Introduction to SageMath

Štěpán Starosta

FIT, Czech Technical University in Prague

SageDays 72, Prague

Outline

Brief overview of SageMath

Key Points

Taken from http://www.sagemath.org/library-press.html

Brief overview of SageMath

1. SageMath aims to provide everything mathematicians, researchers and students need to do their calculations. The basic concept is to combine many established software packages under one umbrella. Even more than that, it provides powerful and unique algorithms in it's own library. SageMath's mission is to "**create a viable free open source alternative to Magma, Maple, Mathematica and Matlab**".

Brief overview of SageMath

2. *Open Source*: SageMath is built upon open-source software and it is fully open-source by itself. It is **free** to use worldwide for private, commercial, governmental, etc. use. Its license is the well known GPL and everyone is allowed to download it, install it on an unlimited number of computers and redistribute copies.

Brief overview of SageMath

3. *Open Development*: SageMath loves curious students and researchers to examine its source code and it is possible to understand how each calculation is done. SageMath fosters a community of developers and encourages them to take part in its development. A vital community of people not only using but also participating in development is key to a **healthy ecosystem** in the field of mathematical software. Additionally, SageMath utilizes the scientific method of peer-review to double check each line of new source code in addition to its strict testing policies to ensure a certain level of quality.

Brief overview of SageMath

4. *Leverages Existing Software*: SageMath does not reinvent the wheel for every known calculation. When possible, it uses existing tools to solve the problem and combine all of them in one unique interface. This concept not only exposes software packages to a wider audience, but also helps to increase the quality by submitting bugs upstream.

Brief overview of SageMath

5. SageMath uses Python as its “glue language” to interface with all its components. Python is also SageMath’s primary interface language and hence SageMath does not invent a new programming language as other mathematical software systems do. Python is well established among research communities and makes interfacing even less complicated.

Brief overview of SageMath

6. *Interface*: There are three basic ways how the user can interact with SageMath: a web-interface, accessible through a web-browser while SageMath is running on your local machine or on a server, a rich command-line interface and as a Python library. Additionally, for example it is possible to embed SageMath in LaTeX documents.

Few comments on open (mathematical) software

You can read Sylow's Theorem and its proof in Huppert's book in the library without even buying the book and then you can use Sylow's Theorem for the rest of your life free of charge, but for many computer algebra systems license fees have to be paid regularly. In order to protect what you pay for, you do not get the source code
(. . .)
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.

1993, Joachim Neubüser
founder of GAP (1986)

[J. NEUBÜSER: *An invitation to computational group theory.* In: C. M. Campbell, T. C. Hurley, E. F. Robertson, S. J. Tobin and J. J. Ward, eds. Groups '93 Galway / St Andrews: Galway 1993, Volume 2. Cambridge University Press, 1995.]

“Why You Do Not Usually Need to Know about Internals”

You should realize at the outset that while knowing about the internals of the Wolfram System may be of intellectual interest, it is usually much less important in practice than you might at first suppose.
(. . .)
For the internals of the Wolfram System are quite complicated, (. . .), it is usually
extremely difficult to reach a reliable conclusion about how the detailed implementation of this algorithm will actually behave in particular circumstances

http://reference.wolfram.com/language/tutorial/WhyYouDoNotUsuallyNeedToKnowAboutInternals.html

“Why You Do Not Usually Need to Know about Internals”

The Misfortunes of a Trio of Mathematicians Using Computer Algebra Systems. Can We Trust in Them? by *Antonio J. Durán, Mario Pérez, and Juan L. Varona*

http://www.ams.org/notices/201410/rnoti-p1249.pdf

The witchcraft frame

I think, fundamentally, open source does tend to be more stable software. It's the right way to do things. I compare it to science vs. witchcraft. In science, the whole system builds on people looking at other people's results and building on top of them. In witchcraft, somebody had a small secret and guarded it – but never allowed others to really understand it and build on it.
Traditional software is like witchcraft. In history, witchcraft just died out. The same will happen in software. When problems get serious enough, you can't have one person or one company guarding their secrets. You have to have everybody share in knowledge.

2004, Linus Torvalds

http://bazaarmodel.net/phorum/read.php?1,7899

Components

http://www.sagemath.org/links-components.html

What's more: SageMath combines other software:

```
sage: from sage.misc.citation import
   get_systems
sage: get_systems("integrate(cos(x^2),x)")
['PARI', 'MPFI', 'FLINT', 'MPFR', 'ginac', '
   GMP', 'Maxima', 'NTL']
```

SageMath interfaces

Interfaces

1. Command line
2. Notebook (local/Internet)
3. Cell server
4. SageMathCloud

Other notable features:

1. SageTeX
2. Interact and animations
3. Cython
4. Parallel computing

Thank you