# The latexalpha2 package*

Yi Liu†
me@yliu.io

2019/03/05

## 1 Introduction

latexalpha2 is a LaTeX package that allows you to embed and execute your Wolfram Language (Mathematica) source codes in a LaTeX document. When the document is compiled, the computation results will be inserted into the compiled file. For example,

```
$$ \wolfram{LaplaceTransform[t^4 Sin[t],t,s]} $$
```

gives the Laplace transform of $t^4 \sin t$ and generates

$$\frac{24\left(5s^4-10s^2+1\right)}{\left(s^2+1\right)^5}.$$

It is also quite easy to generate plots or animations with this package. Moreover, all the embedded codes can be executed either locally or on the cloud. In addition, you can also use Mathics[1] (a free alternative to Mathematica) for computations. The main features of the package are somewhat similar to `SageTeX`[2], but here we use Wolfram Language (Mathematica) instead of Sage.
If you have any questions or comments, you are welcome to raise issues or pull requests through the Github repository for this package[3]. For now, latexalpha2 only supports Unix-like systems.
This package is *not* endorsed by or affiliated with Wolfram Research, Inc. in any way.

*This document corresponds to latexalpha v1.1, date 2019/03/05.
†Personal website: `https://yliu.io`.
[1] `http://mathics.github.io`
[2] `https://ctan.org/pkg/sagetex`
[3] `https://github.com/stevenliuyi/latex-alpha2`

# 2 Installation

The Wolfram Language codes are executed using the WolframScript interpreter[4]. So please make sure that WolframScript is properly installed before using latexalpha2. If you'd like to run your codes on cloud, please authenticate first:

```
wolframscript -authenticate
```

Alternatively, if you are using Mathics for computations, please make sure that Mathics is properly installed. Please refer to the Mathics installation guide[5] for more information.
When compiling your document, LaTeX must be invoked with the `-shell-escape` flag in order to run either WolframScript or Mathics. Currently, this package is only tested with pdfLaTeX and XeLaTeX. After putting `\usepackage{latexalpha2}` in the preamable of your document, you can compile the file as:

```
pdflatex -shell-escape mydocument.tex
```

# 3 Usage

## 3.1 Package options

When importing the package as `\usepackage[`⟨*option*⟩`]{latexalpha2}` in your document, there are several options available.
The first pair of options is `local` (default) and `cloud`. As the names suggest, it controls whether the computations are performed locally (via locally installed Mathematica) or on the cloud (via Wolfram Cloud).
The second pair of options is `cache` (default) and `nocache`, which controls whether or not the computation results are cached. Cached results will not be computed again when you compile the document next time if the corresponding Wolfram Language code and output format are not changed.
There is also an option `mathics`, which tells the package to use Mathics (a free, open-source alternative to Mathematica) for computations. The `mathics` mode only supports `\wolfram`, `\wolframgraphics`, `\wolframsolve` and `\wolframdsolve`. Please note that functions may behave differently in Mathics and Mathematica, and not all Mathematica built-in functions are implemented in Mathics.

## 3.2 Macros

`\wolfram` `\wolfram[`⟨*format*⟩`]{`⟨*code*⟩`}` takes any Wolfram Language code, executes it and insert the result into the document. The options for format are `tex` (default), `wolfram` and `text`. For example,

[4] `https://www.wolfram.com/wolframscript`
[5] `https://github.com/mathics/Mathics/wiki/Installing`

```
$$ \wolfram{Series[Exp[x],{x,0,5}]} $$
```

generates a power series expansion for $e^x$ about $x = 0$ to 5th order, and the result is

$$1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120} + O\left(x^6\right).$$

`\wolframgraphics` `\wolframgraphics[⟨format⟩]{⟨code⟩}{⟨filename⟩}` generates a plot from Wolfram Language code and saves the image in the current folder. The file format options are `pdf` (default), `png` and `jpg`. For example, the 3D plot shown in figure 1 is generated by

```
\begin{figure}
  \wolframgraphics{Plot3D[Sin[x]Cos[y],{x,-2Pi,2Pi},{y,-2Pi,2Pi}]}{example}
  \includegraphics{example.pdf}
  \caption{Plot of $f(x,y)=\sin(x)\cos(y)$}
  \centering
\end{figure}
```

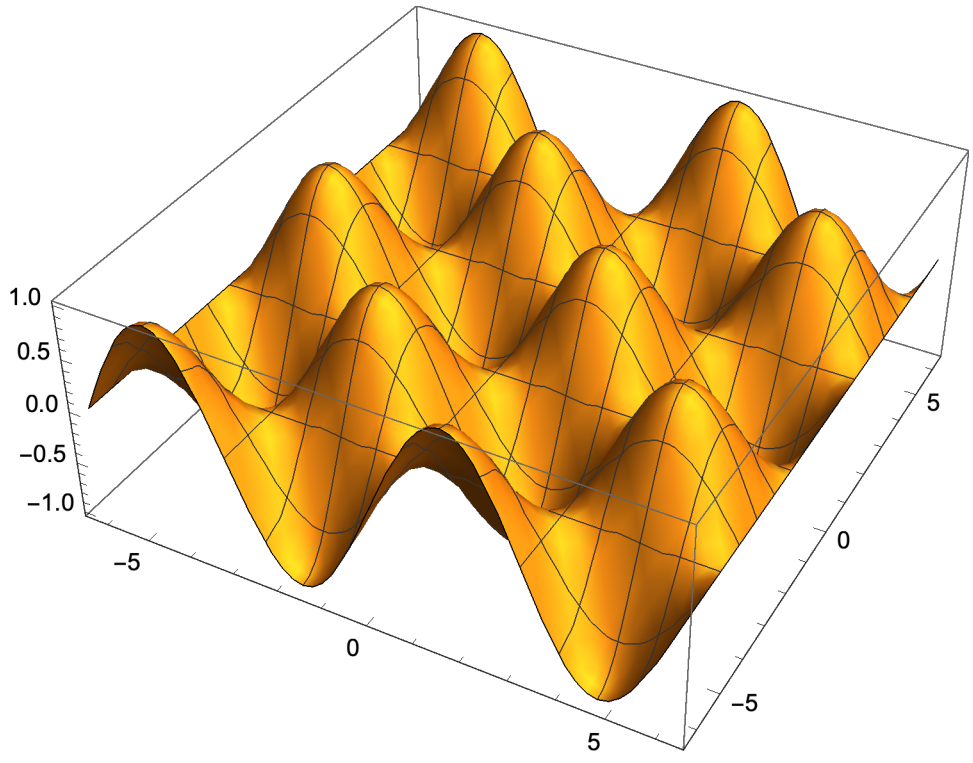


Figure 1: Plot of $f(x, y) = \sin(x)\cos(y)$

In the `mathics` mode, plots are exported as `.asy` files (Asymptote graphics), and

the `asymptote` package[6] is required to import the plots. Here is an example:

```
\begin{figure}
  \wolframgraphics{Plot[Sin[x],{x,0,2Pi}]}{example}
  \input{example.asy}
\end{figure}
```

Note that you need to process the `.asy` files so they can be included in the next compilation. For example,

```
pdflatex -shell-escape mydocument.tex
asy mydocument-*.asy
pdflatex -shell-escape mydocument.tex
```

Please refer to the documentation of the `asymptote` package for more information.

`\wolframalpha` `\wolframalpha[`⟨*format*⟩`]{`⟨*query*⟩`}` sends a query to Wolfram|Alpha and put the result into the document. The options for format are `tex` (default), `wolfram`, `wolfram2` and `text`. The options `wolfram` and `wolfram2` correspond to the pure standard Wolfram Language result and the result generated by free-form input, respectively. In the Wolfram Language documutation[7], the former corresponds to the `WolframResult` format, and the latter corresponds to the `Result` format. The default option `tex` uses the `WolframResult` format and converts the result into the TEX form. Besides, the option `text` generates plain text which is the result of the `ShortAnswer` format. As an example,

```
The population of Shanghai is $\wolframalpha{Shanghai population}$, which
is $\wolframalpha{ratio of Shanghai population and NYC population}$ times the
population of New York City.
```

generates "The population of Shanghai is $2.415 \times 10^7$people, which is 2.814 times the population of New York City."

`\wolframsolve` `\wolframsolve{`⟨*equation*⟩`}{`⟨*variable*⟩`}` solves an equation and display the corresponding results. For example,

```
\wolframsolve{a x^2+b x+c==0}{x}
```

[6] `https://ctan.org/pkg/asymptote`
[7] `https://reference.wolfram.com/language/ref/WolframAlpha.html`

produces

$$x = \frac{-\sqrt{b^2 - 4ac} - b}{2a}$$
$$x = \frac{\sqrt{b^2 - 4ac} - b}{2a}$$

`\wolframdsolve` `\wolframdsolve{⟨equation⟩}{⟨dependent variable⟩}{⟨independent variable⟩}` is similar to `\wolframsolve`, but it solves an differential equation. For example,

```
\wolframdsolve{y'[x]+y[x]==a Sin[x]}{y[x]}{x}
```

produces

$$y(x) = \frac{1}{2}a(\sin(x) - \cos(x)) + c_1 e^{-x}$$

`\wolframtex` `\wolframtex{⟨format⟩}{⟨code⟩}` takes TEX code instead of Wolfram Language code, and performs some simple calculations. The options for format are the same as `\wolfram`, i.e. `tex` (default), `wolfram` and `text`. For example, the result of

```
$$ \wolframtex{\int_a^b\sin(x)\,dx} $$
```

is

$$\cos(a) - \cos(b).$$

`\wolframanimation` `\wolframanimation{⟨code⟩}{⟨foldername⟩}` is similar to `\wolframgraphics`, but it converts any Wolfram Language animation object into a sequence of images, instead of a single image. The images are saved in a subfolder of current folder, named as ⟨*foldername*⟩. You can then use `\animategraphics` from the `animate` package[8] to generate animation. Note that PDF files with animations can only be viewed in a small number of PDF readers, which includes Acrobat Reader. Please refer to the documentation for the `animate` package for more information.

`\wolframtable` `\wolframtable{⟨table⟩}` converts a table in Wolfram Language into TEX form. The macro can be put inside environments such as `tabular`, `tabularx`, etc. For example,

```
\begin{center}
```

[8] `https://ctan.org/pkg/animate`

```
  \begin{tabular}{ccc}
    \hline
    \wolframtable{Join[{{x,x^2,x^3}},Table[{i,i^2,i^3},{i,5}]]}
    \hline
  \end{tabular}
\end{center}
```

generates the following table:

| $x$ | $x^2$ | $x^3$ |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 4 | 8 |
| 3 | 9 | 27 |
| 4 | 16 | 64 |
| 5 | 25 | 125 |

### 3.3 Notes

1. If you want to input backslashs (\) or number signs (#) in your Wolfram Language codes, you could use `\backslash` and `\hash`, respectively. For example, use `\backslash[Alpha]` instead of `\[Alpha]` to represent the Greek letter $\alpha$.

2. Outputs are cached in hidden files named as `.latexalpha2_`$\langle hash \rangle$`.out`, unless the option `nocache` is specified. You can clean the cached outputs manually using the following command if you like:

   ```
   rm .latexalpha2_*.out
   ```

3. `\def` could be utilized to create re-useable code snippets. For example,

   ```
   \def\mat{{{a,b},{c,d}}}
   $$ \wolfram{\mat}^{-1} = \wolfram{Inverse[\mat]}. $$
   ```

   generates

$$\left( \begin{array}{cc} a & b \\ c & d \end{array} \right)^{-1} = \left( \begin{array}{cc} \frac{d}{ad-bc} & -\frac{b}{ad-bc} \\ -\frac{c}{ad-bc} & \frac{a}{ad-bc} \end{array} \right).$$

## 4 Acknowledgement

This package is heavily inspired by LaTeX-Alpha[9], which also explains the name of this package. Unfortunately, LaTeX-Alpha has been down for a while. The objective of this package is to replace LaTeX-Alpha and at the same time provide various new features.

[9] https://github.com/Akollek/LaTeX-Alpha