SAGE: Software for Algebra and Geometry Experimentation

William Stein

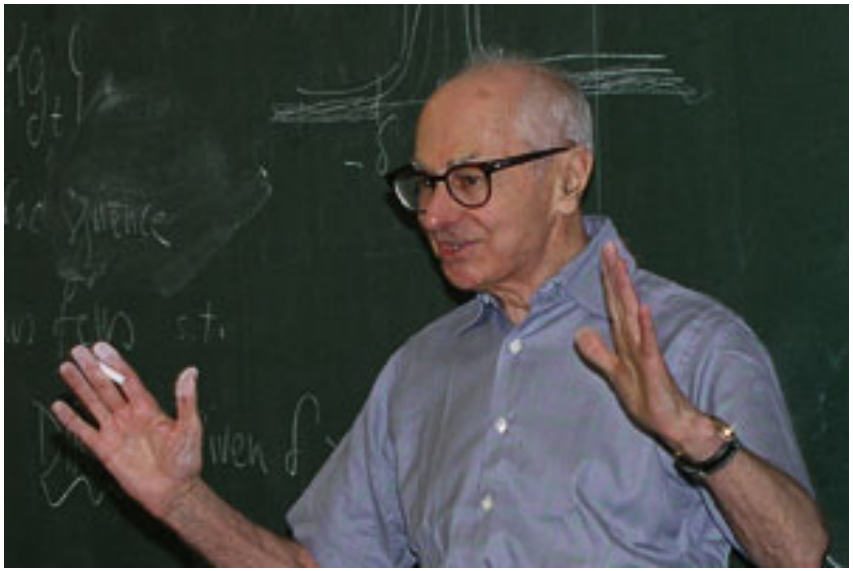
August 28, 2006
MUSA Serge Lang Memorial Lecture

http://sage.math.washington.edu

Outline

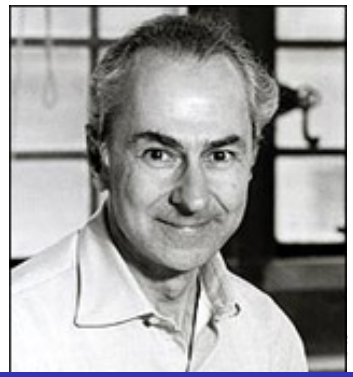
My First Meeting With Serge Lang

Tell an anecdote about meeting Serge Lang at Berkeley in 1995 and learning about restriction of scalars... Piles of books... Algebraic geometry chapter...



Mazur's Quote

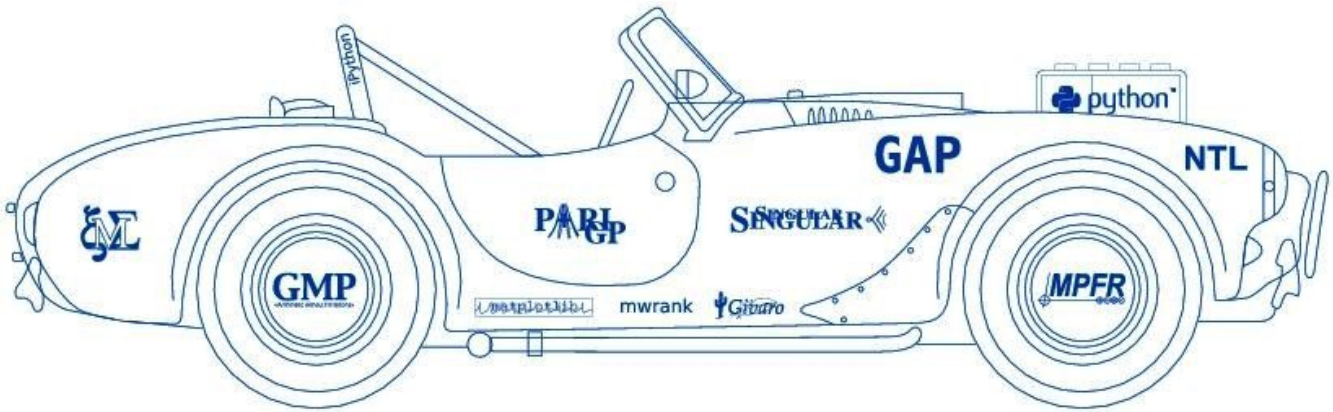
"Serge did this sort of thing, through the decades, with many of the young: he would proffer to them gracious, yet demanding, invitations to engage as a genuine colleague—not teacher to student—but mathematician to mathematician; he did all this naturally, and with extraordinary generosity and success.
That we are personally responsible for the web of compromises that we have all come to accept, and to think are inevitable, is something he would never let us forget. That we, as editors or referees of journals, make our judgments based on some presumed social, or sociable, contract does not let us off the hook when asked to examine without prejudice the underpinnings of our (usually only implicit) social contracts."
– Barry Mazur



Open Source Math Software

SAGE: Free open source Software For Algebra and Geometry Experimentation

SAGE
Building »The Car«



»Every free computer algebra system I've tried has reinvented many times the wheel without being able to build the car.«

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? "

– J. Neubüser (he initiated the free group theory software GAP in 1986).

A few days ago, from the Axiom mailing list:

```
From: "Page, Bill" <Bill.Page@drdc-rddc.gc.ca>
Date: Wed, 16 Aug 2006 07:48:48 -0400
```

"You will find that SAGE is quite different than any other computer algebra system with which you might be familiar - truly a 'new generation'.

I think the SAGE developers were very bold—**maybe even audacious**—to actually attempt this. And they are doing it in a largely pragmatic way without attempting to incorporate the more formal and theoretical ideas developed by the OpenMath community.

One might have been tempted to predict an early failure to this effort but on the contrary **Sage seems to be growing more rapidly than any other computer algebra research and development effort**."

What is SAGE?????

1. **A Distribution** of free open source math software. 52MB source tarball that builds self-contained ("no" dependencies).
2. **A New Collection of code** that fills in the gaps in existing open source free math software.
3. **A New Interface** to your mathematics software: MAGMA, Mathematica, Maple, etc.

SAGE works well on **Linux** and **OS X**, and works on MS Windows.

Who is Writing SAGE?

Professors, Postdocs, Graduate students,
Undergraduates, High school teachers, Professionals,
Retired techies, **Maybe YOU?!**

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

SAGE: Technology Overview

The SAGE Notebook

SAGE has a nontraditional graphical user interface. It is an **"AJAX application"** like Gmail, but it runs completely on your desktop; also a sort of "wiki".

1. **Written from scratch** in Python and Javascript in June by me, Alex C. and Tom B.
2. Uses Python's built-in **BaseHTTPServer** web server.
3. Works well with Firefox, Safari, and Opera.
4. Client/server model which works **over network** or locally.
5. Current version is **stable and usable**.
6. Try it: `http://sage.math.washington.edu:8100`

The Key Components of SAGE

Graphical User Interface	**Notebook (AJAX)**
Interactive Shell	**IPython**
Interpreted language	**Python**
Graphics	**Matplotlib**
Group theory and combinatorics	**GAP**
Symbolic computation / calculus	**Maxima**
Commutative algebra	**Singular**
Number theory	**PARI, MWRANK, NTL**
Numerical Computation	**GSL** and **SciPy**

All (but SciPy) part of the core SAGE install.

Interfaces to Everything!

Continue to use your favorite programs and code from within SAGE (but with a **Pythonic** interface!):

- SAGE includes (mostly pseudo-tty) interfaces to

  GAP, **GP/PARI**, Kash, Macaulay2, Magma, Maple, Mathematica, **Maxima**, Octave, **Singular**, etc.

- Red systems are included standard with SAGE.
- Get uniform flexible access to all of the functionality of the other systems.
- Get tab completion, online help.
- Driven by **demand**, not theory – Result: completely ignores the Open Math and MathML communities.

Pyrex: Fast compiled code

- Is SAGE **"impossible"**???:

  ```
  Date: Mon, 14 Aug 2006 15:26:36 +0100 (BST)
  I've never understood how you can think something that
  sits on top of python can beat some custom code like
  magma. How does SAGE multiply two numbers? Does it use
  python to do it? I was trying to explain SAGE and failed
  miserably.
  ```

- SAGE can be very fast, since it is partly written in **Pyrex**, which is a Python-like language that is converted to C and compiled:
  `http://www.cosc.canterbury.ac.nz/greg.ewing/python/Pyrex/`
  My plan is that all of SAGE's basic arithmetic types will be (re-)written this way in the next few months.
- Martin Albrecht and I are improving Pyrex and making it very easy to use from within SAGE (all build details are automated).

Demo: The Birch and Swinnerton-Dyer Conjecture

"It was whilst I was staying in a lovely hotel in the Black Forest in Germany. I plotted the numbers I got, and lo and behold there were a dozen dots arranged in four parallel lines ... Wonderful! From that point on it was absolutely clear that there was something there."

– Bryan Birch

History and Status Report

History: SAGE 0.1 to SAGE 1.3.6.3

- **1997-2004:** Much C++ and MAGMA for research.
- **Feb 2004:** Extreme frustration with MAGMA.
- **Feb 2005:** I got job offers with **tenure** – *SAGE 0.1*
- **April 2005:** **Interfaces** to Mathematica, Magma, etc.
- **Feb 2006:** SAGE Days 1 workshop – **SAGE 1.0**
- **June 2006:** High school workshop – **SAGE Notebook**
- **August 2006:** Grad student workshop – two week super-intense *graduate student* coding sprint.
- **Today:** SAGE now has a **huge range of functionality**. But **SAGE is not fast enough yet.**

Goals for SAGE 2.0

Main Goals for SAGE 2.0 – December 2006

1. Make the **basic arithmetic**, e.g., finite fields, polynomials, etc., very fast.
2. Make the **everyday *exact* linear algebra** very fast.
3. Core numerical support.
4. Good **3d visualization** in the notebook.
5. Make viewing **source code** of what you're using *extremely easy*.
6. For SAGE 3.0: Big push for parallel algorithms.

Any Questions?