random" walk [3].[2] In this workshop, a talk on non-convex feasibility demonstrated the usefulness of visualizing the convergence of projection and reflection algorithms for this problem [4]. Tools for generation, description, and manipulation of advanced graphic output are also lacking.

Along this line, we should mention the recently organized effort to produce a Digital Mathematical Library (DML), which would encompass not just formulas but, much more broadly, a knowledge-based library of mathematical concepts, theorems and proofs. For details, see [30].

3.2 Best practices for software development

Researchers in the field who are developing large software tools now must adopt practices of software engineering, as appropriate, to manage their codes. "Best practices" in this area include software tools for version control, collaborative protocols for checking out and updating software, validation tests, issues of worldwide distribution and support, and persistence of the code base. Equally important is the development of strong programming interfaces (APIs) that allow interoperability between libraries, extension to new capabilities, and automatic generation of code for specific tasks or wrappers for new languages.

Along this line, the long-term persistence of experimental mathematical software is important. Persistence is important not just for software reuse, but also for reproducibility, in case another team (or even the same team) of researchers wishes to reconstruct earlier published results. Thus the experimental mathematics community, like others, must develop permanent repositories for software and encourage researchers to place their software there. Along this line, for decades, we have told our students and research assistants to document their code; this time we really mean it!

Another area that perhaps is unfamiliar to many research mathematicians, but which will increasingly be essential in the future, is to employ advanced database structures and data management facilities to store and manage data (including managing access to this data by a worldwide community of researchers). Some experimental mathematics projects have produced multi-Tbytes of data; these datasets will only increase in size in future research.

3.3 Programming multi-core and many-core computing systems

The emergence of powerful new computing platforms, particularly the massively parallel multi-core or many-core systems, presents daunting challenges to both

[2] A 108Gb image of a walk on 200 billion bits of Pi is hosted by *Gigapan* and is available through `http://walks.carma.newcastle.edu.au/`.

researchers and commercial vendors, who must now adapt their codes to these new architectural innovations, or else risk being left behind in the scientific computing world. This will be a major challenge of the next decade in the field.

Emerging techniques in experimental mathematics must go hand-in-hand with system developments. Future algorithms must meet the concurrency of multi- and many-core platforms while preserving requirements such as accuracy and reproducibility. The multithreading environment of these platforms may be a major impediment to the latter requirements. Recently, graphics processing units and similar many-core designs have been a fundamental driver and testing platform for many of the above mentioned challenges. Still, to succeed in a long term, it is clear that the experimental mathematics community needs to embrace many-core technology, beyond any particular vendor-specific architecture. Doing so will also create dependability and help meet interoperability requirements for the software libraries.

4 Communities and collaborations

As mentioned above, experimental mathematics can be thought of as being as much a philosophy as a discipline. The field certainly encompasses computer-assisted mathematics, but it also intersects with the realm of computational and numerical mathematics (i.e., techniques mostly applied to other scientific and engineering disciplines). Clearly there is overlap between these communities, and both can learn from the other. Experimental mathematics also has much to offer current thinking on the nature of Mathematics [6].

4.1 Definition of a diverse community

Even within the field of experimental mathematics *per se*, there is a large international, multi-discipline community that embrace the discipline from different points of view. Specifically, some approach the field from the vantage of pure mathematics; some approach it from the computer science perspective; some approach it from within symbolic computing; others approach it from the education angle; yet others approach it from the broader arena of software tools and databases for scientific research. It is clear that each of these overlapping but distinct communities will need to work in concert.

Here is a list of some of the communities that experimental mathematics may interact with in their work:

- *Traditional mathematical fields.* Experimental work has traditionally focused on fields such as combinatorics, computational number theory and group theory.

- *Non-traditional mathematical fields.* New applications include computational geometry and topology.

- *Computer science.* Many working in experimental mathematics already come from computer science backgrounds or are familiar with theoretical computer science.

- *Computational science.* These researchers mostly work in various "applied" disciplines, where work centers on issues of large-scale, highly parallel scientific computation.

- *Probability and statistics.* The areas are increasingly key to much computational research, both theoretical and practical.

- *Physics.* This includes related disciplines such as astronomy, astrophysics and cosmology.

- *Chemistry.* Computational chemistry and materials science increasingly involve rather sophisticated mathematics and computation.

- *Engineering.* Many fields of engineering now involve sophisticated mathematical algorithms and large-scale computations.

- *Biology.* This includes biostatistics, namely statistical methods specifically applied to genomics and biomedicine.

- *Medicine and biochemistry.* This includes advanced geometric imaging and visualization techniques, as well as "data mining" of biomedical data.

- *Social science.* Economics, psychology, sociology, linguistics, anthropology, history all now include significant mathematical computational research.

- *Humanities.* Considerable interest has arisen lately in "digital humanities," which is opening up new areas of research, for example in text analysis to identify the authors of texts or, more broadly, to quantitatively analyze cultural trends such as the evolution of grammar, the adoption of technology and various historical trends. See [21] for details.

- *Finance and investment.* Recently there has been an explosion of activity in this arena, with many top mathematicians and computer scientists working in the field.

The breadth of the disciplines in the list above clearly underscores a major challenge to the field: How can researchers learn at least a modicum of each of these fields, so that one can be moderately conversant with these other communities and explore potential collaborations? Along this line, it is important to keep in mind a related challenge, namely to foster respect for these other disciplines and to expect respect in return. All too often, promising collaborations of this sort founder on the problems of this sort.

We should add a word caution here that in all these fields, mathematicians have an obligation to confront the unsound usage of mathematics. For example, the overselling of quant-based investing before the great economic recession of

2007–08 has led to financial products and even peer-reviewed papers that are, in many cases, mathematically and/or statistically unsound [7]. The questionable application of statistical methods in the social sciences has also recently come under harsh criticism [40].

4.2 Tools to foster collaboration

The Internet has certainly facilitated many of these collaborations. When many of those attending the conference began their careers, mathematics, even experimental mathematics (such as it was back then), research papers typically were authored alone, or perhaps by two or three authors at the same institution. Nowadays none of us think twice about writing a paper with multiple collaborators in several time zones or continents.

Two of the workshop organizers, for instance, reported that they have collaborated for over 30 years, jointly writing dozens of papers, even though they have never been in the same country at the time, let alone the same institution, and for at least the past six years have been on different continents. They exchange computational research and manuscript drafts via email and Dropbox, and communicate mostly by video Skype. They did not meet in person for eight years after their collaboration began, and even now, only occasionally interact in person. Numerous others at the workshop reported similar changes in how they interact with colleagues.

A wide array of software tools and online facilities are employed by researchers in the field to support their multi-institution and multi-national collaboration. Some of the more common items include those listed in [5] and:

- Commercial communication tools: Skype or Google hangouts, Access Grid systems, etc. Most of these now also include facilities for "chat" and multi-person video conferencing. Some of these tools, such as Xoom and FaceTime, are restricted to specific vendors and/or platforms, complicating their wider applicability.
- Data sharing tools (high bandwidth is important): Email, Globus toolkit, Google docs, Dropbox.
- Software sharing tools: Git and GitHub [19], SVN.
- Mathematical community online resources and tools: Mathworld [28], PolyMath [23], Math Overflow, PlanetMath [31], DLMF [11], DRMF [12], OEIS [34], mathLibre [27], swMATH [33] and arXiv, and various high-profile blogs such as those by Gowers and Tao.
- Data Management tools: MySQL.

However, there is clear need for improved tools to foster mathematical collaborations. For example, one can envision an "interactive whiteboard" facility, so two researchers at different institutions can work on a mathematical example as if they were standing side-by-side. Some discussions with computer scientists and engineers may be in order here to design a practical yet effective system.

4.3 Best practices and standards

Given the increasing impact of experimental mathematics, some at the workshop recommended that the field establish some "best practices" and other guidance, both for those working in experimental mathematics, and for those using their results. This would be particularly valuable given the increasing role that experimental mathematicians are playing as participants in larger collaborations: for example detailed Maple computations played a key role in the Polymath 8, project on "Bounded gaps between primes [23]. An analogy might be drawn with "Technology readiness levels, a set of criteria used in defence and other applications to assess the maturity of emerging technologies [37]

- Career advice for young researchers intending to pursue professional work in experimental mathematics — recommended course background, best ways to meet others in the field, how to do real publishable research, etc.
- College-level education: Recommended curricula for courses in experimental mathematics; recommended textbooks, etc.
- Attracting students (high school and college) to experimental mathematics: Successful outreach methods, motivating students to learn, etc.
- Instilling general "experimental" skills: Building intuition, knowing how to check that one is wrong, right (or "not-right," "not-wrong').
- Professional practice for the deployment of experimental mathematics: reproducibility, credit, attribution and provenance of experimental results
- Setting expectations so that collaborators working with experimental mathematics experts had a realistic understanding of the power and limitations of experimental techniques

Along this line, several at the workshop hoped that the U.S. National Institute for Standards and Technology (NIST) might play a role to further develop such practices and standards.

This would build on the strong role that NIST has already played in the area of special functions, where NIST's activities go back at least to the Abramowitz-Stegun *Handbook of Mathematical Functions* in 1964 [1]. More recently, the NIST Digital Library of Mathematical Functions (`http://dlmf.nist.gov`) [11] and the NIST Digital Repository of Mathematical Formulae (`http://gw32.iu.xsede.org/index.php`) [12] continue this tradition. NIST has also been active in the development of algorithms and mathematical software for special functions, with numerous journal publications in the field.

If an organization such as NIST is to take this role, numerous questions arise, such as what resources and types of standards would be established, what participation by others in the mathematical community would be required, what scale of software infrastructure would be required, what specific projects should be embarked upon, and, of course, how any work in this area will be paid for. Clearly this needs some additional discussion. Several at the workshop expressed interest in participating in such discussions, perhaps at a future workshop.

5 Conclusion

In summary, the workshop participants agreed that there is considerable potential for near- and long-term progress in the field, but that significant challenges also lie ahead. Challenges include (a) instilling a culture of reproducibility and reliability in computational experiments, (b) developing new algorithms and software appropriate for execution on highly-parallel, multi-core and many-core platforms, (c) adapting and porting existing software to these platforms, (d) fostering collaboration and interactions with numerous other allied disciplines (especially including the larger high-performance scientific computing community), and (e) providing outreach and career advice to prospective researchers.

More generally, there are high-level questions to be considered in the field. For example, much of the published work to date in experimental mathematics has focused on a few fields that are particularly amenable to computational exploration — combinatorics, classical number theory, analytic number theory, geometry, groups, rings and fields, etc. How can we expand the scope of questions that have been examined with these methodologies?

The discussions on education raised several interesting questions. Can we foster greater interest in the experimental mathematics field by promoting the field as a way to build practical computer literacy and computational science skills? After all, most of the students who we may teach about experimental mathematics will end up in other fields, e.g., science, engineering, technology and finance. Can we craft or stimulate development of instructional material targeted to such persons?

Similarly, the list of allied disciplines above raises the question of whether there are in fact other disciplines, perhaps distinct from some of the groups that experimental mathematicians have traditionally collaborated with, which have the potential for particularly productive interactions?

All of this also raises the question of how all this research work can be paid for. A significant portion of published research in the field of mathematics (pure and applied) not been specifically funded — it has been 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 better with governmental funding agencies to find ways to provide this funding. Perhaps this can more easily be done if projects can be done in collaboration with others, particularly in computer science or other fields that heretofore have been somewhat more generously funded.

References

[1] Milton Abramowitz and Irene A. Stegun, ed., *Handbook of Mathematical Functions*, Dover Publications, New York, 1965.

[2] Sylwester Arabas, Michael R. Bareford, Ian P. Gent, Benjamin M. Gorman, Masih Hajiarabderkani, Tristan Henderson, Luke Hutton, Alexander Konovalov, Lars Kotthoff, Ciaran McCreesh, Ruma R. Paul, Karen E. J. Petrie, Abdul Razaq, Danil Reijsbergen, "An open and reproducible paper on openness and reproducibility of papers in computational science," 9 Aug 2014. Available at `http://arxiv.org/abs/1408.2123`.

[3] F. Aragon, D. H. Bailey, J.M. Borwein and P.B. Borwein, "Walking on real numbers." *Mathematical Intelligencer*. **35** (1) (2013), 42–60. DOI: `http://link.springer.com/article/10.1007\%2Fs00283-012-9340-x`.

[4] F. Aragon, J.M. Borwein, and M. Tam, "Douglas-Rachford feasibility methods for matrix completion problems." *ANZIAM Journal*. Online, 28 pages, July 2014. Available at `http://arxiv.org/abs/1308.4243`.

[5] D.H. Bailey and J.M. Borwein, "Exploratory Experimentation and Computation." *Notices of the AMS*. **58** (2011), 1410–1419.

[6] D.H. Bailey and J.M. Borwein,"Computation as an ontological game changer: The impact of modern mathematical computation tools on theontology of mathematics." Chapter to appear in *Mathematics, Substance and Surmise: Views on the Meaning and Ontology of Mathematics.* Ernest and Philip J. Davis editors. Springer-Verlag, to appear 2015.

[7] D.H. Bailey, J.M. Borwein, M. Lopez de Prado, and Qiji Zhu, "Pseudo mathematics and financial charlatanism: Backtest overfitting and out-of-sample over performance." *Notices of the AMS*. **61** May (2014), 458–471. Online at `http://www.ams.org/notices/201405/rnoti-p458.pdf`.

[8] Jonathan M. Borwein and David H. Bailey, *Mathematics by Experiment: Plausible Reasoning in the 21st Century*, AK Peters, Natick, MA, 2008.

[9] Jonathan M. Borwein and Keith Devlin, *The Computer As Crucible: An Introduction to Experimental Mathematics*, AK Peters/CRC Press, 2008.

[10] Richard P. Brent, "Recent Progress and Prospects for Integer Factorisation Algorithms." Computing and Combinatorics. Lecture Notes in Computer Science Volume 1858, 2000, pp. 3–22.

[11] "NIST Digital Library of Mathematical Functions," National Institute for Standards and Technology, 2014, available at `http://dlmf.nist.gov`.

[12] Howard Cohl, Marje McClain, Bonita Saunders, Moritz Schubotz, "Outgrowths of the DLMF Project: Part 2: NIST Digital Repository of Mathematical Formulae," 21 Jul 2014, available at `http://icerm.brown.edu/materials/Slides/tw-14-5/Outgrowths_of_the_Digital_Library_of_Mathematical_Functions_Project_]_Howard_Cohl_and_Daniel_Lozier,_National_Institute_of_Standards_and_Technology.pdf`.

[13] David H. Bailey, Yozo Hida, Xiaoye S. Li and Brandon Thompson, "ARPREC: An arbitrary precision computation package," Sept 2002, available at `http://www.davidhbailey.com/dhbpapers/arprec.pdf`.

[14] David H. Bailey, Roberto Barrio, and Jonathan M. Borwein, "High precision computation: Mathematical physics and dynamics," *Applied Mathematics and Computation*, vol. 218 (2012), pg. 10106–10121.

[15] Yozo Hida, Xiaoye S. Li and David H. Bailey, "Algorithms for quad-double precision floating point arithmetic," 15th IEEE Symposium on Computer Arithmetic, IEEE Computer Society, 2001, pg. 155-162, available at `http://www.davidhbailey.com/dhbpapers/arith15.pdf`.

[16] "The GNU MPFR Library," not dated but viewed 31 Jul 2014, available at http://www.mpfr.org.

[17] J. Dongarra, "LAPACK," not dated but viewed 31 Jul 2014, available at http://www.netlib.org/lapack.

[18] J. Dongarra, "LINPACK," not dated but viewed 31 Jul 2014, available at http://www.netlib.org/linpack.

[19] "GitHub: Build software better, together," not dated but viewed 31 Jul 2014, available at https://github.com.

[20] Gonthier, G., Asperti, A., Avigad, J., Bertot, Y., Cohen, C., Garillot, F., Le Roux, S., Mahboubi, A., O'Connor, R., Ould Biha, S., Pasca, I., Rideau, L., Solovyev, A., Tassi, E., and Théry, L. "A machine-checked proof of the odd order theorem." In ITP 2013 (2013), vol. 7998 of LNCS, pp. 163–179.

[21] Jean-Baptiste Michel1, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K. Gray, The Google Books Team, Joseph P. Pickett1, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, Steven Pinker, Martin A. Nowak, Erez Lieberman Aiden, "Quantitative Analysis of Culture Using Millions of Digitized Books," *Science*, vol. 331, no. 6014 (14 January 2014), pp. 176-182.

[22] President's Council of Advisors on Science and Technology. Report to the President and Congress "Designing a digital future: federally funded research and development in networking and information technology", December 2010, available at http://www.whitehouse.gov/sites/default/files/microsites/ostp/pcast-nitrd-report-2010.pdf

[23] "The polymath blog," not dated but viewed 31 Jul 2014, available at http://polymathprojects.org.

[24] "The Flyspeck Project," not dated but viewed 31 Jul 2014, available at https://code.google.com/p/flyspeck/wiki/FlyspeckFactSheet.

[25] "Sage," not dated but viewed 31 Jul 2014, available at http://www.sagemath.org.

[26] Victoria Stodden, David H. Bailey, Jonathan M. Borwein, Randall J. LeVeque, William Rider and William Stein, "Setting the default to reproducible: Reproducibility in computational and experimental mathematics," 2 Feb 2013, available at http://www.davidhbailey.com/dhbpapers/icerm-report.pdf.

[27] "MathLibre," not dated but viewed 10 Aug 2014, available at http://www.mathlibre.org/site.

[28] Eric Weisstein, "Wolfram Mathworld," 30 Jul 2014, available at http://mathworld.wolfram.com.

[29] "PETSc: A portable, extensible, toolkit for scientific computation," not dated but viewed 31 Jul 2014, available at http://www.mcs.anl.gov/petsc.

[30] Jim Pitman and Clifford Lynch, "Planning a 21st century global library for mathematics research," *Notices of the AMS*, vol. 61 (2014), pp. 776-777.

[31] "PlanetMath: Math for the people, by the people," not dated but viewed 31 Jul 2014, available at http://planetmath.org.

[32] Xioaye S. Li, "SuperLU," not dated but viewed 31 Jul 2014, available at http://crd-legacy.lbl.gov/~xiaoye/SuperLU.

[33] "swMATH: An information service for mathematical software," not dated but viewed 10 Aug 2014, available at `http://www.swmath.org`.

[34] Neil J. A. Sloane, "The Online Encyclopedia of Integer Sequences (OEIS)," not dated but viewed 31 Jul 2014, available at `http://www.oeis.org`.

[35] "The R Project for Statistical Computing," not dated but viewed 31 Jul 2014, available at `http://www.r-project.org`.

[36] "The Top 500 List," not dated but viewed 31 Jul 2014, available at `http://www.top500.org`.

[37] Technology Readiness Assessment (TRA) Guidance, Prepared by the Assistant Secretary of Defense for Research and Engineering (ASD(R&E)), dated 13 May 2011, available at `http://www.acq.osd.mil/chieftechnologist/publications/docs/TRA2011.pdf`.

[38] Marko Petkovsek, Herbert S. Wilf, Doron Zeilberger, $A = B$, AK Peters/CRC Press, Natick, MA 1996.

[39] David E. Womble. "Is there a Moore's law for algorithms?". Presented at Salishan 2004. Available at `http://www.lanl.gov/orgs/hpc/salishan/salishan2004/womble.pdf`

[40] Ed Yong, "Nobel laureate challenges psychologists to clean up their act," *Nature*, doi:10.1038/nature.2012.11535, 03 October 2012. Available at `http://www.nature.com/news/nobel-laureate-challenges-psychologists-to-clean-up-their-act-1.11535`.