# SAGE Days 2: Status Report and Future

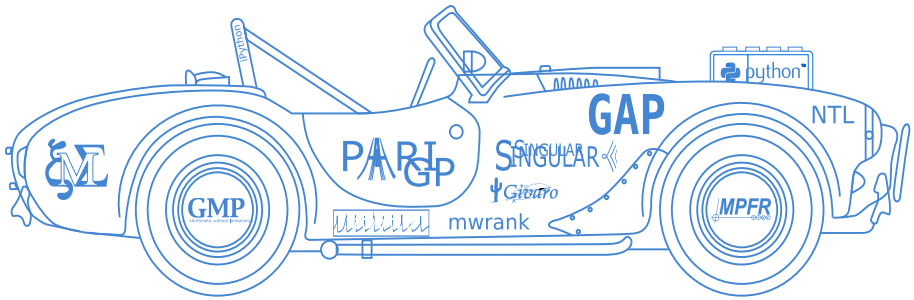
William Stein

October 7, 2006, UW

http://sage.math.washington.edu/sage

SAGE Days 2

Building The Car




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

# SAGE Days 2

http://sage.math.washington.edu/sage/days2

## Welcome!!

Talks Saturday-Sunday and coding sprints Friday, Monday, Tuesday.

# History: to SAGE 1.4

- **1997-1999:** Tons of C++
- **1999-2004:** Tons of 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 Days 2 – SAGE now has a wide range of functionality. SAGE is *not stable* enough. It is *not robust* enough. It *doesn't do what I want*, or what you want. **I want way way more. SOON!**

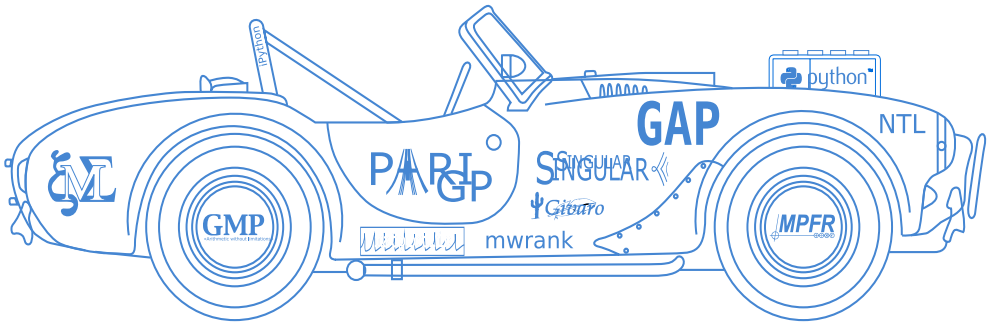
# The GOAL

**GOAL: Create high quality free software for algebra, number theory, and geometry. And don't neglect numerical computation.**

http://sage.math.washington.edu/sage

## SAGE Days 2

### Building The Car




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

# What is SAGE?

1. **Distribution:** 63MB free open source tarball
2. **New Code:** New algorithms; fill gaps
3. **Interfaces:** To your software

# SAGE is also a computer...

http://sage.math.washington.edu

The SAGE "sandbox": **64GB RAM**, **16 processor** Opteron server.
You can browse the developer working directories over the web here!

# The SAGE Notebook: We now have a good GUI

1. An **"AJAX application"** like Google maps or Gmail, **but** can run locally on your computer.
2. **Written from scratch** recently by me, Alex C. and Tom B.
3. Uses Python's built-in **BaseHTTPServer** web server (we may soon switch to Twisted for a more robust security model).
4. Works well with Firefox, Safari, and Opera.
5. Client/server model which works **over network** or locally.
6. Current version is **stable and usable**. There is a big idea list that will influence the next version.
7. Try it: `http://sage.math.washington.edu:8100`

# New SAGE Code: SAGE is getting BIG

Over 100000 lines of new Python and Pyrex code:

```
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
```

```
UNIQUE Source Code Lines (including docstrings):
$ cat */*.py */*/*.py */*/*/*.py */*.pyx \
                */*/*.pyx */*/*.pyx |sort |uniq | wc -l
86102
```

```
UNIQUE Input Docstrings:
$ cat */*.py */*/*.py */*/*/*.py */*.pyx \
          */*/*.pyx */*/*.pyx |sort|uniq|grep "sage:" | wc -l
9612
```

# Core Components of SAGE: All Bases are Covered

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

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

# Singular

Singular includes a *very* restrictively licensed component, namely omalloc. So far, it appears that Singular crucially depends on omalloc in order to build. This may be very nontrivial to get around since, e.g., omalloc is nearly 28000 lines. I am extremely angry at being misled by the Singular website about this!! The omalloc license is *terrible*:

> *"Permission of use within the software SINGULAR is granted by the author. In addition to this permission to modify the sources is granted to the copyright holders of SINGULAR. If you wish to* **use this package outside of SINGULAR** *or* **to modify it in any way**, *please contact the author..."*

Thus omalloc is not allowed to be used in software other than Singular, which could potentially mean its use in SAGE is a violation. Moreover, one is not allowed to "modify it in any way" (even for personal use!).

# Singular?: Options

The options are:

1. Find a way to **remove** the Singular dependence on omalloc (which is over 20000 lines of code!).
2. Get omalloc **released** under a free open source license: Email "Olaf Bachmann" <obachman@mathematik.uni-kl.de>.
3. **Switch to Macaulay2** – it builds easily for SAGE now due to much hard work by Dan Grayson. Collaboration between Macaulay2 and SAGE could be *extremely* productive, and they have a lot of amazing algebraic geometry.

**Plan:** Try 1 and 2 until Oct 23. If they fail, do 3 (I will be at a workshop then with both of the Macaulay2 authors (Dan and Mike), which would be an ideal time to do 3.) Many students at UW know Macaulay2.

# Distribution Model

Target Platforms: Linux, OS X, Windows (I have given up on Solaris and Cygwin, for now).

- **Linux** – Build from source. Still no good .dep and .rpm's. Part of why is not just deciding to make something that install to /opt.
- **OS X $\geq$10.4.x** – Build from source. Precompiled binaries.
- **Windows** – **Colinux as kernel is a reasonable solution**. Gary Zablackis does a great job maintaining this branch.

## 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. **Usability:** Notebook, Pyrex introspection, Documentation browser, Graphics.
4. **Stability:**
   1. SAGE is **not ready for stability** today (timing isn't right).
   2. **Fast robust arithmetic** is crucially important, so we have to finish that ASAP. Stabilize only on a proven design choice.
   3. **SEP**: SAGE Enhancement Proposal (like Python's PEP's) – will be required after SAGE 2.0.

# Next Year (2007) – SAGE 3.0

1. **Parallelism:** Support for **coarse grain parallelism** and use of it for algorithms in algebra, geometry, and number theory. I'm organizing **workshop in January** at MSRI on this, hopefully followed by a series of 2-week coding sprints through the year.

2. **Higher level algorithms**, e.g., computing with modular abelian varieties, quaternion algebra ideal theory, Hilbert modular forms, etc.

# Funding Sources

1. My **Startup** Money.
2. My 3-year **NSF grant**, if I can move it from UCSD.
3. **Apply** to the NSF, e.g., I'm applying for an **undergraduate program** right now.
4. Friends in high places.
5. Create a **SAGE Mathematics Foundation**.
6. A **major expense** is workshops— apply to AIM, MSRI, Banff, IMA, etc., for another SAGE Days!

# Mercurial: SAGE's Source Control System

1. **Written in Python** – crucial for long-term integration into SAGE (it's 15000 lines of code).
2. **VERY FAST and ROBUST**
3. **Active Development Community**
4. From SAGE: `hg_sage.[tab]`
5. From command line: `sage -hg [args]` or use `install_scripts`
6. Quick **tutorial** if time permits: Create a hello world patch and bundle it.

# Python 2.5!

1. **list indices**: `__index__` method; This means `n=2; [1,2,3,4][n]` finally works!! Thanks to Travis Oliphant.
2. **Optimizations** – "Several of the optimizations were developed at the NeedForSpeed sprint, an event held in Reykjavik, Iceland, from May 21-28 2006." E.g., "Exception handling in Python 2.5 is therefore about 30
3. **Memory management** is improved
4. **ctypes** is included standard
5. **conditional expressions** `x = true_value if condition else false_value` context managers: relevant to all the discussion about precision lately.
6. **any and all** – new builtins:

```
sage: any(is_prime(x) for x in range(114,125))
False
sage: all(is_prime(x) for x in prime_range(50,100))
True
```

# Upcoming Relevant Events

- **October 23–27**: I'm speaking about SAGE at an IMA "Software in Algebraic Geometry" in Minnesota later this month.
- **Jan 29–Feb 2**: MSRI Workshop on Interactive Parallel Computation – I'm chair of organizing committee.
- **March 10–14, 2007**: Arizona Winter School (I'm a co-organizer). David Savitt is the organizer.
- **June 3–8, 2007**: Banff workshop on modular forms computation – I'm an organizer
- **July 30-Aug 3, 2007**: AIM: Workshop on Modular Forms and L-functions – I'm an organizer

# Mathematics Software for Everybody!