Approach of lazy calculations in application to topological graph indices in Sage. Introduction to a new molecular class.

Alexander Vasilyev

University of Primorska. Andrej Marušič Institute.

May 17, 2012

The Sage Project

Mission Statement

Create a viable free open source alternative to Magma, Maple, Mathematica, and Matlab

A "viable alternative" will have...

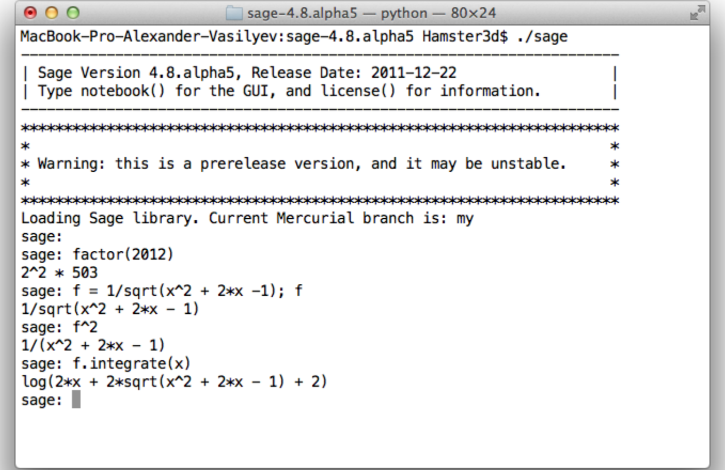
- The mathematical features of Magma, Maple, Mathematica, and Matlab with comparable speed.
- Beautiful interactive 2d and 3d graphics.
- A notebook interface and an IDE.
- Many books and Commercial support (e.g., customized notebook servers)

What is Sage

1. **A self-contained distribution** of over 90 open source packages that is easy to build from source.
2. **Interfaces** that smoothly tie together all these libraries and packages.
3. **A new library** that implements novel algorithms. About a half million lines of code written by a worldwide community of about 200 people over the last 6 years.

Sage in use

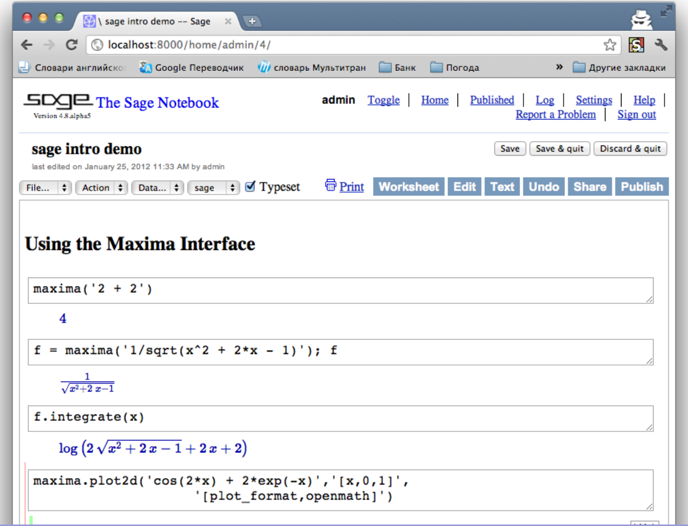
Command Line Sage



```
MacBook-Pro-Alexander-Vasilyev:sage-4.8.alpha5 Hamster3d$ ./sage
----------------------------------------------------------------------
| Sage Version 4.8.alpha5, Release Date: 2011-12-22                  |
| Type notebook() for the GUI, and license() for information.        |
----------------------------------------------------------------------
**********************************************************************
*                                                                    *
* Warning: this is a prerelease version, and it may be unstable.     *
*                                                                    *
**********************************************************************
Loading Sage library. Current Mercurial branch is: my
sage:
sage: factor(2012)
2^2 * 503
sage: f = 1/sqrt(x^2 + 2*x -1); f
1/sqrt(x^2 + 2*x - 1)
sage: f^2
1/(x^2 + 2*x - 1)
sage: f.integrate(x)
log(2*x + 2*sqrt(x^2 + 2*x - 1) + 2)
sage: 
```

Sage in use

Sage notebook



sage intro demo

last edited on January 25, 2012 11:33 AM by admin

Using the Maxima Interface

```
maxima('2 + 2')
```

4

```
f = maxima('1/sqrt(x^2 + 2*x - 1)'); f
```

$\frac{1}{\sqrt{x^2+2\,x-1}}$

```
f.integrate(x)
```

$\log\left(2\sqrt{x^2+2\,x-1}+2\,x+2\right)$

```
maxima.plot2d('cos(2*x) + 2*exp(-x)','[x,0,1]',
                         '[plot_format,openmath]')
```

Try Sage

Sage is crossplatform. Supported platforms:

- Linux
- Apple Mac OS X
- Solaris
- Microsoft Windows (run on virtual machine with Linux)
- Live CD

The best way to try Sage is using on-line Sage notebook.

Links

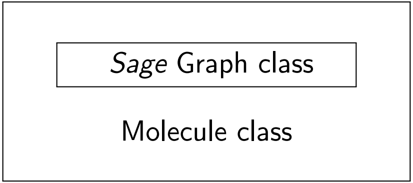
Sage homepage	www.sagemath.org
Sage Notebook	www.sagenb.org
Sage documentation	www.sagemath.org/help.html
Bug report and contribution	trac.sagemath.org/sage_trac/report

Math Chemistry package willl include:

Atom-Bond Connectivity Index; Balaban J Index; Balaban-like indices; Degree Distance; Reverse Degree Distance; Detour Index; Hyper-Detour Index; Wiener Index; Hyper-Wiener Indices; Reverse Wiener Index; Modified Wiener index; Variable Wiener index; Terminal Wiener index; Edge Wiener index; y-Wiener index; Reciprocal complimentary Wiener index; Reciprocal reverse Wiener index; Estrada Index; Laplacian Estrada index; Distance Estrada index; Eccentric Connectivity Index; Eccentric Distance Sum; (Schultz) Molecular Topological Index; Reciprocal Molecular Topological Index; Randic Index; (Zeroth-order) General Randic Index; Variable Randic Index; Augmented Zagreb Index; M1, M2 Zagreb index; Path-Zagreb index; Zagreb Coindices; Reformulated Zagreb indices; Edge-Szeged index; Szeged Index; Revised Szeged Index; (Padmakar-Ivan) PI Index; Vertex PI Index; Connectivity Index; Matching number; Number of perfect matchings; 1th,2th,3th Geometric-Arithmetic index; Harary Index; Kirchhoff Index; Hosoya index; Merrifield-Simmons Indices; **Spectrum, spread, spectral moments, characteristic polynomial, Energy** (Adjacency, Laplacian, Detour, Distance, Reciprocal distance, Normalized laplacian, Signless laplacian, Resistance distance, Harary); **Polynomials** (Hosoya, Omega, Eccentric connectivity, Wiener, PI, Szeged, Edge-Szeged); Anti-Kekule number; Anti-Forcing Number;

First attempt: classical object-oriented approach

Molecule class inherited from the Sage Graph class


Sage Graph class

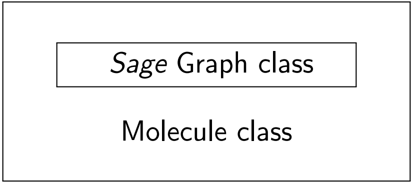
Molecule class

+ Advantages

- Easy to develop
- Inherited methods such as
  - $Graph.show() \rightarrow Molecule.show()$
  - $Graph.spectrum() \rightarrow Molecule.spectrum()$
  - $Graph.min\_cut() \rightarrow Molecule.min\_cut()$
  - $\ldots$
- The package could be included into main Sage repository

First attempt: classical object-oriented approach

Molecule class inherited from the Sage Graph class



Sage Graph class

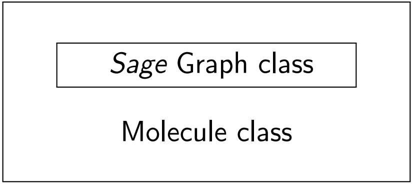
Molecule class

+ Advantages

- Easy to develop
- Inherited methods such as
  - $Graph.show() \rightarrow Molecule.show()$
  - $Graph.spectrum() \rightarrow Molecule.spectrum()$
  - $Graph.min\_cut() \rightarrow Molecule.min\_cut()$
  - ...
- The package could be included into main Sage repository

First attempt: classical object-oriented approach

Molecule class inherited from the Sage Graph class



Sage Graph class

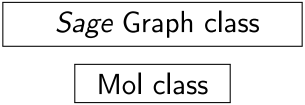
Molecule class

– Disadvantages

- Performance
- Dependence on the framework

Second attempt: lazy calculations

We calculate data only when it is needed and then we save the results.



Sage Graph class

Mol class

+ Advantages

- Performance
- Independence on the framework (the package can be used in any Python-based project)

− Disadvantages

- Hard to code, easy to get bugs
- The package will not be included into main Sage boundle

Second attempt: lazy calculations

We calculate data only when it is needed and then we save the results.

Sage Graph class

Mol class

+ Advantages

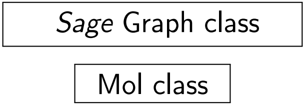
- Performance
- Independence on the framework (the package can be used in any Python-based project)

− Disadvantages

- Hard to code, easy to get bugs
- The package will not be included into main Sage boundle

Second attempt: lazy calculations

We calculate data only when it is needed and then we save the results.



Sage Graph class

Mol class

+ Advantages

- Performance
- Independence on the framework (the package can be used in any Python-based project)

− Disadvantages

- Hard to code, easy to get bugs
- The package will not be included into main Sage boundle

Tests

Graphs for test: all connected simple graps of 7, 8, 9 vertices

	7c	8c	9c
number of graphs	853	11117	261080

1. Zagreb M1 and M2 indices:

$$M1 = \sum_{v \in V(G)} d_v^2 \qquad M2 = \sum_{u \sim v \in V(G)} d_v d_u \tag{1}$$

2. Molecular Topological Index MTI, Degree Distance DD, Reverse Degree Distance rDD:

$$MTI = \sum_{v \in V(G)} (A + D) d_v \qquad DD = \sum_{v \in V(G)} d_v \sum_{u \in V(G)} d_{u,v} \tag{2}$$

$$rDD = 2(n-1) m d_G - DD$$

Where d_v is degree of v, A is adjacency matrix, D is distance matrix, $d_{u,v}$ is distance between u and v, d_G is diameter of G, n is number of vertices, m is number of edges

Tests

Graphs for test: all connected simple graps of 7, 8, 9 vertices

	7c	8c	9c
number of graphs	853	11117	261080

1. Zagreb M1 and M2 indices:

$$M1 = \sum_{v \in V(G)} d_v^2 \quad M2 = \sum_{u \sim v \in V(G)} d_v d_u \tag{1}$$

2. Molecular Topological Index MTI, Degree Distance DD, Reverse Degree Distance rDD:

$$MTI = \sum_{v \in V(G)} (A + D) d_v \quad DD = \sum_{v \in V(G)} d_v \sum_{u \in V(G)} d_{u,v} \tag{2}$$

$$rDD = 2(n-1) m d_G - DD$$

Where d_v is degree of v, A is adjacency matrix, D is distance matrix, $d_{u,v}$ is distance between u and v, d_G is diameter of G, n is number of vertices, m is number of edges

Tests

Graphs for test: all connected simple graps of 7, 8, 9 vertices

	7c	8c	9c
number of graphs	853	11117	261080

1. Zagreb M1 and M2 indices:

$$M1 = \sum_{v \in V(G)} d_v^2 \quad M2 = \sum_{u \sim v \in V(G)} d_v d_u \tag{1}$$

2. Molecular Topological Index MTI, Degree Distance DD, Reverse Degree Distance rDD:

$$\begin{gathered} MTI = \sum_{v \in V(G)} (A + D) d_v \quad DD = \sum_{v \in V(G)} d_v \sum_{u \in V(G)} d_{u,v} \\ rDD = 2(n-1) m d_G - DD \end{gathered} \tag{2}$$

Where d_v is degree of v, A is adjacency matrix, D is distance matrix, $d_{u,v}$ is distance between u and v, d_G is diameter of G, n is number of vertices, m is number of edges

Tests. First test.

Performance test of *Molecule* and *Mol* classes.
Time (in seconds) of calculation Zagreb M1 and M2 indeces for all connected graphs of 7, 8, 9 vertices.

	7c	8c	9c
Molecule	0.48	7.36	206.7
Mol	0.07	1.17	33.0
multiplier	6.85	6.29	6.26

The new *Mol* class is ≈**6 times** faster

Tests. First test.

Performance test of *Molecule* and *Mol* classes.
Time (in seconds) of calculation Zagreb M1 and M2 indeces for all connected graphs of 7, 8, 9 vertices.

	7c	8c	9c
Molecule	0.48	7.36	206.7
Mol	0.07	1.17	33.0
multiplier	6.85	6.29	6.26

The new *Mol* class is ≈**6 times** faster

Tests. Second test.

Performance test of *Molecule* and *Mol* classes.
Time (in seconds) of calculation Molecular Topological Index, Degree Distance, Reverse Degree Distance for all connected graphs of 7, 8, 9 vertices.

	7c	8c	9c
Molecule	0.86	12.72	336.34
Mol	0.75	10.33	263.07
multiplier	1.14	1.23	1.27

The new *Mol* class is ≈**1.2 times** faster

Tests. Second test.

Performance test of *Molecule* and *Mol* classes.
Time (in seconds) of calculation Molecular Topological Index, Degree Distance, Reverse Degree Distance for all connected graphs of 7, 8, 9 vertices.

	7c	8c	9c
Molecule	0.86	12.72	336.34
Mol	0.75	10.33	263.07
multiplier	1.14	1.23	1.27

The new *Mol* class is ≈**1.2 times** faster

Easy link to Sage and NetworkX

Mol().sage_graph()

Mol().NX_graph()

Thank you!