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

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

August 2, 2014

1 Executive summary

"Experimental mathematics" has emerged in the past 25 years or so to become a competing paradigm for research in the mathematical sciences. So what exactly is "experimental mathematics"? While several definitions have been offered (e.g., [6]), perhaps the most succinct definition is 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. [7, pg. 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 advance the state of the art in mathematical research.

While the overall approach and philosophy of experimental mathematics has not changed greatly in the past 25 years, its techniques, scale and sociology have

*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
‡University of Helsinki, Helsinki 00100, Finland olga.caprotti@helsinki.fi.
§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.

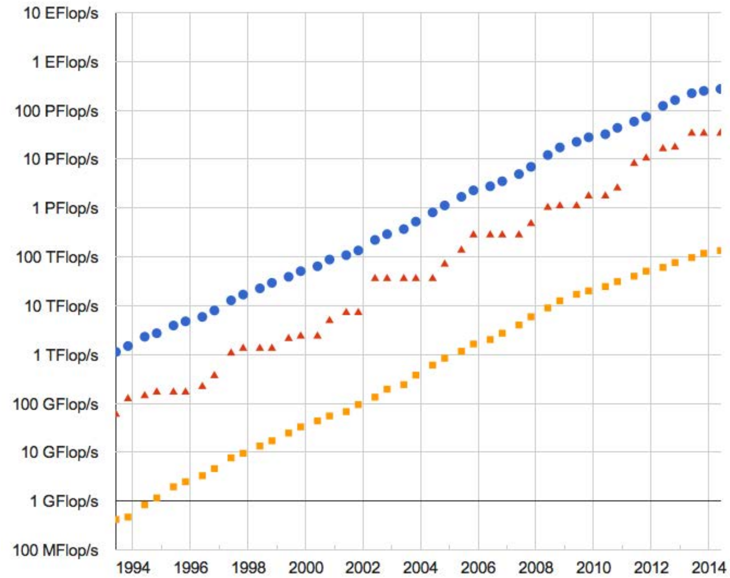

Figure 1: Performance of the Top 500 computers: Red = #1 system; orange = #500 system; blue = sum of #1 through #500.

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. Of particular significance is the development of highly parallel and multi-core computing technology, which allows researchers to harness literally hundreds of thousands of individual computing elements, and to manipulate comparably prodigious amounts of data. The overall impact of these developments is perhaps best illustrated by the Top 500 list [27], which has tracked the world's 500 most powerful scientific computers since 1993 — see Figure 1. Note strikingly, that aggregate performance in 1994 was surpassed by the lowest ranked machine a decade later!

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 [3]. These include the open-source *Sage* [19], 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.

Yet many challenges remain as researchers press the envelope in mathematical computing. To begin with, 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* in the field.

Another challenge is how to ensure reliability and reproducibility of computed results, e.g., to ensure that the results of floating-point computations are numerically reliable and reproducible on other platforms. A related challenge is how to manage the exploding scale of symbolic expressions. 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.

Also, researchers in the field who are developing large software tools now must face *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.

Finally, recent developments in the field have highlighted *the challenge of addressing changing sociological and community issues.* To begin with, 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* [17], 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. Another issue here is the incorporation of *formal methods* into the experimental mathematics enterprise.

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 true 21st century computer-savvy students eager to press forward with these tools. But this is not the first time that technology has promised to reinvent mathematical education. Thus, *a final challenge is to provide evidence-based rationales for experimental mathematics in the classroom* (at elementary and advanced levels).

A workshop held at the Institute for Computational and Experimental Research in Mathematics (ICERM), July 21–25, 2014, explored many of these emerging challenges. This report summarizes much of the workshop findings.

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 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 [20]. 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 adopted in other scientific disciplines.

Some in the workshop questioned whether rock-solid reproducibility is always needed. After all, mathematical equivalence is often sufficient, but even recognizing equality is formally undecidable. Also, in some instances efforts to ensure reproducibility are not as important, since it is possible to directly certify a result (for example, by applying the *Wilf-Zeilberger* algorithm [28]. Some related tools for inequalities exist in *Flyspeck* project [18], although additional work is needed to make them faster. Formal proofs will become more and more useful here, as tools to prove certificates. They are unlikely, however, to be broadly accessible within the next few decades.

Nonetheless, reproducibility is often essential for debugging, if nothing else — if results vary from run to run, how can a researcher be certain that he/she did not introduce a bug in the process?

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 [11]. 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.

2.2 Validity checking

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. Thus, the development of reliable validity checks, or other means for ensuring a very high level of reliability in experimental results, is as important as finding highly efficient algorithms for discovering results in the first place. Such considerations are likely to be even more important as the field presses forward with larger and larger computations on highly parallel computer systems. Some examples of validity checks that are now routine in the field the following:

1. When new formulas are found using "PSLQ" *integer relation algorithms* [6, 3], 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).

2. Very high precision calculations typically employ *fast Fourier transforms* (FFTs) to accelerate multiplication operations. The final inverse complex-to-real FFT values should be very close to whole numbers. If all are close to integer values, this is a good validity check that the FFT-convolution process has worked properly, and that the results are reliable.

3. When mathematical constants (e.g., $\pi, e, \log 2$, etc.) are computed to very high precision, it is now customary to check the results by an independent computation, say using a different algorithm. If the results of the two such

computations agree except perhaps for a few trailing digits, then this is strong evidence that both computations are likely correct.

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. Indeed, large datasets exhibit a quasi-linear complexity that potentially magnifies the chance for error. 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* [25] is an excellent model for how this can be done effectively over a period of decades..

The recently initiated *Digital Repository of Mathematical Formulae* (DRMF) [9] project ties specific LaTeX character sequences to well-defined mathematical objects. The DRMF, like the D*igital Library of Mathematical Functions* (DLMF) [8], is being developed at the National Institute of Standards and Technology (NIST). One suggestion mentioned at the workshop is to incorporate semi-automatic visual, numeric and symbolic validity checks into formulas that are entered DRMF. For example, a formula could be spot-checked for validity by numerically evaluating both sides of the equation at some pseudorandomly chosen values of the arguments.

2.3 Standards

One step that would greatly help foster a culture of reproducibility and greater reliability in the field is to establish some standards:

- Field-specific standards for reports on computational experiments (and insisting that researchers report hunts that found nothing or tests that failed).
- Standards for test suites of mathematical software. Such test suites exist for many different areas but with no coherence.
- 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?
- Standards for preserving data in a rapidly changing software/database environment. Some examples already exist where the data is available (e.g., **Chen-Kauers-Singer, Bostan et al., Almkvist-Zudilin, ...-TBA BRUNO)** may prove useful.
- Standards for transferring data and symbolic expressions between different software environments. For example, it would be very nice to be able to

exchange data and expressions between *Maple* and *Mathematica*.[1] Sometimes this is a challenge even within the same computer algebra system.

One challenge is to agree on a set or sets of standards. A more difficult challenge will be to get the experimental mathematics community to adopt such standards. But we have to start somewhere, so perhaps some individual research groups can adopt one or more of these standards and then report their experience at future workshops and conferences.

3 Computer systems and software

As mentioned above, the power of computer systems used in the field has changed dramatically over the past 25 years. Just as importantly, the range of 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. These include the open-source *Sage*, numerous high-level computational libraries, such as SuperLU for scalable direct factorization of matrices and PETSc for solution of large-scale nonlinear algebraic systems, to ADCIRC for solving time dependent, free surface circulation and transport problems. 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 perform specific tasks, such as very high-precision computation, or which implement a specific algorithm such as PSLQ or the tanh-sinh quadrature algorithm. Currently available packages of this sort include QD, which performs double-double computation (approximately 31-digit accuracy) [12], ARPREC [10] and MPFR [13], which perform arbitrary precision computation, LINPACK [15] and LAPACK [14], which perform linear algebra and matrix computations, ParInt for parallel numerical integration, and R [26], which performs many sophisticated statistical calculations.

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 [1].[2] In this workshop, a talk on non-convex feasibility demonstrated the usefulness of

[1] Currently *Maple* provides a *Mathematica* translator.

[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/`.

visualizing the convergence of projection and reflection algorithms for this problem [2]. Tools for generation, description and manipulation of advanced graphic output are also lacking.

3.1 Challenges ahead

However, the emergence of powerful new computing platforms, particularly those with highly parallel, multi-core or many-core designs, presents daunting challenges to both 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.

Also, 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) which allows 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. For example, a study by the National Science Foundation (NSF) found that the median persistence of software developed in NSF-funded educational projects was nine months [**?**]. 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.[3]

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.

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, graphic 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

[3] We have told our students for decades to document their code. This time we really mean it.

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. It also has much to offer current thinking on the nature of Mathematics [4].

Even within the field of experimental mathematics *per se*, there is a large international, multi-discipline community. Some approach the field from the vantage of pure mathematics; some approach it from computer science; some approach it from within symbolic computing (i.e., Groebner bases, etc.); others approach it from education; 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 sub-communities will need to work with the others.

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

1. Other specialties of mathematics — new applications include computational geometry and topology. More traditional experimental work engages computational number theory and group theory.

2. Computer science — many working in experimental mathematics already come from computer science backgrounds or are familiar with theoretical computer science.

3. Computational science (i.e., researchers whose work in various "applied" disciplines, but whose work centers on issues of large-scale, highly parallel scientific computation).

4. Probability and statistics (both theoretical and practical).

5. Physics (including related disciplines such as astronomy, astrophysics and cosmology).

6. Chemistry (especially computational chemistry and materials science).

7. Engineering (many fields of engineering now involve sophisticated mathematical algorithms and large-scale computations).

8. Biology (including biostatistics, namely statistical methods specifically applied to genomics and biomedicine).

9. Medicine and biochemistry (including advanced geometric imaging and visualization techniques, as well as “data mining” of biomedical data).

10. Social science — economics, psychology, sociology, linguistics, anthropology, history all now include significant mathematical computational research.

11. Finance and investment — there has been an explosion of activity in this arena recently, with many top mathematicians and computer scientists working in the field.[4]

The breadth of these disciplines 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 mind a related challenge, namely to foster respect for these other disciplines and to expect respect respect in return. All too often, promising collaborations of this sort founder on the problems of this sort.

4.1 Tools to foster collaboration

The Internal 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 jointly written dozens of papers, even though they have never been in the same country, let alone the same institution, and for at least 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. Many others reports similar stories.

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 [3] and:

1. 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.[5]

[4] As with the overselling of ‘the Quants’ before the great economic recession of 2007-08, mathematicians have an obligation to confront the bad practice of mathematics, both in finance and in other fields such as the social sciences [5].

[5] Tools like Xoom or FaceTime which are restricted to given hardware or vendors further complicate the issue.

2. Data sharing tools (high bandwidth is important): Email, Globus toolkit, Google docs, Dropbox.

3. Software sharing tools: GitHub [16], SVN.

4. Mathematical community online resources and tools: Mathworld [21], PolyMath [17], (Gowers) Blogs, Math Overflow, PlanetMath [23], DLMF [8], DRMF [9], OEIS [25], arXiv.

5. Data Management tools: MySQL.

6. Computer algebra packages: *Mathematica*, *Maple*, *MATLAB*, Sage [19], R.

7. Arbitrary precision libraries: ARPREC [10], GMP [13], MPFR [13], QD [12].

8. Other commonly used numerical libraries: ADCIRC, LAPACK [14], LINPACK [15], PETSc [22], SuperLU [24].

9. Experimental math tools incorporating specific algorithms: PSLQ and tanh-sinh quadrature (available with ARPREC and QD); *Mathematica* includes the `PadeApproximant` tool (for finding rational function approximations) and the `RootApproximant` tool (for finding the polynomial satisfied by an floating-point value), while *Maple* includes a constant recognition facility (since version 9.5), as well as `numapprox[pade]` and `PolynomialTools [MinimalPolynomial]`.[6]

4.2 Best practices

Given the expanding role of collaborations (and the expanding presence of experimental mathematics in general), some at the workshop recommended that the field establish some "best practices" and other guidance for researchers in different roles:

1. 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.

2. College-level education: Recommended curricula for courses in in experimental mathematics; recommended textbooks, etc.

3. Attracting students (high school and college) to experimental mathematics: Successful outreach methods, motivating students to learn, etc.

4. Instilling general "experimental" skills: Building intuition, knowing how to check that one is wrong, right (or "not-right," "not-wrong").

[6] There are numerous other examples such as *Maple*'s `_Dexp` adaptive doubly exponential integration method that a user may or may not be aware of.

There are several objectives here - TBA URSALA

Along this line, several at the workshop hoped that the U.S. National Institute for Standards and Technology (NIST), which has already sponsored projects such as the DLMF and the DRMF, might play a role to further develop standards for the field of experimental mathematics. Among the questions are 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.

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, mutli-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 a few fields that are particularly amenable to computational exploration — 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. It is well-known in the field of mathematics that most published research has not been specifically funded — it has been done by academic mathematicians as they have time, as a part of their teaching work. But some of the work described above, particularly that which involves substantial software development and maintenance, cannot be done so informally.[7]

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] 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.

[2] 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.

[3] D.H. Bailey and J.M. Borwein, "Exploratory Experimentation and Computation." *Notices of the AMS.* **58** (10) (2011), 1410–1419. Reprinted in Book # **??**. E-published October 13, 2011 at http://www.ams.org/notices/201110/rtx111001410p.pdf. Published in translation in *Mathematical Advances in Translation* (Chinese Academy of Science), June 2012.

[4] 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.

[5] 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.

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

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

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

[9] 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.

[10] 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.

[7] Nor does a royalty model work as it has for traditional publications. The development costs are too great and the academic reward too small.

[11] 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.

[12] 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`.

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

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

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

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

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

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

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

[20] 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`.

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

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

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

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

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

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

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

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