

# 2014-05-07.sagews

May 7, 2014

## Contents

<b>1</b>	<b>Math 480b Sage Course</b>	<b>1</b>
1.1	Overview of Sage	1
1.2	May 7, 2014	1
1.3	Distribution heres whats included	1
1.4	Distribution discuss some notable components of Sage that dont require a Ph.D. to appreciate	2

## 1 Math 480b Sage Course

### 1.1 Overview of Sage

### 1.2 May 7, 2014

Screencast: REMIND ME!?

Plan

- Questions
- Build Sage from source: quick checkup
- Sync changes: some remarks and a live stress test
- Overview of what is in Sage continued

### 1.3 Distribution heres whats included

(see also <http://sagemath.org/packages/standard/>)

/sageatlas-3.10.1.tar.bz2 gsl-1.15.tar.bz2 Pillow-2.2.2.tar.gz boehm\_gc-7.2d.tar.bz2 iconv-1.14.tar.bz2 polybori-0.8.3.tar.bz2 boost\_cropped-1.52.0.tar.bz2 iml-1.0.3.tar.bz2 polytopes\_db-20120220.tar.bz2 bzip2-1.0.6.tar.gz ipython-0.13.2.tar.bz2 ppl-1.1pre9.tar.bz2 cddlib-094f.tar.bz2 jinja2-2.5.5.tar.bz2 pycrypto-2.6.1.tar.gz cephes-2.8.tar.bz2 jmol-12.3.27.tar.bz2 pygments-1.3.1.tar.bz2 cliquer-1.21.tar.bz2 lcalc-1.23.tar.bz2 pynac-0.3.0.tar.bz2 conway\_polynomials-0.4.tar.bz2 libfplll-4.0.4.tar.bz2 python-2.7.5.tar.bz2 cvxopt-1.1.6.tar.bz2 libgap-4.6.4.tar.bz2 r-3.0.2.tar.bz2 cython-0.19.1.tar.bz2 libm4ri-20130416.tar.bz2 ratpoints-2.1.3.tar.bz2 docutils-0.7.tar.bz2 libm4rie-20130416.tar.bz2 readline-6.2.tar.bz2 ecl-12.12.1.tar.bz2 libpng-1.2.35.tar.bz2 rpy2-2.3.8.tar.gz eclib-20120830.tar.bz2 linbox-1.3.2.tar.bz2 rubiks-20070912.tar.bz2 ecm-6.4.4.tar.bz2 lrcalc-1.1.6.tar.bz2 sagenb-0.10.8.2.tar ellip-tic\_curves-0.7.tar.bz2 matplotlib-1.2.1.tar.bz2 sagemath-2.3.4.tar.bz2 fflas\_ffpack-1.6.0.tar.bz2 maxima-5.29.1.tar.bz2 scipy-0.12.0.tar.bz2 flint-2.3.tar.bz2 mpc-1.0.tar.bz2 scons-1.2.0.tar.bz2 flintqs-20070817.tar.bz2 mpfi-1.5.1.tar.bz2

setuptools-0.6.16.tar.bz2 freetype-2.5.2.tar.bz2 mpfr-3.1.2.tar.bz2 singular-3.1.5.tar.bz2 gap-4.6.4.tar.bz2 mpir-2.6.0.tar.bz2 sphinx-1.1.2.tar.bz2 gd-2.0.35.tar.bz2 mpmath-0.17.tar.bz2 sqlalchemy-0.5.8.tar.bz2 gdmodule-0.56.tar.bz2 ncurses-5.9.tar.bz2 sqlite-3.7.17.tar.bz2 genus2reduction-0.3.tar.bz2 networkx-1.8.1.tar.gz symmetrica-2.0.tar.bz2 gf2x-1.1.1.tar.bz2 ntl-5.5.2.tar.bz2 sympow-1.018.1.tar.bz2 gfan-0.5.tar.bz2 numpy-1.7.0.tar.bz2 sympy-0.7.4.tar.gz git-1.8.4.4.tar.bz2 palp-2.1.tar.bz2 tachyon-0.98.9.tar.bz2 givaro-3.7.1.tar.bz2 pari-2.5.5.tar.bz2 zlib-1.2.8.tar.bz2 glpk-4.44.tar.bz2 patch-2.5.9.tar.bz2 zn\_poly-0.9.tar.bz2 graphs-20120404.tar.bz2 pexpect-2.0.tar.bz2

## 1.4 Distribution discuss some notable components of Sage that dont require a Ph.D. to appreciate

More notable components:

- atlas: Automatically Tuned Linear Algebra software = operations with floating point matrices quickly; used by numpy

```
import numpy.random, numpy
n = 1000
a = numpy.random.rand(n,n) # see http://docs.scipy.org/doc/numpy/\
    reference/routines.random.html
b = numpy.random.rand(n,n)
%time c = a*b # component-wise multiplication
%time d = numpy.dot(a,b) # matrix multiplication (probably uses ATLAS \
    heavily)
CPU time: 0.00 s, Wall time: 0.00 s
CPU time: 0.29 s, Wall time: 0.00 s
```

- cvxopt: big convex optimization package; Convex minimization has applications in a wide range of disciplines, such as automatic control systems, estimation and signal processing, communications and networks, electronic circuit design, data analysis and modeling, statistics (optimal design), and finance. Wikipedia

Example here: <http://cvxopt.org/examples/tutorial/lp.html>

```
%python # disable preparsing in this cell; alternatively, do RealNumber=\
    float
from cvxopt import matrix, solvers
A = matrix([ [-1.0, -1.0, 0.0, 1.0], [1.0, -1.0, -1.0, -2.0] ])
b = matrix([ 1.0, -2.0, 0.0, 4.0 ])
c = matrix([ 2.0, 1.0 ])
sol=solvers.lp(c,A,b)
print(sol['x'])
```

	pcost	dcost	gap	pres	dres	k/t
0:	2.6471e+00	-7.0588e-01	2e+01	8e-01	2e+00	1e+00
1:	3.0726e+00	2.8437e+00	1e+00	1e-01	2e-01	3e-01
2:	2.4891e+00	2.4808e+00	1e-01	1e-02	2e-02	5e-02
3:	2.4999e+00	2.4998e+00	1e-03	1e-04	2e-04	5e-04
4:	2.5000e+00	2.5000e+00	1e-05	1e-06	2e-06	5e-06

```

5: 2.5000e+00 2.5000e+00 1e-07 1e-08 2e-08 5e-08
Optimal solution found.
[ 5.00e-01]
[ 1.50e+00]

```

- cython: Cython is an optimising static compiler for both the Python programming language and the extended Cython programming language. It makes writing C extensions for Python as easy as Python itself. Cython gives you the combined power of Python and C. Hundreds of thousands of lines of Sage are written in Cython; we learn the basics very soon.

```

%python
def f0(n):
    k = 0
    for m in range(n):
        k += 2*m+1
    return k

```

```

%cython
def f1(int n):
    cdef long m, k=0
    for m in range(n):
        k += 2*m+1
    return k

```

```

%time f0(10^7)
1000000000000000

```

CPU time: 3.63 s, Wall time: 3.08 s

```

%time f1(10^7)    # very fast, C semantics (so silent overflow), etc.
1000000000000000
CPU time: 0.01 s, Wall time: 0.00 s

```

```

3.63/0.01
363.0000000000000

```

- GAP: Groups, Algorithms and Programming a System for Computational Discrete Algebra

GAP is a system for computational discrete algebra, with particular emphasis on Computational Group Theory. GAP provides a programming language, a library of thousands of functions implementing algebraic algorithms written in the GAP language as well as large data libraries of algebraic objects. <http://www.gap-system.org/>

```

G = SymmetricGroup(4); G
Symmetric group of order 4! as a permutation group

```

```

# This uses nontrivial functionality of GAP behind the scenes...
for H in G.normal_subgroups():
    print H

```

Subgroup of (Symmetric group of order 4! as a permutation group) generated by  $[(1,3)(2,4), (1,4)(2,3)]$   
 Subgroup of (Symmetric group of order 4! as a permutation group) generated by  $[(2,4,3), (1,3)(2,4), (1,4)(2,3)]$   
 Subgroup of (Symmetric group of order 4! as a permutation group) generated by  $[(1,2), (1,2,3,4)]$

```
groups.permutation.<tab>
```

Directly using GAP (see <http://www.gap-system.org/Manuals/doc/tut/chap5.html#X8171DAF2833FF728>)

```
%gap
s8 := Group( (1,2), (1,2,3,4,5,6,7,8) );
Group([ (1,2), (1,2,3,4,5,6,7,8) ])

%gap
a8 := DerivedSubgroup( s8 );
Group([ (1,2,3), (2,3,4), (2,4)(3,5), (2,6,4), (2,4)(5,7), (2,8,6,4)(3,5) ])

%gap
Size( a8 ); IsAbelian( a8 ); IsPerfect( a8 );
20160
false
true

%gap
IsNaturalAlternatingGroup(a8);
true
```

NOTE: You can switch a whole worksheet into gap mode by typing

- Pillow: image processing capabilities to [Sage].

<http://pillow.readthedocs.org/en/latest/> says Pillow:

- provides extensive file format support, an efficient internal representation, and fairly powerful image processing capabilities.
- designed for fast access to data stored in a few basic pixel formats. [] a solid foundation for a general image processing tool.
- create thumbnails, convert between file formats, print images, etc.
- point operations, filtering with a set of built-in convolution kernels, and colour space conversions, image resizing, rotation and arbitrary affine transforms.

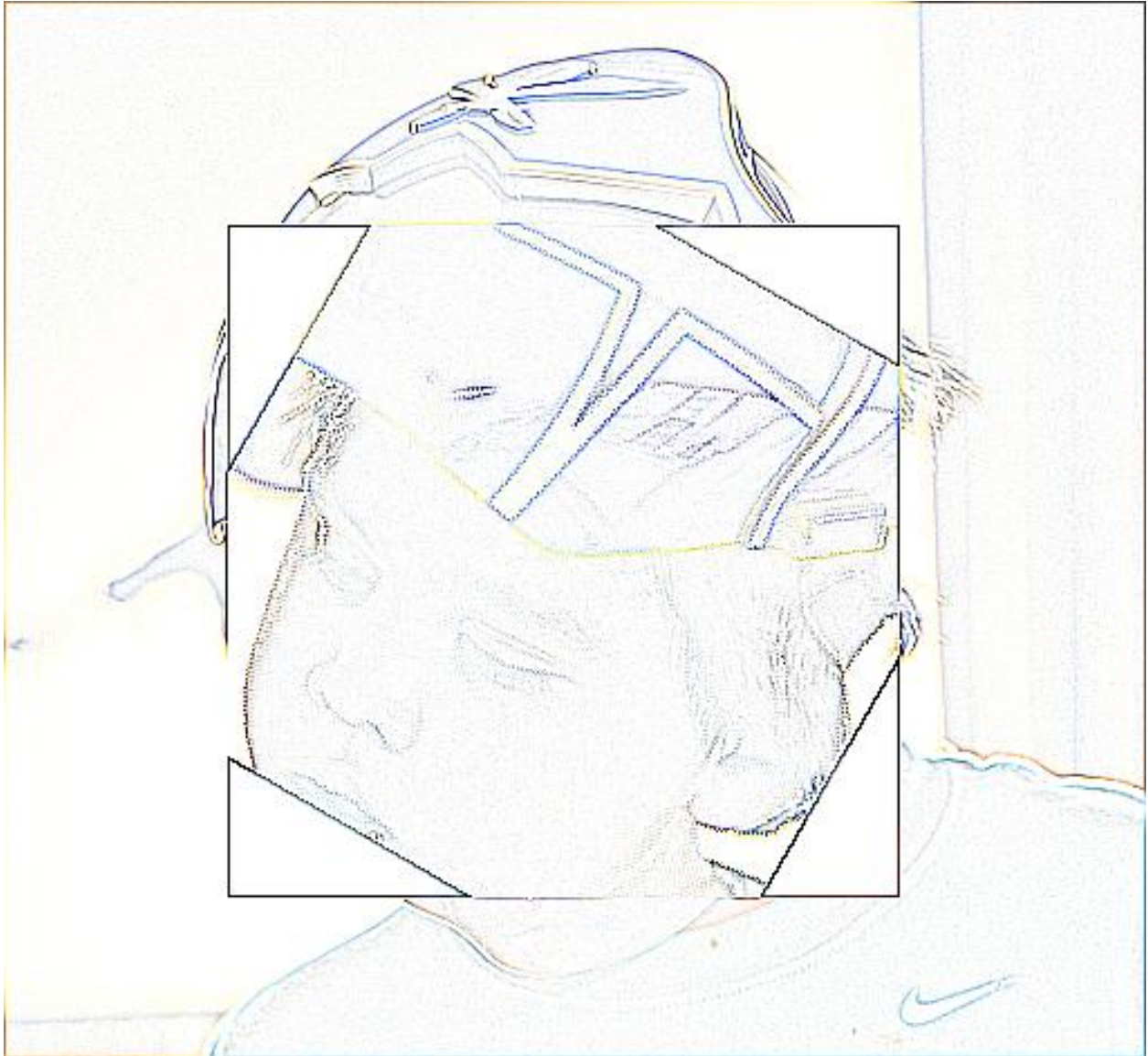
```
from PIL import Image

im = Image.open("santi.png")

print(im.format, im.size, im.mode)
```

```
('PNG', (510, 472), 'RGB')
```

```
# convert file to a JPEG (and view it)  
im.save("tmp.jpg"); salvus.file("tmp.jpg")
```



```
# create a thumbnail  
im = Image.open("santi.png")  
im.thumbnail((128,128))  
im.save('thumb.png'); salvus.file('thumb.png')
```

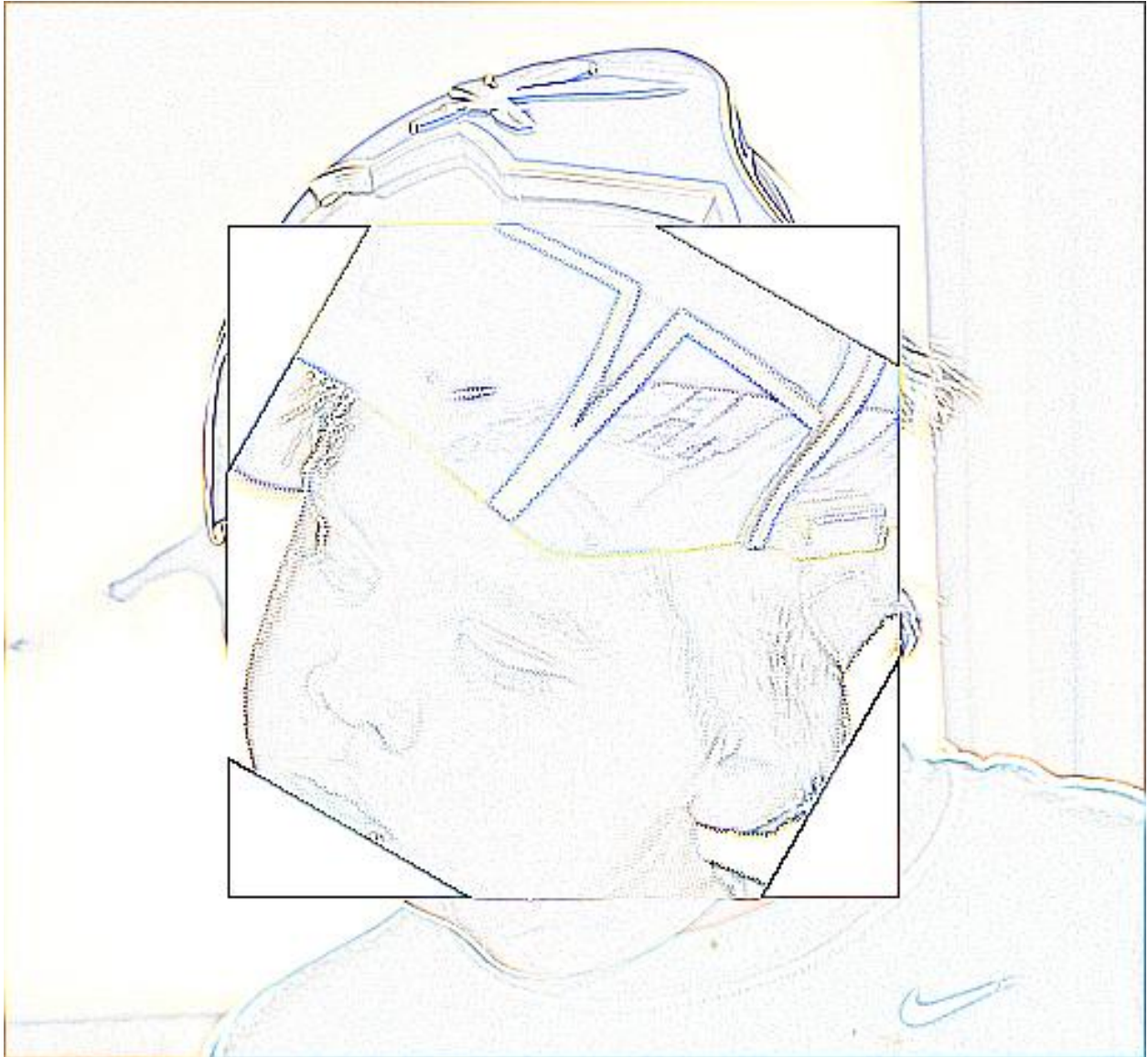


```
# copy, flip, and paste a region
im = Image.open("santi.png")
box = (100, 100, 400, 400)
region = im.crop(box)
region = region.transpose(Image.ROTATE_180)
im.paste(region, box)
im.save("tmp3.jpg"); salvus.file("tmp3.jpg")
```



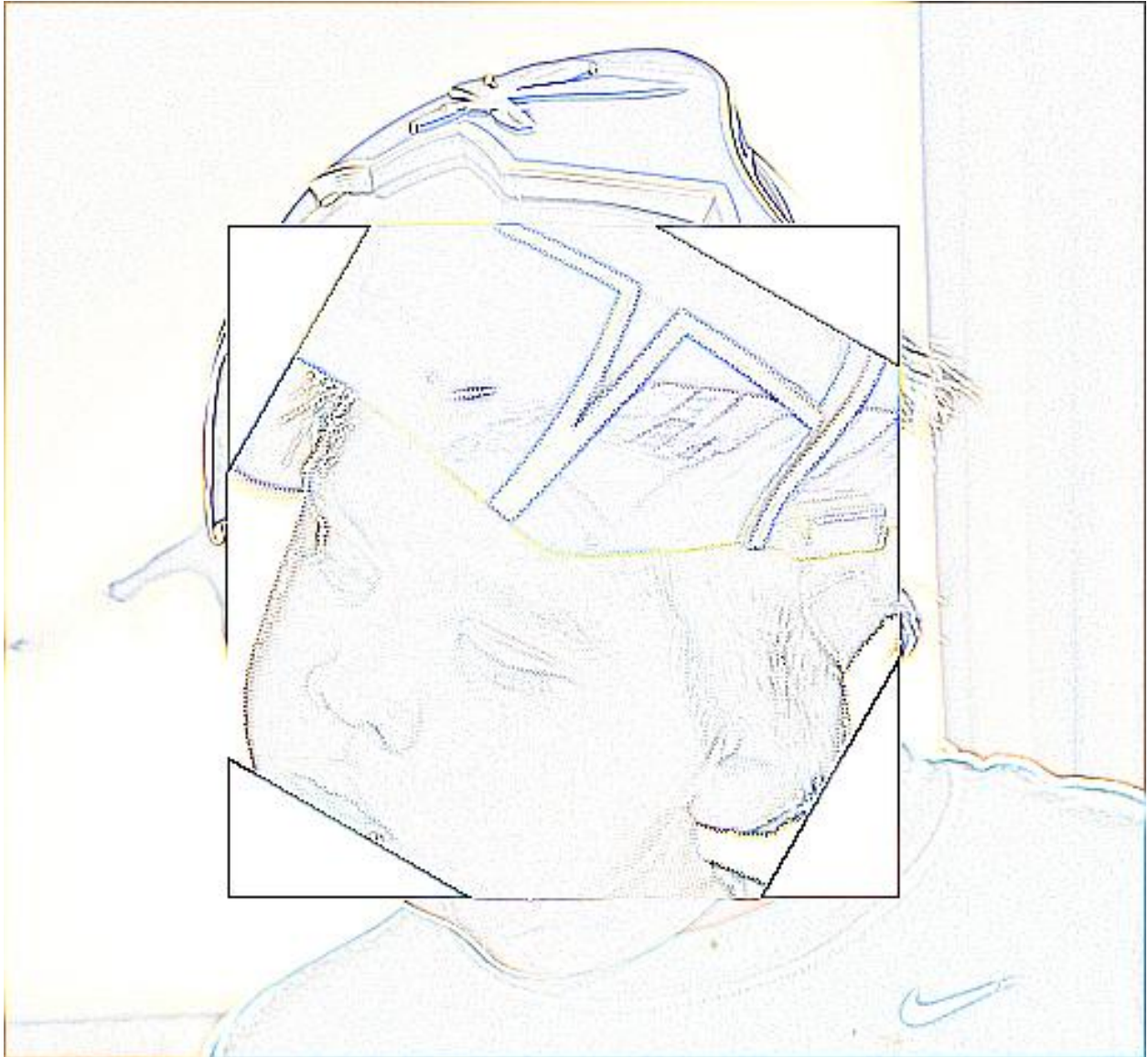
```
# copy, rotate, and paste a region
im = Image.open("santi.png")
box = (100, 100, 400, 400); region = im.crop(box)
region = region.rotate(-30)
im.paste(region, box)
im.save("tmp.jpg"); salvus.file("tmp.jpg")
```





```
from PIL import ImageFilter
out = im.filter(ImageFilter.CONTOUR)
out.save("tmp.jpg"); salvus.file("tmp.jpg")
```





```
from PIL import ImageFilter
filters = [x for x in dir(ImageFilter) if hasattr(getattr(ImageFilter,x),\
    'name')]
@interact
def _(filter = selector(filters,buttons=True)):
    out = im.filter(getattr(ImageFilter, filter))
    out.save("tmp.jpg"); salvus.file("tmp.jpg")
```