DRAFT: A Review of Mathematica Version 6.0

Richard Fateman
Computer Science Division, EECS Department
University of California
Berkeley, CA, USA

October 13, 2007

Mathematica version 6.0, released in May, 2007, is the latest edition of the flagship product of Wolfram Research (WRI). Mathematica is probably the computer algebra system (CAS) in widest distribution. Readers of this review may wonder (a) Should I upgrade from version 5.2? or (b) Should I "side"grade from another CAS, or an interactive numerical package? or (c) Is this finally the time to learn what a CAS can do for me?

We provide suggested answers in the last paragraph of the review.

You can read about the new features in this program from the Wolfram web site (wolfram.com) and in fact you can look online at the complete user documentation for the program. The program package, as well as its earlier versions, includes a large array of algorithmic symbolic, numeric, and graphical tools. Also, as in the past, great attention has been paid to display technology. WRI deserves credit for understanding the importance of notation, even if it is somewhat idiosyncratic. Each new version comes with further enhancements to the presentation of the system, some of it being "eye-candy" but other features show thought and application of graphical design principles.

We offer this caution: the claims of the vendor that this version is a "dramatic breakthrough" should be viewed critically. While some of these notes may highlight significant improvements, it would be prudent to view such claims as marketing fluff. Some of the claimed Mathematica 6.0 novelties or "inventions" are well known to (for example) users of the Macintosh graphing calculator, Tcl/TK or even competing CAS, whose graphing programs were interactive a decade ago.

The program is bulky. Checking on my Windows XP computer, I see version 6.0 more than twice the size of 5.1. This new version occupies about 0.9 gigabytes of disk space occupying some 12,000 files (vs. 0.42 gigabytes in 2,800 files in version 5.1). WRI claims a growth from 1.5 million to 2.5 million lines of code. If nothing else, it is impressive in a sense similar to that of a cruise ship: a large and complex object which floats. The new version has added some additional decks and connecting passages. Some of the code represents the further integration of numerical, symbolic, and communication facilities. This may be appealing to users who would make Mathematica some kind of central interchange for web-based facilities; this would possibly make sense if the such a relied on substantial sophisticated scientific computing, and the licensing of the installed Mathematica permitted it.

Skilled technical labor has gone into some very specific niches, and only a perusal of the documentation will tell if your favorite activities are now better supported. Here we have space to mention a few:

- The numerical integration package is now far more elaborate, and appears quite versatile. While there is a substantial effort to "do the right thing automatically", unfortunately a user with a ticklish problem may still need to figure out how to use the many options for explicit choice of method, monitoring, or accuracy.

- As another niche improvement, there is a much enlarged set of options for pseudo-random number generation.

To continue the cruise ship analogy though, a ship is a simulation that you are building on solid ground, but the simulation is necessarily incomplete. If you go too far below deck you may notice a vibration from the engines, and if you sail for too long without pausing, the kitchen will run out of shrimp, steak, and even pasta; eventually the ship will run out of fuel.

The simulation that Mathematica is doing mathematics has analogous problems, some from the finiteness of computers, but more significantly from programming features that veer away from what critics (including this reviewer) think should be part of mathematical computation.

Most design errors from the earlier editions persist; indeed, some defenders of Mathematica seem to view the inevitable peculiar results of these decisions as correct: they are replacing long-standing problems of mathematics notation or convention with a new understanding. This is an alternative world view in which an answer is said to be correct if it is "generic" and ignores sigularities. This is annoying if, as sometimes happens, the most important aspect is a singularity. Worse is the view offered by fans of the program (not necessarily WRI employees!) that an answer is correct because it is *what Mathematica computes*, and any view to the contrary is the fault of the user (perhaps for misunderstanding the documentation.)

Fortunately, for routine symbolic computations, and for many esoteric ones, the program is surprisingly clever in obtaining neat solutions, using, as necessary, an armamentum of special functions from algebra, analysis, combinatorics. These include pieces of constructive mathematics largely untaught to mathematicians, but often well known in the small world of computer algebra specialists (including builders of similar rival CAS).

Problems noted in our review[1] published more than a decade ago are still part of the design. Regardless of whether WRI "meant to do it" or not, these produce problems for users. Although an important reason for using a CAS is to enable the manipulation of *symbolic* expressions or programs, WRI has continued to expend effort on numerical floating-point computations so as to rival certain numeric "only" systems. We have, in the past, advocating allowing a CAS to grow to include anything that might be useful, and this might include LAPACK, FORTRAN compilers, or theorem provers. However, the inclusion of numerics from well-written public libraries does not overcome the problems inherent in programs written in Mathematica's native arithmetic. In this context we provide the following:

Example: Consider the function `f[x_,a_] := (x^2+a)/(2*x)`. Given a number a and a guess x for $\sqrt{a}$, then $f(x, a)$ will produce a better approximation. Indeed, iterating to find $\sqrt{4}$ starting with the guess $x = 1$ can be done via `{x=1.0; Do[x=f[x,4],{5}];x}`. This returns a list containing the answer 2. Iterating 60 times instead of 5 times still produces 2, as expected. Since Mathematica allows high-precision floats, one would expect this to work also `{x = 1.00000000000000000; Do[x = f[x, 4], {60}]; x}`. Surprisingly, in this case Mathematica returns `Indeterminate`. If one cuts off the iteration at 58, the answer is more bizarre.

`{x = 1.00000000000000000; Do[x = f[x, 4], {58}]; x}`

gives `ComplexInfinity`.

These kinds of unexpected results from using Mathematica's significance arithmetic were pointed out in my 1992 review. A good argument can be made that the conventional standards for floating-point arithmetic, the ideas taught in most numerical analysis courses about absolute and relative error, roundoff and truncation, are inadequate for a CAS. After all, a CAS can devote more than 64 bits for a number, and often does. We dispute, however, that Mathematica's significance arithmetic is a better idea: it certainly has its share of surprises, and they are nearly unique to this program.

A careful programmer schooled in the particulars of the treatment of `Accuracy` in Mathematica can program around these phenomena, just as one can, with sufficient advance knowledge, avoid booking a stateroom on the cruise ship that is too close to the engines. One pays a price, and one also hopes that the

[1] `http://www.cs.berkeley.edu/~ fateman/papers/mmarev.pdf`

elaborate proprietary programs written by WRI staff are carefully constructed to avoid the traps of its own foundational mechanisms.

Returning again to version 6.0: as in any "improved" program, there are aspects for which a user may favor the older version as easier or initially more familiar. In fact the book describing Mathematica is apparently no longer a book, but an online help system, and this is different from the version 5.0 version. The previous help system provided a kind of hierarchical browser, links to book sections, demos, tours, and a master index. In the new system one can select a term in a notebook and then open the "help" system to find the main page (if any) for that term; the thought seems to be that including a variety of "see also" links absolves the authors from organizing the documentation further. The tutorials and guides may fulfill the need of users seeking a broader view of the program, but in a small sampling they seem to suffer from (one of) the banes of CAS documentation: it is impossible to find a single "right level" at which to address the audience. The reader may be a high-school student or a seasoned mathematician, or a computer scientist. What should be explained and what can be assumed?

WRI has extended its facilities for dynamic interaction, though using them may require considerable attention to detail. Given the claims for the Mathematica language's effectiveness, it is curious that, according to WRI statistics, so much of the application commands seem to be written in C. It is a good thing that the trend seems to be toward using Mathematica's language – a necessity for programmers outside Wolfram Research.

I understand that most viewers of Mathematica are most impressed by its graphics, plotting 3-d surfaces in dramatic colors: I apologize for not including a snazzy picture in this review, but judging a computer algebra system by its graphics is like judging an automobile by the comfort of its headrests and the number of cupholders. I personally admire a nicely designed cupholder, but...

Perhaps this reviewer is too sensitive to "attitude" such as this explanation taken from the on-line version 6 tutorial "Why You Do Not Usually Need to Know about Internals":

> 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. And there are indeed occasionally major improvements that you will be able to make in specific computations as a result of such analyses.
>
> 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.

This passage illustrates the (to me) annoying use of sentence fragments. For expressing complex ideas. In bursts. Here the fragments are not even short!

Now look at what it is saying: Mathematica is so complicated that it defies your understanding, so don't bother. Trust us: the answer is right and is being reached in an efficient way.

If it were true, this kind of approach would still be patronizing. Some readers may have taken courses in computer algebra algorithms (yes, there are even several texts now), some know about modern algebra, constructive algorithms in analysis, etc. What this passage suggest is that even if you find a Mathematica computational mathematics application that seems to be reliable, the relevant parts may be broken by slighly different circumstances, or may change upon the introduction of new algorithms, possibly without warning.

It is clear that for an expert in CAS, finding bugs in Mathematica, by exploring the borders of difficult (or silly) problems is still not difficult, and can apparently be automated to some extent. The number and type of bugs known to the vendor is (no surprise) not public, and so some critics have posted their finding on internet newsgroups. It is likely that the domain (essentially all of mathematics) provides such a large cross-section for bugs, and the evident mathematical incorrectness of some behavior too easily observed. It is not apparent that Mathematica is especially unreliable compared to other programs of its size and ambition.

There are few programs quite so ambitious, perhaps providing a justification for its unusual, even for shrink-wrapped software, licenses. Mathematica goes beyond trying to limit its liability, and includes the following disclaimer, "WRI does not recommend the use of the product for applications in which errors and/or omissions could threaten life, injury, or significant loss."

The online newsgroup `comp.soft-sys.math.mathematica` has many comments about the program and its earlier versions, including reports of bugs. However, the newsgroup is moderated, and the moderator apparently filters some criticisms, as well as any messages which explicitly compare Mathematica to another program. The newsgroup `sci.math.symbolic` has the advantages and disadvantages of being unmoderated; it may be a useful resource for persons seeking to collect comments on this and other CAS. Recent messages include examples where version 5.2 provides a correct answer to obscure integration problems, while version 6.0 provides either a much bulkier answer, or one which is wrong. Some of the errors reported include dependencies where the re-ordering of apparently independent commands results in errors: it seems that Mathematica 6.0, in attempting to save time, stores intermediate results for re-use; these may be either wrong or corrupted when recalled later.

While we are mentioning the internet, we should also note that a web site `www.symbolicnet.org` attempts to provide links to various additional resources on CAS. These lead to web sites of competitors as well as to free programs similar to Mathematica.

In the interests of full disclosure, I feel obliged to mention that I worked on the Macsyma CAS dating back to 1968 at MIT. (It is now most easily available as a free version called "Maxima" from `maxima.sourceforge.net`). I consulted briefly with Stephen Wolfram on a predecessor program to Mathematica, namely SMP. Some time ago I published (a free, open source) parser for the Mathematica (version 3) language, including a pattern matcher and rudimentary simplifier, to enable researchers or others to experiment with alternatives to the Mathematica commercial syntax and semantics. For this, WRI threatened to sue me, but eventually dropped the matter. Also, in the interests of full disclosure, I have had fruitful interchanges with the technical staff at WRI, some of whom are excellent; I suspect that some of my suggestions (but certainly not all) have been incorporated in Mathematica.

In answer to the three questions posed in the first paragraph: If you are a committed Mathematica user then upgrading to the latest version probably makes sense; however, do not delete the copy of the older version until you are sure version 6.0 works for you in all respects, including especially the help system.

If you are quite content with an alternative CAS program, you might not find the transition of Mathematica from 5.2 to 6.0 much of an incentive to switch to this product. This reviewer cannot, on the other hand, deny the possibility that one of the new features in version 6.0 unique among CAS fulfills some requirement of your daily scientific computing work. Finally, if you have never used a computer algebra system you are missing out on a technology that broadens the domain of applicability of computing to any discipline that uses mathematics: the sciences and engineering, but also business, education, etc. Perusing Mathematica's documentation online may be an eye-opener: a smorgasbord of capabilities. In spite of various shortfalls perceived by this reviewer and other experts, and the inherent hazards of depending on such a large and complicated system, the vast majority of users of such systems will tread the well-worn and quite reliable paths of computation and never encounter any problems in Mathematica (or other competing CAS).