

SAGE : Open-Source Mathematics Software

Pavel Hubáček
pavel.hubacek@gmail.com

March 27, 2010
Spring School of Department of Algebra

Overview

- SAGE is free open source software for research in algebra, geometry, number theory, cryptography, and numerical computation.
- A way to use all your favorite (commercial or free) mathematics software together.
- It is intended to be a free and open-source alternative to the M's: Mathematica, Maple, Matlab, Maxima, etc.
- The primary goal of SAGE is to make modern research-level algorithms available in an integrated package with a graphical interface.

Basic functionality

Some of the many features of Sage include:

- A notebook document interface, for review and re-use of previous inputs and outputs, including graphics and text annotations usable from most web browsers. A secure connection via HTTPS to the notebook is supported when security or confidentiality are important, and allows Sage to be used both locally and remotely.
- The Python programming language supporting procedural, functional and object oriented constructs.
- Support for parallel processing using both multi-core processors found in many modern computers, multiple processors, in addition to distributed computing.
- Well documented: tutorial, programming guide, reference manual, and how-to, with numerous examples and discussion of background mathematics.

Who is behind sage?

- The project is organized by William Stein, who is an associate professor in mathematics at the University of Washington.
- The first version was released in February 2005 under the GNU GPL.
- Work is done mainly by volunteers. There is an endless list of names of all who have contributed to the included packages so far.

What is inside?

Sage is combination of the following:

- The best existing open source math programs in many functional area: number theory (PARI), group theory (GAP), commutative algebra (Singular), and symbolic computation (Maxima).
- The best open source math libraries: arithmetic (GMP), number theory (NTL), numerical (gsl), finite fields (givaro), linear algebra (linbox, iml, and numpy), graphics (matplotlib and tachyon), etc.
- A mainstream interpreter (Python) and a GUI (the web-based SAGE notebook).
- Interfaces to all major math software.
- Lots and lots of work to put it all together, make it fast, easy to install, and fill in the massive gaping holes in functionality.

What is inside?

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, jmol
Group theory and combinatorics	GAP
Graph theory	Networkx
Interactive programming language	Python (widespread!)
Networking	Twisted
Number theory	Pari
Numerical computation	GSL, Numpy, etc.
Polynomials in $GF()$	NTL/FLINT, Singular
Statistics	R (itself extensible!)
Symbolic computation, calculus	Maxima, sympy

A huge package

SAGE is a rather huge distribution. All included packages are slightly adopted and well tested to work with SAGE.

It can be obtained in different ways:

- the program with sources, version history, ready for development and documentation, for Linux
- Linux binaries
- Mac OS X binaries
- Solaris binaries
- Windows: encapsulated in VMWare image since main development platform of the modules is Linux

Long-term goals

- The aim is to provide software that can be used to explore and experiment with mathematical constructions in algebra, geometry, number theory, calculus, numerical computation, etc.
- SAGE will help make it easier to interactively experiment with mathematical objects.
- SAGE's intended audience is mathematics students (from high school to graduate school), teachers, and research mathematicians.
- fast
- free and open source
- cooperation: SAGE is meant to unify existing math software.

Ways to use SAGE

- ① notebook with a graphical interface as a dynamic website
- ② interactive command line
- ③ writing interpreted and compiled programs
- ④ invoking the SAGE library inside python scripts

The Python programming language

- Python is a powerful modern interpreted programming language
"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*.
- *"Google has made no secret of the fact they use Python a lot for a number of internal projects. Even knowing that, once I was an employee, I was amazed at how much Python code there actually is in the Google source code system."*, said Guido van Rossum, Google, creator of Python.
- *"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 ...

- Easy for you to define your own data types and methods on it. Bitstreams, ciphers, rings and whatever.
- Very clean language that results in easy to read code.
Easy to learn:
 - ▶ Free: Dive into Python <http://www.diveintopython.org/>
 - ▶ Free: Python Tutorial <http://docs.python.org/tut/>
- A huge number of libraries: statistics, networking, databases, bioinformatic, physics, video games, 3d graphics, and serious mathematics (via SAGE).
- Very easy to use any C/C++ libraries from Python.
- Excellent support for string manipulation and bit fiddling.
- Cython - a Python compiler (<http://www.cython.org>).



www.sagemath.org

- open and transparent on all levels
- web centric interface and sharing
- unifies existing efforts
- ease of understanding, development and contribution
- Python everywhere - simple and mainstream
- fast growth