

Software Review

Michael Wester

Cotopaxi, Albuquerque, NM, USA

Wolfram Mathematica 6 for Linux, by Wolfram Research, Inc., 2007

Fifteen years ago, I reviewed Mathematica 2 [1]. I thought that the system then had many very positive and a few very negative aspects. In 1999, I compared Mathematica 3 with six other general purpose computer algebra systems (CASs) [2]. All the systems were quirky (they still are!), but Mathematica performed reasonably well overall and could do some interesting examples, although it still lacked some important capabilities such as being able to make assumptions about variables. Time passes and software changes. Version 6 is now out, a much fancier system with many more functions and features, and an enhanced graphical user interface.

For those readers not that familiar with CASs, an important resource to know about is Symbolic Net (<http://www.symbolicnet.org/>), which provides a variety of information including links to many of the known systems: general purpose and specialized, commercial, shareware, GNU General Public Licensed, free, etc. Some of these CASs (including some of the non-commercial ones like Axiom and Maxima) are quite sophisticated.

I had asked for a review copy of Mathematica that ran on Linux. I received a large box that was mostly empty, containing a CD and a few thin documents. Not long after I registered my copy, I received an email offering a free update to version 6.0.1, a nice gesture as my review copy did not include Premier Service (a year's subscription normally accompanies a purchase of Mathematica), which provides for software updates.

Installation on my Linux server went well. However, when I tried to access Mathematica remotely via ssh and X-windows, I got an error about missing fonts. I visited the Documentation Center at <http://reference.wolfram.com/>, and after some hunting around, I found that I needed to make Mathematica fonts available on any Unix system I used that accessed a remote Mathematica installation. The easiest way to do this was to make a local copy of 4 Mb worth of fonts and then is-

sue a couple of xset commands, thankfully detailed in the document that I had found. Actually, this problem with Unix fonts has been present in previous versions – I had forgotten until it bit me. Using Linux Mathematica remotely from an SGI or a Sun also resulted in other minor funniness, so I tried to work mostly from a remote Linux system.

Back in 1992, I was particularly impressed with Mathematica's documentation. The *Mathematica Book* was very well laid out and easy to use. Alas, with version 6, no written manual is provided or even exists – all system documentation is online, even the installation instructions. The reason for this, I was told, is that the system has become so large and complex that a manual would be huge and unwieldy. In Vernor Vinge's 2006 novel *Rainbows End*, the net savvy folk of the future wear personal communication technology and much of society disdains the printed (as opposed to the electronic) word as it is difficult to search efficiently. However, one character defends the inexact art of library stack searching by noting that "I found answers to questions that I had never thought to ask", often taking him in new valuable directions. Thus I feel that there continues to be a need for a properly indexed printed summary of the core language and functionality of Mathematica, and if Wolfram will not provide it, others will fill the gap. (I recently talked to one of the Mathematica developers, and he told me that he frequently refers to his Mathematica 5 Book [3].) The online documentation does have a table of contents and an alphabetical index of function names, but no topic index, although there is a text search feature which was helpful.

As I find completely online documentation tedious, I ended up printing out a large amount of material. This required a great deal of mouse clicking, including expanding notebook sections and using File->Save As... to convert the discussion into PostScript/PDF (the remote server was not connected to any local printers available to me and using File->Print to print to a file was somewhat erratic – I had to uncheck and check 'Print to file' in the Print dialog box to obtain a 'Browse' button and then a file was not always pro-

duced when I hit ‘Print’). File-> Save As... did not remember its last location, an annoying feature which led to even more clicking.

Mathematica pioneered the notebook interface, which has since been emulated by many other systems, and Mathematica 6 has substantially extended its capabilities. A particularly cool feature I learned about from my ‘First Five Minutes with Mathematica’ is that a selected piece of output will automatically convert to traditional math notation by pressing Control-Shift-T; Control-Shift-N will revert back to Mathematica’s StandardForm. A second cool feature is that Edit-> Copy As... allows one to copy text in a variety of formats, including L^AT_EX and MathML. Intelligent syntax coloring has also been introduced, which is a helpful feature for program development.

Indeed, Mathematica 6 seems to have incorporated a large variety of different data formats from various application fields (e.g., audio, chemistry, genomics, geometry modeling, geospatial modeling, image compression, spreadsheets, web, etc.). In some cases, sample data is included as well. One example is `ChemicalData`, consisting of 18181 compounds with a large number of properties (nomenclature, geometrical, topological, physical, etc.) included. Of course, the public PubChem database (<http://pubchem.ncbi.nlm.nih.gov/>) has about 11.6 million compounds, so one can only do toy problems with `ChemicalData`, but nonetheless, it can serve as a useful educational tool.

A cute command that has been introduced with version 6 is `Manipulate`. `Manipulate[command, {p, pmin, pmax}]` allows one, for example, to create a plot or compute an expression that depends on a parameter, then change the value of that parameter by moving a slider, allowing for easy interactive experimentation. Unfortunately, there are no labels on the slider that indicate the minimum and maximum values of the parameter (and no way that I could discern to easily add some), however, if one clicks the little plus sign to the right of the slider, the current parameter value and various buttons for animating with respect to the parameter appear.

Speaking of plots, Mathematica has finally added the ability to rotate 3D plots in place using a mouse. Maple and Matlab introduced this feature long ago (and both systems have considerably more plot features that can be changed easily via interactive menus). As it has traditionally been easy to interconvert between 3D, contour and density plots within Mathematica, it would be nice to use this ability to systematically

change one’s perspective between different plot types, but in version 6 this does not seem to be possible (or at least trivial). At least, plots can be saved in a great variety of graphics formats.

Version 6 has also introduced `Exclusions`, `RegionPlot` and `RegionPlot3D`, which allow the user to exclude subregions from a plot or numerical integral, or specify the plot region with a predicate. The methodology used is numerical, and so one must be a little careful as certain small regions which satisfy the predicate may be missed (and the `RegionPlot` documentation explicitly states that points and lines certainly will be missed). Nonetheless, this is an extremely useful set of commands that will undoubtedly be welcomed by the user community.

The so-called Standard Packages that have been provided with previous versions of Mathematica have substantially changed. There is a Compatibility Guide, however, the information in it is sometimes a bit sparse. Much functionality has been moved into the kernel. Some functionality also has been lost or obscured. For example, it is no longer easy to compute Fourier series with a non-unity period, so one has to fiddle around to reproduce this functionality (not hard, but irksome since it used to be present).

The add-on package `AuthorTools`, which contained handy functions for processing notebooks for publication, is no longer supported. Developer support, in general, could be better in this new version. For example, there is currently no way to integrate extra help information into the Documentation Center.

Mathematica can smoothly perform some tough calculations. It is especially facile with special functions. For example, Mathematica recognizes that $2^{1-z}\Gamma(z) \times \zeta(z) \cos \frac{z\pi}{2} - \pi^2 \zeta(1-z)$ is equivalent to zero [Gradshteyn and Ryzhik 9.535(3)], and version 6 was able to correctly compute the following integral that comes from doing a two loop Feynman diagram for a QCD problem:

```
Integrate[x^2*PolyLog[3, 1/(x + 1)],
          {x, 0, 1}]
```

On the other hand, Mathematica 6 can also miss some seemingly obvious problems. Simplifying the indeterminate expression

$$\frac{\tan^2 x + 1 - \sec^2 x}{\sin^2 x + \cos^2 x - 1}$$

produces zero! (Mathematica 3 realized something funny was going on and returned `Indeterminate`.)

Differentiating or integrating this expression with respect to x also produces zero. Interestingly, simplifying the reciprocal of the above expression is detected as a problem.

A more subtle problem occurs when computing

$$\int \frac{1}{4 + 3 \cos x + 4 \sin x} dx.$$

The result produced is indeed an antiderivative, however, numerically evaluating the antiderivative at any real x yields a complex value (the imaginary part is at a constant offset of $\frac{i\pi}{3}$, even though the integrand is real for any real x).

Mathematica's programming language is basically functional, resembling Lisp in style. Pattern matching is an important part of the programming paradigm. If one has only used procedural languages previously, programming in Mathematica takes some getting used to. For example, the function application $f(x)$ is represented by `f[x]`, $e^{i\pi}$ by `Exp[I*Pi]`, etc., and many other features are reminiscent of the C language. All this can be both good and bad.

One can write some very compact rules that define an operation. For example, defining one's own toy differentiation operator is easily done with pattern matching:

```
difrules = {
  dif[y_ + z_, x_] ->
    dif[y, x] + dif[z, x],
  dif[y_ * z_, x_] ->
    dif[y, x]*z + y*dif[z, x],
  dif[a_, x_] /; FreeQ[a, x] -> 0,
  dif[x_, x_] -> 1,
  dif[w_^n_, x_] /; !FreeQ[w, x] ->
    n*w^(n - 1)*dif[w, x],
  dif[Exp[n_], x_] ->
    Exp[n]*dif[n, x] }
dif[expr, u] //. difrules
```

Here, rules for derivatives of sums, products, constants, the variable in question, powers and exponentials are all one liners written in an obvious fashion. One can also place rules directly on an operator (with care).

On the other hand, matrix subscripting can be obscure. For example, the Mathematica 6 equivalent of the Matlab assignment

```
B= [A(1:3,2:4), ...
     A([1,2,4],[3,1,4]); ...
     A, [A(1:2,3:4); A([4,1],[3,2])]]
```

with A a 4×4 matrix, is

```
B = Join[
  Join[A[[1 ;; 3, 2 ;; 4]],
        A[{{1,2,4},{3,1,4}}], 2],
  Join[A, Join[A[{{1,2},{3,4}}],
                A[{{4,1},{3,2}}]]],
        2] // MatrixForm
```

Mathematica considers a non-sparse matrix as simply a list of lists, placing a significant syntax burden on the user in situations such as the above.

Mathematica (like most other CASs) has a large variety of transformation/simplification commands. In the beginning, there were just a few like `Expand`, `Factor`, `TrigExpand`, `TrigFactor` and `Simplify`, but that was not sufficient, so `FullSimplify` was added soon after. A plethora of other transformation/simplification commands are now available, including `ComplexExpand`, `FunctionExpand`, `LogicalExpand`, `PiecewiseExpand`, `PowerExpand`, `ToRadicals`, etc., each with its own assumptions and capabilities. This can be daunting to the new user, especially as it not always obvious what names to look for (typing `?*Expand*` found many of the above). `FullSimplify` does seem to work reasonably well in many circumstances, however, so a good strategy when encountering a messy expression is often to try `Simplify` and if that doesn't help, `FullSimplify`, if one isn't sure what else to do immediately. Note that `FullSimplify` can be substantially slower than `Simplify` as it tries many more transformations.

A very important transformation useful to those who work with numerical codes is converting symbolic entities into compiled languages such as C or Fortran (code generation). Here, Mathematica lags behind the curve. Consider a simple function to iteratively compute factorials:

```
myfact[n_] := Module[{i, f},
  If[n < 0, Return["undefined"],
    f = 1; Do[f = i * f, {i, 2, n}];
    f];
```

The following output (edited to fit) is produced when attempting to convert `myfact[n]` (an expression) into Fortran:

```
In[2] := FortranForm[myfact[n]]
```

```
Out[2] // FortranForm =
  If(n.lt.0, Return("undefined"),
  - CompoundExpression(Set(f$14,1),
  - Do(Set(f$14,i$14*f$14),
  - List(i$14,2,n)), f$14))
```

Other systems and packages, such as Maple and the GENTRAN family, can actually directly translate functions as well as expressions into immediately usable code.

Another interesting experiment to try is

```
Do[Print[Timing[Factor[Expand[
  Sum[a[i], {i, 1, n}]10]]]],
  {n, 1, 10}]
```

For $n = 7$ and 8 , the calculations took almost 70 seconds/51+ minutes, respectively, on my 2.7 GHz system. In contrast, Maple 9 on a 1.7 GHz machine with the same amount of memory (1 Gb) did the the whole series through 10 in less than two minutes total.

As I mentioned at the beginning, Mathematica did not originally include assumptions on variables, but now there are many ways to make them: via the function `Assuming`, the option `Assumptions` for various functions, and the global variable `$Assumptions`. Assumptions can also be specified directly as a second argument to commands like `Refine`, `Simplify` and `FullSimplify`, and the domain of the variables to be solved for as a third argument to `Reduce`. For example,

```
Reduce[x^4 + 9 == y^2, {x, y},
  Integers]
```

solves the given Diophantine equation.

In summary, Mathematica 6 is a significant advancement over previous versions. It is also a significantly more complicated piece of software, so one has to take care to use it correctly and effectively. Mathematica 6 can solve some hard problems, but it can

also make mistakes or produce results in cumbersome forms that are difficult to use (as is true for other CASs as well, of course). Sometimes an answer is produced quickly, but multivariate polynomial factorization can be rather slow. The online documentation is helpful, but needs to be supplemented with a real topical index. New users should also check out the many books written about using Mathematica to fully comprehend the features of the user language. Understanding how to write simple functions, patterns and rules is the key to using the system for non-trivial applications. <http://demonstrations.wolfram.com/> is another resource with a large number of open source examples.

Acknowledgement

I wish to thank Daniel Lichtblau for his aid in arranging a copy of the Mathematica 5 Book to be sent to me, and Rolf Mertig of GluonVision for some helpful discussions.

References

- [1] M. Wester, Book review of *Mathematica: A System for Doing Mathematics by Computer*, 2nd edn, by Stephen Wolfram, *SIAM Review* **34** (3) (1992), 519–522.
- [2] M. Wester, A critique of the mathematical abilities of CA systems, in: *Computer Algebra Systems: A Practical Guide*, M.J. Wester, ed., John Wiley & Sons, Chichester, UK, 1999, pp. 25–60.
- [3] S. Wolfram, *The Mathematica Book*, 5th edn, Wolfram Media, Champaign, IL, 2003.