Combinatorial Potlatch 2011

Follow along: http://sagenb.org/home/pub/3698

Sage: Creating a Viable Open Source Alternative to Magma, Maple, Matlab, and Mathematica

Combinatorial Potlach 2011 (Seattle U)

William Stein (University of Washington, Seattle)

Testing, testing, ...

```
1 + 2 + 3
```

```
6
```

Prelude

- **Which do you use more?** Wikipedia or Encyclopædia Britannica?
- Which are you more likely to **contribute to**?
- The English Wikipedia alone has over 1 billion words, which is over 25 times as many as Encyclopædia Britannica.

Potlatch

Pot·latch — n. Chinook for the Native spirit of gift-giving (potlatchfund.org)

Sage Project Mission Statement

"Create A Viable Free Open Source Alternative to Magma, Maple, Mathematica, and Matlab"

- **Mathematical features:** Of Magma, Maple, Mathematica, and Matlab, with comparable speed
- **Graphics:** 2d and 3d
- **Notebook:** Interactive graphical user interface
- **Documentation:** Books, papers, curriculum, etc.

Sage is not a drop-in replacement: does not run programs written in the custom languages of the Ma's.

Sage is not like Octave (versus Matlab).

Sage's culture, architecture, programming language, and feel are different than the Ma's.

Why *not* Magma, Maple, Matlab, Mathematica?

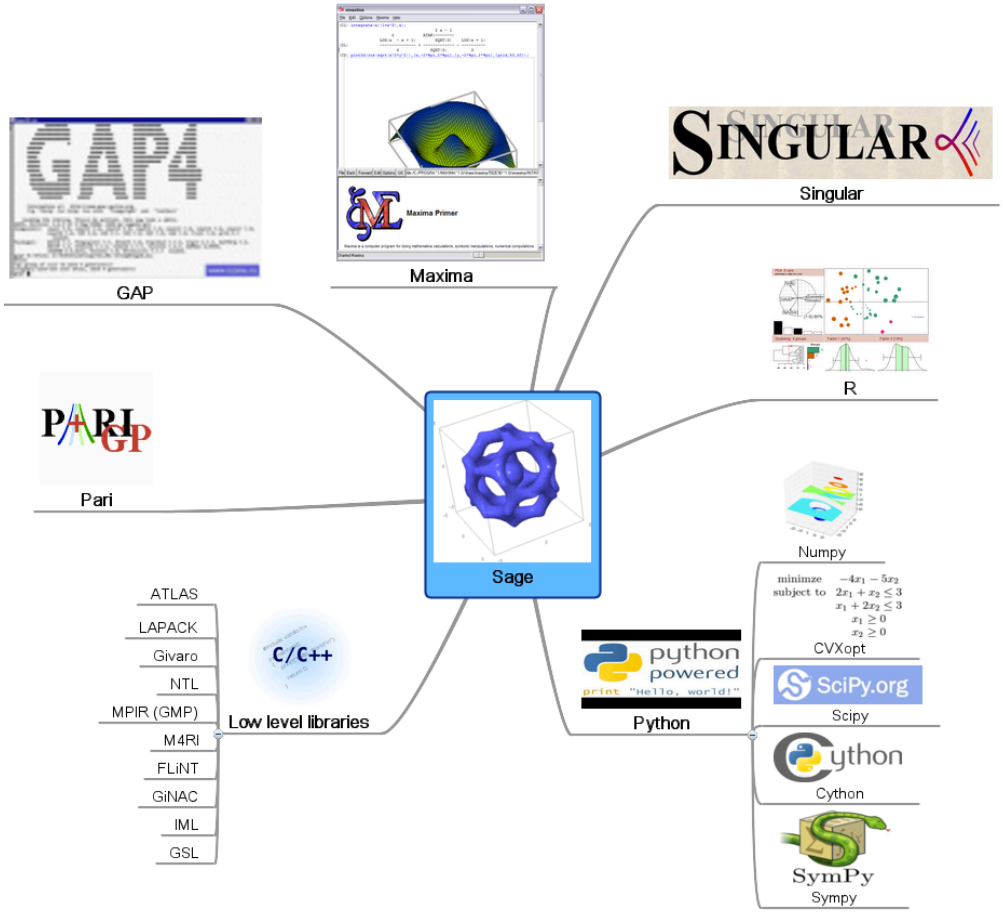
1. **Commercial:** Expensive for my collaborators and students. Not community owned.
2. **Closed:** Implementation of algorithms often secret
3. **Frustrating:** Tight control of development
4. **Copy protection:** Hard to run on supercomputer or my new laptop or after my 1-year license expires.
5. **Programming language:** All use a special math-only language
6. **Bugs:** Bug tracking is secret
7. **Compiler:** Lack of compilers for their math-only languages

Have you ever seen this? I did this morning when preparing this talk:

```
Last login: Fri Nov 18 16:16:32 on ttys008
You have mail.
deep:~ wstein$ math
Mathematica 7.0 for Mac OS X x86 (64-bit)
Copyright 1988-2009 Wolfram Research, Inc.

/Users/wstein/Library/Mathematica/Licensing/mathpass:1:
        The Mathematica license you are using has expired.
        Please contact Wolfram Research or an authorized
        Mathematica distributor to extend your license and
        obtain a new password.

You will need to get a password from your
license certificate or from Wolfram Research
(http://register.wolfram.com).
Machine name:   deep.local
MathID: 5108-53088-04270

You will need a valid license ID and password in order
to proceed. Go to http://register.wolfram.com or
consult your Getting Started documentation.

Enter your name:
```

1. **Python:** a mainstream general purpose programming language (with a compiler: Cython)
2. **Distribution:** about 100 open source packages (**written by you** and your colleagues!)
3. **Interfaces:** smoothly tie together all these libraries and packages
4. **New library:** implements novel algorithms; over a half million lines; worldwide community of several hundred people.

Distribution



Hundreds of Sage Developers

(There are at least 242 contributors in 164 different places from all around the world.)

William Stein, Tim Abbott, Michael Abshoff, Antti Ajanki, Martin Albrecht, Nick Alexander, Bill Allombert, Ethan Van Andel, Ivan Andrus, Pablo Angulo, Benjamin Antieau, André Apitzsch, Maite Aranes, Oscar Gerardo Lazo Arjona, Eviatar Bach, Jennifer Balakrishnan, Jason Bandlow, Gregory Bard, Sébastien Barthélemy, Rob Beezer, Karim Belabas, Arnaud Bergeron, Luis Berlioz, Erin Beyerstedt, Francois Bissey, Jonathan Bober, Tom Boothby, Nicolas Borie, Johan Bosman, Robert Bradshaw, Michael Brickenstein, Nils Bruin, André-Patrick Bubel, Stanislav Bulygin, Dan Bump, Iftikhar Burhanuddin, Paul Butler, Oriol Castejón, Ondrej Certik, Wilson Cheung, Dan Christensen, Craig Citro, Anders Claesson, Francis Clarke, Timothy Clemans, Alex Clemesha, Nathann Cohen, Jenny Cooley, John Cremona, Karl-Dieter Crisman, Fidel Barrera Cruz, Doug Cutrell, Alyson Deines, Vincent Delecroix, Jeroen Demeyer, Tom Denton, Maarten Derickx, Didier Deshommes, Ryan Dingman, Dan Drake, Tom Draper, Alexander Dreyer, Tim Dumol, Nathan Dunfield, Gabriel Ebner, Ben Edwards, Dana Ernst, Burcin Erocal, Ron Evans, Richard J. Fateman, Lars Fischer, Jean-Pierre Flori, Evan Fosmark, Laurent Fousse, Gary Furnish, Alex Ghitza, Andrzej Giniewicz, Alain Giorgetti, Samuele Giraudo, Amy Glen, Daniel Gordon, Chris Gorecki, Jan Groenewald, Rob Gross, Jason Grout, Ryan Grout, Mathieu Guay-Paquet, Alexey U. Gudchenko, Harold Gutch, Jonathan Gutow, Jose Guzman, Anna Haensch, Carlo Hamalainen, Marshall Hampton, Jon Hanke, David Møllerlier Hansen, Mike Hansen, Bill Hart, David Harvey, Leif Hille, Florent Hivert, Ryan Hinton, Neal Holtz, Golam Mortuza Hossain, Sean Howe, Alexander Hupfer, Wilfried Huss, Hamish Ivey-Law, Naqi Jaffery, Peter Jeremy, Peter Jipsen, Fredrik Johansson, Niles Johnson, Timo Jolivet, Benjamin Jones, David Joyner, Michael Kallweit, Josh Kantor, Kiran Kedlaya, Lloyd Kilford, Simon King, Keshav Kini, David Kirkby, Emily Kirkman, David Kohel, Ted Kosan, Ross Kyprianou, Sébastien Labbé, Yann Laigle-Chapuy, Kwankyu Lee, Julien Leroy, Richard Lindner, David Loeffler, Miguel Marco, Michael Mardaus, Robert Mařík, Jason Martin, Alexandre Blondin Massé, Peter McNamara, Gregory McWhirter, Jason Merrill, Matthias Meulien, Robert Miller, Kate Minola, Moritz Minzlaff, Joel Mohler, Thierry Monteil, Peter Mora, Bobby Moretti, Rich Morin, Guillaume Moroz, Gregg Musiker, Tobias Nagel, Brett Nakashima, Pablo De Nápoli, Johan Sebastian Rosenkilde Nielsen, Minh Van Nguyen, Andrey Novoseltsev, Christopher Olah, Johan Oudinet, Bill Page, Ronan Paixão, Willem Jan Palenstijn, John Palmieri, Dmitrii Pasechnik, Javier López Peña, Paulo César Pereira de Andrade, David Perkinson, Clement Pernet, John Perry, Pearu Peterson, David Poetzsch-Heffter, Viviane Pons, Bill Purvis, Julien Puydt, Yi Qiang, Jordi Quer, Gustavo Rama, Jens Rasch, Martin Raum, Dorian Raymer, Stefan Reiterer, R. Rishikesh, David Roe, Bjarke Hammersholt Roune, Gordon Royle, Serge A. Salamanka, Franco Saliola, Leonardo Sampaio, Kyle Schalm, Ed Scheinerman, Anne Schilling, Harald Schilly, Jack Schmidt, Michael Schneider, Christopher Schwan, Dag Sverre Seljebotn, Dan Shumow, Denis Simone, Steven Sivek, Nils-Peter Skoruppa, Jaap Spies, Jonathan Spreer, Armin Straub, Marco Streng, Kevin Stueve, Christian Stump, Blair Sutton, Chris Swierczewski, Luis Felipe Tabera Alonso, Glenn Tarbox, Philippe Theveny, Nicolas Thiery, Griffen Thoma, Emmanuel Thomé, John Thurber, Igor Tolkov, Gonzalo Tornaria, Kiminori Tsukazaki, Charlie Turner, Michel Vandenbergh, Joris Vankerschaver, Soledad Villar, John Voight, Felipe Voloch, Steve Vonn, Justin Walker, Mark Watkins, Georg S. Weber, Eric Webster, Ralf-Philipp Weinmann, Joe Wetherell, Carl Witty, Cristian Wuthrich, Soroosh Yazdani, Dal S. Yu, Gary Zablackis, Mike Zabrocki, Bin Zhang, Paul Zimmermann, Mao Ziyang

```
bernoulli??
```

History

- **2005:**SAGE-0.1 released February 1, 2005; SAGE=Software for Arithmetic Geometry Experimentation
- **2006:** (2 Sage Days workshops); Sage is not just for number theory

- **2007:** (4 Sage Days) 100% test requirements; peer review of all new code (see trac); industry funding; NSF; Trophees du Libre
- **2008:** (7 Sage Days) Release managers besides me.
- **2009:** (8 Sage Days) Better foundations; 3d graphics; more developers (e.g., sage-combinat)
- **2010:** (13 Sage Days) More devs and users; nontrivial NSF grants
- **2011:** (12 Sage Days) Much faster http://sagenb.org with >50,000 accounts; very stable releases; undergrad curriculum development
- **2012:** (5 Sage Days already planned)
- **2013:** (1 Sage Days already planned)

See this article for more details about the (pre-)history of Sage.

Question Break

????

```
factorial(3)
```

```
6
```

```
A = random_matrix(QQ, 4); b = A^(-1); b
```

```
[   0 -1/2    0 -1/2]
[-2/3   -1 -1/3 -1/2]
[   0  1/2    0    0]
[-1/3   -1  1/3 -1/2]
```

```
show(b)
```

$$\begin{pmatrix} 0 & -\frac{1}{2} & 0 & -\frac{1}{2} \\ -\frac{2}{3} & -1 & -\frac{1}{3} & -\frac{1}{2} \\ 0 & \frac{1}{2} & 0 & 0 \\ -\frac{1}{3} & -1 & \frac{1}{3} & -\frac{1}{2} \end{pmatrix}$$

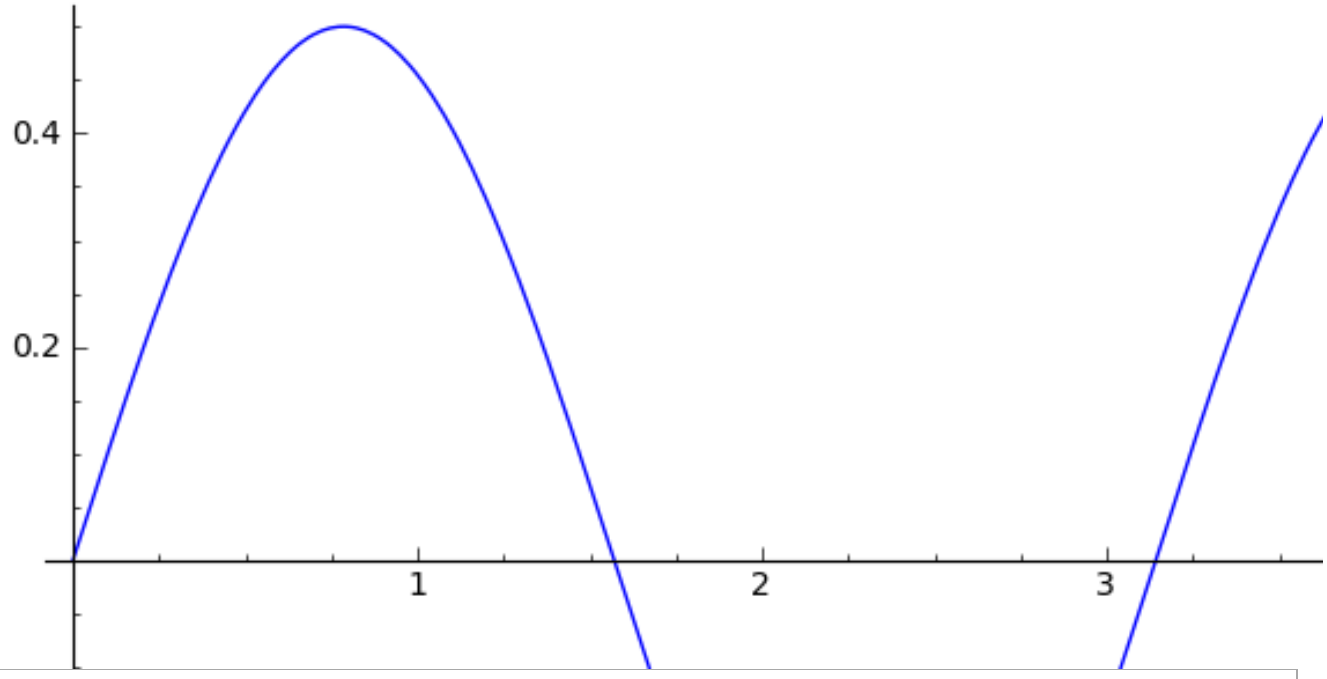
```
latex(b)
```

```
\left(\begin{array}{rrrr}
0 & -\frac{1}{2} & 0 & -\frac{1}{2} \\
-\frac{2}{3} & -1 & -\frac{1}{3} & -\frac{1}{2} \\
0 & \frac{1}{2} & 0 & 0 \\
-\frac{1}{3} & -1 & \frac{1}{3} & -\frac{1}{2}
\end{array}\right)
```

```
show(integrate(sin(x)*cos(x), x))
```

$$-\frac{1}{2}\cos\left(x\right)^{2}$$

```
G = plot(sin(x)*cos(x), (0, 4))
G
```




```
G.save('a.pdf')
```

a.pdf

```
sage.interfaces.
```

Interfaces

Use a single Mathematica session from within Sage. Here, Mathematica is running on a computer at UW, since my laptop license expired.

```
mathematica = Mathematica(server='sage.math.washington.edu',
                          server_tmpdir='/tmp/')
f = mathematica('Integrate[Sin[x],x]'); f
```

^C

```
f.Integrate(x)
```

```
    -Sin[x]
```

```
f + mathematica(sin(x)*cos(x^2))
```

```
    -Cos[x] + Cos[x^2]*Sin[x]
```

```

```

```

```

```

```

```

```

Interactive Image Compression

(using numpy)

```
import pylab, numpy

X = pylab.imread(DATA + 'wyckoff.png')
A_image = numpy.mean(X, 2)
u,s,v = numpy.linalg.svd(A_image)
S = numpy.zeros(A_image.shape)
S[:len(s),:len(s)] = numpy.diag(s)
n = A_image.shape[0]

@interact
def svd_image(i = ("Eigenvalues (quality)",
                   (20,(1..A_image.shape[0]//2)))):
```

```
    A_approx = numpy.dot(numpy.dot(u[:,:i], S[:i,:i]), v[:i,:])
    g = graphics_array([matrix_plot(A_approx),
                        matrix_plot(A_image)])
    show(g, axes=False, figsize=(6,3))
    html("%sx%s image compressed to %.1f%% of size using %s
eigenvalues."%(
            A_image.shape[0], A_image.shape[1],100*
(2.0*i*n+i)/(n*n), i))
```

Number Theory

```
factor(2009201020112012)      # uses PARI
```

```
    2^2 * 43 * 2269 * 5148259709
```

```
# Jon Bober - Rademacher's formula
time number_of_partitions(10^6)
```

```
    147168498635822339863100476060989594348403048443914212533461274
    661174189186182763301488739835975558420153741306002880959293873
    232270327849578001932784396072064228659048713020170971840761025
    986084690814282935670692978599129051989944549067221999782345287
    740222882298501367675662947818874946878790038246999881977292006
    668735996662273816798266213482417208446631027428001918132198177
    651123454259502672842445259229678119344813999466473010574256435
    949891814852853513705513994767199816914590220155991019596014174
    715430750022184895815209339012481734469448319323280150665384042
    417958775176129491624814247999880293650719525707448504757166277
    033914424951138232981952630083364898260458377122024553049963821
    028531832004519046591968302787537418118486000612016852593542741
    504626724547323732184583342751252422746539913017407694128084740
    422179992860711083363033162982891024446496968053954167918754800
    636774022023128467646919775022348562520747741843343657801534130
    197553037516970799928704028567784161934747236817177215404666430
    15630003467104673818
    Time: CPU 0.03 s, Wall: 0.03 s
```

```
@interact
def _(n=(25..10000)):
    plot(prime_pi, 0, n, gridlines='minor').show(figsize=[8,3])
```

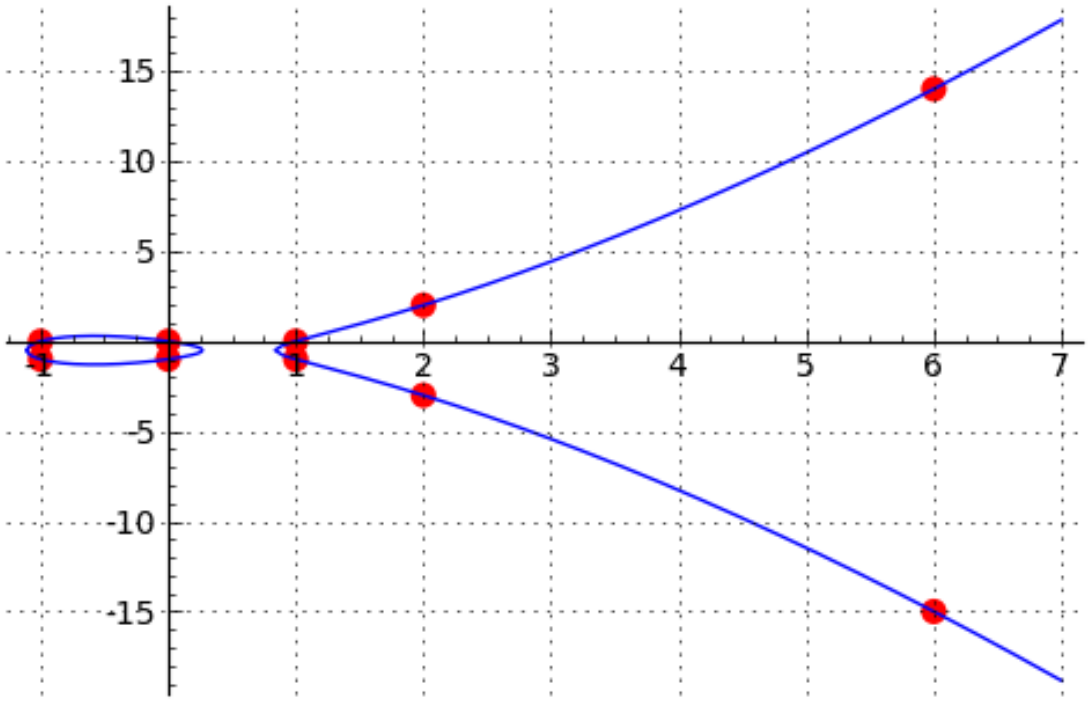
```
E = EllipticCurve('37a'); E
```

```
Elliptic Curve defined by y^2 + y = x^3 - x over Rational Field
```

```
v = E.integral_points(both_signs=True); v     # mwrank, new code
```

```
[(-1 : -1 : 1), (-1 : 0 : 1), (0 : -1 : 1), (0 : 0 : 1), (1 : -1
1), (1 : 0 : 1), (2 : -3 : 1), (2 : 2 : 1), (6 : -15 : 1), (6 :
1)]
```

```
plot(E, xmax=7, gridlines=True) + points([z[:2] for z in v],
          color='red', pointsize=50)
```

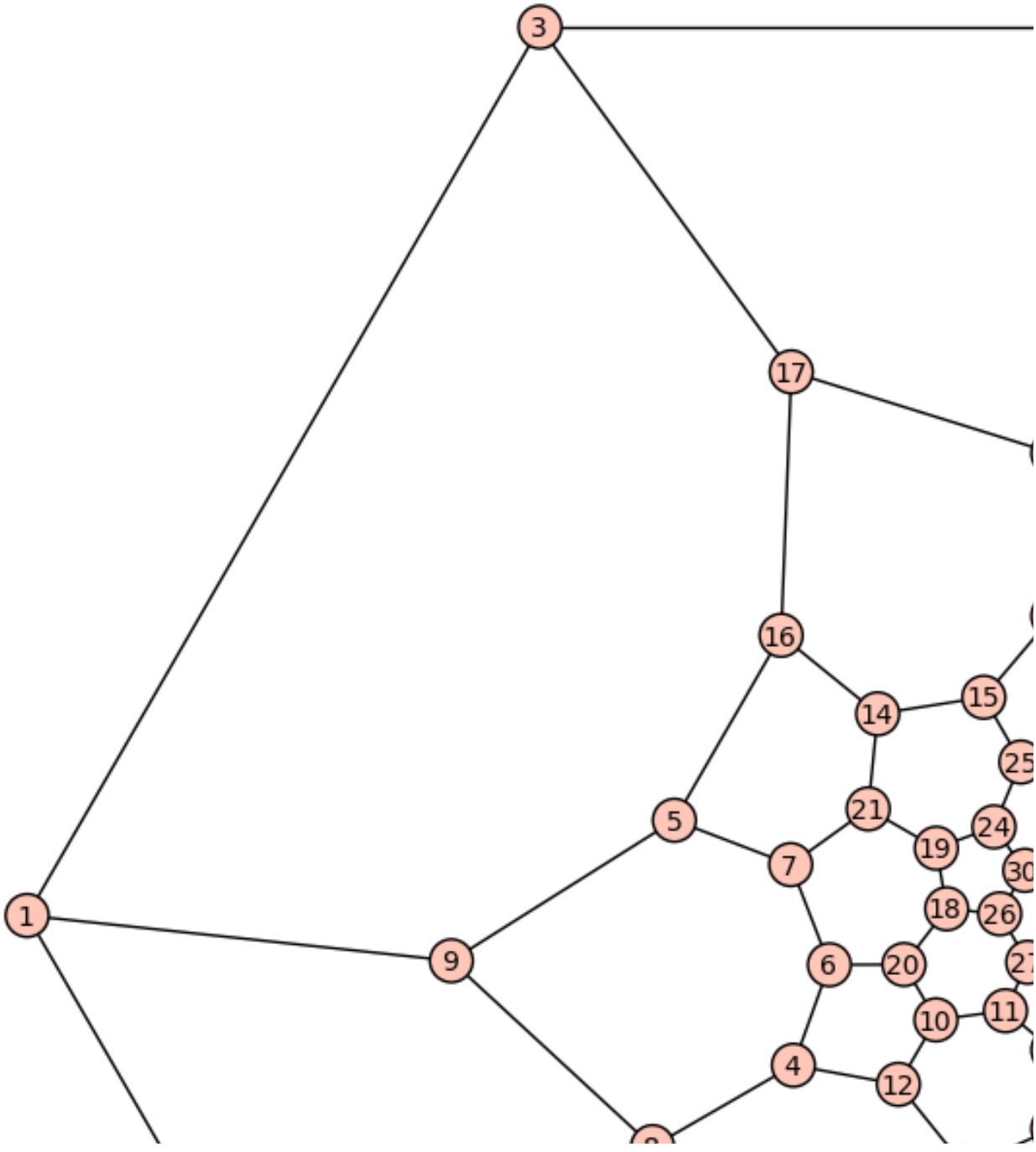


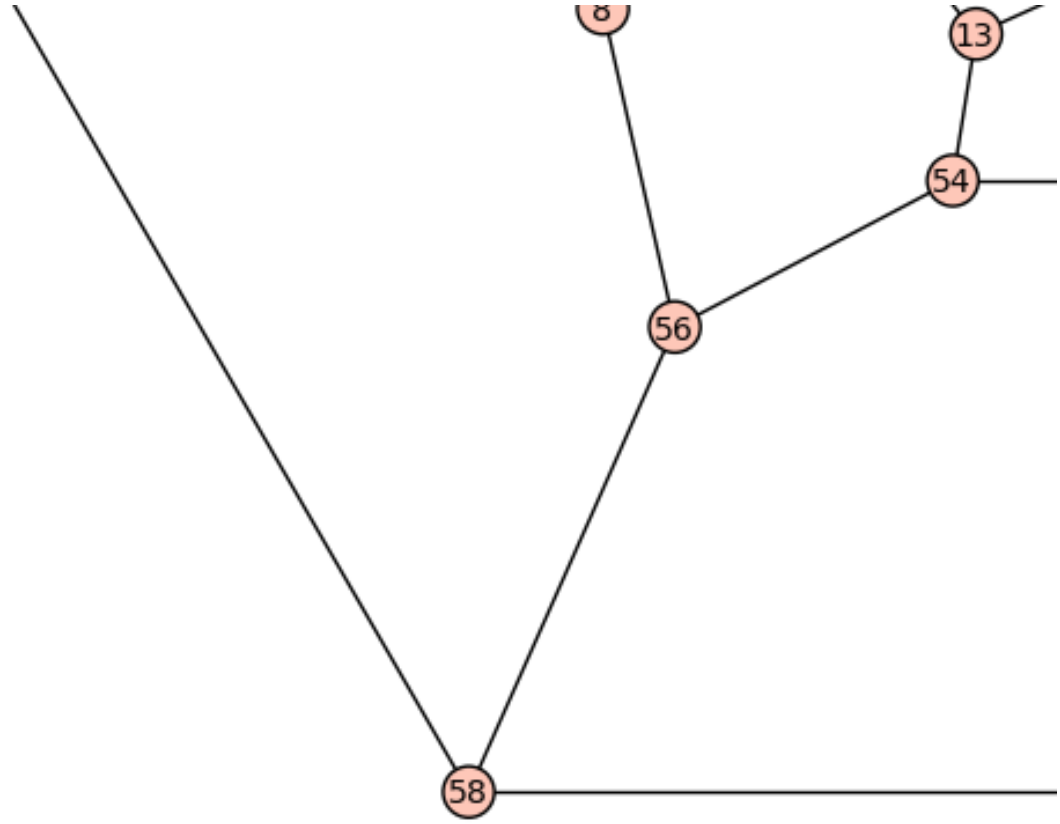
Graph Theory

"Sage's graph theory crushes anything I have met from the point of view of methods implemented. I would say: 'if you found a proprietary graph library and you are convinced that it is better for your needs than Sage, your license is on me.' But those licenses are really expensive :-D"

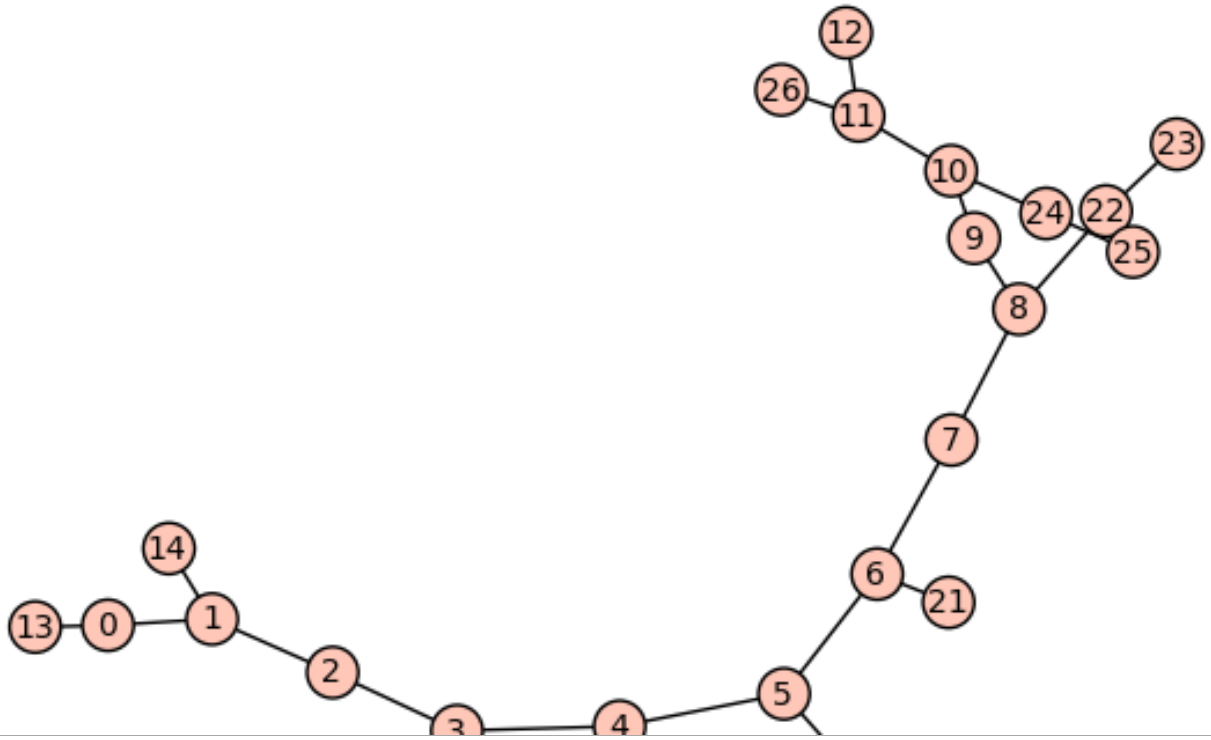
```
graphs.ClawGraph(
```

```
G = graphs.BuckyBall()
G.plot().show(figsize=8)
```






```
set_random_seed(1)
G = graphs.RandomLobster(8, .6, .3)
show(G, figsize=7)
```




```
G.automorphism_group()
```

```
    Permutation Group with generators [(12,26)]
```

```
G.chromatic_number()
```

```
    2
```

```
G.shortest_path(13,20)
```

```
    [13, 0, 1, 2, 3, 4, 5, 19, 20]
```

Algebraic Combinatorics

The sage-combinat project: "improve the open source mathematical system Sage as an extensible toolbox for computer exploration in (algebraic) combinatorics, and foster code sharing between researchers in this area."

For example, see Combinatorics Chapter of the Reference Manual

```
s = SFASchur(QQ)
h = SFAHomogeneous(QQ)
p = SFAPower(QQ)
e = SFAElementary(QQ)
m = SFAMonomial(QQ)
```

```
s
```

```
Symmetric Function Algebra over Rational Field, Schur symmetric
functions as basis
```

```
a = s([3,1]); a
```

```
s[3, 1]
```

```
a*a
```

```
s[3, 3, 1, 1] + s[3, 3, 2] + s[4, 2, 1, 1] + s[4, 2, 2] + 2*s[4,
1] + s[4, 4] + s[5, 1, 1, 1] + 2*s[5, 2, 1] + s[5, 3] + s[6, 1,
s[6, 2]
```

```
p(a)
```

```
1/8*p[1, 1, 1, 1] + 1/4*p[2, 1, 1] - 1/8*p[2, 2] - 1/4*p[4]
```

```
a.inner_plethysm(a)
```

```
s[1, 1, 1, 1] + 2*s[2, 1, 1] + s[2, 2] + 2*s[3, 1]
```

```

```

```

```

Cython

- Smooth transition between Python and compiled C code.
- Make code that involves lots of manipulation of C-level data structures optimally fast
- Heavily used in scientific computing using Python.

```
def python_sum(n):
    s = int(0)
    for i in xrange(1, n+1):
        s += i*i
    return s
```

```
python_sum(3)
```

```
14
```

```
time python_sum(2*10^6)
```

```
2666668666667000000
Time: CPU 0.16 s, Wall: 0.16 s
```

```
timeit('python_sum(2*10^6)')
```

```
5 loops, best of 3: 164 ms per loop
```

```
def python_sum2(n):
    return sum(i*i for i in xrange(1,n+1))
```

```
time python_sum2(2*10^6)
```

```
2666668666667000000
Time: CPU 0.19 s, Wall: 0.19 s
```

```
%cython
def cython_sum(long n):
    cdef long long i, s = 0
    for i in range(1, n+1):
        s += i*i
    return s
```

__Users_ws...9_code_sage81_spyx.c __Users_ws...ode_sage81_spyx

```
cython_sum(3)
```

```
14L
```

```
time cython_sum(2*10^6)
```

```
    266666866666700000L
```

```
timeit('cython_sum(2*10^6)')
```

```
    625 loops, best of 3: 671 ns per loop
```

```
165/.663
```

```
    248.868778280543
```

```
%cython
def cython_sum2(long n):
    cdef long long i
    return sum(i*i for i in range(1,n+1))
```

```
    __Users_ws...9_code_sage89_spyx.c    __Users_ws...ode_sage89_spyx
```

```
time cython_sum2(2*10^6)
```

```
    266666866666700000L
    Time: CPU 0.16 s, Wall: 0.16 s
```

```

```

Of course, it is better to choose a different algorithm:

```
var('k, n')
factor(sum(k^2, k, 1, n))
```

```
    1/6*(n + 1)*(2*n + 1)*n
```

```
def sum2(n):
    return n*(2*n+1)*(n+1)/6
```

```
sum2(2*10^6)
```

```
    2666668666667000000
```

Even then, Cython provides a speedup:

```
%cython
def c_sum2(long long n):
    return n*(2*n+1)*(n+1)/6
```

```
    __Users_ws...9_code_sage97_spyx.c    __Users_ws...ode_sage97_spyx
```

```
c_sum2(3)
```

```
    14L
```

```
c_sum2(2*10^6)
```

```
    -407786678951258603L
```

```
n = 2*10^6
```

```
timeit('sum2(n)')
```

```
625 loops, best of 3: 2.01 µs per loop
```

```
timeit('c_sum2(n)')
```

```
625 loops, best of 3: 218 ns per loop
```

```
2.01/.218
```

```
9.22018348623853
```

But at a cost!

```
c_sum2(2*10^6)        # WARNING: overflow -- it's just like C...
```

```
-4077886789512586O3L
```

```
n=2*10^6; n*(2*n+1)*(n+1) > 2^63
```

```
True
```

Solving Equations

Solve a ***cubic equation***:

```
x = var('x'); show(solve(x^3 + x - 1==0, x)[0])
```

$$x = -\frac{1}{2}\left(i\sqrt{3}+1\right)\left(\frac{1}{18}\sqrt{3}\sqrt{31}+\frac{1}{2}\right)^{\left(\frac{1}{3}\right)} + \frac{-i\sqrt{3}+1}{6\left(\frac{1}{18}\sqrt{3}\sqrt{31}+\frac{1}{2}\right)^{\left(\frac{1}{3}\right)}}$$

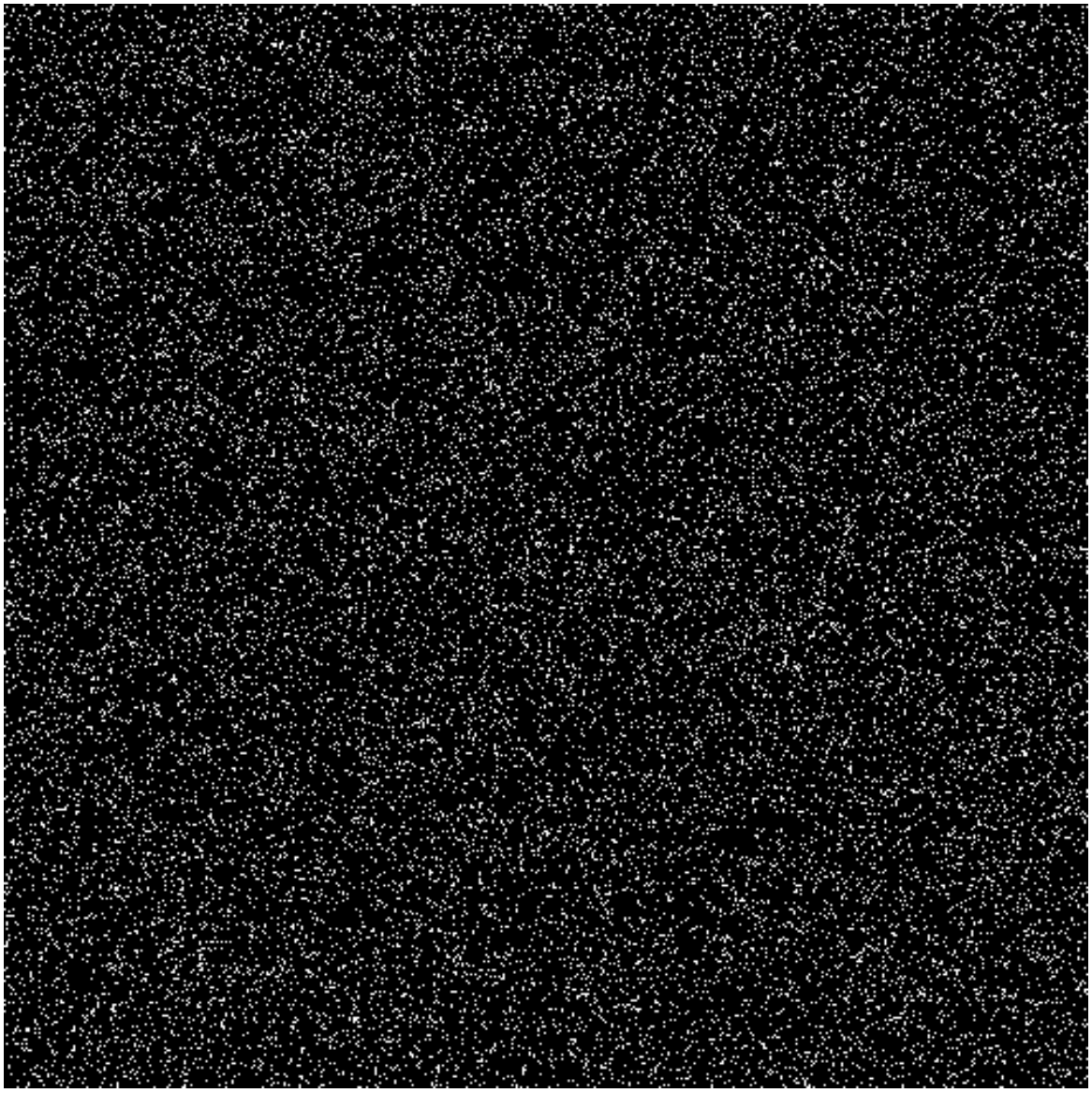
Solve a system of ***two linear equations*** with one unknown coefficient α:

```
var('alpha, y')
show(solve([3*x + 7*y == 2, alpha*x + 3*y == 8], x,y)[0])
```

$$\left[x = \frac{50}{7\,\alpha - 9}, y = \frac{2\,(\alpha - 12)}{7\,\alpha - 9}\right]$$

Solve a ***system of 500 linear equations*** exactly over the rational numbers:

```
n = 500
A = random_matrix(QQ,n,n,num_bound=10, den_bound=10)
v = random_matrix(QQ,n,1,num_bound=10, den_bound=10)
A.visualize_structure()
```



```
# IML is used -- http://www.cs.uwaterloo.ca/~astorjoh/iml.html
time w = A \ v
```

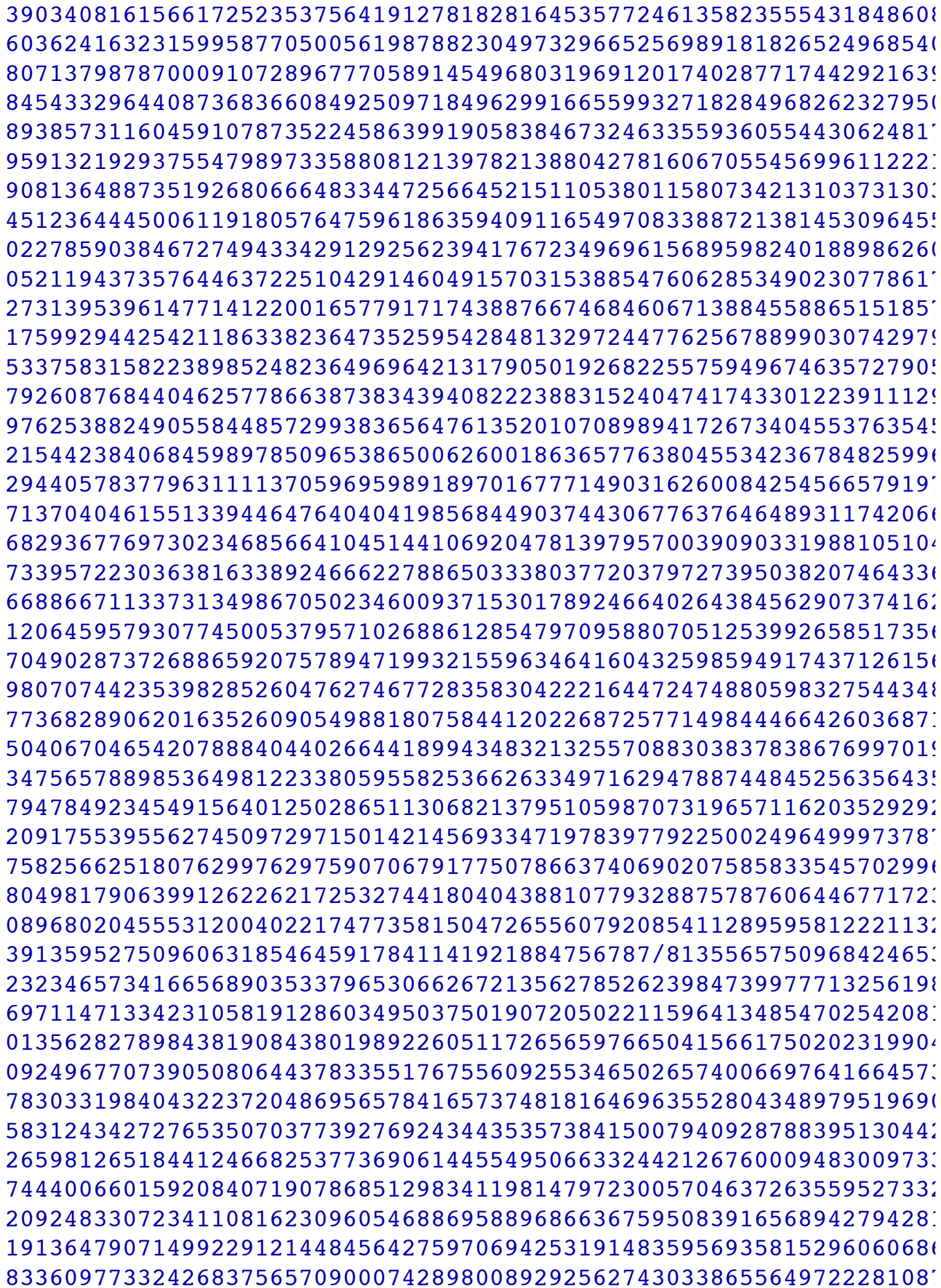
Time: CPU 3.08 s, Wall: 2.04 s

```
print w.str()
```

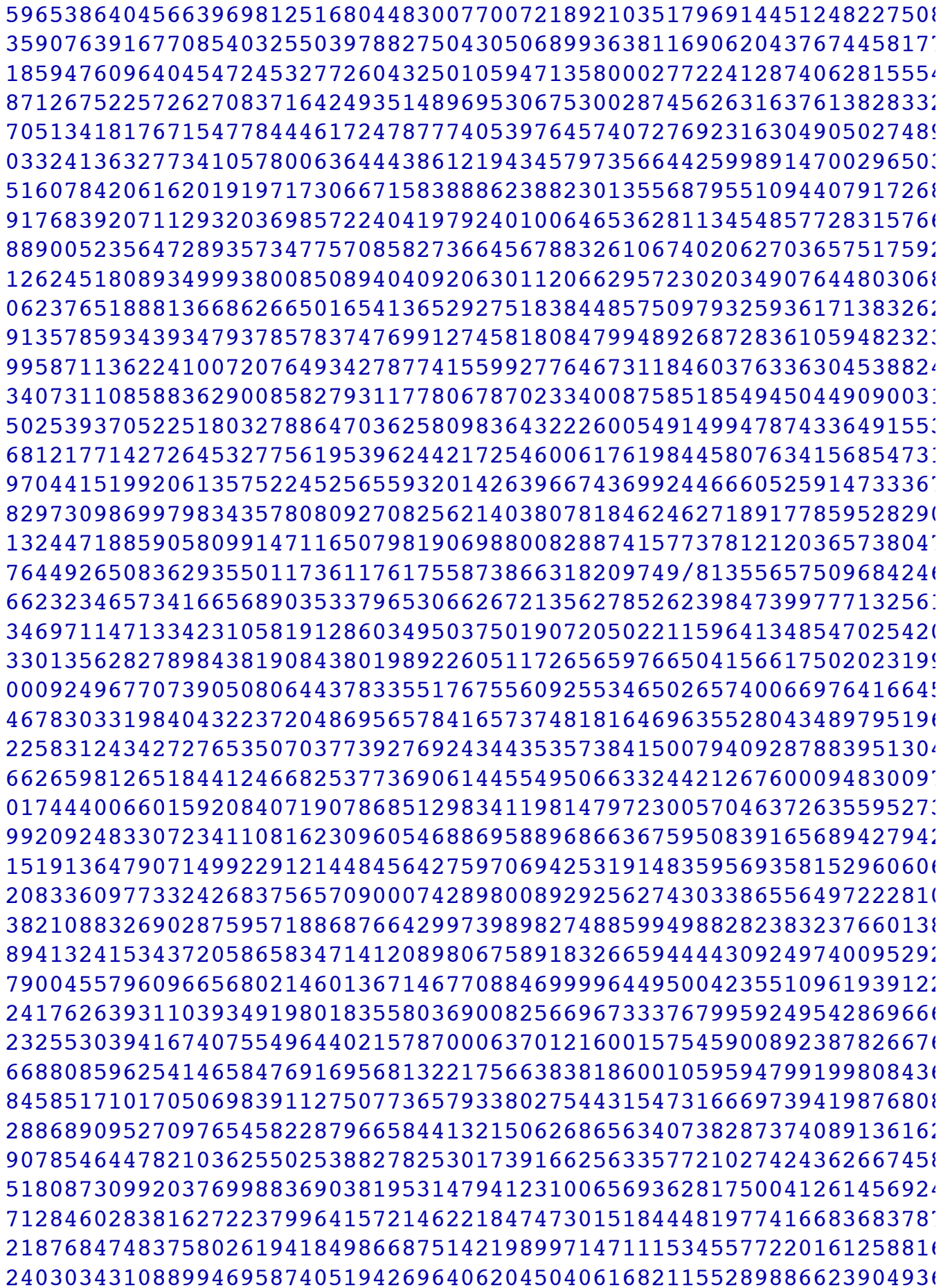
WARNING: Output truncated!
full_output.txt

[
-351702876068802906437552420686055850326076457904676962275100830
401420955261319182505472655403446871133375351213855350226815464
541678542779937822420479756487655377896847799471633977576723782
752013079903081977572402942342855631847634917334375567725200949
981494332871024651254158873412391649819482714094296169547438994
166187839273830840430242204133808650932888101185375789831810538
749302374084369069440792237427868339146115514095572829020266899
134355557855057198503380254232298641549695798842795869295699735
920100244120646553550505898020495083583179450653843216750029252
795806490140294417796752695156539524124805975616026523319266984
226942792117185857783063808851623050866691080434331157140193997
656405532372637424988128302930037221573841204291852109644817711
003582668463237232745438803646481729385618653719290569550431763
479911712917380077097901887716153891517164989985540840321675523
530360179417770975752951610003948983037774750319911117156732832
778539539244149073095868925297296767522098873014512743624494378
988592956277583816139788377093357552496933776948081224203156447
737484105107310982124866738518101800129697107516438736576977941
437927029350938265391137496664227427915828658742276488248475686
797308951586142050635510616384914735047939100560830346367036281
909909358743617446014742435439813694562291652176283751189754664
361180619892765924138532602101463250422678650464862058688845651
378935711191095339677561251599756523993225002326911920613591375
810713117303327205420957785613280646524680178422545699670542002
185206878446429324207574638947936431387935201327903186688186655
889749881958253860610185138652298499093125007922510871073642155
160192682022566179204513802804002649348052390662029135279718239
571118081333104312614968911252810197919260114909352478204286878
399735896224340600057359854531576146926251200089897481071278163
047295612770389853217738467738241538454558869964299842463856929
637878107844843582575044052722770755619169229082572645352435097
865291508953639079652724169694694026175524388621142937766905589
623925909303675286193416452595298612780914411142216266018732516
619208172280955826296174872463796580072215353832080758361490836
805804530940354095353962448272508728028384915074672326844898676

```
0800665429165430814274592676665502787407335232444362008294879391
3811194688142700374028190883369685908216732077/4067782875484212
3311617328670832844517668982653313360678139263119923699888566280
6734855735667115529095643017475187509536025110579820674273512710
1650678141394921909542190099461302558632829883252078308751011599
0004624838536952540322189167758837780462767325132870033488208322
7339151659920216118602434782892082868740908234817764021744897598
6129156217136382675351886963846217217678692075039704643941975652
3313299063259220623341268868453072277475331662210633800047415048
0087220033007960420359539342564917059907398615028523186317797630
9960462416536170554081154802734434794484331837975419582844713971
0759568239535749614560724228213798534712659574179784679076480303
1041680488666213418782854500371449004464628137151693278248611405
1910544163451437978594343832149869949137442997494141191618830069
4470662076718602932917357060449033794591633297222154624870047640
3950227898048328401073006835733854423499982247502117755480969561
1208813196555196745990091779018450412834836668839979624771434833
1162765197083703777482201078935003185060800078772950446193913338
8344042981270732923845847840661087831919093000529797399599904218
9229258550852534919556375386828966901377215773658334869709938404
1443445476354882729114398329220660753134328170369143687044568081
4539273223910518127512694139126508695831281678860513712181333729
2590436549601884994184519097657397061550328468140875020630728462
3564230141908136118998207860731109237365075922240988708341841893
1093842374187901309709249334375710994985735557672788610080629408
1201517155444973479370259713482031022520308410577644943311952468
1238661819696470413701424802298593548776978832108656234910264109
2205911593318386075749335920440095933187160459299339643209960996
9544012564454057364306563040653021522038450273701898168250233560
1542331594468641179158138428277349731703737219963018417632363299
1494386783276795049901144295223363777310171682399431489512674283
8722166638871500680396177707298901665698448682925479970938945488
3015207286373345737063021706179180185939364399949813105084680631
5909826757786706357829130980704081337842853100001277249846738997
6878582385201810484324518892638117928226308648220890311998167122
9766497072354298239001901713874648443916904411649363916870517743
6627139496012306231401783236358878349778653017133170776393229190
3962899031016050228081413485309678808196586762986239037632308148
0814472095156234891023401
[
7126109666514325766865816281323217854027936420028241233110182944
6957296194203263097154402938277462517133906629031143528461511612
0441097120169029621264201559926948837404463199725065392914261993
6363726571652597606910738417927277709146849430771949030118855670
```



```
210883269028759571886876642997398982748859949882823832376601385
413241534372058658347141208980675891832665944443092497400952921
004557960966568021460136714677088469999644950042355109619391224
176263931103934919801835580369008256696733376799592495428696663
255303941674075549644021578700063701216001575459008923878266761
880859625414658476916956813221756638381860010595947991998084363
585171017050698391127507736579338027544315473166697394198768086
868909527097654582287966584413215062686563407382873740891361625
785464478210362550253882782530173916625633577210274243626674582
808730992037699883690381953147941231006569362817500412614569240
284602838162722379964157214622184747301518444819774166836837870
876847483758026194184986687514219899714711153455772201612588162
030343108899469587405194269640620450406168211552898866239049361
773236393929408274028496045971870975539576642173124698205282191
118231866367721514986718408801918663743209185986792864199219921
880251289081147286131260813060430440769005474037963365004671213
846631889372823583162768565546994634074744399260368352647265987
887735665535900998022885904467275546203433647988629790253485675
443332777430013607923554145978033313968973658509599418778909770
304145727466914741260434123583603718787287998996262101693612751
196535155734127156582619614081626756857062000025544996934779955
571647704036209686490377852762358564526172964417806239963342458
329941447085964780038034277492968878338088232987278337410354876
542789920246124628035664727177566995573060342663415527864583801
257980620321004561628269706193576163931735259724780752646162972
289441903124697820468 0]
[
-160368478412945774060217542266565614741480339664473882003496792
752897849430720531415548290532939309218576071020154022957094625
428590597947081758626095111584217558171924149959990476644701661
038712093916153308039910777872838474144477153790765392231975989
989203718917773532983442999895333411071060834557959189343034901
381788149001754798357219163777955690624243090780940038394378744
339766207403967477832079106079326901230674181667029083585928270
426638506252451467844695758873804899221014531349829577420754990
789195750769348073242033773798477949923643941006395796361486072
387038378286670215845854667646188249874827894474553349754712203
003197697832503905138418102378193703590224644315609939459211656
851110031181085277800724768081489399381001096579409475861721879
236334471139887152312263238253619950898878488951112166498298946
482754937825057055164371245566732363301246448434407173594758959
905162150877680661132588489739431934491509649514421694064324595
807563959797726791175328941361817993760343741477898086610146642
022526463051329946957720535443508803014679725351903425522645486
```



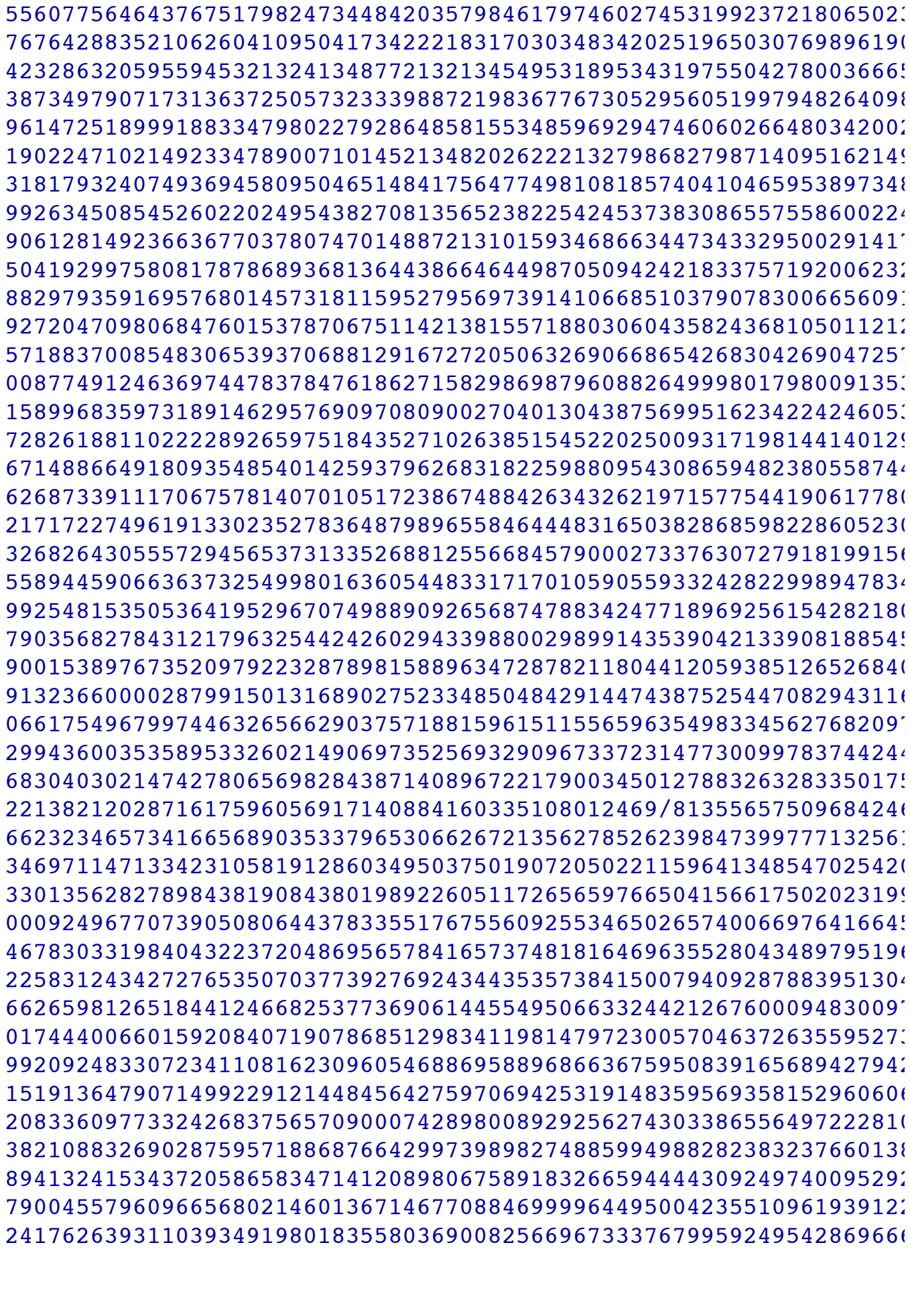
```
24773236393929408274028496045971870975539576642173124698205282195
44118231866367721514986718408801918663743209185986792864199219
90880251289081147286131260813060430440769005474037963365004671
30846631889372823583162768565546994634074744399260368352647265
29887735665535900998022885904467275546203433647988629790253485
74443332777430013607923554145978033313968973658509594187789097
60304145727466914741260434123583603718787287998996262101693612
18196535155734127156582619614081626756857062000025544996934779
37571647704036209686490377852762358564526172964417806239963342
95329941447085964780038034277492968878338088232987278337410354
32542789920246124628035664727177566995573060342663415527864583
79257980620321004561628269706193576163931735259724780752646162
16289441903124697820468]
[
67078090575460996786265039822017713117798283113493478926653538
46757733075646620738656926481812339943316973838468394971753389
98323513187899868079464157734043255602408681967520161201925055
46490357047190466819818026490143454145094083716988539956877866
75848187448041515208657811571942064579633839589404530385701787
35502120297975155121858149317360619754079230429560868422017095
45634084786897995745483722385346543743765433110166546345397142
59552701728276074098883057445870278068844711516024669217022715
79556828705860943892266695837122690078564177545904879325344663
41413565889606142593164456278072831567090592707061437774490499
90224015420623682037735613078465672194993959964381294642064204
47382646396571610758270980794351970912021233545865456886524596
46683547087871879256758033390855412169531132531870259169605495
62355777216894300695420152571284827680413861524846430866849097
07395420172135538285912319239017008565968032526765724608123929
...
92831442924436949460303688963954833367367574022443753694967516
36997337502843979942942230691154440322517632584048060686217798
81038853928124090081233642310579773247809872390449546472787858
05699209194374195107915328434624939641056438224088236463732735
97343681760383732748641837197358572839843984303581760502578162
26252162612086088153801094807592673000934242679261693263778745
32553713109398926814948879852073670529453197418859775471331071
04577180893455109240686729597725958050697135164748866655548600
84710829195606662793794731701919883755781954026520608291454933
52086824716720743757457599799252420338722550281836393070311468
16523922816325351371412400005108999386955991141075143295408072
98075570552471712905234592883561247992668491709990659882894171
```

...

```
92831442924436949460303688963954833367367574022443753694967516
36997337502843979942942230691154440322517632584048060686217798
81038853928124090081233642310579773247809872390449546472787858
05699209194374195107915328434624939641056438224088236463732735
97343681760383732748641837197358572839843984303581760502578162
26252162612086088153801094807592673000934242679261693263778745
32553713109398926814948879852073670529453197418859775471331071
04577180893455109240686729597725958050697135164748866655548600
84710829195606662793794731701919883755781954026520608291454933
52086824716720743757457599799252420338722550281836393070311468
16523922816325351371412400005108999386955991141075143295408072
98075570552471712905234592883561247992668491709990659882894171
```

```
076068554985937756676176465974556674820709753490650855798404922
071329454355133991146120685326831055729167603109585159612406420
256539412387152327863470519449561505292325945552325788838062493
936]
[
945260438484163563044816553270592918001394751912601774911825365
210186951649246661639559021114717376260331481162829454649734526
636664155830950900670329940696913349120038708974764850263142036
918632939627113750449192588841359026968866079516989072845040071
399129515132555537815586198188713257058948928710679504407301812
940414668298404351855540952490861059182703733512440699503048216
848574794292595412083193782255633569430951497593694134968328311
211162667393710608200155087818900639054239313234029693460423489
441405538529537696013160246766809186278924539876304196200204970
598327996829810101649304391646190740797656277617083226079919395
421088844404238582953699439321045065119972867039415349335513738
933631896577491253730096499620593044973519755639461232043143165
152264789694619954476090375233361538701533649288116764115532254
854890114949465865750404859164490723955956375892869511866664079
147779808254947501465030222002823761654856395430512577247784211
490532187380404634254768989064965878170952631359925451301370623
105437700060458729323040583128863823396182531650819946998154635
110407963903605891932720040600908011713193872812237006246495537
338223923345889066439761405289990288428639070654876921929568265
290576863230320277827306546290762432457034387209476515845520952
653243365303253758669744504659483762701775332917273973568382856
531657541338612055138174753528355821001604691566163573615831527
453159590368748658409948587747041001806751755499789686883415778
275481645239053135957456545695246919956709635964434442580270975
552645917906701767500506143166159568497263134244759938183252653
447415460077612095353409486005855386850439643956698189806363674
987323197108015864016590417090517492078533207706981439333576739
557367359279288126574362896160724072947728043556896394907309356
920892474962624204819772628457129922633055024709546943683165402
193714259645310447344332904810199100595329076190131943841631440
729069342567803937185325801377563775644505083555319782084434941
697286480783955427711755968149276252408940243994775192084812675
812517599625605228591101892002706366617404918893980717977359587
053573297443175261338656116565852050642004020743136552986786830
457858732004847917685636992226257847548078323792583349793725124
836945089819773438848969641207868175331302983599152734842122754
089851844417605157177929842231947390251620333/406778287548421232
311617328670832844517668982653313360678139263119923699888566280
734855735667115529095643017475187509536025110579820674273512710
```

```
650678141394921909542190099461302558632829883252078308751011599
004624838536952540322189167758837780462767325132870033488208322
339151659920216118602434782892082868740908234817764021744897598
129156217136382675351886963846217217678692075039704643941975652
313299063259220623341268868453072277475331662210633800047415048
087220033007960420359539342564917059907398615028523186317797636
960462416536170554081154802734434794484331837975419582844713971
759568239535749614560724228213798534712659574179784679076480303
041680488666213418782854500371449004464628137151693278248611405
910544163451437978594343832149869949137442997494141191618830069
470662076718602932917357060449033794591633297222154624870047646
950227898048328401073006835733854423499982247502117755480969561
208813196555196745990091779018450412834836668839979624771434833
162765197083703777482201078935003185060800078772950446193913338
344042981270732923845847840661087831919093000529797399599904218
229258550852534919556375386828966901377215773658334869709938404
443445476354882729114398329220660753134328170369143687044568081
539273223910518127512694139126508695831281678860513712181333729
590436549601884994184519097657397061550328468140875020630728462
564230141908136118998207860731109237365075922240988708341841893
093842374187901309709249334375710994985735557672788610080629408
201517155444973479370259713482031022520308410577644943311952468
238661819696470413701424802298593548776978832108656234910264109
205911593318386075749335920440095933187160459299339643209960960
544012564454057364306563040653021522038450273701898168250233560
542331594468641179158138428277349731703737219963018417632363299
494386783276795049901144295223637773101716823994314895126742837
722166638871500680396177707298901665698448682925479970938945488
015207286373345737063021706179180185939364399949813105084680637
909826757786706357829130980704081337842853100001277249846738997
878582385201810484324518892638117928226308648220890311998167122
766497072354298239001901713874648443916904411649363916870517743
627139496012306231401783236358878349778653017133170776393229190
962899031016050228081413485309678808196586762986239037632308148
8144720951562348910234 0]
[
-358669728503033965955046155185932503931883887379872277455118210
839102543863181438534347689798327235493745396550354862873325284
123204745893692383498832669526276786683673904466835238639111088
176473066762370230956445440152466475196273329776775660504684381
821573429292001039205618744405254772514486944264759109061833637
162688648567335403604919306230696926882577929210556910121365156
592500723474741715873211918157674895500365603131503393909304553
543668772530242757398365829942964382287353102256878852782630392
```



```
2325530394167407554964402157870006370121600157545900892387826676
6688085962541465847691695681322175663838186001059594799199808436
8458517101705069839112750773657933802754431547316669739419876808
2886890952709765458228796658441321506268656340738287374089136162
9078546447821036255025388278253017391662563357721027424362667458
5180873099203769988369038195314794123100656936281750041261456924
7128460283816272237996415721462218474730151844481977416683683787
2187684748375802619418498668751421989971471115345577220161258816
2403034310889946958740519426964062045040616821155289886623904936
2477323639392940827402849604597187097553957664217312469820528219
4411823186636772151498671840880191866374320918598679286419921992
9088025128908114728613126081306043044076900547403796336500467121
3084663188937282358316276856554699463407474439926036835264726598
2988773566553590099802288590446727554620343364798862979025348567
7444333277743001360792355414597803331396897365850959941877890971
6030414572746691474126043412358360371878728799899626210169361275
1819653515573412715658261961408162675685706200002554499693477995
3757164770403620968649037785276235856452617296441780623996334245
9532994144708596478003803427749296887833808823298727833741035487
3254278992024612462803566472717756699557306034266341552786458380
7925798062032100456162826970619357616393173525972478075264616297
1628944190312469782046801
[
9484560833309713317340188817817645805883209097863060441392248421
6288566062991682255926553835169465103206007869525612855843836725
2370444557928845542502800318016033743491602294306312049277866092
6080582601373682982577551842626964818788336342113685296608521542
2476122015757418481391942924820517687214300185279060122694264880
8265747513494798641566802199827783229808348585230164912908128805
6427998014100279887329923231326618166127032801953363599292381068
1502353967309321293752304668044678714812907049518162597961100401
0836486219583991994588336766319125808902170852173227591600459058
5799771288051953939706646330801624759011694185212596731657586701
4162596464798216733415944141253675294713122451907274139750788252
2367837976434085495114687857050066610252083336737515050272209950
1187593198512945970606025448484237048902437335556765388797629050
0646924691249260232907817155205358753874709316834417595347734910
1786808259343647755603278279938041928352841299637769724250061850
1169778771499506237102092848603955533107241569277420812142222128
8942541562609759606279776379376647753253121940351755870942189780
2902303201317200546213304055596547273229543258328466136934551237
8005111419502362665593762155765389632864455031023666951471242370
3970043843884191626429015975078989834706618427205158006735274771
6906480601610644925093160066953220006986065422049226212008252162
```

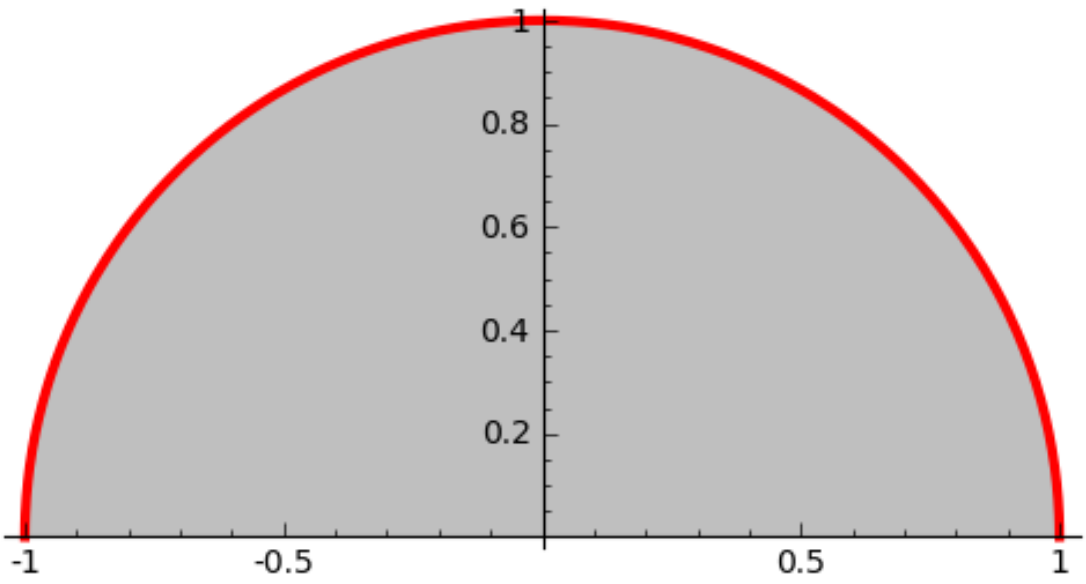
```
339885718786936207969256237041382613823161350119119429459091502
087963748646415195807484336944765521070550962312638973564387487
956617645497124616853081094950235251864257209358302822682381838
741391151438679870522123091154030115425650722039163042527468471
728556538800193064808839182079147256993832713170403686970379758
850055765710815089206639349980448370748033579146000230115852408
995510261420620024711857820370964177245278356271585944194739806
330636050397092879438893520526866531369671015628647982314388141
390391151262584465141695984982364444981061689940426402678696883
628623163181399562948689541589417374862328392215950494778011076
168915288975095394927862912695483022350031666986962394011941027
862267001848167890754404152582912592274949659557728846373227036
141101871946136303255638946038956530286598425547576754297880487
909166842506531697578529691272393821106179926353716178985129634
966201903031749806863622741998706119661273804076243986703538996
761962783334570101233626149346796901551547 01/406778287548421232
116173286708328445176689826533133606781392631199236998885662809
348557356671155290956430174751875095360251105798206742735127104
506781413949219095421900994613025586328298832520783087510115995
046248385369525403221891677588377804627673251328700334882083228
391516599202161186024347828920828687409082348177640217448975984
291562171363826753518869638462172176786920750397046439419756522
132990632592206233412688684530722774753316622106338000474150486
872200330079604203595393425649170599073986150285231863177976366
604624165361705540811548027344347944843318379754195828447139714
595682395357496145607242282137985347126595741797846790764803034
416804886662134187828545003714490044646281371516932782486114054
105441634514379785943438321498699491374429974941411916188300692
706620767186029329173570604490337945916332972221546248700476460
502278980483284010730068357338544234999822475021177554809695612
088131965551967459900917790184504128348366688399796247714348331
627651970837037774822010789350031850608000787729504461939133380
440429812707329238458478406610878319190930005297973995990421818
292585508525349195563753868289669013772157736583348697099384043
434454763548827291143983292206607531343281703691436870445680812
392732239105181275126941391265086958312816788605137121813337291
904365496018849941845190976573970615503284681408750206307284620
642301419081361189982078607311092373650759222409887083418418935
938423741879013097092493343757109949857355576727886100806294081
015171554449734793702597134820310225203084105776449433119524680
386618196964704137014248022985935487769788321086562349102641095
059115933183860757493359204400959331871604592993396432099609960
440125644540573643065630406530215220384502737018981682502335606
423315944686411791581384282773497317037372199630184176323632993
```

```
943867832767950499011442952233637773101716823994314895126742837
221666388715006803961777072989016656984486829254799709389454885
152072863733457370630217061791801859393643999498131050846806375
098267577867063578291309807040813378428531000012772498467389977
785823852018104843245188926381179282263086482208903119981671229
664970723542982390019017138746484439169044116493639168705177438
271394960123062314017832363588783497786530171331707763932291900
```

Symbolic Calculus

Symbolic Calculus makes use of **Maxima** and **Ginac** under the hood.

```
var('x')
f = sqrt(1 - x^2)
plot(f, thickness=3, color='red', aspect_ratio=1, fill=True)
```



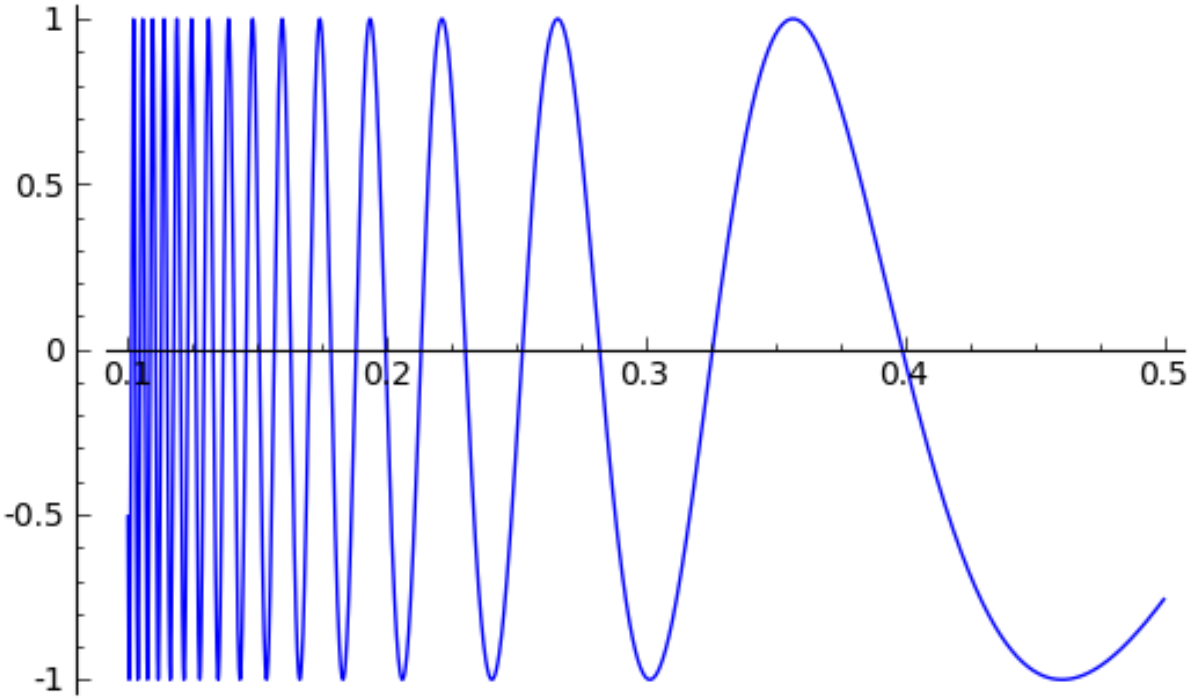

```
var('t')
assume(t+1 > 0)
g = integrate(f, (x, -1, t)); show(g)
```

```
show(g(t=1) - g(t=-1))
```

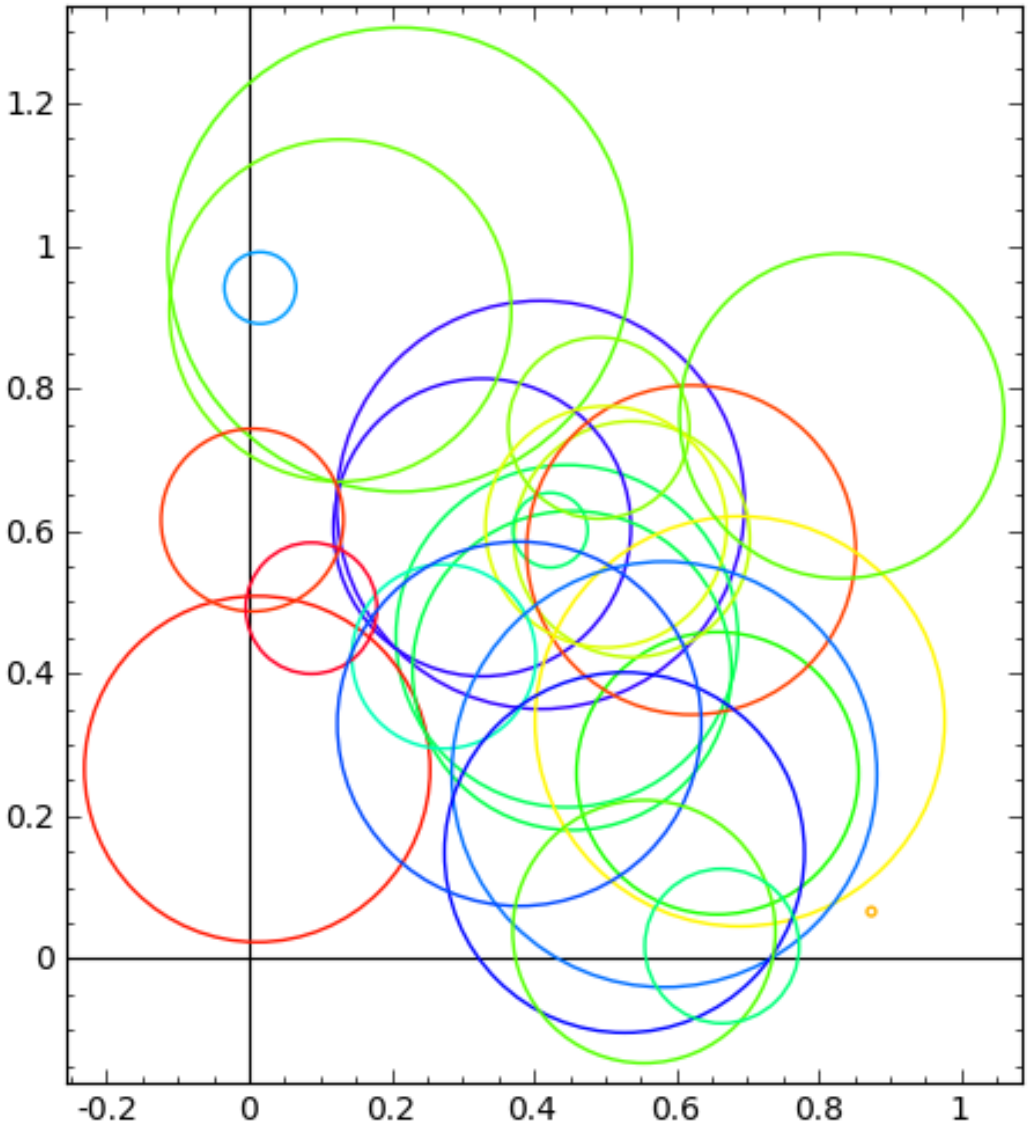
$$\frac{1}{2}\pi$$

Plotting in 2D

```
plot(sin(1/x^2), (x,.1,.5))
```



```
G = sum(circle((random(), random()), random()/3,
        color=hue(random())) for i in range(25))
G.show(aspect_ratio=1, frame=True)
```



Plotting in 3D

```
f(x,y) = sin(x - y) * y * cos(x)
plot3d(f, (x,-3,3), (y,-3,3), opacity=.9, color='red')
```

Get Image

```
G = sum(sphere((random(), random(), random()), random()/4,
        color=hue(random()), opacity=.6)
        for i in range(20))
G.show(aspect_ratio=1, frame=True)
```

Get Image

Plotting a 3D Model

See http://www.davidson.edu/math/chartier/Starwars/

```
# Yoda! (53,756 vertices)
```

```
from scipy import io
yoda = io.loadmat(DATA + 'yodapose.mat')

from sage.plot.plot3d.index_face_set import IndexFaceSet
V = yoda['V']; F3=yoda['F3']-1; F4=yoda['F4']-1
Y = IndexFaceSet(F3,V,color=Color('#444444')) +
IndexFaceSet(F4,V,color=Color('#007700'))
Y = Y.rotateX(-1)
Y.show(aspect_ratio=1, frame=False, zoom=1.2)
```

Get Image

Questions

????