SageTEX tutorial

Pauline Hubert

FPSAC software days, July 2019

Why use SageTEX?

SageTEX is a Sage package that allows you to add Sage code and run it directly in a LATEX document.

Why use SageTEX?

SageTEX is a Sage package that allows you to add Sage code and run it directly in a LATEX document.

- Print Sage code
- Print the result of Sage commands without having to copy and paste
- Plot graphs
- Generate multiple versions of the same document

How to use it?

1. Copy the file `sage.sty` in your working directory.
   This file can be found at

   `$SAGE_ROOT/local/share/texmf/tex/latex/sagetex`

   where `$SAGE_ROOT` is where you installed sage on your computer.
   Or click on this link to download it.
2. Add the package `sagetex` in your tex file.
3. Add SageTex commands in your latex.
4. Compile normally. This will generate a `.sagetex.sage` file.
5. Execute this file in a terminal.

   `sage FILE_NAME.sagetex.sage`
6. Compile your latex once again.

Inline Sage

Include a Sage output in the text in maths mode.

What I write:

```
This is an example $2+2 = \sage{2+2}$.
The integer $150$ admits
$\sage{number_of_partitions(150)}$ partitions.
```

What I get:

This is an example $2 + 2 = 4$.
The integer 150 admits 40853235313 partitions.

Sage block

You can print sage code using `sageblock`. For example, I write:

```
To get the list all the integer from 5 to 25 and
define the function f, I type in Sage :
\begin{sageblock}
        [i for i in range(5, 26)]
        f(x) = exp(x) * sin(2*x)
        print f
        f
\end{sageblock}
```

And I get :

To get the list all the integer from 5 to 25, I type in Sage :

```
[i for i in range(5, 26)]
f(x) = exp(x) * sin(2*x)
print f
f
```

Sage commandline

If you want to print Sage code and its output use `sagecommandline`.

What I write

```
\begin{sagecommandline}
        sage: [i for i in range(5, 26)]
        sage: f(x) = exp(x) * sin(2*x)
        sage: print f
        sage: f
\end{sagecommandline}
```

Sage commandline

If you want to print Sage code and its output use `sagecommandline`.

What I get

```
sage: [i for i in range(5, 26)]                                    1
[5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,                        2
   17, 18, 19, 20, 21, 22, 23, 24, 25]
sage: f(x) = exp(x) * sin(2*x)                                     3
sage: print f                                                      4
sage: f                                                            5
x |--> e^x*sin(2*x)                                               6
```

Sage silent

You may want to run some Sage commands but without printing them in the text. This is what `sagesilent` does.

I can write :

```
\begin{sagesilent}
        var('x,y')
        M = matrix([[i*x+j*y for i in range(3)]
                            for j in range(3)])
\end{sagesilent}

And later in the text you may want to define
the matrix $M$ as
$$M:= \sage{M}$$
and print its determinant which is
$\sage{M.determinant()}$.
```

Sage silent

You may want to run some Sage commands but without printing them in the text. This is what `sagesilent` does.

And I get :

And later in the text you may want to define the matrix M as

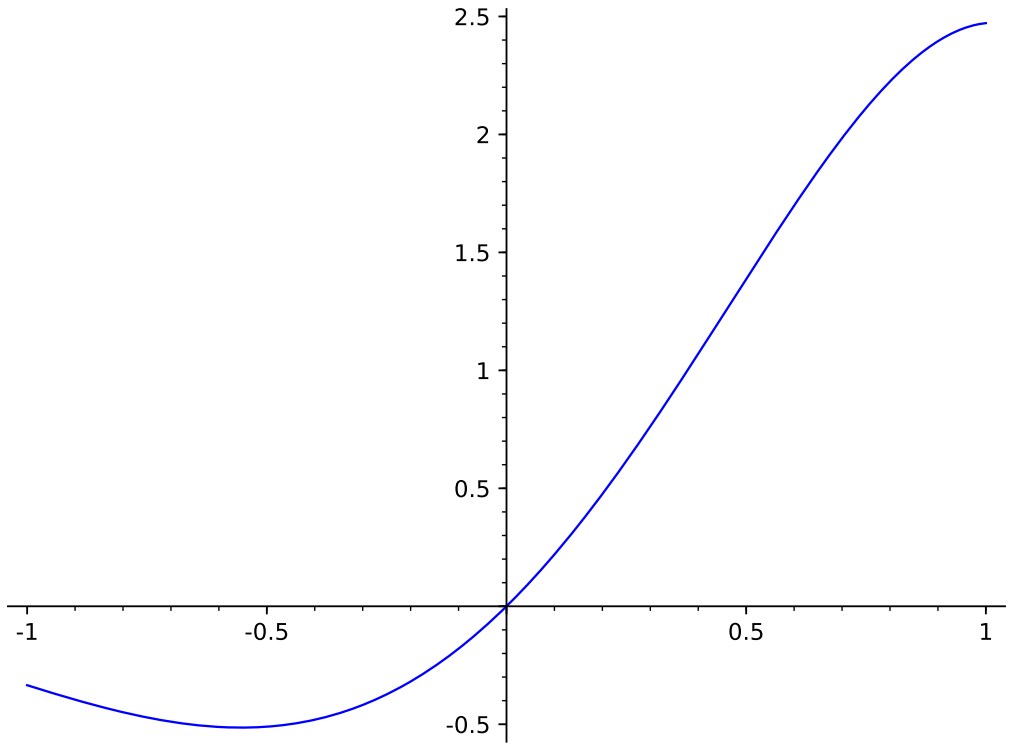
$$M := \begin{pmatrix} 0 & x & 2\,x \\ y & x+y & 2\,x+y \\ 2\,y & x+2\,y & 2\,x+2\,y \end{pmatrix}$$

and print its determinant which is $2\,((2\,x+y)x-2\,(x+y)x)y+2\,((x+2\,y)x-(x+y)x)y$.

Sage plot

You can also use SageTex commands to plot functions and graphs.

```
\begin{center}
        \sageplot[height=5cm]{plot(f, -1, 1)}
\end{center}
```



Sage plot

You can also use SageTex commands to plot functions and graphs.

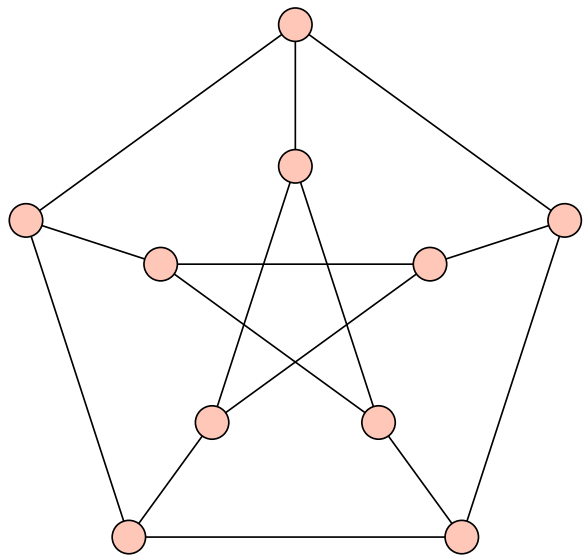
```
\begin{sagesilent}
        G = graphs.PetersenGraph()
        c = G.coloring(hex_colors=True)
\end{sagesilent}

Let's print the Petersen graph.
\begin{center}
\sageplot[height=5cm]{G.plot(vertex_labels=False,
        vertex_size=400)}
\end{center}
```

Sage plot

You can also use SageTex commands to plot functions and graphs.

Let's print the Petersen graph.



Automatically generated files

You can also combine Latex and Sage features. For example ramdom matrices in Sage and file generating commands in Latex, can be combined to automatically obtain several versions of the same homework.

From Aram's website

Semaine 4
26 septembre

Devoir 1

à remettre le 10 octobre

- Aucun travail rédiger au crayon plomb ne sera corrigé.
- Il faut bien écrire votre réponse (e.g. sans trop de ratures).
- Si vous voulez transmettre votre devoir par courriel, il faut envoyer le fichier en format PDF. Aucun scan ni document Word ne sera accepté.
- Vous pouvez remettre votre devoir en français ou en anglais.
- Vous remettez une copie par personne.
- Notez votre code permanent sur la première page.

Faites le devoir qui correspond à votre code permanent. Sinon, vous aurez avoir 0 / 100 sur votre devoir.

Devoir 1 - Code Permanent **ABD - AMO**
Devoir 1 - Code Permanent **BAC - BERN**
Devoir 1 - Code Permanent **BERR - BOUG**
Devoir 1 - Code Permanent **BOUJ - CAR**
Devoir 1 - Code Permanent **CHE - CHI**
Devoir 1 - Code Permanent **COT - DES**
Devoir 1 - Code Permanent **DJO - DOU**
Devoir 1 - Code Permanent **DUF - FAR**
Devoir 1 - Code Permanent **FIL - GAGA**
Devoir 1 - Code Permanent **GAGM - GEO**
Devoir 1 - Code Permanent **GHA - HAM**
Devoir 1 - Code Permanent **HAS - KAS**
Devoir 1 - Code Permanent **LAB - LAM**
Devoir 1 - Code Permanent **LAV - MAN**
Devoir 1 - Code Permanent **MAR - MON**
Devoir 1 - Code Permanent **NUG - PEL**
Devoir 1 - Code Permanent **PEM - RAC**
Devoir 1 - Code Permanent **RAM - YAH**

Automatically generated files

You can also combine Latex and Sage features. For example ramdom matrices in Sage and file generating commands in Latex, can be combined to automatically obtain several versions of the same homework.

For Aram's website

Conclusion

Some references

- SageTEX documentation
  http://ctan.ijs.si/tex-archive/macros/latex/contrib/sagetex/sagetex.pdf
- A nice example
  https://github.com/sagemath/sagetex
- Nadia's notes on SageTEX (french)
  https://nadialafreniere.github.io/sage
- Slides