Can We Create a Viable Free Open Source Alternative to Magma, Maple, Mathematica and Matlab?

William Stein
Department of Mathematics
University of Washington
Seattle, Washington
USA
http://wstein.org
wstein@gmail.com

ABSTRACT

The goal of the Sage project (`http://sagemath.org`) is to create a truly viable free open source alternative to Magma, Maple, Mathematica and Matlab. Is this possible?

Categories and Subject Descriptors

D.2.8 [**Software Engineering**]: Miscellaneous

General Terms

Design

1. INTRODUCTION

For over a decade I have primarily done research in number theory that often involves computation, mainly using Magma. In 2004 I realized that it was stupid for me to continue building all of my work on Magma because Magma is proprietary, the development model is closed, Magma is expensive which is bad for students, and the language itself lacked many features (e.g., user defined classes) that I had requested repeatedly for over 5 years. Thus to continue to use *only* Magma unacceptably limited my potential in both research and education.

Having used Magma for many years, I simply could not switch to an existing open source system. The only serious free open source software for number theory is PARI, whose capabilities are far behind that of Magma in numerous critical areas of interest to me, including exact linear algebra, commutative algebra, and algebraic curves. And no other free system–GAP, Singular, Axiom, Maxima, Macaulay 2, etc.–even comes close in *all* these areas. In fact, after a decade of patiently waiting, I doubt they ever will.

Magma is the result of decades of hard work by extremely talented mathematicians and programmers such as John Cannon, Allan Steel, Claus Fieker, David Kohel, and many others. I've worked with them and they are *simply amazing*, as is their software. The situation for me to find something similar but open source seemed hopeless. And all attempts to convince the Magma group to open source Magma failed.

In 2004, frustrated that there was no way to solve my problem, and driven by nothing but a naive compulsion, I started the Sage project as a free open source alternative to Magma, and spent a large amount of time working on it even though I was convinced that there was no hope of Sage ever succeeding. The first version of Sage consisted of the Python interpreter and a few scripts for doing number theory, with a design modeled on Magma.

My first real feedback from the computer algebra community came from Richard Fateman in December 2005 when he posted his opinion of the Sage project to `sci.math.symbolic`:

> "By avoiding applications (say, to engineering design, finance, education, scientific visualization, etc etc) the activity [Sage] is essentially doomed. Why? Government funding for people or projects will be a small percentage of the funding for pure mathematics. That's not much. And the future is pretty grim."

It is now nearly three years later and the Sage project currently has over 100 contributors and around 10,000 users. In November 2007, Sage won first place in the scientific category of the Trophées du Libre

`http://www.tropheesdulibre.org/`

a major international free software competition. Sage is funded by the US National Science Foundation, the US Department of Defense, the University of Washington, Microsoft Research, Google and private donations. Sage has new releases every two weeks, and typically 30–40 people contribute to each release. All new code contributions to Sage are peer reviewed, and every new function must be documented with tests that illustrate its usage. The documentation has over 50,000 lines of input examples.

ISSAC'08, July 20–23, 2008, Hagenberg, Austria.
ACM 978-1-59593-904-3/08/07.

2. WHAT IS SAGE?

1. a *huge distribution* of free open source mathematical software that is surprisingly easy to build from source,
2. a *set of interfaces* to most other mathematical software systems, and
3. a *new Python library* that fills in the numerous gaps in other open source math software included in Sage, unifies everything offering a smooth user experience, and provides a modern web-based graphical notebook interface with math typesetting and integrated 2D and 3D graphics.

3. WHAT MAKES SAGE UNIQUE?

3.1 Python and Cython

Sage is the first large general purpose mathematics software system that uses a mainstream programing language (Python) as the end user language. Python—easily one of the world's top 10 programming languages—is a powerful and beautiful modern interpreted programming language with an organized and professional developer base and millions of users. Sage also makes extensive use of a Python-to-C compiler called Cython (see `http://www.cython.org`).

> "It is our belief that this [Cython] is something missing in the world, and not for any good technical reasons. That it is eminently possible to have a language that gets down to the iron, runs at C speeds, and has no surprises in generated assembly, but at the same time guides you along to a clear, succinct and correct expression of complicated systems and algorithms." – Dan Gindikin

By building on Python and Cython we see that Sage has a tremendous longterm advantage over every other general purpose computer algebra system. Sage gets excellent support for compiled code, thousands of third party Python libraries, object serialization, superb database support, and excellent mature numerical libraries (numpy and scipy).

3.2 Building the Car Instead of Reinventing the Wheel

Instead of reinventing the wheel, Sage combines many of the best existing open source systems that have been developed over the last 40 years (about *5 million lines of code*) with about *250,000 lines of new code*. Every single copy of Sage includes all of the following software (and much much more):

- Algebra and calculus: Maxima, SymPy
- High precision arithmetic: GMP, MPFR, MPFI, quad-double, Givaro
- Commutative algebra: Singular
- Number theory: PARI, NTL, mwrank, ECM, FLINTQS, GMP-ECM
- Exact linear algebra: LinBox, IML
- Group theory: GAP
- Scientific computation: GSL, SciPy, NumPy, cvxopt
- Statistical computation: R
- Graphics (2d and 3d): Matplotlib, Tachyon3d, Jmol

Sage is thus the first system to combine together such a wide range of libraries and programs in a meaningful way. This huge range of programs is tied together using Python's excellent extensibility via C libraries and also in some cases using pseudo-tty's. Sage has a highly developed unified collection of pseudo-tty based interfaces that make it is possible to make extensive use of Maple, Mathematica, Magma, Matlab, GAP, Maxima, Singular, PARI, and many other systems from anywhere within a single Sage program.

Curious? If you also *want* a viable open source alternative to Magma, Maple, Mathematica or Matlab, drop everything, try out Sage now and become a Sage developer.

`http://www.sagemath.org`