“An Introduction to Sage” or “Why I learned to stop worrying and love py-thon”

Mike Zabrocki
York University
What is Sage?

Sage is a computer algebra system based on python which ties together many specialized open source mathematics software packages.
What is Sage?

Sage is a computer algebra system based on python which ties together many specialized open source mathematics software packages.
What is Sage?

Sage is a computer algebra system based on python which ties together many specialized open source mathematics software packages.
What is Sage?

Sage is a computer algebra system based on python which ties together many specialized open source mathematics software packages.
What Sage does well

• fast/vast libraries of mathematical tools
• share mathematics
• free and easy to access
• based on python
What Sage does well

- fast/vast libraries of mathematical tools
- share mathematics
- free and easy to access
- based on python
»Every free computer algebra system I’ve tried has reinvented many times the wheel without being able to build the car.«
What Sage does well

• fast/vast libraries of mathematical tools

• **share mathematics**

• free and easy to access

• based on python
‘zero knowledge’ answer key

Step 1: teach students basic commands of Sage such as taking a coefficient in a taylor series or numerical integration

Step 2: If answer is function f(x) then give as answer key the coefficient of x^{100} in the taylor expansion

does it work

Example:

What is the generating function for the number of solutions to the equation

\[ x_1 + x_2 + 2x_3 + 3x_4 + x_5 = n \]

with \( x_i \geq 0 \) and where \( x_1 + x_2 + 2x_3 \) is even and \( x_1 + 2x_3 \) is less than or equal to 10, and \( x_4 \neq x_5 \)?

Answer key: the number of solutions with \( n = 100 \) is 23779
where the element $s_i$ represents the permutation which interchanges $i$ and $i + 1$. We will refer to the left cosets of $\tilde{S}_n/S_n$ as affine Grassmannian elements and they will be identified with their minimal length coset representatives, that is, the elements of $w \in \tilde{S}_n$ such that either $w = id$ or $s_0$ is the only elementary transposition such that $\ell(ws_0) < \ell(w)$.

**Remark 1.7.** The definition of affine Grassmannian elements are the special case of a more general definition. The $l$-Grassmannian elements are the minimal length coset representatives of $\tilde{S}_n/S_n^l$ where $S_n^l$ is the group generated by $\{s_0, s_1, s_2, \ldots, s_{n-1}\}\{s_l\}$ and the affine Grassmannian elements are the 0-Grassmannian elements. Due to the cyclic symmetry of the affine type $A$ Dynkin diagram, these constructions are of course all equivalent.

**Sage Example 1.8.** We can create the affine symmetric group and its generators in SAGE as

```sage
sage: W = WeylGroup(['A',4,1])
sage: S = W.simple_reflections()
sage: [s.reduced_word() for s in S]
[[0], [1], [2], [3], [4]]
```

For a given element, we can ask for its reduced word or create it from a word in the generators and ask whether it is Grassmannian:

```sage
sage: w = W.an_element(); w
[ 2 0 0 1 -2]
[ 2 0 0 0 -1]
[ 1 1 0 0 -1]
[ 1 0 1 0 -1]
[ 1 0 0 1 -1]
sage: w.reduced_word()
[0, 1, 2, 3, 4]
sage: w = W.from_reduced_word([2,1,0])
sage: w.is_affine_grassmannian()
True
```
share mathematics - through worksheets
share mathematics - through collaborations
share mathematics - by contributing to Sage

Sage Days: turning first time users into developers
What Sage does well

- fast/vast libraries of mathematical tools
- share mathematics
- free and easy to access
- based on python
Unable to determine the Host ID of this system. Please contact Maplesoft Support.
Why one should use sage

• fast/vast libraries of mathematical tools
• share mathematics
• free and easy to access
• based on python
YOU'RE FLYING! HOW?

I DUNNO...
DYNAMIC TYPING?
WHITESPACE?

COME JOIN US!
PROGRAMMING IS FUN AGAIN!
IT'S A WHOLE NEW WORLD UP HERE!

BUT HOW ARE YOU FLYING?

I JUST TYPED
import antigravity

THAT'S IT?

...I ALSO SAMPLED EVERYTHING IN THE MEDICINE CABINET FOR COMPARISON.

BUT I THINK THIS IS THE PYTHON.
Sage is based on python (a very nice language) and has syntax close to mathematics

\[
\sum_{i=1}^{10} i^2
\]

\[
\text{sum}(i^2 \text{ for } i \text{ in range}(1,11))
\]

\[
\{17x|x \in \{0, 1, 2, \ldots, 9\} \text{ and } i \text{ is odd} \}
\]

\[
[17\times x \text{ for } x \text{ in range}(10) \text{ if } x\%2==1]
\]
Who Sage is not good for

- experts
- beginners
Who Sage is not good for

- experts doing high efficiency specialized computation
- beginners who are turned off by user friendly issues
Who Sage is not good for

- experts doing high efficiency specialized computation
- beginners who are turned off by user friendly issues
CHECK IT OUT! AN ELECTRIC LONGBOARD!

SWEET!

I FEEL LIKE WE'RE MISSING SOMETHING...

Yeah...

SKATING UPHILL LIKE THIS IS AMAZING. YEARS OF GLIDING DOWNHILL AND PUSHING UPHILL, AND NOW SUDDENLY IT'S GLIDING BOTH WAYS.

IT'S LIKE GOING FROM C TO PYTHON. YOU DON'T REALIZE HOW MUCH TIME YOU WERE SPENDING ON THE BORING PARTS UNTIL YOU DON'T HAVE TO DO THEM ANYMORE.

BUT CODING C OR ASSEMBLY MAKES YOU A BETTER PROGRAMMER.

MAYBE THE BORING PARTS BUILD CHARACTER.

WHAM!

Yeah... but it depends how you want to spend your life. See, my philosophy is—
CHECK IT OUT! AN ELECTRIC LONGBOARD!

SWEET!

I FEEL LIKE WE'RE MISSING SOMETHING...

YEY...

WHAM!

SKATING UPHILL LIKE THIS IS AMAZING. YEARS OF GLIDING DOWNHILL AND PUSHING UPHILL, AND NOW SUDDENLY IT'S GLIDING BOTH WAYS.

It's like going from C to Python. You don't realize how much time you were spending on the boring parts until you don't have to do them anymore.

But coding C or assembly makes you a better programmer.

Yeah... but it depends how you want to spend your life. See, my philosophy is—

Maybe the boring parts builds character.
Who Sage is not good for

- experts doing high efficiency specialized computation
- beginners who are turned off by user friendly issues
for i from 1 to 10:
    print i
for i from 1 to 10:
    print i

Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "_sage_input_3.py", line 10, in <module>
    exec compile(u'open("___code___\.py","w")\n''\n    File "", line 1, in <module>

    File "/private/var/folders/_3/qq_ptsxd0bd645cgmt_ssrm0000gn/T/tmpUXQVWB/___code___\.py", line 3
      for i from _sage_const_1 to _sage_const_10 :

SyntaxError: invalid syntax
for i from 1 to 10:
    print i

Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "_sage_input_3.py", line 10, in <module>
    exec compile(u"open("__code__.py","w").write("# -*- coding: utf-8 -*-\n" + support.preparse_worksheet_cell(support.load_worksheet_cell(
  File "", line 1, in <module>

  File "/private/var/folders/_3/qq_ptsxd0bd645cgmgt_ssrm0000gn/T/tmpUxQVWB/__code__.py", line 3
    for i from _sage_const_1  to _sage_const_10 :

SyntaxError: invalid syntax

TMI!
for from 1 to 10:
    print i

'for i from ...' Bad syntax in for loop. Help on for loops
for i in range(10):
    print factor(x^i-1)
for i in range(10):
    print factor(x**i-1)

Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "_sage_input_9.py", line 10, in <module>
    exec compile(u'open("__code__.py","w").write("# -*- coding: utf-8 -*-\n" + _support_.preparse_worksheet_cell(r"%%s\n" % code))', "", line 1, in <module>
  File "/private/var/folders/_3/qq_ptsx0bd645cgmgt_ssrm0000gn/T/tmpAkQLpF/__code__.py", line 3, in <module>
    exec compile(u'for i in range(_sage_const_10 ):\n    print factor(x**i-_sage_const_1 )', ", line 2, in <module>
  File "/Applications/sage/local/lib/python2.7/site-packages/sage/rings/arith.py", line 2478, in factor
    return n.factor(**kwds)
  File "expression.pyx", line 8466, in sage.symbolic.expression.Expression.factor (sage/symbolic/expression.cpp:364)
  File "polynomial_element.pyx", line 2892, in sage.rings.polynomial.polynomial_element.Polynomial.factor (sage.../arith.py:71)
ValueError: factorization of 0 not defined
for i in range(10):
    print factor(x^i-1)

Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
    File "_sage_input_9.py", line 10, in <module>
        exec compile(u"open("code"."w").write("# -*- coding: utf-8 -*-\n" + _support_.preparse_worksheet_code(\n            _code_, 2)
        File "", line 1, in <module>

    File "/private/var/folders/_3/qq_ptsxd0bd645cgmgt_ssrm0000gn/T/tmpAkQLpF/__code__.py", line 3, in <module>
        exec compile(u'for i in range(_sage_const_10):
            print factor(x**i-_sage_const_1)
        File ", line 2, in <module>

    File "/Applications/sage/local/lib/python2.7/site-packages/sage/rings/arith.py", line 2478, in factor
        return n.factor(**kwds)
    File "expression.pyx", line 8466, in sage.symbolic.expression.Expression.factor (sage/symbolic/expression.cpp:9376)
    File "polynomial_element.pyx", line 2892, in sage.rings.polynomial.polynomial_element.Polynomial.factor (sage/rings/polynomial/polynomial_element.cpp:3817)   ValueError: factorization of 0 not defined

WTF?
for i in range(10):
    print i
Demonstration

1. How to get started
2. How to get help
3. How to do something cool
Sage is a free open-source mathematics software system licensed under the GPL. It combines the power of many existing open-source packages into a common Python-based interface.

Mission: Creating a viable free open source alternative to Magma, Maple, Mathematica and Matlab.

The new SageMathCloud is in Beta Test. Work with Sage, or run Python, R, GAP, M2 and more in the Cloud.

Try Sage Online
other: Cloud, KAIST, Test
devel: alpha

Download 5.12
Changelog · Source 5.12 · Packages

Help/Documentation
Video · Lists · Tutorial · FAQ · Ask

Feature Tour
Quickstart · Research · Graphics

Library
Testimonials · Books · Publications · Press Kit

Search
Welcome!

Sage is a different approach to mathematics software.

The Sage Notebook

With the Sage Notebook anyone can create, collaborate on, and publish interactive worksheets. In a worksheet, one can write code using Sage, Python, and other software included in Sage.

General and Advanced Pure and Applied Mathematics

Use Sage for studying calculus, elementary to very advanced number theory, cryptography, commutative algebra, group theory, graph theory, numerical and exact linear algebra, and more.

Use an Open Source Alternative

By using Sage you help to support a viable open source alternative to Magma, Maple, Mathematica, and MATLAB. Sage includes many high-quality open source math packages.

Use Most Mathematics Software from Within Sage

Sage makes it easy for you to use most mathematics software together. Sage includes GAP, GP/PARI, Maxima, and Singular, and dozens of other open packages.

Use a Mainstream Programming Language

You work with Sage using the highly regarded scripting language Python. You can write programs that combine serious mathematics with anything else.

Acknowledgement

The Sage Notebook is based upon work supported by the National Science Foundation under grants DMS-0821725, DMS-1020378, DMS-0713225, DMS-0555776, DMS-0545904, DMS-0838212, DMS-0757627, DUE-1020378, DUE-1022574, DMS-1015114, etc. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. See also http://sagemath.org/development-ack.html.
Welcome to Sage! You can create a new worksheet, view published worksheets, or read the documentation.
Untitled

last edited Nov 3, 2013 5:52:44 AM by Mike_Zabrocki

Rename worksheet

Please enter a name for this worksheet.

Untitled

Rename
the meaning of life, universe ...

6 * 7

42
Sage is a free open-source mathematics software system licensed under the GPL. It combines the power of many existing open-source packages into a common Python-based interface.

Mission: Creating a viable free open source alternative to Magma, Maple, Mathematica and Matlab.

The new SageMathCloud is in Beta Test.
Work with Sage, or run Python, R, GAP, M2 and more in the Cloud.

Try Sage Online
other: Cloud, KAIST, Test
devel: alpha

Download 5.12
Changelog · Source 5.12 · Packages

Help/Documentation
Video · Lists · Tutorial · FAQ · Ask

Feature Tour
Quickstart · Research · Graphics

Library
Testimonials · Books · Publications · Press Kit

Search
Download Binaries and Media

There is a low traffic announcement mailing list and an RSS feed to stay informed about new releases.

Email: [Enter email] Subscribe

Please select a download server close to your location below.

Africa
- Stellenbosch University, South Africa
- Polytechnic of Namibia

America, North
- Université du Québec à Montréal, Québec, Canada
- Hub.co Services Inc., NY, USA
- XMission, Utah, USA
- Simon Fraser University, B.C., Canada
- University of Washington, Seattle, WA, USA

America, South
- University of São Paulo, Brazil
- Universidade Federal do Paraná, Brazil

Asia
- Jember University, Java, Indonesia
- Yandex, Russia
- Huazhong University of Science and Technology, Hubei, China
- KAIST, Republic of Korea
- Indian Institute of Technology Madras, India
- Riken, Japan
- Huazhong University of Science and Technology, China
- Tsinghua University, Beijing, China
- WIDE Project Tsukuba NCC, Japan
- Nanyang Technological University, Singapore

Australia
- University of Sydney

Europe
- GARR, Italy
- Freie Universität Berlin, Germany
- Mirrorservice Network, United Kingdom
- Tallinn, Estonia
- SWITCHmirror, Zurich, Switzerland
- Université Pierre et Marie Curie, Paris, France
- Universidade do Porto, Portugal
- Sofia, Bulgaria

Distributed and P2P
- BitTorrent — is a P2P protocol for fast and reliable downloads of huge files over the Internet (read more).
- Metalinks — provide fast, stable and resumeable downloads via a download client (read more).
Sage is a free open-source mathematics software system licensed under the GPL. It combines the power of many existing open-source packages into a common Python-based interface.

Mission: Creating a viable free open source alternative to Magma, Maple, Mathematica and Matlab.

The new SageMathCloud is in Beta Test. Work with Sage, or run Python, R, GAP, M2 and more in the Cloud.

Try Sage Online
other: Cloud, KAIST, Test
devel: alpha

Download 5.12
Changelog · Source 5.12 · Packages

Help/Documentation
Video · Lists · Tutorial · FAQ · Ask

Feature Tour
Quickstart · Research · Graphics

Library
Testimonials · Books · Publications · Press Kit

Search
Help and Support

- Sage standard documentation
- Further resources
- Support and discussion groups
- uSage User Groups

Sage standard documentation

The Sage standard documentation consists of the following documents, in both HTML and PDF versions.

View the online documentation.

- A Tour of Sage, PDF — a brief tour of some of Sage's features
  - Sage en quelques mots (Français), PDF
- Tutorial (Printed & Bound), PDF — information for beginners, recommended
  - Sage Tutorial (Deutsch), PDF
  - Sage thematische Anleitungen (Deutsch), PDF
  - Manual de Sage para principiantes (Español)
  - Introduccion: Matemáticas Elementales con Sage (Español)
  - Tutoriel Sage (Français), PDF
  - учебное пособие Sage (русский язык), PDF
  - Persian Tutorial (Farsi): امورش فارسی (PDF) (DOCX)
- Thematic Tutorials, PDF
  - Introduction to Sage
    - Logging on to a Sage Server and Creating a Worksheet (PREP)
    - Introductory Sage Tutorial (PREP)
    - Tutorial: Using the Sage notebook, navigating the help system, first exercises
    - Sage's main tutorial
- Calculus and plotting
  - Tutorial: Symbolics and Plotting (PREP)
  - Tutorial: Calculus (PREP)
  - Tutorial: Advanced-2D Plotting (PREP)
- Algebra
  - Linear Programming (Mixed Integer)
  - Group Theory and Sage
  - Lie Methods and Related Combinatorics in Sage
- Number Theory
  - Number Theory and the RSA Public Key Cryptosystem
  - Introduction to the p-adics
- Combinatorics
  - Introduction to combinatorics in Sage
- Algebraic Combinatorics
  - Tutorial: Symmetric Functions
  - Lie Methods and Related Combinatorics in Sage
  - Abelian Sandpile Model
- Programming and Design
  - Tutorial: Sage Introductory Programming (PREP)
  - Tutorial: Programming in Python and Sage
  - Tutorial: Comprehensions, Iterators, and Iterables
  - Tutorial: Objects and Classes in Python and Sage
  - Functional Programming for Mathematicians
- Modeling Mathematics on a computer
  - How to implement new algebraic structures in Sage
  - Elements, parents, and categories in Sage: a (draft of) primer
  - Implementing a new parent: a (draft of) tutorial
- FAQ, PDF — frequently asked questions with answers
- Reference Manual, PDF — details about built-in functions
- Installation Guide, PDF — helps you install Sage (download)
- Developer's Guide, PDF — guidelines for current and prospective Sage developers
- Constructions (Printed & Bound), PDF — describes how to explain specific mathematical objects to Sage
- Numerical Sage, PDF — how to do numerical computing with Sage
- Explicit Methods in Number Theory, PDF — computing with number fields and modular forms

You can download the entire standard documentation as a compressed file.
\[ f(x) = e^{-x} \sin(x) \]
\[ \dot{f}(x; 0) = x + \mathcal{O}(x^2) \]

\[
\frac{\sin(x^2 + y^2)}{(1 + y + xy)} > 0
\]

\[
\sin(x^2 + y^2) \cdot \cos(x + y^2) \cdot \sin(y)
\]