Synergy Effects

A Sage Introduction – http://www.sagemath.org

Martin Albrecht (M.R.Albrecht@rhul.ac.uk)

Royal Holloway, University of London

Château de Villeneuve Saint Germain,
29.November 2007

Outline

Outline

Why? I

"Neubüser's Law"

"You can read [some] Theorem and its proof in [some] book in the library [. . .] then you can use [that] 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 (1993) (he started GAP in 1986).

Why? II

Neil Sloane

```
From: N. J. A. Sloane <XXXX@XXXXXXXXXXX.XXX.XXX>
Date: 8 Nov 2007 06:28
Subject: Re: dumb question about installing pari-gp with fink

I would like to thank everyone who responded to
my question about installing PARI on an iMAC.

The consensus was that it would be simplest to install sage,
which includes PARI and many other things.

I tried this and it worked!

Thanks!

Neil

(It is such a shock when things actually work!!)
```

Mission Statement

Mess with the Best

Provide an open source, high-quality, and free viable alternative to **Magma, Mathematica, Maple** and **MATLAB**.

In other words: create a unified mathematics software package for algebra, calculus, elementary to very advanced number theory, cryptography, numerical computation, commutative algebra, group theory, combinatorics, graph theory, exact linear algebra and more.

To achieve this do not reinvent the wheel but **reuse** as much **existing building blocks** as possible and make sure the result is **rigorously tested**, **easy to modify** by the end user and **very well documented**.

Also create a **helpful environment** for users to get help (mailinglists, irc-channel, meetings, coding sprints).

What is Sage?

Sage is a mathematics software package developed by a **worldwide** community of developers.

1. a **distribution** of the best free, open-source mathematics software available that is easy to compile or install from binaries.
2. an **interface** to most free and commercial mathematics software packages (e.g. Magma, Mathematica)
3. a **huge new library**, which uniformly covers the widest area of **functionality**, including several new implementations not yet found elsewhere.

Welcome to Sage:

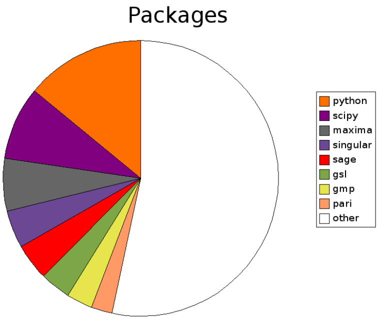
```
----------------------------------------------------------------------
| SAGE Version 2.8.12, Release Date: 2007-11-06
| Type notebook() for the GUI, and license() for information.
----------------------------------------------------------------------

sage: 2 + 3
5
```

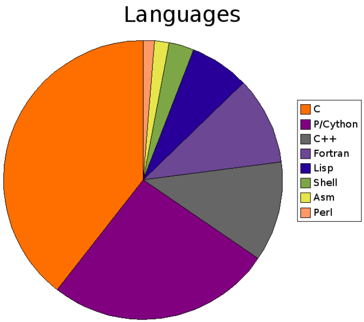
Outline

A Distribution I

Arithmetic	**GMP, MPFR, Givaro**
Commutative Algebra	**SINGULAR** (libSINGULAR)
Linear Algebra	**LinBox, M4RI, IML, fpLLL**
Cryptosystems	**GnuTLS, PyCrypto**
Integer Factorization	**FlintQS, ECM**
Group Theory	**GAP**
Combinatorics	**Symmetrica**
Graph Theory	**NetworkX**
Number Theory	**PARI, NTL, Flint, mwrank**
Numerical Computation	**GSL, Numpy, Scipy**
Calculus, Symbolic Comp.	**Maxima, Sympy**
Lattice Polytopes	**PALP**
User Interface	Sage Notebook, **jsmath, Moin wiki, IPython**
Graphics	**Matplotlib, Tachyon, libgd, Java3d**
Networking	**Twisted**
Databases	**ZODB, SQLite**, Python Pickles
Programming Language	**Python, Cython** (compiled)

A Distribution II





Overall VERY roughly 4.5 million lines of source code and estimated several hundred person-years.

A Unified Interface

- SAGE **interfaces to**: Axiom, GAP, GP/PARI, Kash, Macaulay2, Magma, Maple, Mathematica, MATLAB, Maxima, MuPad, Octave, Singular, etc.
- This gives SAGE a wide range of **functionality**.
- Unified **command completion and help**.

Example:

```
sage: x = gp('9+6')     # the GP/PARI math software
```

This fires up one copy of GP/PARI and sends the line `'sage[1] = 9+6'` to it

```
sage: x, x.name()
15, 'sage[1]'
sage: x.factor()
[3, 1; 5, 1]
```

Another Example:

```
sage: I.groebner_basis(algorithm='singular:slimgb')
```

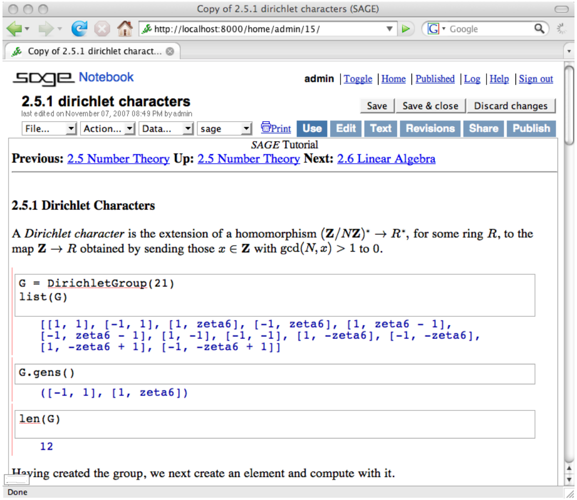
The Interface is a General Purpose Languange: Python

http://www.python.org and http://www.cython.org

- easy for you to define **your own data types** and methods on it (bitstreams, ciphers, r ings, whatever).
- very clean language that results in **easy to read code**.
- a **huge number of libraries**: statistics, networking, databases, bioinformatic, physic s, video games, 3d graphics, numerical computation (scipy), and serious "pure" mathematics (via Sage)
- easy to **use existing C/C++ libraries** from Python.
- Cython – an almost Python compiler.

```
On 11/1/07, Jack <XXXX.XXXXXXXXX@XXXXX.XXX> wrote:
> Is it possible to fetch information from a database (say, PostgreSQL)
> with sage?  If so, how is it done?  If not, is there  a workaround
> that people use?

Using Psycopg2 is the most popular way of using PostgreSQL from python,

see:
http://www.initd.org/tracker/psycopg/wiki/PsycopgTwo
...
```

Web-based Notebook Interface
public notebooks available at `http://www.sagenb.org`


Copy of 2.5.1 dirichlet characters (SAGE)
http://localhost:8000/home/admin/15/
Google
Copy of 2.5.1 dirichlet charact...
sage Notebook
admin | Toggle | Home | Published | Log | Help | Sign out
Save | Save & close | Discard changes
File... | Action... | Data... | sage | Print | Use | Edit | Text | Revisions | Share | Publish
Done

2.5.1 dirichlet characters
last edited on November 07, 2007 08:49 PM by admin

SAGE Tutorial
Previous: 2.5 Number Theory **Up:** 2.5 Number Theory **Next:** 2.6 Linear Algebra

2.5.1 Dirichlet Characters

A *Dirichlet character* is the extension of a homomorphism $(\mathbf{Z}/N\mathbf{Z})^* \to R^*$, for some ring R, to the map $\mathbf{Z} \to R$ obtained by sending those $x \in \mathbf{Z}$ with $\gcd(N, x) > 1$ to 0.

```
G = DirichletGroup(21)
list(G)
```

```
[[1, 1], [-1, 1], [1, zeta6], [-1, zeta6], [1, zeta6 - 1],
[-1, zeta6 - 1], [1, -1], [-1, -1], [1, -zeta6], [-1, -zeta6],
[1, -zeta6 + 1], [-1, -zeta6 + 1]]
```

```
G.gens()
```

```
([-1, 1], [1, zeta6])
```

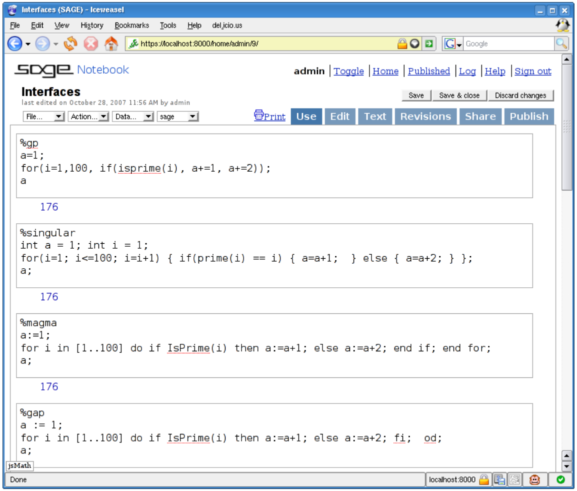
```
len(G)
```

```
12
```

Having created the group, we next create an element and compute with it.

- graphical user interface
- 2d plotting
- LaTeX typesetting
- web service (**AJAX**, SSL) inspired by Google Docs
- worksheet sharing
- worksheet up-/download

GUI to Many Mathematics Packages


Interfaces (SAGE) - Iceweasel
File Edit View History Bookmarks Tools Help del.icio.us
https://localhost:8000/home/admin/9/
sage Notebook
admin | Toggle | Home | Published | Log | Help | Sign out
Interfaces
last edited on October 28, 2007 11:56 AM by admin
Save | Save & close | Discard changes
File... Action... Data... sage
Print Use Edit Text Revisions Share Publish
jsMath
Done
localhost:8000

```
%gp
a=1;
for(i=1,100, if(isprime(i), a+=1, a+=2));
a
```

```
176
```

```
%singular
int a = 1; int i = 1;
for(i=1; i<=100; i=i+1) { if(prime(i) == i) { a=a+1;  } else { a=a+2; } };
a;
```

```
176
```

```
%magma
a:=1;
for i in [1..100] do if IsPrime(i) then a:=a+1; else a:=a+2; end if; end for;
a;
```

```
176
```

```
%gap
a := 1;
for i in [1..100] do if IsPrime(i) then a:=a+1; else a:=a+2; fi;  od;
a;
```

Examples:

- Pari
- Maxima
- Singular
- Gap
- Mathematica
- Maple
- ...

SAGE: Lots of new code

Unique lines of code and docstrings:

```
$ cat *.py */*.py ... */*/*/*.pxd |sort |uniq |wc -l
189082
```

Unique input lines of autotested examples:

```
$ cat *.py */*.py ... */*/*/*.pxd | grep "sage:"
              | sort |uniq |wc -l
26711
```

Doctesting coverage:

```
$ sage -coverage .
...
Overall weighted coverage score:  34.3%
Total number of functions:  17424
```

Also, close **collaboration with upstream** authors:

build fixes, porting, testing on a wide range of platforms, end user support, contribution (e.g. libSingular)

Live Demo

Outline

How Did Sage Come About?

Sage is:

1. A huge amount of **extremely hard mostly volunteer work** and
2. **Refusal to acknowledge reality**, i.e., that Sage is impossible.

Seriously, how did Sage really come about?

1997–1999	(**Berkeley**) **HECKE** – C++ (modular forms)
1999–2005	(**Berkeley**, **Harvard**) William Stein wrote over 25,000 lines of **Magma** code **Magma is incredibly powerful!** but got frustrated because: ■ Magma's, Mathematica's, and Maple's languages are **old-fashioned and painful** compared to Python. ■ We need to be able to **see inside and change anything** in our software in order to get our research done.
Feb 2005	**SAGE-0.1** released — a Python math library.
Feb 2006	**UCSD SAGE Days 1** – SAGE 1.0.
October 2006	**U Washington SAGE Days 2** workshop.
March 2007	**UCLA SAGE Days 3** workshop.
June 2007	**U Washington SAGE Days 4** workshop.
October 2007	**Clay Math Institute SAGE Days 5** workshop.
November 2007	**U. Bristol and the Heilbronn Institute SAGE Days 6**
Now	**SAGE-2.8.14**; over **100 contributors to SAGE**.

A Selection of Contributers

Alexandru Ghitza, Alex Clemesha, Andrey Novoseltsev, Bill Page, Bobby Moretti, Burcin Erocal, Carl Witty, Christian Wuthrich, Craig Citro, David Harvey, David Joyner, David Kohel, David Roe, Didier Deshommes, Dorian Raymer, Nathan Dunfield, Emily A. Kirkman, Gonzalo Tornaria, Iftikhar A. Burhanuddin, Jaap Spies , Jack Schmidt, Jason Grout, Jennifer S. Balakrishnan, Joel B. Mohler, Joe Wetherell, Jonathan Bober, Jonathan Hanke, Justin C. Walker, John Voight, Kate Minola, Kiran S. Kedlaya, Michael Abshoff, Martin Albrecht, Michel Van den Bergh, Mike Hansen, Nathan Dunfield, Nick Alexander, Nils Bruin, Pablo De Nápoli, Paul Dehaye, Clement Pernet, Robert Bradshaw, Robert L. Miller, Sean Howe, Soroosh Yazdani, Steven Sivek, Timothy Clemans, Tom Boothby, William Stein, Yi Qiang

[sage-devel]

200 people are subscribed to our development mailinglist.

Communication

sage-devel	development discussions, 213 subscribers, ca. 800 messages per month
sage-support	support requests, 187 subscribers, ca. 200 messages per month
sage-forum	general discussions, 148 subscribers, low traffic
sage-newbie	basic questions about e.g. programming, 33 subscribers, low traffic
#sage-devel@freenet	IRC channel, very busy during bug squashes, usually at least two Sage developers around

The Sage Development Guidelines

1. Don't work on anything that is not on trac. All enhancement proposals, bug reports and tasks are available on `http://trac.sagemath.org`.
2. All discussions happen in the open on public mailing lists and/or on a public chat channel (IRC).
3. One release per week on average: release often, release early. Several different release managers.
4. Changes happening to the main repository can be tracked in real time online
5. Code that goes into Sage is refereed by at least one other developer
6. If code is rejected by the release maintainer every developer can appeal to a board of editors of the **Journal of Sage** and they vote on the inclusion of the patch.

Also, developer's $HOME directories on our main development server `http://sage.math.washington.edu` are world readable through a web browser.

Outline

Shortcomings of Sage

1. There are currently about a thousand users of Sage; our goal is to have ten thousand serious users by 2009.
2. Sage is not very well known yet, but this event certainly helps!
3. Sage is not robust enough.
4. Sage is sometimes much slower than Magma, Mathematica, etc. (and sometimes faster, to be fair).
5. Sage is new – there are too many bugs.

Advantages of Sage

1. Sage is the only serious general purpose mathematics software that uses a mainstream programming language (Python).
2. Sage is the only program that allows you to use Maxima, Pari, GAP, Singular, Maple, Mathematica, Magma, etc., all together.
3. Sage has more functionality out of the box than any other open source mathematics software.
4. Sage has a huge, active, and well rounded developer community: [sage-devel] mailing list has over 200 subscribers, working very hard on everything from highly optimised arithmetic, to high school education, to computing modular forms.
5. Sage development is done in the open. You can read about why all decision are made, have input into decisions, see a list of every change anybody has made, etc.

Getting Started With Sage

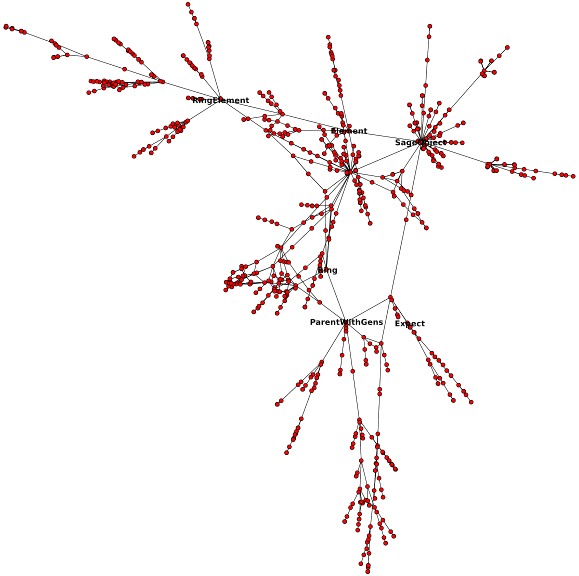
Ways to use Sage:

- download source; type 'make', wait, run (Linux, OSX, Solaris (soon))
- download binaries for a wide range of platforms
- download VMWare image (e.g. for Windows)
- try it online: `http://www.sagenb.org`

Documentation:

- Sage installation guide
- interactive tutorial
- comprehensive reference manual
- "How Do I Construct . . ." manual
- Sage programming guide
- "Sage programming for Newbies" book
- `http://wiki.sagemath.org`

Questions?



Thank You!