Open Source Mathematical Software

Mathematical software has greatly contributed to mathematical research, enabling exciting advances in mathematics and providing extensive data for conjectures. Perhaps three of the most well-known applications of computation to mathematical research are the resolution of the *four-color conjecture* by Appel and Haken in 1976 (though it is now reproven with less need for computer verification by N. Robertson, D. P. Sanders, P. D. Seymour and R. Thomas), Thomas Hales's proof of *Kepler's conjecture*, and the formulation of the *Birch and Swinnerton-Dyer conjecture*, which grew out of extensive numerical computation.

Open source software, such as TEX, Mozilla Firefox, and Linux has had a profound effect on computing during the last decade, and we hope that open source mathematical software will have a similar positive impact on mathematics.

> I think we need a symbolic standard to make computer manipulations easier to document and verify. And with all due respect to the free market, perhaps we should not be dependent on commercial software here. An open source project could, perhaps, find better answers to the obvious problems such as availability, bugs, backward compatibility, platform independence, standard libraries, etc. One can learn from the success of TEX and more specialized software like Macaulay2. I do hope that funding agencies are looking into this.
>
> —*Andrei Okounkov, 2006 Fields medalist* (see "Interviews with three Fields medalists" *Notices of the AMS*, **54**(3) (2007), 405–410).

The term *open source* is defined at `http://www.opensource.org/`, but basically it means anyone (including commercial companies or the defense department) should be able to inspect open source software, modify it, and share it with others.

One key difference between mathematical theorems and software is that theorems require little maintenance, whereas *mathematical software requires substantial and potentially expensive maintenance* (bug fixes, updates when algorithms or languages change, etc.). Mathematical research usually generates no direct revenue for researchers, and likewise open source mathematical software is free to share and extend, so it rarely generates revenue. Volunteer effort, donations, and financial support from the NSF and other organizations is thus critical to the success of open source mathematical software.

There is a proof in the article by Campbell et al. in *The Atlas of Finite Groups—Ten Years On* (1998) that describes how many separate software packages were "easily used" to deduce various mathematical facts—no code is given, and some of the programs are proprietary software that runs only on hardware many years out of date. Such proofs may become increasingly common in mathematics if something isn't done to reverse this trend.

Suppose Jane is a well-known mathematician who announces she has proved a theorem. We probably will believe her, but she knows that she will be required to produce a proof if requested. However, suppose now Jane says a theorem is true based partly on the results of software. The closest we can reasonably hope to get to a rigorous proof (without new ideas) is the open inspection and ability to use all the computer code on which the result depends. If the program is proprietary, this is not possible. We have every right to be distrustful, not only due to a vague distrust of computers but because even the best programmers regularly make mistakes.

If one reads the proof of Jane's theorem in hopes of extending her ideas or applying them in a new context, it is limiting to not have access to the inner workings of the software on which Jane's result builds. For example, consider the following quote from the Mathematica tutorial[1]:

> Particularly in more advanced applications of Mathematica, it may sometimes seem worthwhile to try to analyze internal algorithms in order to predict which way of doing a given computation will be the most efficient. [...] But most often the analyses will not be worthwhile. For the internals of Mathematica are quite complicated, and even given a basic description of the algorithm used for a particular purpose, it is usually extremely difficult to reach a reliable conclusion about how the detailed implementation of this algorithm will actually behave in particular circumstances.

No journal would make a statement like the above about the proofs of the theorems they publish. Increasingly, proprietary software and the algorithms used are an essential part of mathematical proofs. To quote J. Neubüser, "*with this situation two of the most basic rules of conduct in mathematics are violated: In mathematics information is passed on free of charge and everything is laid open for checking.*"

Full disclosure: The second author started a new mathematics software system in 2005 called SAGE (see `www.sagemath.org`), which combines Python, GAP, Singular, PARI, Maxima, SciPy, etc. with several hundred thousand lines of new code. SAGE receives contributions from many mathematicians worldwide that synthesize the latest algorithms from a broad range of topics into a comprehensive toolkit for mathematical research.

—*David Joyner*
U. S. Naval Academy, Annapolis
`wdj@usna.edu`
—*William Stein*
University of Washington, Seattle
`wstein@u.washington.edu`

[1] `http://reference.wolfram.com/mathematica/tutorial/WhyYouDoNotUsuallyNeedToKnowAboutInternals.html`