Example Integration of LyX with Sage

Original by : Murat Yildizoglu
Modified by : Thomas Coffee

March 30, 2012

1 Overview

Sage (http://sagemath.org) is an open source computer algebra system that integrates a wide variety of open source mathematics software, with the mission of "creating a viable free open source alternative to Magma, Maple, Mathematica, and Matlab." Sage uses Python as the glue for integrating nearly 100 open source packages for mathematics, programming, and visualization. Sage speaks LaTeX and can be used for literate programming that mixes normal text with Sage instructions, directly including results in PS/PDF files. The `sagetex.sty` package allows the input of Sage instructions in a LaTeX file (in a manner similar to Sweave and R-Project).

The file sage.module provides the following SageTeX elements in LyX:

- `sageblock` and `sagesilent` environments for blocks of code made visible or invisible in the final document, respectively

- `sageplot` and `sageplotcenter`[1] environments that can be used for plotting code inside a Figure float

- `sagecommand` and `sageinline` insets[2] that can be used inline, where results of the latter appear in math mode

- `sagedisplay` environment, whose results appear in display math mode[3]

The SageTeX commands `\sage` and `\sageplot`, along with other custom commands defined in sage.module, can also be used as ERT. This can be useful in math mode when results from Sage should be combined seamlessly with other equation components. Note that LyX's special handling of math mode formatting can cause problems for some Sage input: for this reason, it may be necessary to assign a result to a named variable using a `sagesilent` environment and include the variable by name in ERT where the result should appear.

---

[1] a customized call to `sageplot` defined in the sage.module file; you can define similar environments with different options to conveniently generate multiple plots of different styles

[2] accessible via the menu item Insert >> Custom Insets

[3] you can define similar environments in sage.module to support other math environments

When a SageTEX document `mydocument.tex` is compiled with (pdf)latex, it automatically generates a Sage file `mydocument.sage` that can be run through Sage using the command `sage mydocument.sage`. This compiles the results of the Sage instructions and a supplementary run through (pdf)latex includes these results (computations, plots, etc.) in the final document. The files included with this example provide a basic capability to combine this functionality with LyX, which you can extend to meet your specific needs.

Using this module, to compile a document with embedded Sage code one must view the output as "PDF (pdflatex+sagetex)", located by default under "Other Formats." To update only the LATEX without updating any results from Sage, one can use the default output format "PDF (pdflatex)". This default can be changed in the included `preferences` file.

If you are working under GNU/Linux, the included shell script `setup.sh` can be customized to place the necessary components where they belong. This assumes you have already installed and tested LyX and Sage independently, and includes the necessary commands to include SageTEX in a common LATEX installation.[4] After establishing the configuration, you will need to run Tools >> Reconfigure from the LyX menu. This configuration has been successfully tested using the following setup:

- Ubuntu 11.10 x86_64

- LyX 2.0.0 with TEXLive 2009-13 (but should work with LyX 1.6+)

- Sage 4.8 [2012-01-20]

Note that in earlier versions of Sage, SageTEX may produce files with extension `.sage` rather than `.sagetex.sage`. You may need to modify the `preferences` and `compile-pdf-sage.sh` files accordingly.

2 Examples

Sage can output LATEX: $\begin{pmatrix} 7 & 10 \\ 15 & 22 \end{pmatrix}$

Sage can do plots : Figure 1.

SageTEX can include blocks for defining and computing elements. In this case, the block is not typeset in the final document, since we use the `sagesilent` environment. We declare x as a variable and $y = x^2 + 5x + 2$.

We can now compute an equation with an unknown x: the roots of equation $x^2 + 5x + 2$ are hence given by (computed by Sage):

$$\left[x = -\frac{1}{2}\sqrt{17} - \frac{5}{2}, x = \frac{1}{2}\sqrt{17} - \frac{5}{2} \right]$$

Another equation, third degree of x this time: roots of $x^3 - 10$ are:

$$\left[x = \frac{1}{2}i \cdot 10^{\frac{1}{3}}\sqrt{3} - \frac{1}{2} \cdot 10^{\frac{1}{3}}, x = -\frac{1}{2}i \cdot 10^{\frac{1}{3}}\sqrt{3} - \frac{1}{2} \cdot 10^{\frac{1}{3}}, x = 10^{\frac{1}{3}} \right]$$
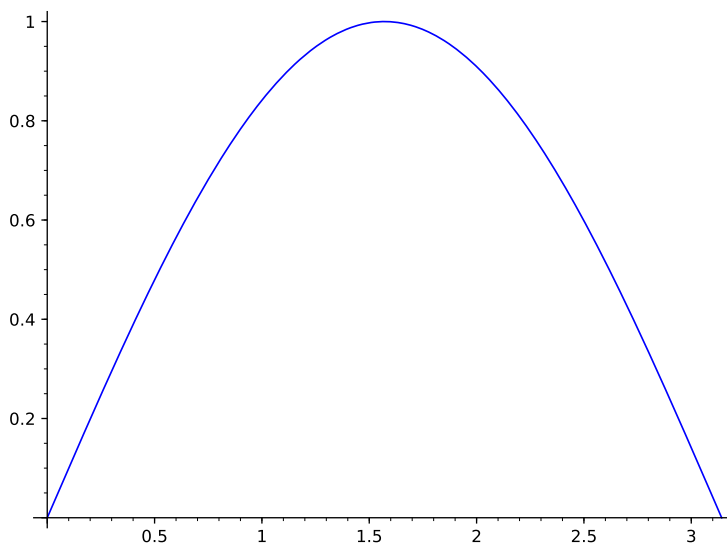
---

[4]based on http://www.sagemath.org/doc/installation/sagetex.html

A system of two equations:

```
x,y=var('x y')
z=solve([x+y==6,x-y==4],x,y)
```

The roots of the system

$$\begin{aligned} x + y &= 6 \\ x - y &= 4 \end{aligned}$$

are given by :

$$[[x = 5, y = 1]]$$

Sage can do differentiation:

```
y=sin(x^2)
```

$$\frac{\partial^4 y}{\partial x^4} = 16\,x^4 \sin\left(x^2\right) - 48\,x^2 \cos\left(x^2\right) - 12\,\sin\left(x^2\right)$$

Sage can also do integrals:

$$\int_0^1 \frac{x}{x^2 + 1}$$

is given by $\frac{1}{2} \log(2)$.

Sage can solve differential equations: we declare a variable t, a function of t, $x(t)$ and a differential equation

$$\frac{dx}{dt} + x - 1 = 0$$

3

```
t = var('t')
x = function('x',t)
DE = diff(x, t) + x - 1
```

We can now solve the equation:

$$x(t) = (C + e^t)e^{(-t)}$$