

软设终审答辩

L^AT_EXHelper

王与进 王治 朱健维

清华大学电子工程系

2022 年 4 月 17 日

① 项目背景

② 项目功能

③ 架构总览

④ 项目创新

① 项目背景

Why L^AT_EX?

Why L^AT_EXHelper?

② 项目功能

③ 架构总览

④ 项目创新

① 项目背景

Why L^AT_EX?

Why L^AT_EXHelper?

② 项目功能

③ 架构总览

④ 项目创新

Why L^AT_EX?

Microsoft® Word

最普遍的文本编辑工具

相对更易上手

WYSIWYG¹

兼顾内容与格式

公式排版不够友好

需要付费获取商业许可

L^AT_EX

专业排版用语言

相对更难上手

WYTIWYG²

无需担心格式，专心于内容

尤其擅长公式排版

开源语言，可自由免费使用

¹What You See Is What You Get, 所见即所得

²What You Think Is What You Get, 所想即所得

① 项目背景

Why L^AT_EX?

Why L^AT_EXHelper?

② 项目功能

③ 架构总览

④ 项目创新

Why L^AT_EXHelper?

- T_EXWorks、T_EXStudio 等目前主流的 L^AT_EX 编译器界面不够美观
- L^AT_EX 文档的结构类似，代码可复用性强，但并不受支持，编写时存在重复性劳动
- 模板的获取依赖于在线网站³，不能方便地在本地改写
- ...

L^AT_EXHelper

- 提高在 VS Code 上编写 L^AT_EX 文件生产力的工具
- 基于 Python、bash 语言编写
- 支持 Windows、Linux 系统

³如 Overleaf:<https://www.overleaf.com>

1 项目背景

2 项目功能

```
latexhelper -h | --help
latexhelper -i | --init <template name>
latexhelper -l | --list
latexhelper -t | --table <rows> <columns> <style>
latexhelper -j | --json
latexhelper -c | --csvreader <csv_name> <style>
latexhelper -m | --md <markdown_name>
generate_pdf.sh & generate_pdf.ps1
```

3 架构总览

4 项目创新

说明

使用中可通过以下命令行程序调用各项功能⁴

```
latexhelper -h | --help
latexhelper -i | --init <template name>
latexhelper -l | --list
latexhelper -t | --table <rows> <columns> <style>
latexhelper -j | --json
latexhelper -c | --csvreader <csv_name> <style>
latexhelper -m | --md <markdown_name>
```

⁴基于 argparse 库实现

1 项目背景

2 项目功能

```
latexhelper -h | --help
latexhelper -i | --init <template name>
latexhelper -l | --list
latexhelper -t | --table <rows> <columns> <style>
latexhelper -j | --json
latexhelper -c | --csvreader <csv_name> <style>
latexhelper -m | --md <markdown_name>
generate_pdf.sh & generate_pdf.ps1
```

3 架构总览

4 项目创新

latexhelper -h | --help

功能

显示帮助信息

运行实例

```
86181@Ther > E:\P_6
~#> latexhelper
usage: latexhelper [-h] [-i FILENAME] [-l] [-t ROW COLUMN STYLE] [-c CSVNAME STYLE] [-j]

A tool for editing LaTeX

optional arguments:
  -h, --help            show this help message and exit
  -i FILENAME, --init FILENAME
                        generate a TeX template with your scheduled name
  -l, --list            show your TeX templates
  -t ROW COLUMN STYLE, --table ROW COLUMN STYLE
                        create a table in your TeX work
  -c CSVNAME STYLE, --csvreader CSVNAME STYLE
                        transform a csv file into a table in your TeX work
  -j, --json            generate a .json file for the auto-completion in VSCode
```

1 项目背景

2 项目功能

```
latexhelper -h | --help
latexhelper -i | --init <template name>
latexhelper -l | --list
latexhelper -t | --table <rows> <columns> <style>
latexhelper -j | --json
latexhelper -c | --csvreader <csv_name> <style>
latexhelper -m | --md <markdown_name>
generate_pdf.sh & generate_pdf.ps1
```

3 架构总览

4 项目创新

```
latexhelper -i | --init <template name>
```

功能

生成一个 \LaTeX 模板文件并存放于指定目录

Example. 需要一个书写 Maxwell 方程组的 \LaTeX 文档，则输入命令 `latexhelper -i maxwell.tex`，并按照提示输入文档类型与内容类型，程序将在指定位置创建此模板文件以方便未来的查找

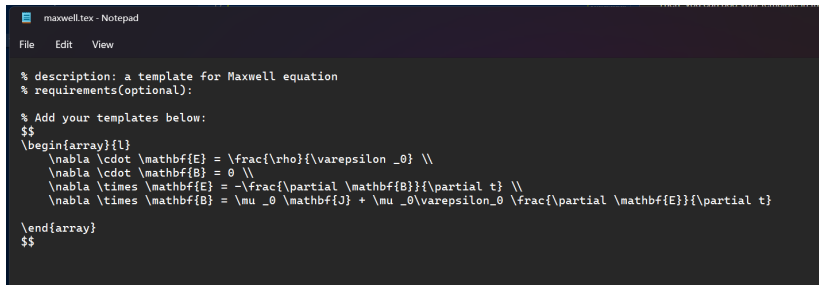
运行实例

```
~#> latexhelper -i maxwell.tex
Please input your document type:article
Please input your content type:math
Add your templates according to the format in the empty file:
█
```

```
latexhelper -i | --init <template name>
```

运行实例（续）

创建模板文件后，编辑器将自动打开，用户在其中输入文档介绍、依赖宏包与具体代码



```
maxwell.tex - Notepad
File Edit View

% description: a template for Maxwell equation
% requirements(optional):

% Add your templates below:
$$
\begin{array}{l}
\backslash nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0} \\
\backslash nabla \cdot \mathbf{B} = 0 \\
\backslash nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \\
\backslash nabla \times \mathbf{B} = \mu_0 \mathbf{J} + \mu_0 \epsilon_0 \frac{\partial \mathbf{E}}{\partial t}
\end{array}
$$
```

```
latexhelper -i | --init <template name>
```

运行实例（续）

关闭文件后，将会自动保存，同时程序将会提示具体保存路径

```
Please input your document type:article
Please input your content type:math
Add your templates according to the format in the empty file:
template saved in: C:\Users\86181\.latexhelper\LaTeX-templates\article\math\maxwell.tex
maxwell.tex
```

1 项目背景

2 项目功能

```
latexhelper -h | --help
latexhelper -i | --init <template name>
latexhelper -l | --list
latexhelper -t | --table <rows> <columns> <style>
latexhelper -j | --json
latexhelper -c | --csvreader <csv_name> <style>
latexhelper -m | --md <markdown_name>
generate_pdf.sh & generate_pdf.ps1
```

3 架构总览

4 项目创新

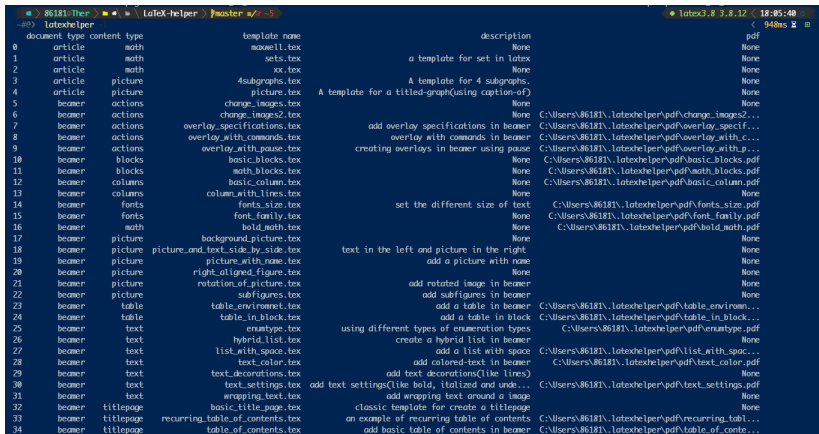

```
latexhelper -l | --list
```

功能

以表格形式打印所有本地保存的 \LaTeX 模板信息，包括文档类型、内容类型、文档介绍、预览 PDF 文件的路径（如果存在的话）

latexhelper -l | --list

运行实例



document type	content type	template name	description
0	article	math	maxwell.tex
1	article	math	sets.tex
2	article	math	xx.tex
3	article	picture	4subgraphs.tex
4	article	picture	A template for a titled-graph(using caption-of)
5	beamer	actions	change_images.tex
6	beamer	actions	change_images2.tex
7	beamer	actions	overlay_specifications.tex
8	beamer	actions	overlay_with_commands.tex
9	beamer	actions	overlay_with_pause.tex
10	beamer	blocks	basic_blocks.tex
11	beamer	blocks	math_blocks.tex
12	beamer	columns	basic_column.tex
13	beamer	columns	column_with_lines.tex
14	beamer	fonts	font_size.tex
15	beamer	fonts	font_family.tex
16	beamer	math	bold_math.tex
17	beamer	picture	background_picture.tex
18	beamer	picture	picture_and_text_side_by_side.tex
19	beamer	picture	picture_with_name.tex
20	beamer	picture	right_aligned_figure.tex
21	beamer	picture	rotation_of_picture.tex
22	beamer	table	subfigures_in_beamer
23	beamer	table	table_environment.tex
24	beamer	table	table_in_block.tex
25	beamer	text	enumtype.tex
26	beamer	text	hybrid_list.tex
27	beamer	text	list_with_space.tex
28	beamer	text	text_color.tex
29	beamer	text	text_decorations(tex
30	beamer	text	text_settings(tex
31	beamer	text	wrapping_text.tex
32	beamer	titlepage	basic_title_page.tex
33	beamer	titlepage	recurring_table_of_contents.tex
34	beamer	titlepage	table_of_contents.tex

1 项目背景

2 项目功能

```
latexhelper -h | --help
latexhelper -i | --init <template name>
latexhelper -l | --list
latexhelper -t | --table <rows> <columns> <style>
latexhelper -j | --json
latexhelper -c | --csvreader <csv_name> <style>
latexhelper -m | --md <markdown_name>
generate_pdf.sh & generate_pdf.ps1
```

3 架构总览

4 项目创新

```
latexhelper -t | --table <rows> <columns> <style>
```

功能

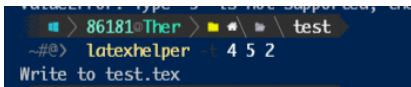
打印指定格式的表格代码块到当前文件夹下的.tex 文件中，其中 `<style>` 参数可选，分别对应不同的表格样式

表格样式

<code><style></code> 参数值	指代样式
1	"no_lines"
2	"full_lines"
3	"three_lines"
4	"bold_three_lines"

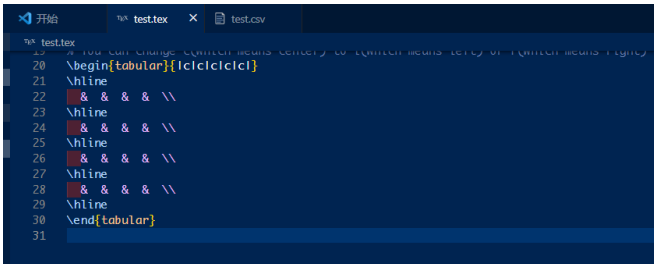
```
latexhelper -t | --table <rows> <columns> <style>
```

运行实例



```
> 86181@Ther > \ test
~#> latexhelper -t 4 5 2
Write to test.tex
```

此命令即代表生成一个"full_lines"格式的 4 行 5 列的表格代码块，如下图所示



```
20 \begin{tabular}{|c|c|c|c|c|}
21 \hline
22 & & & & \\
23 \hline
24 & & & & \\
25 \hline
26 & & & & \\
27 \hline
28 & & & & \\
29 \hline
30 \end{tabular}
```

1 项目背景

2 项目功能

```
latexhelper -h | --help
latexhelper -i | --init <template name>
latexhelper -l | --list
latexhelper -t | --table <rows> <columns> <style>
latexhelper -j | --json
latexhelper -c | --csvreader <csv_name> <style>
latexhelper -m | --md <markdown_name>
generate_pdf.sh & generate_pdf.ps1
```

3 架构总览

4 项目创新

23 / 38

latexhelper -j | --json

运行实例（续）

为了代码可读性，将文件按.json 的格式排版，如图

```
C:\Users\86181> AppData\Roaming\Code\User\snippets\{} latex.json > {} jacobian > {} body
1  {
2    "Pythoncode": {
3      "prefix": "Pythoncode",
4      "body": "\n% description:\n% requirements(optional):\n\n% Add your templates below:\n\n% Homework template for Inference and Infor
5      "description": "% requirements(optional):"
6    },
7    "jacobian": {
8      "prefix": "jacobian",
9      "body": "\n% description:\n% requirements(optional):\n\n% Add your templates below:\n\n% \\begin{bmatrix}\\n \\frac{\\partial
10     "description": "% requirements(optional):"
11   },
12   "math": {
13     "prefix": "math",
14     "body": "\n% description:\n% requirements:\n% write a brief description of your template\n% If your template needs special package
15     "description": "% requirements:"
16   },
17   "maxwell": {
18     "prefix": "maxwell",
19     "body": "% description: a template for Maxwell equation\n% requirements(optional):\n\n% Add your templates below:\n\n% \\begin{arr
20     "description": "a template for Maxwell equation"
21   },
22 }
```


latexhelper -j | --json

运行实例（续）

自动代码补全功能如图

The screenshot shows a LaTeX editor interface with a dark theme. On the left, a file explorer shows a file named 'maxwell'. The main editor area displays the content of this file, which is a LaTeX template for Maxwell's equations. The text is as follows:

```

49
50 \max
51 \en\maxwell

a template for Maxwell equation (用 ×
户代码片段)

% description: a template for Maxwell e
% requirements(optional):

% Add your templates below:
$$
\begin{array}{l}
\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0} \\
\nabla \cdot \mathbf{B} = 0 \\
\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \\
\nabla \times \mathbf{B} = \mu_0 \mathbf{j}
\end{array}
$$

```

1 项目背景

2 项目功能

```
latexhelper -h | --help
latexhelper -i | --init <template name>
latexhelper -l | --list
latexhelper -t | --table <rows> <columns> <style>
latexhelper -j | --json
latexhelper -c | --csvreader <csv_name> <style>
latexhelper -m | --md <markdown_name>
generate_pdf.sh & generate_pdf.ps1
```

3 架构总览

4 项目创新

```
latexhelper -c | --csvreader <csv_name> <style>
```

功能

将.csv 文件转换为 \LaTeX 格式的表格。

运行实例

生成的表格如图

```
13 % You can change c(which means center) to l(which means left) or r(which means right) according to your need.
14 \begin{tabular}{ccc}
15 1 & 2 & 3 \\
16 4 & 5 & 6 \\
17 7 & 8 & 9 \\
18 \end{tabular}
```

1 项目背景

2 项目功能

```
latexhelper -h | --help
latexhelper -i | --init <template name>
latexhelper -l | --list
latexhelper -t | --table <rows> <columns> <style>
latexhelper -j | --json
latexhelper -c | --csvreader <csv_name> <style>
latexhelper -m | --md <markdown_name>
generate_pdf.sh & generate_pdf.ps1
```

3 架构总览

4 项目创新

```
latexhelper -m | --md <markdown_name>
```

功能

将.md 文件转换为 \LaTeX 风格的 PDF（支持中文、格式美观）。

运行实例

生成的 PDF 如图

3. C++ 标准委员会

1

3. C++ 标准委员会

国际 C++ 标准委员会正式名称为 **ISO/IEC JTC1/SC2/WG21**，它是 C++ 发展的核心。自 1991 年成立以来，就一直如此。而从 1989 年开始到它成立之前，C++ 开发的中心则是美国国家标准学会（ANSI）的 C++ 标准委员会 [Stroustrup 1993]。C++ 没有腰缠万贯的所有者，也没有其他重要的资金提供来源，因此社区依赖于企业开发和开源项目。对于很多处于相互竞争的组织中的人来说，WG21 和各个国家的标准委员会是他们能够会面并联合解决问题的唯一场合。

委员会成员都是志愿者，也没有带薪的秘书处，虽然许多委员确实以其工作组织的代表身份出现。在每次会议上，都会有人自豪地声称就代表“自己”。也就是说，他们没有得到赞助，只代表自己。有些人换工作后，就会代表新组织，这种情况并不少见。许多人以“参加 C++ 标准委员会”作为接受新工作的条件。有人加入了委员会来学习 C++，有人则把“C++ 委员会成员”当作资格来引用（并非一定是真的）。

有些人仅参加过几次会议，不那么经常。另一方面，也有人一直参加了大多数会议，数十年没有间断。一开始时，还有现在，我们一年开三次会。在 1998 年标准之后的几年里，我们一年只开两次会。目前，除了面对面的会议，还有很多的电话会议进行补充，以及天天都有大量电子邮件。

1 项目背景

2 项目功能

```
latexhelper -h | --help
latexhelper -i | --init <template name>
latexhelper -l | --list
latexhelper -t | --table <rows> <columns> <style>
latexhelper -j | --json
latexhelper -c | --csvreader <csv_name> <style>
latexhelper -m | --md <markdown_name>
generate_pdf.sh & generate_pdf.ps1
```

3 架构总览

4 项目创新

generate_pdf.sh & generate_pdf.ps1

功能

将目前所有的代码模板片段生成为 PDF。

运行实例

```
> generate_pdf.sh
working in /home/nullptr/.latexhelper/LaTeX-templates/article
all the existing tex files: ./picture/picture.tex
./picture/4subgraphs.tex
./math/matrix.tex
./math/sets.tex
./math/maxwell.tex
→ generating picture.pdf ...
✓ generate picture.pdf successfully!
→ generating 4subgraphs.pdf ...
✓ generate 4subgraphs.pdf successfully!
→ generating matrix.pdf ...
✗ can not generate matrix.pdf!
→ generating sets.pdf ...
✗ can not generate sets.pdf!
→ generating maxwell.pdf ...
✓ generate maxwell.pdf successfully!
preview the pdf files in /home/nullptr/.latexhelper/pdf/
```


① 项目背景

② 项目功能

③ 架构总览

④ 项目创新

架构总览

- `\bin`
 - `latex_helper.py` 对接命令行程序与核心逻辑程序
- `\src`
 - `TeXinit.py` 生成一个 \LaTeX 模板文件并存放于指定目录
 - `TeXlist.py` 显示模板文件夹下的文件名与基本信息
 - `TeXtable.py` 基于用户设置或指定的.csv 文件生成表格代码块
 - `TeXjson.py` 基于已有模板文件生成.json 文件，用于代码补全
 - `TeXmarkdown.py` 基于.md 文件生成 \LaTeX 风格的.pdf 文件⁵
- `\utils`
 - `csvreader.py` 解析.csv 文件
 - `fileio.py` 通用的文件读写接口
 - `text_replacement.py` 通用的文本替换接口

⁵Typora 可实现类似功能，但不支持中文，且标题、子标题编号存在问题，我们基于 pandoc 库改进了此功能

架构总览

- `setup.py` 安装部署 L^AT_EXHelper 工具
- `generate_pdf.ps1` & `generate_ppt.ps1`
在 Windows 平台上将模板中的.tex 文件编译成.pdf 文件
- `generate_pdf.sh` & `generate_ppt.sh`
在 Linux 平台上将模板中的.tex 文件编译成.pdf 文件
- `requirements.txt` 记录需要依赖的 Python 库，供部署时使用
- `config.yaml` 记录用户的默认配置
- `README.md` & `LICENSE.txt`
README 文件与遵循的授权文件（MIT 许可）

① 项目背景

② 项目功能

③ 架构总览

④ 项目创新

项目创新

- 开源性：我们的模板库完全开源，供所有人下载，也欢迎其他人的贡献，一起为 \LaTeX 生态做贡献
- 扩展性：可以内嵌图片转公式等功能
- 完整性：我们以一个完整工程的角度去设置代码和架构
- 跨平台性：可以在 Windows 和 Linux 以及 macOS 下运行

Thanks!