

Free Math!

The Sage Software Project for Free and Open-Source Computation

Marshall Hampton
University of Minnesota, Duluth

Abstract: Three years ago William Stein released the first version of Sage, a free and open source computational platform based on the popular scripting language Python. Sage unifies a huge collection of mathematical software projects into a coherent and powerful system for mathematics, statistics, and scientific computation. In this talk I will give a brief overview of its present capabilities and future directions.

Thoughts on Open-Source Software

“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 in 1993 (founded GAP in 1986).

Timeline

- 1 1980 Maple created as an accessible alternative to Macsyma.
- 2 1984 Matlab-based company MathWorks founded.
- 3 1984 Maple becomes commercial.
- 4 1988 Mathematica releases version 1.0.
- 5 1993 Magma 1.0 released.

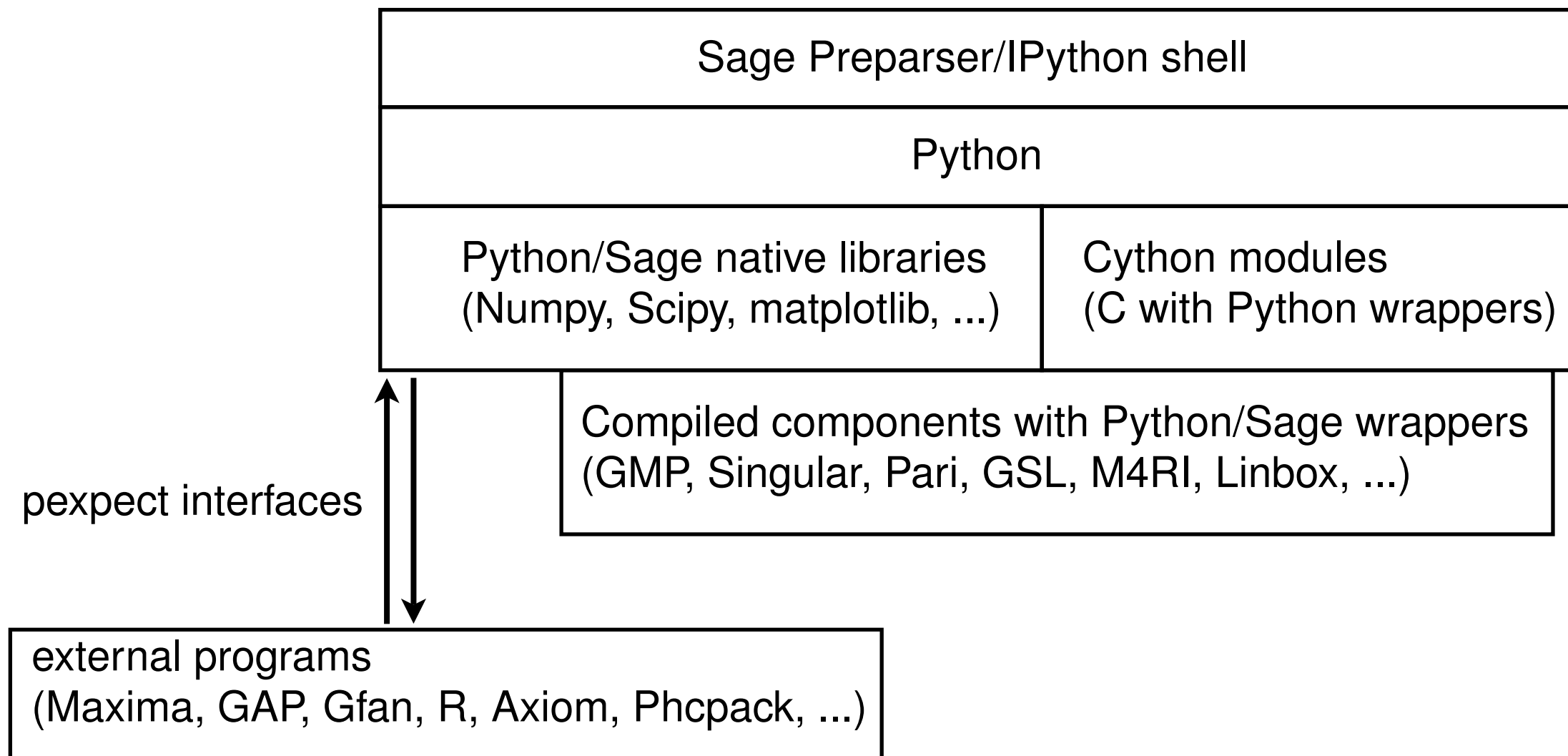
Timeline

- ① 1980 Maple created as an accessible alternative to Macsyma.
- ② 1984 Matlab-based company MathWorks founded.
- ③ 1984 Maple becomes commercial.
- ④ 1988 Mathematica releases version 1.0.
- ⑤ 1993 Magma 1.0 released.
- ⑥ 2005 Sage created
- ⑦ 2005 (November) Version 0.9.10 of Sage made publicly available.
- ⑧ 2006 (February) version 1.0.0 of Sage (38th release).
- ⑨ 2007 (January) Mathematica 6.0: Rewrite of graphics for Manipulate/interactive displays.

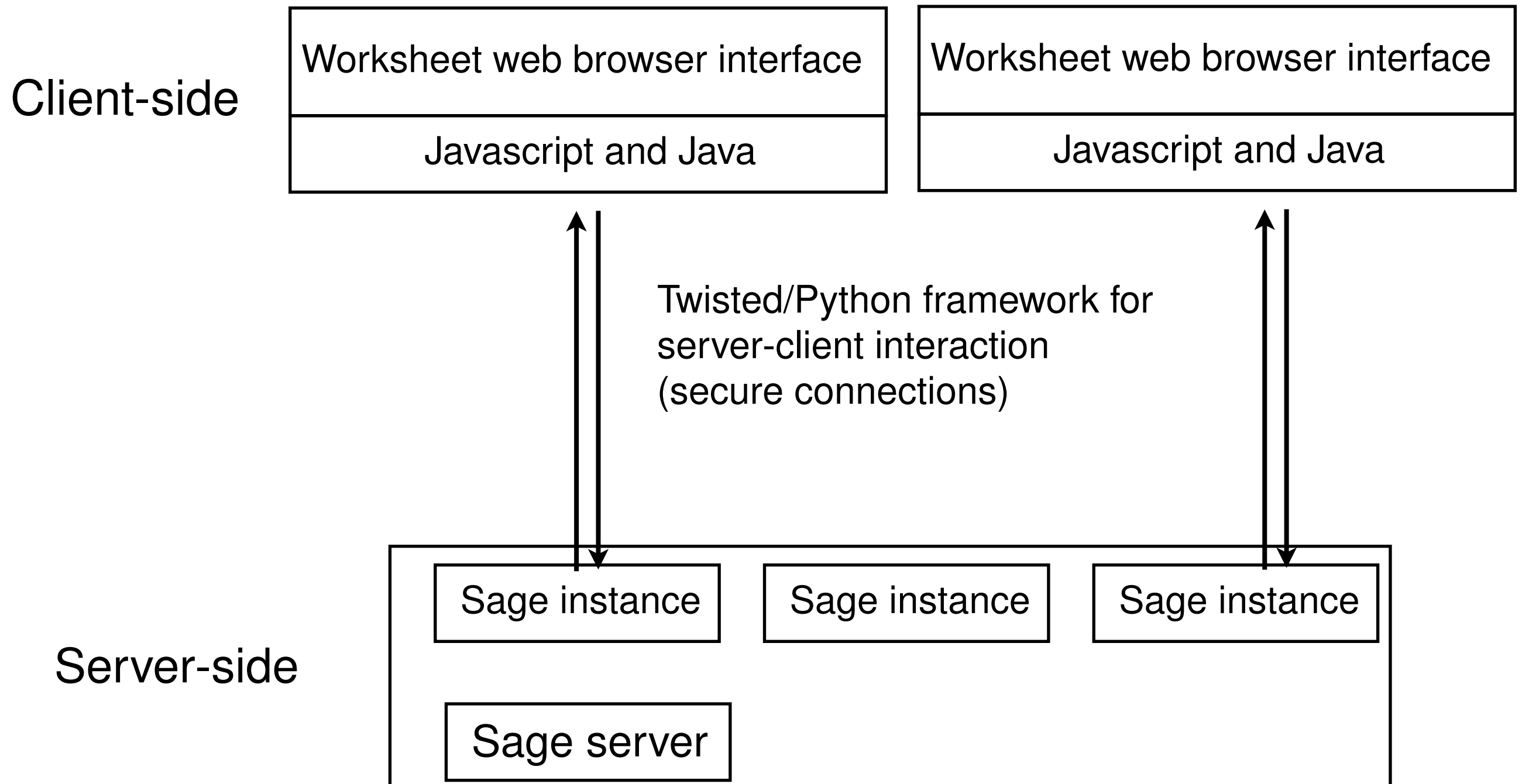
Timeline

- 1 1980 Maple created as an accessible alternative to Macsyma.
- 2 1984 Matlab-based company MathWorks founded.
- 3 1984 Maple becomes commercial.
- 4 1988 Mathematica releases version 1.0.
- 5 1993 Magma 1.0 released.
- 6 2005 Sage created
- 7 2005 (November) Version 0.9.10 of Sage made publicly available.
- 8 2006 (February) version 1.0.0 of Sage (38th release).
- 9 2007 (January) Mathematica 6.0: Rewrite of graphics for Manipulate/interactive displays.
- 10 2007 (January) version 2.0.0 of Sage (118th release). Interactive 3D graphics.
- 11 2007 (November) Sage wins 1st place in the Trophees du Libre competition.
- 12 2008(March) Interact command created.
- 13 2008 (April) version 3.0.0 of Sage (176th release). R support and massively improved testing.
- 14 2008 (November) 188th release of Sage, version 3.2.

Sage Architecture



Sage notebook architecture



Python Quotes

YouTube.com

"Python is fast enough for our site and allows us to produce maintainable features in record times, with a minimum of developers," said Cuong Do, Software Architect, [YouTube.com](#).

Industrial Light & Magic

"Python plays a key role in our production pipeline. Without it a project the size of Star Wars: Episode II would have been very difficult to pull off. From crowd rendering to batch processing to compositing, Python binds all things together," said Tommy Burnette, Senior Technical Director, [Industrial Light & Magic](#).

"Python is everywhere at ILM. It's used to extend the capabilities of our applications, as well as providing the glue between them. Every CG image we create has involved Python somewhere in the process," said Philip Peterson, Principal Engineer, Research & Development, [Industrial Light & Magic](#).

Google

"Python has been an important part of Google since the beginning, and remains so as the system grows and evolves. Today dozens of Google engineers use Python, and we're looking for more people with skills in this language." said Peter Norvig, director of search quality at [Google, Inc.](#)

Python Quotes

Firaxis Games

"Like XML, scripting was extremely useful as both a mod tool and an internal development tool. If you don't have any need to expose code and algorithms in a simple and safe way to others, you can argue that providing a scripting language is not worth the effort. However, if you do have that need, as we did, scripting is a no brainer, and it makes complete sense to use a powerful, documented, cross-platform standard such as Python." -- Mustafa Thamer of Firaxis Games, talking about Civilization IV. Quoted on page 18 of the August 2005 [Game Developer Magazine](#).

"Python, like many good technologies, soon spreads virally throughout your development team and finds its way into all sorts of applications and tools. In other words, Python begins to feel like a big hammer and coding tasks look like nails." -- Mustafa Thamer of Firaxis Games, talking about Civilization IV. Quoted on page 18 of the August 2005 [Game Developer Magazine](#).

"We chose to use python because we wanted a well-supported scripting language that could extend our core code. Indeed, we wrote much more code in python than we were expecting, including all in-game screens and the main interface. It was a huge win for the project because writing code in a language with garbage collection simply goes faster than writing code in C++. The fact that users will be able to easily mod the interface is a nice plus as well. The downside of python was that it significantly increased our build times, mostly from linking with Boost."

-- Soren Johnson, lead designer, Civilization IV. Quoted in [a Slashdot interview](#).

Python Quotes

EVE Online

"Python enabled us to create [EVE Online](#), a massive multiplayer game, in record time. The EVE Online server cluster runs over 25,000 simultaneous players in a shared space simulation, most of which is created in Python. The flexibilities of Python have enabled us to quickly improve the game experience based on player feedback," said Hilmar Veigar Petursson of [CCP Games](#).

Thawte Consulting

"Python makes us extremely productive, and makes maintaining a large and rapidly evolving codebase relatively simple," said Mark Shuttleworth.

University of Maryland

"I have the students learn Python in our undergraduate and graduate Semantic Web courses. Why? Because basically there's nothing else with the flexibility and as many web libraries," said Prof. James A. Hendler.

EZTrip.com

"The travel industry is made up of a myriad supplier data feeds all of which are proprietary in some way and are constantly changing. Python repeatedly has allowed us to access, build and test our in-house communications with hundreds of travel suppliers around the world in a matter of days rather than the months it would have taken using other languages. Since adopting Python 2 years ago, Python has provided us with a measurable productivity gain that allows us to stay competitive in the online travel space," said Michael Engelhart, CTO of [EZTrip.com](#).

Cython

A killer-app spinoff of the Sage project.

Used when there is a need for speed.



About Cython

Documentation

Community

Download

Development

About Cython

Cython is a language that makes writing C extensions for the Python language as easy as **Python** itself. Cython is based on the well-known **Pyrex**, but supports more cutting edge functionality and optimizations.

The Cython language is very close to the **Python** language, but Cython additionally supports calling **C functions** and declaring **C types** on variables and class attributes. This allows the compiler to generate very **efficient C code** from Cython code.

This makes Cython the ideal language for **wrapping** for external C libraries, and for **fast C modules** that speed up the execution of Python code.

Documentation

- [Cython Users Guide](#) (also available as a [pdf](#))
- [The Cython Wiki](#).
- [Pyrex FAQ](#) and [Cython FAQ](#)

If you still have questions, feel free to send an email to the [mailing list](#).

License

Sage components are GPL version 2 compatible.

This means that anyone is free to install and modify all parts of Sage.
If Sage is used in a commercial product, AND the product uses a modified version of Sage, then the modified Sage code must be made available (a reasonable fee can be charged for distribution).

Sage Components

- ① ATLAS: Automatically Tuned Linear Algebra Software
- ② BLAS: Basic Fortran 77 linear algebra routines
- ③ Bzip2: High-quality data compressor
- ④ Cddlib: Double Description Method of Motzkin
- ⑤ Common Lisp: Multiparadigm and general-purpose programming language
- ⑥ CVXOPT: Convex optimization, linear programming, least squares, etc.
- ⑦ Cython: C-Extensions for Python
- ⑧ F2c: Converts Fortran 77 to C code
- ⑨ Flint: Fast Library for Number Theory
- ⑩ FpLLL: Euclidian lattice reduction
- ⑪ FreeType: A Free, High-Quality, and Portable Font Engine
- ⑫ G95: Open source Fortran 95 compiler
- ⑬ GAP: Groups, Algorithms, Programming
- ⑭ GD: Dynamic graphics generation tool
- ⑮ Genus2reduction: Curve data computation
- ⑯ Gfan: Grbner fans and tropical varieties
- ⑰ Givaro: C++ library for arithmetic and algebra

Sage Components

- 18 GMP: GNU Multiple Precision Arithmetic Library
- 19 GMP-ECM: Elliptic Curve Method for Integer Factorization
- 20 GNU TLS: Secure networking
- 21 GSL: Gnu Scientific Library
- 22 JsMath: JavaScript implementation of LaTeX
- 23 IML: Integer Matrix Library
- 24 IPython: Interactive Python shell
- 25 LAPACK: Fortran 77 linear algebra library
- 26 Lcalc: L-functions calculator
- 27 Libgcrypt: General purpose cryptographic library
- 28 Libgpg-error: Common error values for GnuPG components
- 29 Linbox: C++ linear algebra library
- 30 M4RI: Linear Algebra over GF(2)
- 31 Matplotlib: Python plotting library
- 32 Maxima: computer algebra system
- 33 Mercurial: Revision control system
- 34 MoinMoin Wiki
- 35 MPFI: Multiple Precision Floating-point Interval library

Sage Components

- 36 MPFR: C library for multiple-precision floating-point computations with correct rounding
- 37 ECLib:Cremona's Programs for Elliptic curves
- 38 NetworkX: Graph theory
- 39 NTL: Number theory C++ library
- 40 Numpy: Numerical linear algebra
- 41 OpenCDK: Open Crypto Development Kit
- 42 PALP: A Package for Analyzing Lattice Polytopes
- 43 PARI/GP: Number theory calculator
- 44 Pexpect: Pseudo-tty control for Python
- 45 PNG: Bitmap image support
- 46 PolyBoRi: Polynomials Over Boolean Rings
- 47 PyCrypto: Python Cryptography Toolkit
- 48 Python: Interpreted language
- 49 Qd: Quad-double/Double-double Computation Package
- 50 R: Statistical Computing
- 51 Readline: Line-editing
- 52 Rpy: Python interface to R

Sage Components

- 53 Scipy: Python library for scientific computation
- 54 Singular: fast commutative and noncommutative algebra
- 55 Scons: Software construction tool
- 56 SQLite: Relation database
- 57 Sympow: L-function calculator
- 58 Symmetrca: Representation theory
- 59 Sympy: Python library for symbolic computation
- 60 Tachyon: lightweight 3d ray tracer
- 61 Termcap: Simplifies the process of writing portable text mode applications
- 62 Twisted: Python networking library
- 63 Weave: Tools for including C/C++ code within Python
- 64 Zlib: Data compression library
- 65 ZODB: Object-oriented database

Sage developers around the world

This is a map of all contributors to the Sage project. There are currently 111 contributors in 62 different places from all around the world.

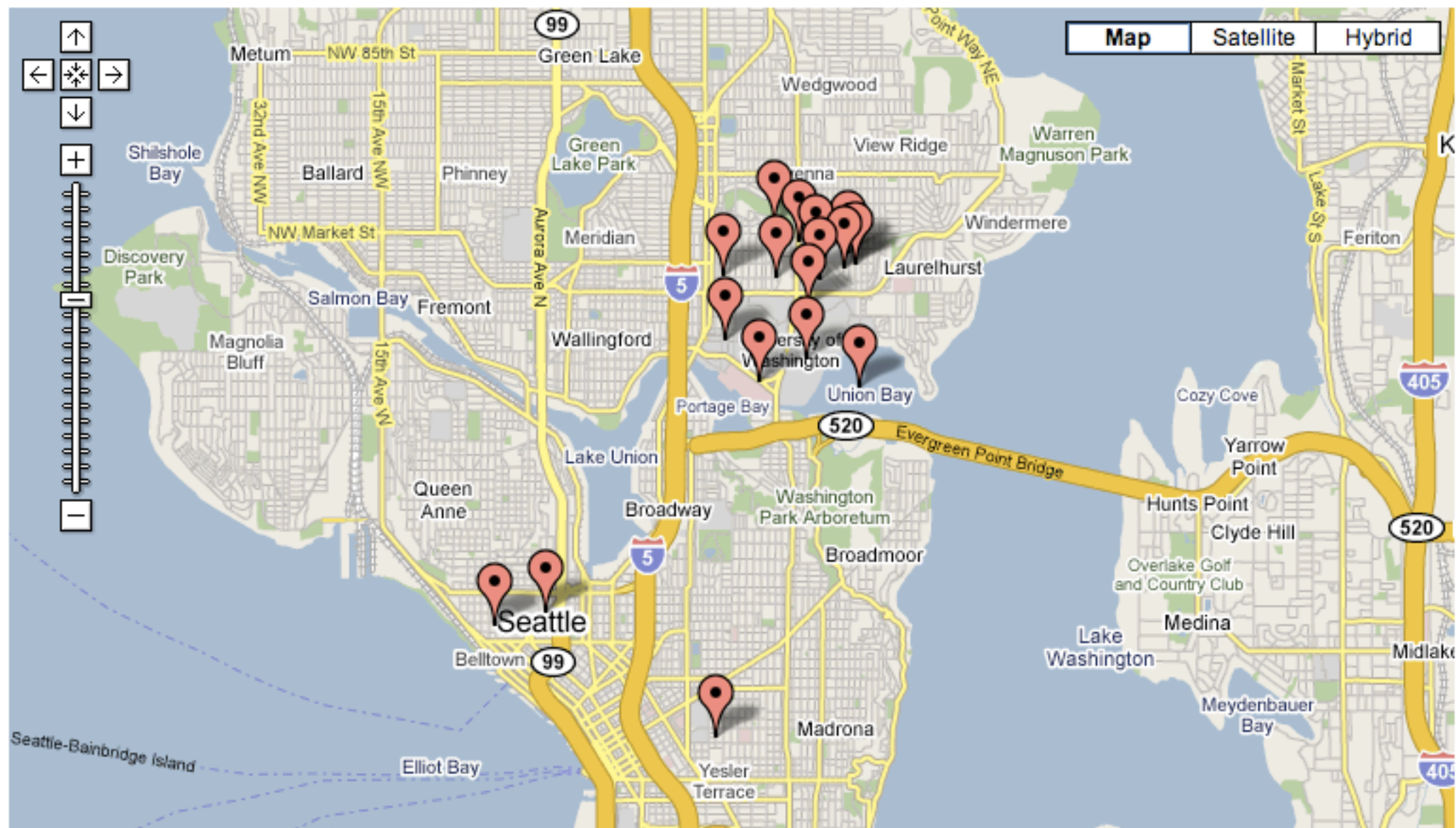
Map Zoom: [Earth](#) - [USA](#) ([UW](#), [West](#), [East](#)) - [Europe](#)



Sage developers around the world

This is a map of all contributors to the Sage project. There are currently 111 contributors in 62 different places from all around the world.

Map Zoom: [Earth](#) - [USA](#) ([UW](#), [West](#), [East](#)) - [Europe](#)





sage-devel

*-Combinat: MuPAD-Combinat join forces with Sage

[Options](#)

☆ 2 messages - [Collapse all](#)

➤ **Mike Hansen** [View profile](#)

[More options](#) Jun 20, 6:54 am

A message from Florent and Nicolas:

Dear Sage developers,

After months of discussions and experiments we, MuPAD-Combinat developers, have finally reached a conclusion:

We are joining you!

Thank you for all the support you readily provided us. This decision is a risky one for us: it will be a massive investment to migrate our code and, even more important, our community. We really count on your support to make this profitable for both sides in the long run!

In the coming days, we will start building a roadmap by posting tickets on Sage trac for both:

- sage low level features we would like to rely on
- features we want to port

Sage developers around the world

This is a map of all contributors to the Sage project. There are currently 111 contributors in 62 different places from all around the world.

Map Zoom: [Earth](#) - [USA](#) ([UW](#), [West](#), [East](#)) - [Europe](#)





Sage Demo

SAGE Notebook

Version 3.2.alpha0

admin | [Toggle](#) | [Home](#) | [Published](#) | [Log](#) | [Settings](#) | [Report a Problem](#) | [Help](#) | [Sign out](#)

Free Math Demo

last edited on October 30, 2008 12:27 PM by admin

[Save](#) | [Save & quit](#) | [Discard & quit](#)

File... Action... Data... sage ☐ Typeset



Print

Worksheet

Edit

Text

Undo

Share

Publish

```
2+2
```

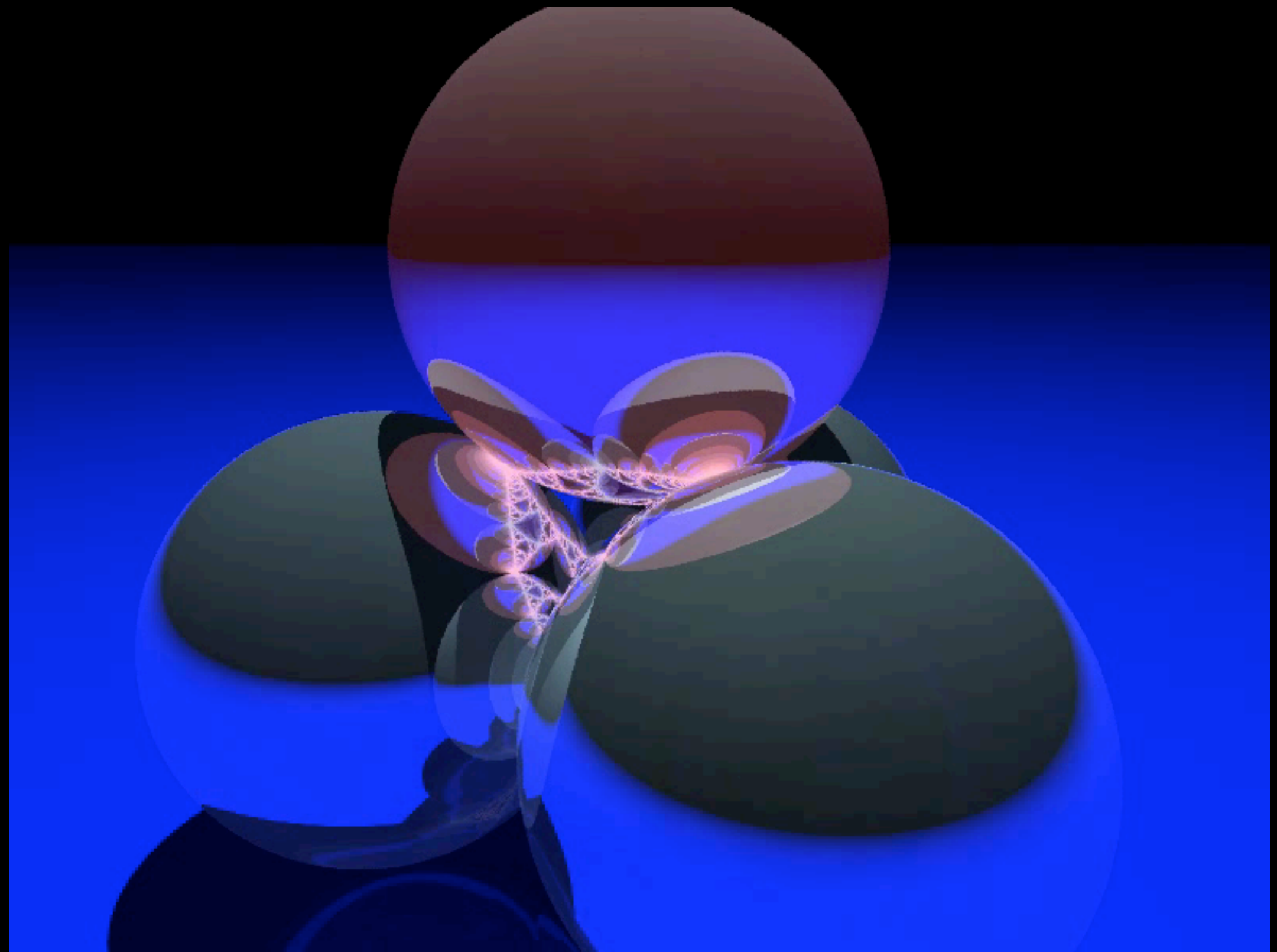
```
factorial(1000)
```

```
a = RR(sqrt(2)); a
```

```
b = sqrt(RealField(500)(2)); b
```

```
b.algdep(2)
```

```
a = integrate(sin(x^2),x);  
a.show()
```



Trac: bug and feature tracker

<http://trac.sagemath.org/sage>

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.»

[Login](#) | [Settings](#) | [Help/Guide](#) | [About Trac](#)

	Wiki	Timeline	Roadmap	View Tickets	Search
--	----------------------	---------------------------------	-------------------------	------------------------------	------------------------

Timeline

10/29/2008:

- 12:26 Ticket [#4381](#) (defect) created by jsp
sage -wthread not passed correctly to ipython
- 06:47 Ticket [#2103](#) (enhancement) closed by mabshoff
fixed: Merged In Sage 3.2.alpha2
- 05:56 Ticket [#4380](#) (defect) closed by mabshoff
fixed: Merged In Sage 3.2.alpha2
- 04:54 Ticket [#4379](#) (defect) closed by mabshoff
fixed: Merged In Sage 3.2.alpha2 Mike: This patch changes one word in the ...
- 02:27 Ticket [#4380](#) (defect) created by SimonKing
[with patch, positive review] fix Memory Leak In libsingular's reduce()


View changes from and days back.

- ☒ Milestones
- ☒ Ticket changes
- ☒ Wiki changes

Trac: bug and feature tracker

<http://trac.sagemath.org/>

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.»

logged in as mhampton | [Logout](#) | [Settings](#) | [Help/Guide](#) | [About Trac](#)

[Wiki](#) | [Timeline](#) | [Roadmap](#) | **[View Tickets](#)** | [New Ticket](#) | [Search](#)

[Available Reports](#) | [Custom Query](#)

{10} Tickets needing review (54 matches)

sage-3.2

Ticket	Status	Owner	Created	Summary
#3900	new	mabshoff	08/19/2008	[with patch; needs review] make testing an official pickle jar a part of "make check"
#4261	new	mabshoff	10/10/2008	[with patch, needs review] sympow Configure fails to handle aliases
#4372	new	was	10/26/2008	[with patch, needs review] Repair to totallyreal_dsage

sage-3.2.1

Ticket	Status	Owner	Created	Summary
#3898	assigned	mabshoff	08/19/2008	[with spkg, needs review] Make an optional, self contained gcc 4.3.2.spkg
#4205	new	was	09/27/2008	[with patch, needs review] Fix deprecation warnings from numpy 1.2

Sage Wiki: General Coordination and Information

Bug Days

- [Sage Bug Day 1](#) (August 18th, 2007)
- [Sage Bug Day 2](#) (September 6th, 2007)
- [Sage Bug Day 3](#) (September 20th, 2007)
- [Sage Bug Day 4](#) (October 20th, 2007)
- [Sage Bug Day 5](#) (November 3rd, 2007)
- [Sage Bug Day 6](#) (December 1st, 2007)
- [Sage Bug Day 7](#) (December 14th, 2007)
- [Sage Bug Day 8](#) (January 2nd, 2008)
- [Sage Bug Day 9](#) (January 19th, 2008)
- [Sage Bug Day 10](#) (February 16th, 2008)
- [Sage Bug Day 11](#) (April 5, 2008)
- [Sage Bug Day 12](#) (May 10, 2008)
- [Sage Bug Day 13](#) (August 23rd, 2008)
- [Sage Bug Day 14](#) (October 23, 2008)
- [Sage Bug Day 15](#) (October 30, 2008)

Events/Upcoming Research Workshops

- [Sage Days 11](#) -- Austin Texas, November 7 -- 10, 2008.
- Sage Events at the national AMS-MAA joint meeting in Washington DC, Jan 5-8, 2009:
 - [Sage and Mathematical Research](#) -- AMS Special Session.
 - [Sage and Teaching](#) -- MAA Panel discussion on open source software in math education.
 - [Exhibit Booth](#) -- Sage in the exhibition hall.
- [Sage Days 12](#) -- San Diego, Wed-Sat, Jan 21--24, 2009.
- [Sage Days 13](#) -- Quadratic Forms and Lattices --- Athens, Georgia, March 7 -- 13, 2009.
- [Sage Days 14](#) -- MSRI (Berkeley, CA), March 9--12, 2009, [Official MSRI Page](#).
- [Sage Days 15](#) -- Seattle, WA (just got funded; dates TBA).
- [Sage Days 16](#) -- [CRM-UPC](#), Barcelona, June 22--27, 2009.
- [Hosting](#) --- Advice to future hosts

Future directions for Sage

For the project as a whole:

- ① Speed up symbolic computation.
- ② Improve documentation (switch to Sphinx/ReST).
- ③ Better integration between components (R, scipy, JMol,...).
- ④ Convenient API for sage server separate from notebook interface.
- ⑤ Better distributed computing (DSage).
- ⑥ Compile cleanly on more architectures (OS X 64 bit, Windows, Solaris, AIX, ...).

Future directions for Sage

For me personally:

- 1 Expand its use in education.
- 2 Add more dynamical systems functionality (PyDSTool, AUTO).
- 3 Create interfaces for CellML models (mathematical biology).
- 4 Make movie-making easier.
- 5 Improve polyhedral functionality.

Thank you!