Symbolic Computation: Introduction of Sage

Introduction: Textbook Example

In mathematics, a **Mersenne** prime is a prime **number** that is one less than a power of two. That is, it is a prime **number** of the form $M_n = 2^n - 1$ for some integer n. They are named after Marin **Mersenne**, a French Minim friar, who studied them in the early 17th century.
Marin Mersenne found these: n=2, 3, 5, 7, 13, 17, 19, 31, 67, 127, 257. [?]

- In 1903 at a meeting of the American Sagel Society, F.N. Cole read a paper entitled "On the Factorization of Large Numbers." When called upon to speak, Cole walked to the board and, saying nothing, raised two to its sixty-seventh power and subtracted one from the answer. Then he multiplied, longhand, 193,707,721 by 761,838,257,287 and the answer agreed. Without having said a word, Cole sat down to a standing ovation.
- Afterwards he announced that it has taken him twenty years of Sunday afternoons to factorize the Mersenne number 2^{67}-1.
- Now, after 100 years, we have Sage/Sage/Maple, and the situation is totally different. It now takes only a fraction of a second to do the factorization.

Getting Started and Getting Help

- Getting Sage: http://mirrors.mit.edu/sage/index.html

- Opening Sage
  - –In Windows and Macintosh environments: open Sage as other programs
  - –In Unix with X window: type sage —notebook=jupyter

- When you open Sage in a notebook environment, you get an empty notebook.

- At a basic level, we type in a command and Sage displays the result.

- Type 2^67-1 and then press Shift+Enter in the key pad or Enter in the number pad.

- Type Factor[2^67-1].
  Be sure to type the capital letters and square bracket exactly as they appear.

- In the notebook, you can call help()

Overview of Sage

https://en.wikipedia.org/wiki/SageMath

SageMath (previously **Sage** or **SAGE**, "System for Algebra and Geometry Experimentation"[3]) is a computer algebra system with features covering many aspects of mathematics, including algebra, combinatorics, graph theory, numerical analysis, number theory, calculus and statistics.

The first version of SageMath was released on 24 February 2005 as free and open-source software under the terms of the GNU General Public License version 2, with the initial goals of creating an "open source alternative to Magma, Maple, Mathematica, and MATLAB".[4] The originator and leader of the SageMath project, William Stein, is a mathematician at the University of Washington.

SageMath uses a syntax resembling Python's,[5] supporting procedural, functional and object-oriented constructs.

- A browser-based notebook for review and re-use of previous inputs and outputs, including graphics and text annotations. Compatible with Firefox, Opera, Konqueror, Google Chrome and Safari. Notebooks can be accessed locally or remotely connection can be secured with HTTPS.
- A text-based command-line interface using IPython.
- Support for parallel processing using multi-core processors, multiple processors, or distributed computing.
- Calculus using Maxima and SymPy.
- Numerical linear algebra using the GSL, SciPy and NumPy.
- Libraries of elementary and special mathematical functions.
- 2D and 3D graphs of symbolic functions and numerical data.
- Matrix manipulation, including sparse arrays.
- Multivariate statistics libraries, using R and SciPy.
- A toolkit for adding user interfaces to calculations and applications[7].
- Graph theory visualization and analysis tools

https://en.wikipedia.org/wiki/SageMath

- A browser-based notebook for review and re-use of previous inputs and outputs, including graphics and text annotations. Compatible with Firefox, Opera, Konqueror, Google Chrome and Safari. Notebooks can be accessed locally or remotely connection can be secured with HTTPS.
- A text-based command-line interface using IPython.
- Support for parallel processing using multi-core processors, multiple processors, or distributed computing.
- Calculus using Maxima and SymPy.
- Numerical linear algebra using the GSL, SciPy and NumPy.
- Libraries of elementary and special mathematical functions.
- 2D and 3D graphs of symbolic functions and numerical data.
- Matrix manipulation, including sparse arrays.
- Multivariate statistics libraries, using R and SciPy.
- A toolkit for adding user interfaces to calculations and applications[7].
- Graph theory visualization and analysis tools.
- Libraries of number theory functions.
- Support for complex numbers, arbitrary-precision arithmetic, and symbolic computation.
- Technical word processing including formula editing and embedding SageMath within LaTeX documents[8].
- The Python standard library, including tools for connecting to SQL, HTTP, HTTPS, NNTP, IMAP, SSH, IRC, FTP and oth
- Interfaces to some third-party applications like Mathematica, Magma, R, and Maple.
- MoinMoin as a Wiki system for knowledge management.
- Documentation using Sphinx.
- An automated test-suite.
- Execution of Fortran, C, C++, and Cython code[9].
- SageMath can pull up Mathematica within a program.[10] Interfacing this way is documented officially to Sage.[11]

Getting Started and Getting Help

- Like in MATLAB, the execution (or evaluation) results can be suppressed by a semicolon. Try 34^34 and 34^34; but assigning a variable will not print a value!
- Sage is case sensitive. Try plot(sin(x),(x,0,10)) and plot(Sin(x),(x,0,10))
- In a notebook environment, you can do standard editing as with a word processor.
  - For example, you can move the cursor to the desired location by pressing the arrow keys or with the mouse.
  - After the editing, you can leave the cursor where it is and execute the command.

Naming and Decimals

- Sage will not forgive you even the smallest error in syntax.
- Be especially careful with small and capital letters.
- You probably will occasionally get error message and wrong results because the syntax was not correct.
- Getting used to Sage takes time.
- Important conventions:
  - Built-in Sage names begin usually with a lowercase letter, such as sin or integrate.
  - Variables need to specified using the var statement for example: var('x','y') or var('x y')
  - All arguments are given in parantheses () such as sin(x) and integrate(a+bx,x).
  - Lists are specified the same way as in Python: modifiable [], tuples (), dictionaries etc {}, to make executable lists like in matlab we will need to specify them as vector or matrix.

Grouping

- a/b*c is interpreted as (a/b)*c not as a/(b*c). Try [e^-1, e^-1/2, e^(-1/2)].
- Remember to write the necessary parentheses.
- If you are uncertain whether you should use parentheses or not, go ahead and use them, because unnecessary parentheses are harmless.

Lists

- lists are as in python

Naming and Decimals

- Giving names
  - Assign **value** to **a**: **a** = **value**
  - Show the value of a: a
  - Clear the value of a: a =.
- Decimal values
  - A decimal value is not automatically computed. Try 10/4.
  - If the expression contains a decimal number, the result is also a decimal number. Try 10./4.
  - Calculate a decimal value for **expr**: N(expr)
    Try N(10/4).

Symbolic and Numerical Computation

- Sage distinguishes between exact integers and approximate real numbers.
- When given exact input, Sage will not make approximations, even if it means nothing can be done with the input.
  - sqrt(2): Sage does not compute the square root because there is no simpler way to represent the square root of two *exactly* other than by saying that it is the square root of two.
  - sqrt(2.): The input included a decimal point, indicating that it was meant as an approximate real number. This causes Sage to give an approximate real-number result.
- There is no arbitrary limit to the number of digits in exact integer or gloating-point numbers in Sage. Try N(pi, 500000) and factorial(100000).

Basic Calculations and Plotting

- Basic arithmetic
  - Plus, minus, division, and power: a+b, a-b, a/b, and a^b
  - **Multiplications** must be expressed by an asterisk (*): a*b. Try a=5; b=3; a*b
- Basic constants
  - Try a= vector([pi, e, I, Infinity]) and N(a)
- Basic functions
  - sqrt(z) and exp(z)
  - log(z) and log(b,z): natural log and logarithm to base b
  - sin(z): z is in radians
  - Try exp(log(sqrt(z)))
- Basic plotting
  - plot(x^3 - 2*x - 5,(x,-3,3))

Some Past Projects

Analyzing and Comparing Different Numerical Techniques for solving the Heat Equation: Eigenfunction Expansion Versus Finite Difference

Mathematical Modeling of Arterial Blood Flow

ECG Analysis

Computer Modeling of Groundwater Contaminant Transport

Projects

- first part: presentation, I expect a presentation with powerpoint, keynote, or PDF for about 10 minutes, the presentation includes: introduction, model/problem, method of analysis/approach, results, conclusions.

- Written report

- If you work in a group, I need an independent certification what you have contributed to the group project sent to me by direct email (not canvas).

Format of Project Report

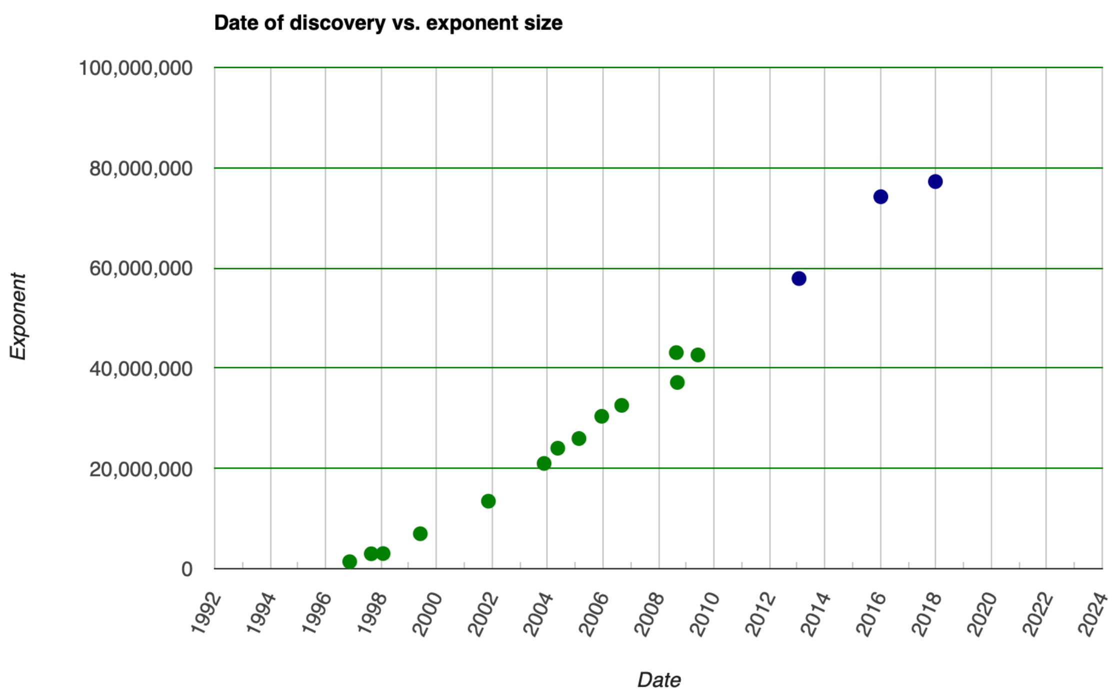
- Single-spaced, typed report. A report generated using the publishing is not enough.
- Problem description (20 points)
  - Project goal
  - Objects of project tasks
- Implementation (40 points)
  - How do you answer each question of the project?
  - You need you include your code as an appendix, and your code should be well documented.
- Results and analysis (30 points)
  - How do you use your code to answer the project questions?
- Conclusions (10 points)
  - What have you learned or concluded from this study?

List of Known Mersenne Prime Numbers

#	2^p-1	Digits	Date Discovered	Discovered By	Method / Hardware	Perfect Number
1	$2^{2}-1$	1	c. 500 BCE	Ancient Greek mathematicians		$2^{1} \cdot (2^{2}-1)$
2	$2^{3}-1$	1	c. 500 BCE	Ancient Greek mathematicians		$2^{2} \cdot (2^{3}-1)$
3	$2^{5}-1$	2	c. 275 BCE	Ancient Greek mathematicians		$2^{4} \cdot (2^{5}-1)$
4	$2^{7}-1$	3	c. 275 BCE	Ancient Greek mathematicians		$2^{6} \cdot (2^{7}-1)$
5	$2^{13}-1$	4	1456	Anonymous	trial division	$2^{12} \cdot (2^{13}-1)$
6	$2^{17}-1$	6	1588	Pietro Cataldi	trial division	$2^{16} \cdot (2^{17}-1)$
7	$2^{19}-1$	6	1588	Pietro Cataldi	trial division	$2^{18} \cdot (2^{19}-1)$
8	$2^{31}-1$	10	1772	Leonhard Euler	Enhanced trial division	$2^{30} \cdot (2^{31}-1)$
9	$2^{61}-1$	19	1883	Ivan Mikheevich Pervushin	Lucas sequences	$2^{60} \cdot (2^{61}-1)$
10	$2^{89}-1$	27	1911 Jun	R. E. Powers	Lucas sequences	$2^{88} \cdot (2^{89}-1)$
11	$2^{107}-1$	33	1914 Jun 11	R. E. Powers	Lucas sequences	$2^{106} \cdot (2^{107}-1)$
12	$2^{127}-1$	39	1876 Jan 10	Édouard Lucas	Lucas sequences	$2^{126} \cdot (2^{127}-1)$
13	$2^{521}-1$	157	1952 Jan 30	Raphael M. Robinson	L-L / SWAC	$2^{520} \cdot (2^{521}-1)$
14	$2^{607}-1$	183	1952 Jan 30	Raphael M. Robinson	L-L / SWAC	$2^{606} \cdot (2^{607}-1)$

43	$2^{30,402,457}-1$	9,152,052	2005 Dec 15	GIMPS / Curtis Cooper & Steven Boone	L-L / Prime95 on 2 GHz Pentium 4 PC	$2^{30,402,456} \cdot (2^{30,402,457}-1)$
44	$2^{32,582,657}-1$	9,808,358	2006 Sep 04	GIMPS / Curtis Cooper & Steven Boone	L-L / Prime95 on 3 GHz Pentium 4 PC	$2^{32,582,656} \cdot (2^{32,582,657}-1)$
45	$2^{37,156,667}-1$	11,185,272	2008 Sep 06	GIMPS / Hans-Michael Elvenich	L-L / Prime95 on 2.83 GHz Core 2 Duo PC	$2^{37,156,666} \cdot (2^{37,156,667}-1)$
46	$2^{42,643,801}-1$	12,837,064	2009 Jun 04	GIMPS / Odd M. Strindmo	L-L / Prime95 on 3 GHz Core 2 PC	$2^{42,643,800} \cdot (2^{42,643,801}-1)$
47	$2^{43,112,609}-1$	12,978,189	2008 Aug 23	GIMPS / Edson Smith	L-L / Prime95 on Dell Optiplex 745	$2^{43,112,608} \cdot (2^{43,112,609}-1)$
48*	$2^{57,885,161}-1$	17,425,170	2013 Jan 25	GIMPS / Curtis Cooper	L-L / Prime95 on Intel Core2 Duo E8400 @ 3.00GHz	$2^{57,885,160} \cdot (2^{57,885,161}-1)$
49*	$2^{74,207,281}-1$	22,338,618	2016 Jan 07	GIMPS / Curtis Cooper	L-L / Prime95 on Intel i7-4790 @ 3.60GHz	$2^{74,207,280} \cdot (2^{74,207,281}-1)$
50*	$2^{77,232,917}-1$	23,249,425	2017 Dec 26	GIMPS / Jon Pace	L-L / Prime95 on Intel i5-6600 @ 3.30GHz	$2^{77,232,916} \cdot (2^{77,232,917}-1)$

* Provisional ranking, not all candidates between M43,112,609 and M77,232,917 have been eliminated



Perfect number

From Wikipedia, the free encyclopedia

For the 2012 film, see Perfect Number (film).

In number theory, a **perfect number** is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself (also known as its aliquot sum). Equivalently, a perfect number is a number that is half the sum of all of its positive divisors (including itself) i.e. $\sigma_1(n) = 2n$.

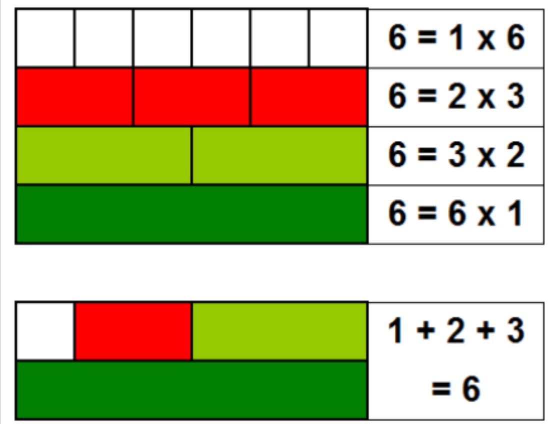


Illustration of the perfect number status of the number 6

This definition is ancient, appearing as early as Euclid's Elements (VII.22) where it is called τέλειος ἀριθμός (*perfect*, *ideal*, or *complete number*). Euclid also proved a formation rule (IX.36) whereby $q(q+1)/2$ is an even perfect number whenever q is a prime of the form $2^p - 1$ for prime p—what is now called a Mersenne prime. Two millenia later, Euler proved that all even perfect numbers are of this form.[1] This is known as the Euclid–Euler theorem.

It is not known whether there are any odd perfect numbers, nor whether infinitely many perfect numbers exist.

Basic Calculus

- Derivative of **expr** with respect to x: diff(expr,x). Try diff(x * sin(x), x).
- Infinite integral of **expr** with respect to **x**: integrate(expr,x). Try integrate(x*cos(x) + sin(x), x).
- Definite integral of **expr** with respect to **x** from **a** to **b**: integrate(expr,(x,a,b)).

  Try integrate(x * sin(x), (x,0,1)).

Getting Help

- If you already know or can guess the function you need, and just want to be reminded of its arguments, the ? Operator is a quick way to get information without opening a new window.
- Generate a random number between 1 and 6.
- Throw a die 10 times and find the maximum of the results.

Tutorials

https://doc.sagemath.org/html/en/tutorial/

https://wiki.sagemath.org/quickref?action=AttachFile&do=get&target=quickref-linalg.pdf

http://doc.sagemath.org/html/en/index.html

http://www.sagemath.org/tour-quickstart.html

Exercises

Find roots

how to solve equations with 1, 2, 3, … unknowns