Open source software and Sage[1]

http://www.sagemath.org/

David Joyner

Math Dept
USNA, Annapolis, MD

May, 2009

[1] Presentation for NSF CDI workshop

Outline

Computers in Mathematics

Many major math problems have been solved with the help of computers:

- Four color problem
- Kepler's conjecture,
- the formulation of the Birch and Swinnerton-Dyer conjecture,
- ...

There are too many examples to list - the impact has been profound.

More and more published papers are using computers as part of their research. Unfortunately, *many of these also use expensive proprietary computer algebra systems where key details cannot be verified but must be believed on faith.*

Computers in Mathematics, 2

To quote J. Neubüser, the "father" of GAP,

... 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.*

What standards should be applied to allow computational research to enter into rigorous mathematics?

Open Source in mathematics

From a recent interview published in the AMS Notices:

> *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*

Open Source in mathematics, 2

The current system has several drawback:

- Commercial mathematics software is analogous to a math journal that only publishes the statements of (presumed) theorems but not their proofs!
- Implementations of mathematical algorithms whose development was funded by taxpayers, is sometimes absorbed into commercial programs.
- If mathematical research were published in exactly this way, the NSF would likely refuse to fund it.

Advantages of Sage

Possible solution: Sage.

Advantages:

- **All** Sage (Python/Cython) code is *refereed*.
- Based on Python, a popular OO language,
- Free and open source
- efficient and fast (when Cython/C/C++ are used)
- robust interfaces to many other computer algebra systems (Maxima, Pari, GAP, R, Magma, ..., some of which are distributed with Sage)
- easy to compile on many OS platforms (Linux, Mac, Solaris; under Windows, Sage runs in a virtual machine but should have a native port this year...)
- ...

Advantages of Sage

1. Sage is a serious general purpose CAS that uses a **mainstream programming language** (Python).
2. Sage allows you to use Maxima, Singular, etc., all **together**.
3. Sage has **more functionality out of the box** than any other open source mathematics software.
4. Sage has a large, **active**, and well rounded **developer community**: `sage-devel` mailing list has over **850** subscribers (1130 for sage-support), working very hard on everything from highly optimized arithmetic, to symbolic computation, to plotting fuctionality. It averages 25 messages a day.
5. Sage **development is done in the open**. You can read about why all decision are made, have input into decisions, see a list of every change anybody has made, etc. This is **totally different than the situation with expensive proprietary CA systems.**

Advantages of Sage

1. With Sage **everything is open source** and the system is setup to strongly encourage looking at code. In fact, arbitrary modifications and redistribution of every single line must be allowed.

2. Just as mathematicians gain a deeper understanding of a theorem by carefully reading or at least skimming the proof, people who do computations should be able to understand how the calculations work by reading documented source code.
   Sage should be **well-documented**.

3. Give proper credit to the authors of all packages which Sage includes.

Pexpect and Pseudotty's

- **Pseudotty:** A device which appears to an application program as an ordinary terminal but which is *in fact* connected to a different process. Pseudo-ttys have a slave half and a control half.

  `maxima_console()` **brings up a Maxima prompt in** Sage

- **Pexpect:** *makes Python a better tool for controlling other applications.* (`pexpect.sourceforge.net`)
  Pexpect is a pure Python module for spawning child applications; controlling them; and responding to expected patterns in their output.

  `maxima.eval('maximacommand')` **sends** `'maximacommand'` **to Maxima**

Core Components of Sage: All Bases are Covered

Basic Arithmetic	**GMP, PARI, NTL**
Command Line	**IPython**
Commutative algebra	**Singular**
Graphical Interface	**SAGE Notebook**
Graphics	**jmol, Matplotlib, Tachyon**
Group theory and combinatorics	**GAP**
Interpreted programming language	**Python**
Networking	**Twisted**
Numerical computation	**SciPy, GSL, etc.**
Source control system	**Mercurial**
Symbolic computation, calculus	**sympy**, **Maxima**

There are about 90 standard packages and 41 optional packages total (plus about 58 experimental packages).

To be a component of Sage, the software must be: **free, open source, robust, high quality, and portable**. Nothing else should be included in the core Sage package.

Interfaces – You can use "anything" from Sage

Continue to use your favorite programs and code from within SAGE (orange/red systems are included standard with Sage):

- Sage includes (mostly pseudo-tty) interfaces to
  - **SciPy**, **Maxima**, **R**, **GAP**, **GP/PARI**, **Singular**, **LinBox**, etc,
  - Kash, Macaulay2, etc (available as "optional" packages),
  - Magma, Maple, Mathematica, (commercial),
  - Octave, gnuplot (open source but separate),

  etc.
- Get access to **100%** of the functionality of the other systems via interfaces. (But there is some overhead.)
- Get tab completion and online help.

The Sage Notebook

Interesting FACT: Most people polled **vastly prefer** using a good GUI for interacting with math software, if available.

Sage **has one.**

1. The Sage Notebook – An "AJAX application" like Google maps or gmail: lots of `CSS`, `Javascript`, and `XMLHTTPRequest`.
2. Uses Python's `Twisted web2` web server to provide a GUI.
3. Client/server model which works over network.
4. A very usable and robust version done.
5. Online version: `http://www.sagenb.com/`

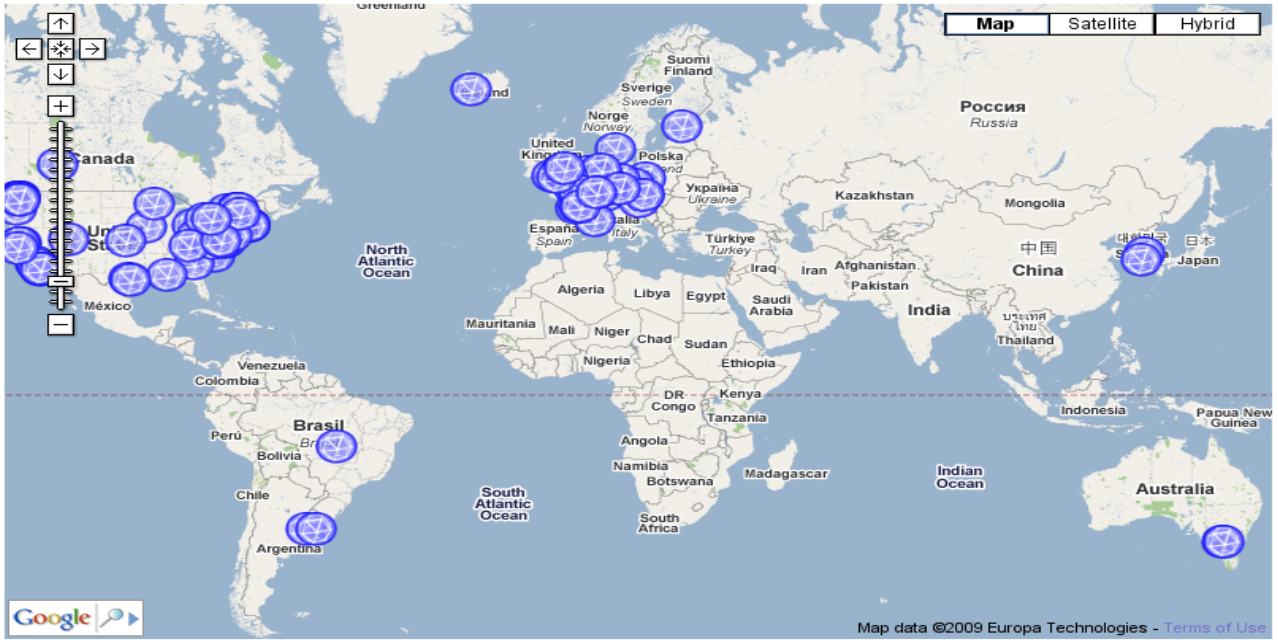
Who is Writing Sage?

Professors, Postdocs, Graduate students, Undergraduates, Retired tech workers (and **You**?) - about 60% of current developers are students. The barrier to becoming a more active Sage developer is very low.

Sage developers around the world

This is a map of all contributors to the Sage project. There are currently 143 contributors in 86 different places from all around the world.

Map Zoom: Earth - USA (UW, West, East) - Europe - Asia - S. America - Australia



Sage is a *community*

Sage users are a **community** - please join us!

SAGE site:

`http://www.sagemath.org/` - plus lots of mirror sites.

`http://www.sagenb.com/` - where you can use Sage online.

`http://sage.math.washington.edu/home/` - where Sage developers have accounts and directories (most are viewable from the www).

`http://wiki.sagemath.org/` - Sage wiki.