Solving Equations and Inequalities

Akram Kalantari

Department of Mathematics
Yazd University

In Sage, equations and inequalities are defined using the operators $==, <=,$ and $>=$ and will return either True, False, or, if there is a variable, just the equation/inequality.

definition

sage: $9 == 9$

True

sage: $9 <= 10$

True

sage: $3 * x - 10 == 5$

$3 * x - 10 == 5$

To solve an equation or an inequality we use using the, aptly named, solve() command. For the moment, we will only solve for x. The section on variables below explains how to use other variables.

solution

sage: solve $(3 * x - 2 == 5, x)$
$[x == (7/3)]$
sage: solve$(2 * x - 5 == 1, x)$
$[x == 3]$
sage: solve$(2 * x - 5 >= 17, x)$
$[[x >= 11]]$
sage: solve $(3 * x - 2 > 5, x)$
$[[x > (7/3)]]$

Equations can have multiple solutions, Sage returns all solutions found as a list.

multiple solutions

sage: solve$\left(x^2 + x == 6, x\right)$

$[x == -3, x == 2]$

sage: solve $\left(2 * x^2 - x + 1 == 0, x\right)$

$[x == -1/4 * I * sqrt\,(7) + 1/4, x == 1/4 * I * sqrt\,(7) + 1/4]$

sage: solve $\left(exp\,(x) == -1, x\right)$ $[x == I * pi]$

The solution set of certain inequalities consists of the union and intersection of open intervals.

inequalities solution

sage: solve $\left(x^2 - 6 >= 3, x\right)$
$[[x <= -3], [x >= 3]]$
sage: solve $\left(x^2 - 6 <= 3, x\right)$
$[[x >= -3, x <= 3]]$

The $solve()$ command will attempt to express the solution of an equation without the use of floating point numbers. If this cannot be done, it will return the solution in a symbolic form.

solve()

sage: solve $(\sin(x) == x, x)$
$[x == \sin(x)]$
sage: solve $(exp(x) - x == 0, x)$
$[x == e^{x}]$
sage: solve $(\cos(x) - \sin(x) == 0, x)$
$[\sin(x) == \cos(x)]$
sage: solve $(\cos(x) - exp(x) == 0, x)$
$[cos(x) == e^{x}]$

To find a numeric approximation of the solution we can use the $find\_root()$ command. Which requires both the expression and a closed interval on which to search for a solution.

$find\_root()$

sage: $find\_root\left(\sin\left(x\right) == x, -pi/2, pi/2\right)$

0.0

sage: $find\_root\left(\sin\left(x\right) == \cos\left(x\right), pi, 3 * pi/2\right)$

3.926990816987241

This command will only return one solution on the specified interval, if one exists. It will not find the complete solution set over the entire real numbers. To find a complete set of solutions, the reader must use $find\_root()$ repeatedly over cleverly selected intervals. Sadly, at this point, Sage cannot do all of the thinking for us. This feature is not planned until Sage 10.

In the previous section we only solved equations in one variable, and we always used x. When a session is started, Sage creates one symbolic variable, x, and it can be used to solve equations. If you want to use an additional symbolic variable, you have to declare it using the $var()$ command. The name of a symbolic variable can be a letter, or a combination of letters and numbers:

$var()$

sage:

$$y, z, t = var("y\ z\ t")$$

sage:

$$phi, theta, rho = var("phi\ theta\ rho")$$

sage:

$$x1, x2 = var("x1\ x2")$$

Note Variable names cannot contain spaces, for example square root is not a valid variable name, whereas $square\_root$ is.

Attempting to use a symbolic variable before it has been declared will result in a NameError.

NameError

sage: u

...

NameError: name 'u' is not defined

sage: solve $(u^2 - 1, u)$

NameError

Traceback (most recent call last)

NameError: name 'u' is not defined

We can un-declare a symbolic variable, like the variable phi defined above, by using the $restore()$ command.

$restore()$

sage: restore($'phi'$)
sage: phi
...
NameError: name 'phi' is not defined

Small systems of linear equations can be also solved using $solve()$, provided that all the symbolic variables have been declared. The equations must be input as a list, followed by the symbolic variables. The result may be either a unique solution, infinitely many solutions, or no solutions at all.

solve()

sage: solve $([3*x-y==2,-2*x-y==1],x,y)$
$[[x==(1/5),y==(-7/5)]]$
sage: solve $([2*x+y==-1,-4*x-2*y==2],x,y)$
$[[x==-1/2*r1-1/2,y==r1]]$
sage: solve $([2*x-y==-1,2*x-y==2],x,y)$
[]

In the second equation above, $r1$ signifies that there is a free variable which parametrizes the solution set. When there is more than one free variable, Sage enumerates them $r1, r2, \cdots, rk$.

free variable

sage: solve $([2*x+3*y+5*z == 1, 4*x+6*y+10*z == 2, 6*x+9*y+15*z == 3], x, y, z)$

$[[x == -5/2*r1 - 3/2*r2 + 1/2, y == r2, z == r1]]$

$solve()$ can be very slow for large systems of equations. For these systems, it is best to use the linear algebra functions as they are quite efficient.

Solving inequalities in several variables can lead to complicated expressions, since the regions they define are complicated. In the example below, Sages solution is a list containing the point of interesection of the lines, then two rays, then the region between the two rays.

solve()

sage: solve

$$([x - y >= 2, x + y <= 3], x, y)$$

$$[[x == (5/2), y == (1/2)], [x == -y+3, y < (1/2)], [x == y+2, y < (1/2)],$$

sage: solve

$$([2 * x - y < 4, x + y > 5, x - y < 6], x, y)$$

$$[[-y + 5 < x, x < 1/2 * y + 2, 2 < y]]$$

Finding the roots of an equation (also known as the zeros of the equation) is closely related to solving an equation. Callable symbolic expressions in Sage have a special method that finds their roots.

roots()

$f(x) = 2 * x^2 + 3 * x - 1$
$f.roots()$
$[(-1/4 * sqrt(17) - 3/4, 1), (1/4 * sqrt(17) - 3/4, 1)]$
$f(x) = x^2 - 2 * x + 1$
$f.roots()$
$[(1, 2)]$

There are many ways to manipulate relational expressions with Sage.

manipulating expressions

$exp1 = (x+3)^3 == (x-1)^2$

$exp1.lhs()$

$(x+3)^3$

$exp1.rhs()$

$(x-1)^2$

$exp1.multiply\_both\_sides(x^2)$

$x^2 * (x+3)^3 == x^2 * (x-1)^2$

$exp1.add\_to\_both\_sides(7*x)$

$(x+3)^3 + 7*x == (x-1)^2 + 7*x$

$exp1.divide\_both\_sides(7)$

$1/7 * (x+3)^3 == 1/7 * (x-1)^2$

Expanding a polynomial is the process of converting the polynomial from a product of sums to a sum of products. Factoring is the opposite process, in which a sum of products is converted into a product of factors.

expanding and factoring polynomials

$expr = (y-7)/(x^2+1) == x^3 - 5$

$expr.expand()$

$y/(x^2+1) - 7/(x^2+1) == x^3 - 5$

$expr.expand('left').lhs()$

$y/(x^2+1) - 7/(x^2+1)$

$expr.expand('right').rhs()$

$x^3 - 5$

$expr.factor()$

$(y-7)/(x^2+1) == x^3 - 5$

working with rational functions

$var('s')$
$f(s) = (s+1)/(s^2 * (s+2)^3)$
$f.numerator()$
$s- > s+1$
$f.denominator()$
$s- > (s^2 * (s+2)^3)$
$f.expand\_rational()$
$s- > 1/((s+2)^3 * s) + 1/((s+2)^3 * s^2)$
$f.partial\_fraction()$
$s- > 1/16/(s+2) - 1/16/s + 1/8/s^2 - 1/4(s+2)^3$

We defined a rational function, and demonstrated the utility methods numerator and denominator to obtain different parts of the expression.The method expand_rational separates the expression into a sum of terms by multiplying out products of sums and exponentials, splitting the numerator into terms, and distributing multiplication over addition. The method partial_fraction returns the partial fraction expansion of the expression.

Expressions involving trigonometric functions can be difficult to manipulate because of the numerous trigonometric identities that can be used. Fortunately, Sage can automate this process.

manipulating trigonometric expressions

$var('x\ y')$

$f(x,y) = \sin(x) * \cos(x)^3 + \sin(y)^2$

$g(x,y) = f.reduce\_trig()$

$(x,y) -> -\frac{1}{2}\cos(2y) + \frac{1}{8}\sin(4x) + \frac{1}{4}\sin(2x) + \frac{1}{2}$

$g.expand\_trig() \qquad or \qquad g.trig\_expand()$

$-\frac{1}{2}\sin(x)^3 * \cos(x) + \frac{1}{2}\sin(x) * \cos(x)^3$
$+\frac{1}{2}\sin(x) * \cos(x) + \frac{1}{2} * \sin(y)^2 - \frac{1}{2} * \cos(y)^2$

$g.expand\_trig().simplify\_trig()$

$or \qquad g.expand\_trig().trig\_simplify()$

$\cos(x)^3\sin(x) + \sin(y)^2$

There are special rules for manipulating logarithms and radicals, so Sage has special methods for expanding and simplifying expressions involving these operations. There is also a separate method for simplifying rational expressions.

simplifying expressions : Logarithms, rational functions, and radicals

$f(x) = \log(x^2 * \sin(x)/sqrt(1+x))$

$f.expand\_log()$

$-1/2\log(x+1) + 2 * \log(x) + \log(\sin(x))$

$f.simplify\_log()$

$x- > \log(x^2 * \sin(x)/sqrt(1+x))$

$f(x) = (x+1)/(x^2+x)$

$f.simplify\_rational()$

$x- > 1/x$

$f(x) = sqrt(x^2+x)/sqrt(x)$

$f.simplify\_radical()$

$x- > sqrt(x+1)$

Method(s)	Description
`expand_log log_expand`	Expands logarithms of powers, logarithms of products, and logarithms of quotients
`simplify_log`	Attempt to simplify an expression involving logarithms
`log_simplify`	
`simplify_rational rational_simplify`	Attempt to simplify an expression involving rational expressions
`radical_simplify simplify_radical exp_simplify`	Attempt to simplify an expression involving radicals
`simplify_exp`	
`simplify_factorial`	Simplify an expression by combining factorials and expanding binomials into factorials
`factorial_simplify`	
`simplify_full`	Applies the following operations, in order: `simplify_factorial`, `simplify_trig`, `simplify_rational`, `simplify_radical`, `simplify_log`, and again `simplify_rational`
`full_simplify`	

Figure: The methods available in Sage for simplifying and expanding expressions

THE END