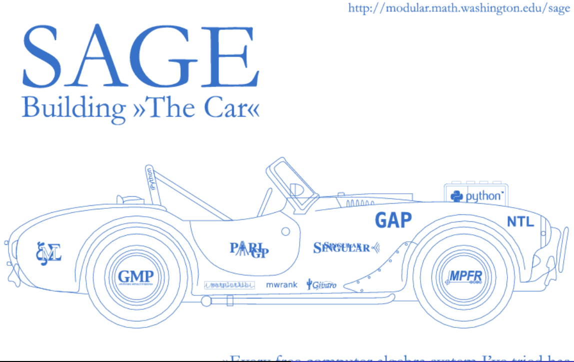
SAGE: Software for Algebra and Geometry Experimentation

William Stein

January 11, 2007, Undergraduate Math Sciences Seminar


http://modular.math.washington.edu/sage
SAGE
Building »The Car«
python
python
GAP
NTL
PARI GP
SINGULAR
GMP
MPFR
matplotlib
mwrank
Givaro
»Every free computer algebra system I've tried has

SAGE: The Goal

1. Create a free open source **viable alternative** to the **proprietary expensive** systems Maple, Mathematica, MATLAB, and Magma.

Nobody has figured out how to do this and not sold out.

I intend to do this or die trying. Soon. **You can help.**

(Remind me to record this lecture if I'm not already!)

The Typical Math Software Lifecycle

The big math software MATLAB, Maple, Mathematica, MuPad, REDUCE, Axiom, and Macsyma have mostly followed this lifecycle:

1. Government or public supported **academic research project**.
2. Form a **commercial company** (lawsuits).
3. **Annoy** mathematicians (software patents, lawsuits, etc.):

   > *When Mathematica first came out, academics were used to the idea that any software they cared about was free–at least to them. I thought there was a serious market for Mathematica in the academic market. We had to dig in our heels and say this is going to cost you real money. People got very upset about that.*
   > *– Stephen Wolfram (1996)*

4. **Profit!!!** Take a lot of money from universities and students – tens of millions per year; enough to support a serious company...

Macsyma, REDUCE and MuPad not so healthy; sometimes *old* academic version emerges as open source.

The GPL and “open source”

1. Mathematicians became **aware** of the GPL software license after MATLAB, Maple, Mathematica, MuPad, REDUCE, Axiom, and Macsyma were all well into production.
2. PARI, GAP, Maxima, Singular, and Macaulay2 are all very **original exciting math software projects** that are developed at universities and available for **free**.
3. All are licensed under the **GPL: Gnu Public License**.
4. They will **never be proprietary** expensive software because their license forbids it and their copyright is widely distributed.
5. They are **the foundation for SAGE**, along with Python and some libraries.

Does Open Source Matter for Math Research?

“You can read Sylow’s Theorem and its proof in Huppert’s book in the library [...] 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 [...]. You press buttons and you get answers in the same way as you get the bright pictures from your television set but you cannot control how they were made in either case.

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**. Not applying these rules to computer algebra systems that are made for mathematical research [...] means **moving in a most undesirable direction**. Most important: Can we expect somebody to believe a result of a program that he is not allowed to see? Moreover: Do we really want to charge colleagues in Moldava several years of their salary for a computer algebra system?”

– J. Neubüser (1993) (he started GAP in 1986).

By the way, you can find Huppert’s book *Endliche Gruppen I* on Amazon.com – it costs **$158** from Springer.

Example: Maple

There is a new PDE solver that will be in **Maple**, written for free by a mathematician. My student found out about it at a conference, and wanted to create something similar for SAGE. Someone remarked *“I imagine this would be quite difficult but don’t see that "copying" would be an issue.”* **This opinion about Maple is common...** We wrote to Maple to be sure; they said that once anyone includes their routines in Maple it becomes **illegal to use them as a basis for doing anything anywhere else ever**.

```
Reproducing and redistribution of Maple code is a violation of
the license agreement.  this is a direct violation of the EULA
[...] Without the express written permission of Maplesoft,
Licensee shall not, and shall not permit any Third Party to:
(a) reproduce, transmit, modify, adapt, translate or create
any derivative work of, any part of the Software, in whole
or in part ...
(b) reverse engineer, disassemble, or decompile the Software,
create derivative works based on the Software, or otherwise
attempt to gain access to its method of operation or source;
Sincerely, Maplesoft Technical Support
```

SAGE makes new things possible

The **free** and very **web-browser friendly** nature of SAGE is creating new possibility for mathematical education.

1. Web sites can use SAGE as a backend for arbitrary mathematics calculations.
2. Licenses for all the commercial programs forbid this, no matter how much you pay them.

For example...

Date: Mon, 19 Dec 2005 16:54:09 -0800
From: "John Cannon" <john@maths.usyd.edu.au>
Subject: Magma calculator
William,

This is to formally advise you that your permission to run a general-purpose calculator based on Magma ends on Dec 31, 2005. This was originally set up at your request so students in your courses at Harvard could have easy access to Magma.

Your making a calculator available has been an interesting experiment and we plan to continue it in modified form out of Sydney (at least for for the time being).

Please confirm receipt of this letter.
Wishing you a happy Christmas,

John

Background: From HECKE 0.1 to SAGE 2.0

- **1997–1999:** HECKE – my free C++ program for **modular forms** (I wrote an interpreter for it).
- **1999–2004:** I wrote $> 25,000$ lines of Magma code.
- **Feb 2004:** Wanted something with a broader scope (not just algebraic research mathematics), a modern interpreter, and **open source** (!).
- **Feb 2005:** I got job offers with **tenure** – **SAGE 0.1**.
- **Feb 2006: SAGE Days 1** workshop – **SAGE 1.0**.
- **June 2006: High school** workshop – Notebook.
- **August 2006: MSRI Grad student** workshop.
- **October 2006: SAGE Days 2** workshop.
- **Now:** Push for SAGE 2.0 by end of month!

What is SAGE?

SAGE is:

1. **A Distribution** of free open source math software. 75MB source tarball that builds self-contained.
2. **New Readable Code** that fill in gaps in functionality; implement new algorithms.
3. **A Unified Mainstream Interface** to math software: to **Magma**, **Macaulay2**, Singular, **Maple**, MATLAB, Mathematica, Axiom, etc.

Who is Writing SAGE?

Contributors Include: Martin Albrecht, Tom Boothby, Robert Bradshaw, Iftikhar Burhanuddin, Craig Citro, Alex Clemesha, John Cremona, Didier Deshommes, David Harvey, Naqi Jaffery, David Joyner, Josh Kantor, Kiran Kedlaya, David Kirkby, Emily Kirkman, David Kohel, Jon Hanke, Bill Hart, Robert Miller, Bobby Moretti, Gregg Musiker, Bill Page, Fernando Perez, Yi Qiang, David Roe, Michael Rubinstein, Nathan Ryan, Kyle Schalm, Steven Sivek, Jaap Spies, Gonzalo Tornaria, Justin Walker, Mark Watkins, Joe Weening, Joe Wetherell, ...

- **Undergraduates**: have many **extremely interesting** ideas; superb at researching available free software; good programmers.
- **Many graduate students**: excellent at implementing optimized code and finding fast algorithms.
- **Faculty and computer professionals**: general direction, great writing, and quality control.

SAGE Days 2 at UW: Coding Sprints...

Bobby Moretti (UW undergrad), Robert Miller (UW grad), David Harvey (Harvard grad), Joel Mohler (grad), David Joyner (USNA), Bill page (Axiom).

Upcoming SAGE-related Workshops I'm Organizing

- **Sage Days 2.5 – Parallel Computation Workshop** at MSRI, Jan 29–Feb 2, 2007. Yi Qiang (UW undergrad!) is an invited speaker.
- **SAGE Days 3** at IPAM (in LA) Feb 17–21, 2007.

Software
SAGE
Experimentation
Geometry
SAGE DAYS 3
February 17-21, 2007 UCLA
http://modular.math.washington.edu/sage/

- **AIM** workshop, databases in SAGE, July 30-Aug 3, 2007.

The SAGE Website

The SAGE Website

1. **Website:** `http://modular.math.washington.edu/sage`
2. **Free online SAGE notebook:** `http://sage.math.washington.edu:8100`
3. **Documentation:** Tutorial, Install Guide, Programming Guide, Reference Manual, Constructions.
4. **Targeted Platforms:** OS X, Linux, and Windows (Cygwin).
5. **Mailing Lists:** sage-devel (hundreds of messages/month), sage-announce, sage-forum, sage-support, and **sage-uw**.
6. **Wiki:** `http://sage.math.washington.edu:9001/`
7. **Bug Tracker:** `http://sage.math.washington.edu:9002/sage_trac`
8. **IRC Chatroom:** `#sage-dev` on irc.freenode.net

A Distribution

1. A Distribution

Basic Arithmetic	**GMP, NTL, MPFR, PARI**
Command Line	**IPython**
Commutative algebra	**Singular** (libcf, libfactory)
Database	**ZODB**, Python Pickles
Graphical Interface	**SAGE Notebook, jsmath**
Graphics	**Matplotlib, Tachyon, GD**
Group theory and combinatorics	**GAP**
Graph theory	**Networkx**
Interactive programming language	**Python** (mainstream !!!)
Networking	**Twisted**
Numerical computation	**GSL, Numpy, etc.**
Symbolic computation, calculus	**Maxima**

All core components are **free and open source** (mostly GPL'd). You may **read the code** and **change anything** in SAGE or any of the core libraries it includes, and redistribute the result.

The SAGE Library – new code we've written

2. New Code

Python and Pyrex code — **designed to be readable**:

```
algebras    edu         lfunctions  monoids          sets
categories  ext         libs        plot             structure
coding      functions   matrix      quadratic_forms  tests
combinat    geometry    misc        rings
crypto      groups      modular     schemes
databases   interfaces  modules     server
```

```
99500 UNIQUE Source Code Lines (including docstrings):
$ cat */*.py */*/*.py */*/*/*.py */*.pyx \
                */*/*.pyx */*/*.pyx |sort |uniq | wc -l
99500
11567 UNIQUE Input Documentation Examples:
$ cat */*.py */*/*.py */*/*/*.py */*.pyx \
          */*/*.pyx */*/*.pyx |sort|uniq|grep "sage:" | wc -l
11567
```

A Unified Interface

3. A Unified Interface

- SAGE **interfaces to**: Axiom, GAP, GP/PARI, Kash, Macaulay2, Magma, Maple, Mathematica, MATLAB, Maxima, Octave, Singular, etc.
- Wide range of **functionality**.
- Unified **command completion and help**.

Some UW Undergraduates Who Have Contributed Substantially to SAGE

1. **Tom Boothby** – The SAGE Notebook, powering algorithms, elliptic curves
2. **Emily Kirkman** – some linear algebra; lots of work on the SAGE graph theory package.
3. **Bobby Moretti** – SAGE Calculus, some group theory, packaging, chroot jails.
4. **Yi Qiang** – Distributed SAGE

How You Can Help

1. Become a **SAGE Developer**:
   - There are employment and research opportunities with me. (Apply for VIGRE or Mary Gates funding, use my grant, volunteer, etc.)
   - Apply to the UW summer REU – I will advise some SAGE development.
2. **Use SAGE** (and report any bugs you find).
3. **Tell people** about SAGE. Put a link on **your website**.
4. Take **my undergrad number theory course** this spring.
5. I am working on creating a general math software course for next year. **Express interest** in the creation of such a course.
6. Come by my office (423) with your laptop and **install SAGE**.