

# THE SAGE ENGINE

RICHARD KOCH

## 1. SAGE

Sage is a mathematical software system providing an open source alternative to Magma, Maple, Mathematica, and Matlab. It is available from <http://www.sagemath.org/> as a free download. The program provides all of the standard features expected from such a system: arbitrary precision arithmetic, symbolic integration and differentiation, two-dimensional plotting of functions, matrix algebra, and much more.

Download sage from the above web site. As of April, 2019, SageMath 8.7 is provided. Different binaries are provided for different versions of macOS. For this note, we used the Mojave version. Even then, users have the choice of downloading a command line version or a version packaged as a Macintosh application. We picked the Macintosh application.

Move the resulting “Sage” program to /Applications. The application will have a version specific name, currently SageMath-8.7.app. Long ago, the name of this application could be changed, and we recommended changing the name to SageMath.app so upgrades could easily be installed. Then for a time the application broke if renamed. Now we get varying reports about renaming. So we’ll give instructions below for two cases: the case when the application can be renamed SageMath.app and the case when the original name must be used.

In ~/Library/TeXShop/Engines/Inactive/Sage we provide a typesetting engine for Sage. The last lines of this engine read as follows:

```
if [ "$run_sage" = "yes" ]
then
  echo Running Sage, please wait a moment...
  # New versions of Sage cannot be renamed; thus users need to read the
  # documentation in ~/TeXShop/Engines/Inactive/Sage whenever they update Sage;
  # for instance, the line below must contain the current name of Sage

  /Applications/SageMath-8.7.app/Contents/Resources/sage/sage "$sagename"
  pdflatex --file-line-error --synctex=1 "$filename"
else
  echo No Sage commands have changed, so running Sage is unnecessary.
fi
```

---

*Date:* April 4, 2019.

Notice the line five lines from the bottom containing the string *SageMath-8.7.app*. If the Sage program name can be changed, we recommend changing this string to *SageMath.app*. After that, the engine need never be changed as long as the user changes the name of updates to just SageMath.app. But if the name of the Sage application cannot be changed, then this engine will have to be updated with the new name whenever you update Sage.

SageTeX is a piece of the Sage download. It is basically a LaTeX style file, which allows users to embed and process Sage code from within TeX files. The last three pages of this document show this style file in action. The source code on page four is followed by LaTeX output on pages five and six.

## 2. HOW SAGETEX WORKS

In the source file, the initial line

```
% !TEX TS-program = sage
```

tells TeXShop to process the file using the sage engine; this engine first calls pdfflatex, then calls sage, and finally calls pdfflatex again. The remaining lines in the preamble are standard LaTeX commands, except the required line

```
\usepackage{sagetex}
```

In the remaining source, sage commands are entered within lines of the form

```
\sage{...}
```

These line cause sage to process commands and output LaTeX source fragments, which become part of the LaTeX document.

Notice in particular that sage can plot standard functions. Sage can also compute integrals symbolically; for example, look carefully at the command which processes  $\int \frac{x^2+x+1}{(x-1)^3(x^2+x+2)}$ . This command contains standard LaTeX code to display the integral, but then Sage integrates and returns a typeset copy of the result.

## 3. SETTING UP THE ENGINE

This folder contains an engine file named “sage.engine”. Move this file to the active portion of  $\sim$ /Library/TeXShop/Engines. This engine will work unchanged for SageMath-8.7. Otherwise it must be edited as explained earlier in this document. Dan Drake, who is responsible for SageTeX, wrote this engine.

The “Sage” program contains a style file named “sagetex.sty” and a number of support files. This style file is supposed to be copied to your TeX distribution. The file depends on other features of sage, so whenever you upgrade sage, you might also need to upgrade sagetex.sty in your TeX distribution. It is easy to forget to do this.

But there is an ingenious way to work around this upgrade problem. Rather than copying the style file, we just create a symbolic link from it's location in TeX Live to its location in Sage. If we always rename the sage program to "SageMath", then this link will remain valid after updates and need not be changed. Otherwise, it will need to be changed every time Sage is updated. Here is how to create the link:

Run Terminal in /Applications/Utilities. Type the following

```
cd /usr/local/texlive/texmf-local/tex/latex
```

Then type the following on a single line. Terminal may make a linefeed when the line grows too long and that is fine, but do not for instance make a linefeed after the top "local" and before the bottom "/share".

```
sudo ln -s /Applications/SageMath-8.7.app/Contents/Resources/sage/local
/share/texmf/tex/latex/sagetex/sagetex.sty sagetex.sty
```

Change SageMath-8.7.app to SageMath.app if renaming is possible, and change it to the appropriate version number if renaming is not possible and you update Sage.

Finally issue the command

```
sudo mktexlsr
```

#### 4. A DEBUGGING WARNING

This document was revised in April, 2019 for a new version of SageTeX.

The sample document at the end of this report failed completely; no sage command worked. Ultimately that taught us a significant lesson we are passing along here.

The problem in the sample document was a single sage command. We wrote

```
\sage{integrate((x^2+x+1)/((x-1)^3*(x^2+x+ 2)) )}
```

and should have written

```
\sage{integrate((x^2+x+1)/((x-1)^3*(x^2+x+ 2)), x )}
```

Here Sage is being asked to symbolically integrate a rational function. If that function contained both  $x$  and  $y$ , Sage would need to know which is the variable to integrate against. In earlier versions of Sage, apparently, Sage assumed the independent variable was  $x$  if this information was omitted. But now the extra comma and  $x$  are required to indicate the variable.

The key point here is that breaking just one sage command caused them all to fail. Consequently, if you intend to use Sage together with TeX and suddenly nothing works, a little clever debugging will be required to determine and fix the bad sage command.

## 5. FINAL REMARKS

A Sage tutorial is available at the Sage page <http://www.sagemath.org/help.html>. It is definitely recommended. Extensive additional documentation is available at the same web page.

```
% !TEX TS-program = sage
```

```
% The following lines are standard LaTeX preamble statements.
```

```
\documentclass[11pt, onside]{amsart}
\usepackage{geometry}
\geometry{letterpaper}
\usepackage[parfill]{parskip}
\usepackage{graphicx}
\usepackage{amssymb}
\usepackage{epstopdf}
\title{Brief Article}
\author{The Author}
```

```
% Only one command is required to use Sage within the LaTeX source:
```

```
\usepackage{sagetex}
```

```
\begin{document}
\maketitle
\section{Introduction}
```

This is an example of using Sage within a  $\TeX$  document. We can compute extended values like

```
$$32^{31} = \sage{32^31}$$
```

We can plot functions like  $x \sin x$ :

```
\sageplot[width=4in]{plot(x * sin( 30 * x), -1, 1)}
```

We can integrate:

```
$$\int \frac{x^2 + x + 1}{(x - 1)^3 (x^2 + x + 2)} dx = \sage{integrate( (x^2 + x + 1) / ((x - 1)^3 * (x^2 + x + 2)), x)}$$
```

```
\newpage
```

We can perform matrix calculations:

```
$$\sage{matrix([[1, 2, 3], [4, 5, 6], [7, 8, 9]]^3)}$$
```

```
$$AB= \sage{Matrix([[1, 2], [3, 4]])} \sage{Matrix([[5, 6], [6, 8]])} = \sage{Matrix([[1, 2], [3, 4]]) * Matrix([[5, 6], [6, 8]])}
$$
```

Plots are fun; here is a second one showing  $x \ln x$ . The "width" command in the source is sent to the include graphics command in LaTeX rather than to Sage.

```
\sageplot[width=5in]{plot(x * ln(x), 0, 2)}
```

Sage understands mathematical constants and writes them symbolically unless it is told to produce a numerical approximation. The term  $e \pi$  below is not in the LaTeX source; instead it is the result of a Sage calculation, as is the numerical value on the other side of the equal sign.

```
The product of  $e$  and  $\pi$  is  $\sage{\pi * e} = \sage{N(\pi * e)}.$ 
```

```
\end{document}
```

## BRIEF ARTICLE

THE AUTHOR

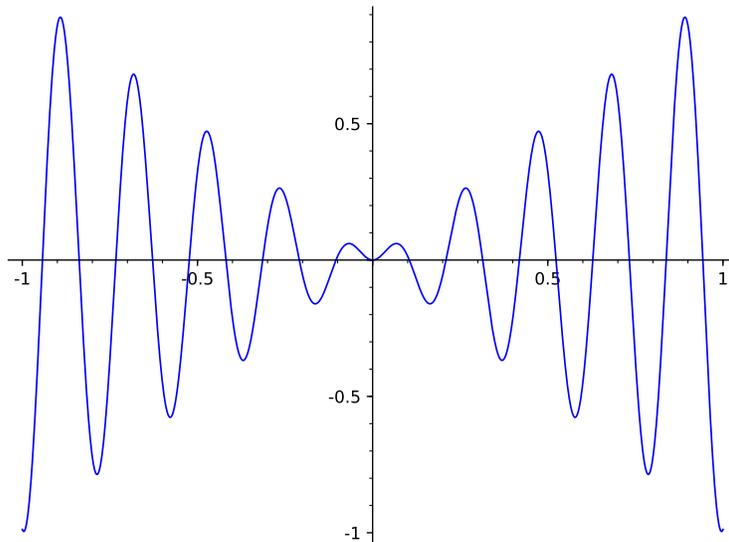
### 1. INTRODUCTION

This is an example of using Sage within a  $\text{\TeX}$  document. We can compute extended values like

$$32^{31} = 45671926166590716193865151022383844364247891968$$

$$1324^9 = 12502356138591322345667559424$$

We can plot functions like  $x \sin x$ :



We can integrate:

$$\int \frac{x^2 + x + 1}{(x-1)^3(x^2+x+2)} dx$$
$$= -\frac{9}{448} \sqrt{7} \arctan\left(\frac{1}{7} \sqrt{7}(2x+1)\right) - \frac{3(x+1)}{16(x^2-2x+1)} + \frac{5}{128} \log(x^2+x+2) - \frac{5}{64} \log(x-1)$$

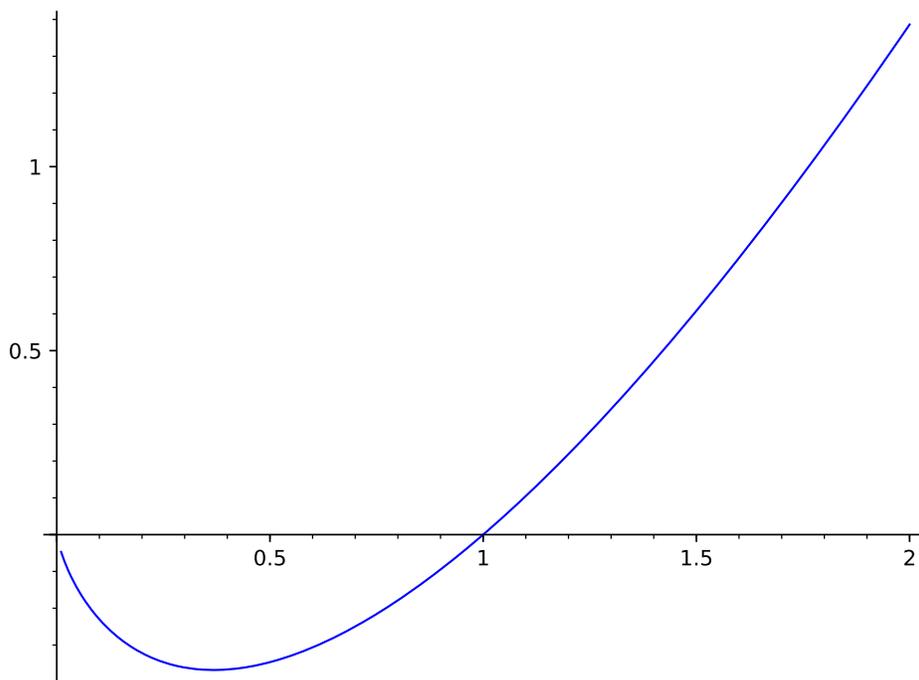
1

We can perform matrix calculations:

$$\begin{pmatrix} 468 & 576 & 684 \\ 1062 & 1305 & 1548 \\ 1656 & 2034 & 2412 \end{pmatrix}$$

$$AB = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 5 & 6 \\ 6 & 8 \end{pmatrix} = \begin{pmatrix} 17 & 22 \\ 39 & 50 \end{pmatrix}$$

Plots are fun; here is a second one showing  $x \ln x$ . The “width” command in the source is sent to the include graphics command in LaTeX rather than to Sage.



Sage understands mathematical constants and writes them symbolically unless it is told to produce a numerical approximation. The term  $e\pi$  below is not in the LaTeX source; instead it is the result of a Sage calculation, as is the numerical value on the other side of the equal sign.

The product of  $e$  and  $\pi$  is  $\pi e = 8.53973422267357$ .