

APMCM 亚太地区大学生数学建模竞赛

L^AT_EX 模板使用基础入门

<https://www.latexstudio.net> 倾情制作
本文档代码不公开, 若有需要联系网站管理员.



2020 年 11 月

Team Number:	2020XXXXXXXXXX
Problem Chosen:	A

2020 APMCM summary sheet

apmcmthesis 是 <https://www.latexstudio.net> 为 2020 年 APMCM 亚太地区大学生数学建模竞赛 <http://www.apmcm.org/> 编写的 L^AT_EX 模板, 旨在让大家专注于论文的内容写作, 而不用花费过多精力在格式的定制和调整上.

需要注意, 使用者需要有一定的 L^AT_EX 的使用经验, 至少要会使用常用宏包的一些功能, 比如参考文献, 数学公式, 图片使用, 列表环境等等. 模板已经添加了常用的宏包, 无需用户再额外添加.

模板库在 <https://github.com/latexstudio/APMCMThesis>, 新格式可以到这里更新即可.

关注微信公众号: LaTeX 工作室, 获取 L^AT_EX 学习免费电子书和免费视频.



Keywords: 数学建模 L^AT_EX Keywords3

Contents

1. 使用说明	1
1.1 编译方式	1
1.2 控制页生成	1
1.3 论文编写	2
1.4 章节标题	2
2. 字符输入与文本排版	3
2.1 保留字符	3
2.2 空格、换行与分段	3
2.3 段落对齐	3
2.4 基本列表	4
3. (La)TeX 数学排版如何入门?	4
3.1 我该读什么书?	4
3.2 我该怎么读书?	5
3.3 数学模式	5
3.4 希腊字母	6
3.5 上下标和根号	6
3.6 分数	6
3.7 运算符	7
3.8 箭头	8
3.9 注音和标注	8
3.10 分隔符	8
3.11 省略号	9
3.12 空白间距	9
3.13 矩阵	10
3.14 多行公式	10
3.14.1 长公式	10
3.14.2 公式组	11
3.14.3 分支公式	11
3.15 数学字体	11
3.16 工具帮助你	11
4. 插图指南	12
4.1 位图与矢量图	13
4.1.1 位图	13
4.1.2 矢量图	13
4.2 编译方式与图形格式	13
4.2.1 插图基本命令	13
4.2.2 图形操作参数说明	14
4.2.3 文件名与路径	14
4.2.4 <i>Figure</i> 环境	14
4.2.5 插入多图	15

4.2.6 并排摆放，各有标题	15
4.2.7 并排摆放，共享标题，各有子标题	16
4.3 绘图工具介绍	16
4.3.1 <i>MetaPost</i> 绘图	16
4.3.2 <i>PSTricks</i> 绘图	17
4.3.3 <i>TikZ</i> 绘图	17
5. 表格制作	18
5.1 简单表格	18
5.2 宽度控制	19
5.3 跨行跨列	20
5.4 长表格	21
5.5 宽表格	23
5.6 工具帮助你	23
6. 参考文献	24
7. References	25
8. 附录	25

I. 使用说明

1.1 编译方式

APMCM 亚太地区大学生数学建模采用英文写作, 因此, 推荐使用 PDFLaTeX 编译:

→ 注意

请安装最新版 TeXLive: <https://www.latexstudio.net/page/texsoftware/>, 测试版本是 TeX Live 2019, 新版本 MacTeX 均可使用.

1. 文件默认的编码为 UTF-8 对于 windows, 请选用支持 UTF-8 编码的编辑器, 如 TeXStudio, WinEdt 新版等。
2. 若是模板有什么问题, 请及时与我们取得联系, Email: latexstudio@qq.com。
3. 可以到 <https://wenda.latexstudio.net> 提问

1.2 控制页生成

只需要用对应的命令填写内容就可以自动生成到封面中去, 分别为题号, 报名号, 组员, 指导老师等等, 可生成对应的控制页内容:

```
\documentclass{apmcmthesis}
%%填写相关信息%%
\tihao{A} %选题
\baominghao{4321} %报名号
\schoolname{XXX University}
\membera{First Member} %队员1
\memberb{Second Member} %队员2
\memberc{Third Member} %队员3
\supervisor{Supervisor} %教练
\filldate{December 15, 2019}
\begin{document}
\makecontrolsheet
\end{document}
```

→ 注意

控制页是要独立生成的, 不能从 main.tex 进行生成. 可以直接在模板的 ControlSheet.tex 修改, 编译生成控制页即可.

效果图如下:

2019 APMCM Control Sheet

Each team member must sign the statement below:

(Failure to obtain signatures from each team member will result in disqualification of the entire team.)

Each of us hereby testifies that our team abided by all of the contest's rules and did not consult with anyone who was not on this team in developing the enclosed solution paper. Our submission and all rights to its publication become the property of APMCM. APMCM may use, edit, excerpt, and publish this submission for promotional use or any other purpose, including placing it online, distributing it electronically or otherwise, without compensation of any kind. APMCM reserves the right to use in materials relating to this contest, the names of the team members, their advisor(s), and their affiliations, without further notification, permission, or compensation. Team members assert that All images, figures, photographs, tables, and drawings in their submission were either created by the team or else, if reproduced from another source, the submission cites a specific reference for each at its location in the submission. All direct quotations in the submission are enclosed in quotation marks or otherwise identified as such, with a specific reference cited for each at its location in the submission.

Problem chosen is (A or B): A

Team control number is (team number): 4321

School (Please fill in the full name): XXX University

Member name (Handwriting signature): 1. First Member
2. Second Member
3. Third Member

Team adviser name: Supervisor

(Please check the above content carefully, **fill in English**, and will not be allowed to make any changes after submission. Such as fill in error, the thesis may be canceled qualification awards.)

Date: December 15, 2019

1.3 论文编写

在 `main.tex` 代码里, 可以用其他信息, 直接参考模板的样例进行编写即可。

1.4 章节标题

一篇结构化的、条理清晰文档一定是层次分明的, 通过不同的命令分割为章、节、小节。L^AT_EX 的三个标准文档类 `article`、`report` 和 `book` 提供了一系列命令, 用以划分章节、生成章节标题并自动编号, 本模板集成的是 `article` 类:

```
\section{ title }
\subsection{ title }
\subsubsection{ title }
\paragraph{ title }
\subparagraph{ title }
```

上述命令除了生成带编号的标题之外, 还向目录中添加条目, 并影响目录的内容。每个命令有两种变体:

- 带可选参数的变体: `\section[⟨short title⟩]{⟨title⟩}`
标题使用 `⟨title⟩` 参数, 在目录和页眉页脚中使用 `⟨short title⟩` 参数;
- 带星号的变体: `\section*{⟨title⟩}`
标题不带编号, 也不生成目录项和页眉页脚。

II. 字符输入与文本排版

2.1 保留字符

L^AT_EX 中有许多字符有着特殊的含义, 在你生成文档时不会直接打印。例如每个命令的第一个字符: 反斜杠。单独输入一个反斜杠在你的行文中不会有任何帮助, 甚至可能产生错误。L^AT_EX 中的保留字符有:

\$ % & _ { } \

它们的作用分别是:

- #: 自定义命令时, 用于标明参数序号。
- \$: 数学环境命令符。
- %: 注释符。在其后的该行命令都会视为注释。如果在回车前输入这个命令, 可以防止行末 L^AT_EX 插入一些奇怪的空白符。
- ^: 数学环境中的上标命令符。
- &: 表格环境中的跳列符。
- _: 数学环境中的下标命令符。
- {与}: 花括号用于标记命令的必选参数, 或者标记某一部分命令成为一个整体。

以上除了反斜杠外, 均能用前加反斜杠的形式输出。即你只需要键入:

```
\# \$ \% \^ \& \_ \{ \}
```

2.2 空格、换行与分段

在 L^AT_EX 中, 多个空格会被视为一个, 多个换行也会被视为一个。

通常的换行方法非常简单: L^AT_EX 会自动转行, 然后在每一段的末尾, 只需要输入两个回车即可完成分段。如果需要一个空白段落 (实质是一个空白行), 先输入两个回车, 再输入 `\mbox{}`, 最后再输入两个回车即可。你可以用 `\par` 来产生一个带缩进的新段。

在下划线一节的例子中已经给出了强制换行的方式, 即两个反斜: `\\`。不过这样做的缺点在于下一行段首缩进会消失, 这个命令也的确一般不用于正文换行, 新人慎用; 正文中想要换行, 请直接使用两个回车。

2.3 段落对齐

L^AT_EX 中的段落缺省两端对齐 (fully justified), 下面的三个环境可以让段落分别居左、居右或居中对齐。另有三个命令 (`\raggedright`, `\centering`, `\raggedleft`) 可以完成同样功能。

<pre>\begin{flushleft} 居左\\ 段落 \end{flushleft}</pre>	居左 段落
<pre>\begin{flushright} 居左\\ 段落 \end{flushright}</pre>	居左 段落
<pre>\begin{center} 居左\\ 段落 \end{center}</pre>	居左 段落

2.4 基本列表

L^AT_EX 有三种基本列表环境：无序列表、有序列表、描述列表。这些列表可以单独使用，也可以互相嵌套。

<pre>\begin{itemize} \item C++ \item Java \item HTML \end{itemize}</pre>	<ul style="list-style-type: none"> • C++ • Java • HTML
<pre>\begin{enumerate} \item C++ \item Java \item HTML \end{enumerate}</pre>	<ol style="list-style-type: none"> 1. C++ 2. Java 3. HTML
<pre>\begin{description} \item[C++] 编程语言 \item[Java] 编程语言 \item[HTML] 标记语言 \end{description}</pre>	<p>C++ 编程语言</p> <p>Java 编程语言</p> <p>HTML 标记语言</p>

III. (L^A)T_EX 数学公式排版

(L^A)T_EX 以其优异的数学排版能力而闻名遐迩，也是目前世界上公认排版数学公式最为优秀的系统。对于数学排版入门，需要如何做？

3.1 我该读什么书？

学习 T_EX 公式排版，网络已有的资源已经非常之丰富，且都是十分优秀的作品。细细读来，对基本入门大有裨益。切忌不了解基本知识就粗暴地使用。

这里简单介绍些电子书资源，其简介和说明均是个人浅见，欢迎指正。

- 《More Math into L^AT_EX》，这本书洋洋洒洒写了六百多页，去掉非数学排版的部分，也有近三百页的内容，从公式的基本元素的输入到复杂公式的输入，逐层深入，抽丝剥茧，

娓娓道来。这是我首推的一本入门书，这本书有配套视频，若是您有兴趣可以去下载观瞻观瞻¹。

- 《Math mode》，这本书是我的入门书。介绍相对上一本书要精细要深入一点，也仅仅是我个人观点。不管做怎么说都是吐血推荐的好书。
- 《L^AT_EX Companion》Ch8，如果说高老头 T_EX 的书是论语，那么这本书算是一本史记，全面而精妙，是所有 L^AT_EX 书中的精品，当然其数学部分—Higher Mathematics，也值得拜读一下。

这些电子书的地址: <https://www.latexstudio.net/page/tex-documents/>

3.2 我该怎么读书？

对于读书，但凡学习 (L^A)T_EX，很多时候需要我们去阅读相关电子书，有时也需要利用搜索引擎去搜索相关问题，实际从很多学习者经验来说，我们遇到的很多问题，在书中都已经给出了解答。

第一，认真研读一本书。基本上，但凡能称得起一本书，其内容都会覆盖到我们所需的基本知识。这一步很重要。因为很多用户入门时不愿读书，记住，(L^A)T_EX 不欢迎临时抱佛脚的莽撞汉。第二，亲自输入代码上机实验。建议初学者亲自输入代码，而不是拷贝电子书的代码来运行。第三，材料输入，就是自己找一个公式较多的书籍，或者就是自己的论文，对照着——输入。第四，实践中扩展知识，这是比较高级的阶段了，会调节公式以至更美观。

多多练习才是学习 (L^A)T_EX 公式排版的王道。由于这个手册并非入门的书，首先简单介绍下基础知识，粗枝大叶而不是面面俱到，仅作为我们手册的前奏。具体知识大家还是要去各个电子书去逐步学习。主要内容摘自 L^AT_EX Notes.

3.3 数学模式

L^AT_EX 的数学模式有两种形式：行间 (inline) 模式和独立 (display) 模式。前者是指在正文中插入数学内容；后者独立排列，可以有或没有编号。简单数学公式的输入方法见下表。

Table 1 简单数学公式的输入

	T _E X 命令	L ^A T _E X 命令	L ^A T _E X 环境	amsmath 环境
行间公式	<code>\$...\$</code>	<code>\(...\)</code>	<code>math</code>	
无编号独立公式	<code>\$\$...\$\$</code>	<code>\[...\]</code>	<code>displaymath</code>	<code>equation*</code>
有编号独立公式			<code>equation</code>	<code>equation</code>

行间公式和无编号独立公式都有多种输入方法，新手也许会看花了眼。懒人包老师的秘诀是用最短的：行间公式用 `$...$`，无编号独立公式用 `\[...\]`。建议不要用 `$$...$$`，因为它和 AMS-L^AT_EX 有冲突。amsmath 版本的 `equation` 环境可以嵌入次环境 (见 3.14.1 节)。

¹http://www.ctan.org/tex-archive/info/examples/Math_into_LaTeX-4

```
Einstein's $E=mc^2$
\[ E=mc^2 \]
\[ \boxed{E=mc^2} \]
\begin{equation}
E=mc^2
\end{equation}
```

Einstein's $E = mc^2$

$$E = mc^2$$

$$\boxed{E = mc^2}$$

$$E = mc^2 \quad (1)$$

3.4 希腊字母

英文字母在数学模式下可以直接输入，希腊字母则需要用下表中的命令输入，注意大写希腊字母的命令首字母也是大写。

Table 2 希腊字母

α	<code>\alpha</code>	θ	<code>\theta</code>	ϕ	<code>\phi</code>	τ	<code>\tau</code>
β	<code>\beta</code>	ϑ	<code>\vartheta</code>	π	<code>\pi</code>	υ	<code>\upsilon</code>
γ	<code>\gamma</code>	ι	<code>\iota</code>	ϖ	<code>\varpi</code>	ϕ	<code>\phi</code>
δ	<code>\delta</code>	κ	<code>\kappa</code>	ρ	<code>\rho</code>	φ	<code>\varphi</code>
ϵ	<code>\epsilon</code>	λ	<code>\lambda</code>	ϱ	<code>\varrho</code>	χ	<code>\chi</code>
ε	<code>\varepsilon</code>	μ	<code>\mu</code>	σ	<code>\sigma</code>	ψ	<code>\psi</code>
ζ	<code>\zeta</code>	ν	<code>\nu</code>	ς	<code>\varsigma</code>	ω	<code>\omega</code>
η	<code>\eta</code>	ξ	<code>\xi</code>				
Γ	<code>\Gamma</code>	Λ	<code>\Lambda</code>	Σ	<code>\Sigma</code>	Ψ	<code>\Psi</code>
Δ	<code>\Delta</code>	Ξ	<code>\Xi</code>	Υ	<code>\Upsilon</code>	Ω	<code>\Omega</code>
Θ	<code>\Theta</code>	Π	<code>\Pi</code>	Φ	<code>\Phi</code>		

3.5 上下标和根号

指数或上标用 `^` 表示，下标用 `_` 表示，根号用 `\sqrt` 表示。上下标如果多于一个字母或符号，需要用一对 `{}` 括起来。

```
\[ x_{ij}^2 \quad \sqrt{x} \quad \sqrt[3]{x} \]
```

$$x_{ij}^2 \quad \sqrt{x} \quad \sqrt[3]{x}$$

3.6 分数

分数用 `\frac` 命令表示，它会根据环境自动调整字号，比如在行间公式中小一点，在独立公式中则大一点。我们可以人工设置分数字号，比如 `\dfrac` 命令把分数的字号设置为独立公式中的大小，而 `\tfrac` 命令则把字号设为行间公式中的大小。

```
$ \frac{1}{2} \dfrac{1}{2} $
\[ \frac{1}{2}
\frac{1}{2} \]
```

$$\frac{1}{2}\frac{1}{2}$$

$$\frac{1}{2}\frac{1}{2}$$

3.7 运算符

有些小运算符例如 $+$ $-$ $*$ $/$ $=$ 等可以直接输入，另一些则需要特殊命令。更多的数学符号可参考 Pakin 的符号列表。

```
\[ \pm; \times; \div; \cdot; \cap; \cup;
\geq; \leq; \neq; \approx; \equiv ]
```

$$\pm \times \div \cdot \cap \cup \geq \leq \neq \approx \equiv$$

和、积、极限、积分等大运算符用 \sum \prod \lim \int 等命令表示，它们的上下标在行间公式中被压缩，以适应行高。我们也可以用 \limits 和 \nolimits 命令显式指定是否压缩上下标。

```
$ \sum_{i=1}^n i \quad \prod_{i=1}^n \quad \lim_{x \rightarrow 0} x^2 \quad \int_a^b x^2 dx $
\sum\limits_{i=1}^n i \quad \prod\limits_{i=1}^n \quad \lim\limits_{x \rightarrow 0} x^2 \quad \int\limits_a^b x^2 dx $
\[ \sum_{i=1}^n i \quad \prod_{i=1}^n \quad \lim_{x \rightarrow 0} x^2 \quad \int_a^b x^2 dx \]
\[ \sum\limits_{i=1}^n i \quad \prod\limits_{i=1}^n \quad \lim\limits_{x \rightarrow 0} x^2 \quad \int\limits_a^b x^2 dx \]
```

$$\sum_{i=1}^n i \quad \prod_{i=1}^n \quad \lim_{x \rightarrow 0} x^2 \quad \int_a^b x^2 dx$$

$$\sum_{i=1}^n i \quad \prod_{i=1}^n \quad \lim_{x \rightarrow 0} x^2 \quad \int_a^b x^2 dx$$

$$\sum_{i=1}^n i \quad \prod_{i=1}^n \quad \lim_{x \rightarrow 0} x^2 \quad \int_a^b x^2 dx$$

$$\sum_{i=1}^n i \quad \prod_{i=1}^n \quad \lim_{x \rightarrow 0} x^2 \quad \int_a^b x^2 dx$$

部分追求完美的同学可能会觉得积分公式末尾的积分变量 dx 改成 $\mathrm{d}x$ 比较好看；另外积分函数和积分变量之间需要拉开点距离。那么我们可以用如下代码中的方法自己定义一个积分变量命令。

```
\newcommand{\myd}{\mathrm{d}}
\[ \int x dx \quad \int x \myd x ]
```

$$\int x dx \quad \int x \mathrm{d}x$$

多重积分如果用多个 \int 来输入的话，积分号之间的距离会过宽。正确的方法是用 \iint , \iiint , \iiiiiint , \idotsint 等命令输入。从如下代码中我们可以看到两种方法的差异。

```
\[ \int\int\quad \int\int\int\quad
\int\int\int\int\quad \int\dots\int \]
\[ \iint\quad \iiint\quad \iiint\quad \idotsint \]
```

$$\begin{array}{ccccccc} \int \int & \int \int \int & \int \int \int \int & \int \dots \int \\ \iint & \iiint & \iiint & \int \dots \int \end{array}$$

3.8 箭头

Table 3 给出了一些箭头的输入方法。`\xleftarrow` 和 `\xrightarrow` 命令生成的箭头可以根据内容自动调整长度。

Table 3 箭头

\leftarrow	<code>\leftarrow</code>	\longleftarrow	<code>\longleftarrow</code>
\rightarrow	<code>\rightarrow</code>	\longrightarrow	<code>\longrightarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>
\Lleftarrow	<code>\Lleftarrow</code>	\Longleftarrow	<code>\Longleftarrow</code>
\Rrightarrow	<code>\Rrightarrow</code>	\Longrightarrow	<code>\Longrightarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\Longleftrightarrow	<code>\Longleftrightarrow</code>

```
\[ \xleftarrow{x+y+z}\quad
\xrightarrow{x<y}\{a*b*c\} \]
```

$$\xleftarrow{x+y+z} \quad \xrightarrow[a<y]{a*b*c}$$

3.9 注音和标注

Table 4 列出一些数学注音符号 (accent)，Table 5 列出一些长的标注符号。

Table 4 数学注音符号

\bar{x}	<code>\bar{x}</code>	\acute{x}	<code>\acute{x}</code>	\mathring{x}	<code>\mathring{x}</code>
\vec{x}	<code>\vec{x}</code>	\grave{x}	<code>\grave{x}</code>	\dot{x}	<code>\dot{x}</code>
\hat{x}	<code>\hat{x}</code>	\tilde{x}	<code>\tilde{x}</code>	\ddot{x}	<code>\ddot{x}</code>
\check{x}	<code>\check{x}</code>	\breve{x}	<code>\breve{x}</code>	\dddot{x}	<code>\dddot{x}</code>

3.10 分隔符

各种括号用 `() [] \{\} \langle\rangle` 等命令表示；注意花括号通常用来输入命令和环境的参数，所以在数学公式中它们前面要加 `\`。因为 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 中 `|` 和 `\|` 的应用过于随意，

Table 5 长标注符号

\overline{xxx}	<code>\overline{xxx}</code>	\overleftrightarrow{xxx}	<code>\overleftrightharrow{xxx}</code>
\underline{xxx}	<code>\underline{xxx}</code>	$\underleftrightarrow{xxx}$	<code>\underleftrightharrow{xxx}</code>
\overleftarrow{xxx}	<code>\overleftarrow{xxx}</code>	\overbrace{xxx}	<code>\overbrace{xxx}</code>
\underleftarrow{xxx}	<code>\underleftarrow{xxx}</code>	\underbrace{xxx}	<code>\underbrace{xxx}</code>
\overrightarrow{xxx}	<code>\overrightarrow{xxx}</code>	\widehat{xxx}	<code>\widehat{xxx}</code>
\underrightarrow{xxx}	<code>\underrightarrow{xxx}</code>	\widetilde{xxx}	<code>\widetilde{xxx}</code>

amsmath 宏包推荐用 `\lvert``\rvert` 和 `\lVert``\rVert` 取而代之。

我们可以在上述分隔符前面加 `\big` `\Big` `\bigg` `\Bigg` 等命令来调整其大小。LaTeX 原有的方法是在分隔符前面加 `\left` `\right` 来自动调整大小，但是效果不佳，所以 amsmath 不推荐用这种方法。

```
\[ \Bigg(\bigg(\Big(\big((x)\big)\Big)\bigg)\Bigg)\quad
\Bigg[\bigg[\Big[[[x]]\Big]\bigg]\Bigg]\quad
\Bigg\{\bigg\{\Big\{\{\{x\}\}\}\}\Bigg\}
\]
\Big\langle\bigg\langle\Big\langle\big\langle\langle x
\rangle\big\rangle\Big\rangle\rangle\big\rangle\rangle\quad
\Big\lvert\bigg\lvert\Big\lvert\big\lvert\lvert x
\rvert\big\rvert\Big\rvert\rvert\big\rvert\rvert\quad
\Big\lVert\bigg\lVert\Big\lVert\big\lVert\lVert x
\rVert\big\rVert\Big\rVert\rVert\big\rVert\rVert\]
```

$$\left(\left(\left(\left(x\right)\right)\right)\right)\quad \left[\left[\left[\left[x\right]\right]\right]\right]\quad \left\{\left\{\left\{\left\{\left\{x\right\}\right\}\right\}\right\}\right\}$$

$$\left\langle\left\langle\left\langle\left\langle x\right\rangle\right\rangle\right\rangle\right\rangle\quad \left\|\left\|\left\|\left\|\left\|x\right\|\right\|\right\|\right\|\quad \left|\left|\left|\left|\left|\left|\left|\left|\left|\left|\left|x\right|\right|\right|\right|\right|\right|\right|\right|\right|\right|\right|$$

3.11 省略号

省略号用 `\dots` `\cdots` `\vdots` `\ddots` 等命令表示。`\dots` 和 `\cdots` 的纵向位置不同；前者一般用于有下标的序列。

```
\[ x_1,x_2,\dots,x_n\quad 1,2,\cdots,n\quad
\vdots\quad \ddots\]
```

$$x_1, x_2, \dots, x_n \quad 1, 2, \dots, n \quad \vdots \quad \ddots$$

3.12 空白间距

在数学模式中，我们可以用下表中的命令生成合适的空白间距，注意负间距命令 `\!` 可以用来减小间距。

Table 6 空白间距

$\backslash,$	3/18em		$\backslash quad$	1em		
$\backslash:$	4/18em		$\backslash qqquad$	2em		
$\backslash;$	5/18em		$\backslash!$	-3/18em		

3.13 矩阵

数学模式下可以用 `array` 环境来生成矩阵，它提供了外部对齐和列对齐的控制参数。外部对齐是指整个矩阵和周围对象的纵向关系，有三种方式：居顶、居中（缺省）、居底，分别用 `t`, `c`, `b` 来表示；列对齐也有三种方式：居左、居中、居右，分别用 `l`, `c`, `r` 表示。`\\` 和 `&` 用来分隔行和列。

其语法如下：

```
\begin{array}[外部对齐]{列对齐}
  行列内容
```

```
\[ \begin{array}{ccc}
x_1 & x_2 & \dots \\
x_3 & x_4 & \dots \\
\vdots & \vdots & \ddots \\
\end{array} \]
```

$$\begin{array}{ccc} x_1 & x_2 & \dots \\ x_3 & x_4 & \dots \\ \vdots & \vdots & \ddots \end{array}$$

`amsmath` 的 `pmatrix`, `bmatrix`, `Bmatrix`, `vmatrix`, `Vmatrix` 等环境可以在矩阵两边加上各种分隔符，但是它们没有对齐方式参数。

```
\[ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \begin{Bmatrix} a & b \\ c & d \end{Bmatrix} \quad \begin{vmatrix} a & b \\ c & d \end{vmatrix} \quad \begin{Vmatrix} a & b \\ c & d \end{Vmatrix} \]
```

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \begin{Bmatrix} a & b \\ c & d \end{Bmatrix} \quad \begin{vmatrix} a & b \\ c & d \end{vmatrix} \quad \begin{Vmatrix} a & b \\ c & d \end{Vmatrix}$$

`\smallmatrix` 命令可以生成行间矩阵。

```
Marry has a little matrix $ ( \begin{smallmatrix} a & b \\ c & d \end{smallmatrix} ) $.
```

Marry has a little matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$.

3.14 多行公式

有时一个公式太长一行放不下，或几个公式需要写成一组，这时我们就要用到 `amsmath` 提供的一些多行公式环境。

3.14.1 长公式

无须对齐的长公式可以使用 `multline` 环境。需要对齐的长公式可以使用 `split` 环境，它本身不能独立使用，必须包含在其它数学环境内，因此也称作次环境。它用 `\\` 和 `&` 来分

行和设置对齐的位置。

```
\begin{multline}
x = a+b+c+{} \\\
d+e+f+g
\end{multline}
```

$$x = a + b + c + \\ d + e + f + g \quad (2)$$

```
\[ \begin{split}
x = {} & a+b+c+{} \\\
& d+e+f+g
\end{split} \]
```

$$x = a + b + c + \\ d + e + f + g$$

3.14.2 公式组

不需要对齐的公式组可以使用 `gather` 环境，需要对齐的公式组用 `align` 环境。

```
\begin{gather}
a = b+c+d \\\
x = y+z
\end{gather}
```

$$a = b + c + d \quad (3)$$

$$x = y + z \quad (4)$$

```
\begin{align}
a &= b+c+d \\\
x &= y+z
\end{align}
```

$$a = b + c + d \quad (5)$$

$$x = y + z \quad (6)$$

`multline`, `gather`, `align` 等环境都有带 * 的版本，不生成公式编号。

3.14.3 分支公式

分段函数通常用 `cases` 次环境写成分支公式。

```
\[ y=\begin{cases}
-x, \quad x \leq 0 \\\
x, \quad x > 0
\end{cases} \]
```

$$y = \begin{cases} -x, & x \leq 0 \\ x, & x > 0 \end{cases}$$

3.15 数学字体

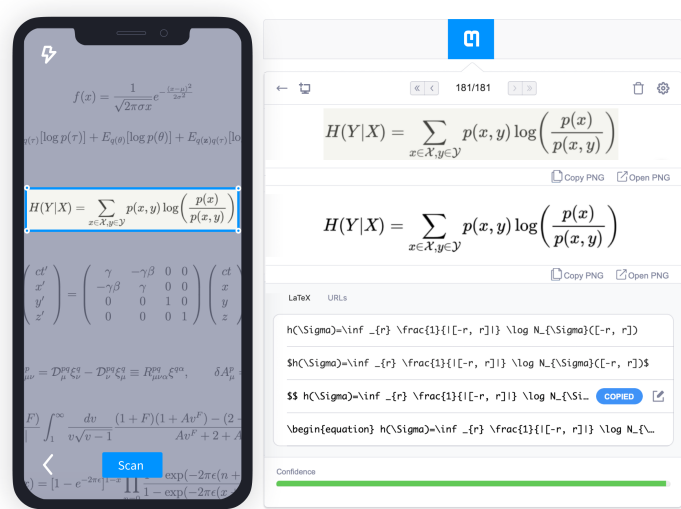
和文本模式类似,我们在数学模式下也可以选用不同的字体样式。`\mathbb` 和 `\mathfrak` 需要 `amsfonts` 宏包, `\mathscr` 需要 `mathrsfs` 宏包。

3.16 工具帮助你

Mathpix 是一款跨平台 (Windows、macOS、Linux) 的 OCR 工具,它能够识别复杂的数学公式,并将其转换为 LaTeX 语法。Mathpix 能够直接截取数学公式的图片,转换为 LaTeX 语法,非常简单方便,并且支持三大主流系统。使用方法是运行 Mathpix 后,使用快捷键 `Shift + Ctrl + M` 选取包含数学公式的屏幕区域,然后就自动生成 LaTeX 语法了。不过该软件现在开始收费了,但是费用单月不高,临时使用还是非常不错的。

Table 7 数学字体

缺省	$ABCXYZ$	<code>\mathbf</code>	$ABCXYZ$
<code>\mathrm</code>	$ABCXYZ$	<code>\mathit</code>	<i>$ABCXYZ$</i>
<code>\mathsf</code>	$ABCXYZ$	<code>\mathbb</code>	\mathbb{ABCXYZ}
<code>\mathtt</code>	$ABCXYZ$	<code>\mathfrak</code>	\mathfrak{ABCXYZ}
<code>\mathcal</code>	\mathcal{ABCXYZ}	<code>\mathscr</code>	\mathscr{ABCXYZ}



IV. 插图指南

西方学者常说：A picture is worth a thousand words。译作：一图胜千言，意思是说，一幅图形可以简单明确地表达很多错综复杂，千言万语都难以描述的信息。所以说一篇优秀的论文应该是图文并茂。

L^AT_EX 自身绘图功能比较简单，相关的各种绘图宏包：METAPOST、PSTricks、PGF 等，借助这些工具我们可以画图非常复杂的图形，但缺点是不直观，不易掌握。通常我们是用 Matlab、R、Visio 等功能强大的绘图工具先把图形画好，然后插入到 L^AT_EX 源文件中。

举个例子, 比如需要插入的图片（logo.pdf）在我们的主文件内，使用 pdfL^AT_EX 编译，图片的宽度为 0.3 * 文档宽度（\textwidth），标题为“这是一个 LOGO”，为方便引用，设置标签为“fig1:logo”。使用到的命令如下：

```
\begin{figure}[!htbp]
\centering
\includegraphics[width=0.25\textwidth]{logo.pdf}
\caption{这是一个 LOGO}\label{fig1:logo}
\end{figure}
```

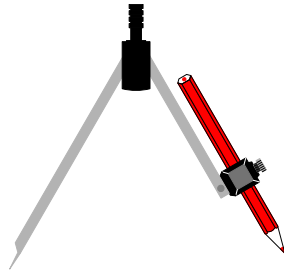



Figure 1 这是一个 LOGO

4.1 位图与矢量图

图形的存储格式很多，一般分为两大类：位图与矢量图，都是以数字形式存储，解释方法各不相同。

4.1.1 位图

位图，也称点阵图，栅格图象，像素图。

- 最小单位由像素（Pixel）构成的图，缩放会失真；
- 由像素阵列的排列来实现其显示效果的；
- 每个像素有自己的颜色信息，可操作对象为像素（HSB）。
- 每英寸所拥有的像素数目用 PPI（分辨率的单位）表示。

4.1.2 矢量图

矢量图（vector），也叫做向量图。

- 由数学公式定义的线段和曲线组成；
- 纪录了元素形状及颜色的算法；
- 可以将其任意缩放和旋转，都不会失真；
- 矢量图与分辨率无关；
- 矢量图文件尺寸一般比较小

4.2 编译方式与图形格式

我们通常使用 LaTeX、PDFLaTeX、XeLaTeX 编译源文件。各种编译方式下图形格式支持如下：

- LaTeX 直接支持 EPS、PS 图形文件，间接支持 JPEG、PNG 等格式；
- PDFLaTeX 直接支持 PNG、PDF、JPEG 格式图形文件，间接支持 EPS；
- XeLaTeX 直接支持 BMP、JPEG、PNG、EPS 和 PDF 图形格式。

4.2.1 插图基本命令

我们插图一般使用到的宏包是 `graphicx`，插图的基本命令如下：

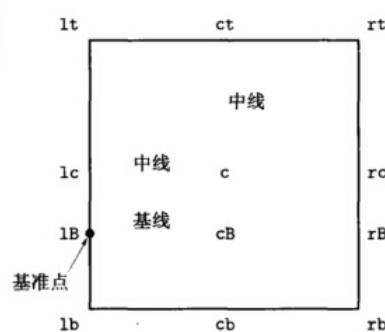
```
\usepackage{graphicx}
\includegraphics[width=0.5\textwidth]{fig.png}
```

本模板默认已经提供了该宏包, 用户不用重复添加.

4.2.2 图形操作参数说明

命令有一些参数选项可以用于缩放、旋转、裁剪等图形操作, 简要说明如下:

- `width=x,height=y`: 宽度和高度, 绝对尺寸, 可用任意长度单位;
- `scale=s`: 缩放比, 相对尺寸, 使用上面参数与缩放时, 绝对尺寸起作用;
- `keepaspectratio`: 保持长宽比;
- `angle=a`: 旋转角度;
- `origin=hv`: 旋转中心, (参考下图);
- `trim=l b r t`: 左下右上裁剪值;



4.2.3 文件名与路径

若想省略文件后缀或路径名, 可以使用下面的命令:

```
\DeclareGraphicsExtensions{.eps,.mps,.pdf,.jpg,.png}
\DeclareGraphicsRule{*}{eps}{*}{}
\graphicspath{{first_dir/}{second_dir/}{third_dir/}}
```

→ 说明如下:

- 第一行指定后缀列表让编译程序自行查找;
- 第二行指出未知后缀的都是 EPS;
- 第三行设置缺省搜索路径。

4.2.4 Figure 环境

插图通常需要占据大块空间, 所以在文字处理软件中用户经常需要调整插图的位置。figure 环境可以自动完成这样的任务; 这种自动调整位置的环境称作浮动环境 (float), 还有一个常用到的浮动环境是 table。

```
\begin{figure}[htbp]
\centering
\includegraphics[width=3cm]{gongzhonghao.jpg}
\caption{扫描有惊喜}
\label{fig:myphoto}
\end{figure}
```



Figure 2 扫描有惊喜

4.2.5 插入多图

当我们需要两幅图片并排摆放,并共享标题时,可以在 figure 环境中使用两个 `\includegraphics` 命令:

```
\begin{figure}[htbp]
\centering
\includegraphics[scale=0.5]{leftfoot.png}
\includegraphics[scale=0.5]{rightfoot.png}
\caption{向左走向右走}
\end{figure}
```



Figure 3 向左走向右走

4.2.6 并排摆放, 各有标题

如果想要两幅并排的插图各有自己的标题,可以在 figure 环境中使用两个 minipage 环境,每个里面插入一幅图。不用 minipage 的话,因为插图标题的缺省宽度是整个行宽;两幅插图就会上下排列。

```
\begin{figure}[htbp]
\centering
\begin{minipage}{5cm}
\centering
\includegraphics[scale=0.4]{leftfoot.png}
\caption{向左走}
\end{minipage}
\hspace{10pt}
\begin{minipage}{5cm}
\centering
\includegraphics[scale=0.4]{rightfoot.png}
\caption{向右走}
\end{minipage}
\end{figure}
```

效果如下:

`\caption` 命令会把环绕它的 minipage 环境“变成”figure 环境。



Figure 4 向左走



Figure 5 向右走

4.2.7 并排摆放，共享标题，各有子标题

如果想要两幅并排的图片共享一个标题，并且各有自己的子标题，可以使用 Steven D. Cochran 开发的 `subfig` 宏包。它提供的 `\subfloat` 命令，并且总图和子图可以分别有标题和引用。

```
\begin{figure}[htbp]
\centering
\subfloat[向左走]{
\label{fig:subfig_a}
\includegraphics[scale=0.4]{leftfoot.png}
}
\hspace{10pt}
\subfloat[向右走]{
\label{fig:subfig_b}
\includegraphics[scale=0.4]{rightfoot.png}
}
\caption{向左走向右走}
\label{fig:subfig}
\end{figure}
```

效果如下：

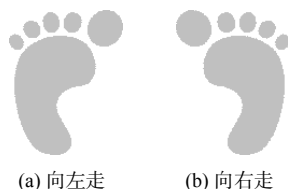


Figure 6 向左走向右走

4.3 绘图工具介绍

4.3.1 MetaPost 绘图

1980 年代末 John D. Hobby 设计了一种绘图语言以及编译器，这就是 MetaPost，MetaPost 灵感来源于 Knuth 的 MetaFont。它的输出是 EPS，支持彩色，可以在图形中做文字标注，并且插入 TeX 源码，不过也继承了 MetaFont 的一些缺点：数值变量精度较低，且绝对值不能超过 4096；只支持部分 PostScript 功能。

- 一个 MetaPost 中可以包含多个图形，注意开始与结束声明；
- 使用 `mpost` 生成的文件是 MPS（特殊 EPS）；
- 借助 EMP 宏包可以在 LaTeX 中直接用 MetaPost 绘图。

4.3.2 PSTricks 绘图

PSTricks 是一个基于 PostScript 的绘图包，有了它用户就可以直接在 LaTeX 文档中插入绘图命令。PSTricks 作者是 van Zandt, 1997 年之后由 Herbert Voß 以及 Denis Girou 等人在维护。

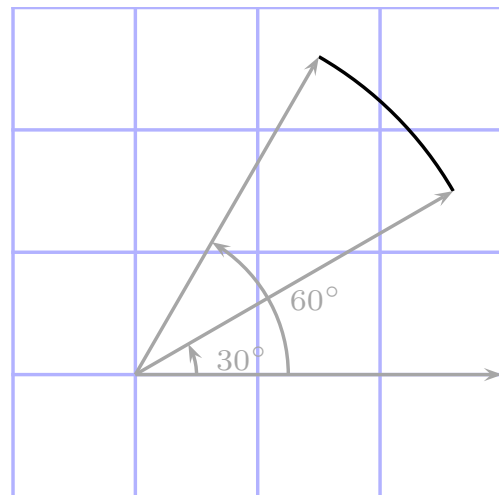
- PSTricks 中缺省长度单位是 1cm;
- 绘图命令放在 `pspicture` 环境里;
- 需要指定画布的大小，即作图左下右上的坐标;
- 注意这个矩形要能容纳所有图形对象;
- 支持 LaTeX, XeLaTeX 在线编译。

更多内容参考: [Graphics with PSTricks](#)。

实例:

```
\begin{pspicture}(-1,-1)(3,3)
\psgrid[gridcolor=blue!30,
        gridlabelcolor=black!40,
        subgriddiv=1]

\psarc(0,0){3}{30}{60}
\SpecialCoor
\psline[linecolor=mygray]{->}(0,0)(3,0)
\psline[linecolor=mygray]{->}(0,0)(3;30)
\psline[linecolor=mygray]{->}(0,0)(3;60)
\psarc[linecolor=mygray]{->}(0,0){0.5}{0}{30}
\psarc[linecolor=mygray]{->}(0,0){1.25}{0}{60}
\uput[r](0.5;15){
  \color{mygray}\scriptstyle 30^\circ}
\uput[r](1.25;30){
  \color{mygray}\scriptstyle 60^\circ}
\end{pspicture}
```



4.3.3 TikZ 绘图

PGF 和 Beamer 的作者都是 Till Tantau。Tantau 当初开发 Beamer 是为了准备 2003 年他的博士学位论文答辩，之后它在 CTAN 上流行开来。2005 年 PGF 从 Beamer 项目中分离出来，成为一个独立的宏包。而 TikZ 是 PGF 的前端，我们一般都是用 TikZ。

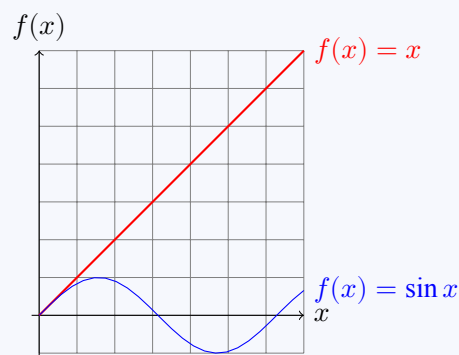
- 配合恰当算法可以得到非常精确的结果;
- 支持 PDFLaTeX 与 XeLaTeX 等;
- 编译速度快，非常舒服的体验;
- 学习难度大，绘图代码不直观更加复杂。

实例:

```

\begin{tikzpicture}[domain=0:7,scale=0.5]
%-----start graphics code -----
\draw[very thin,color=gray] (0,-1) grid (7,7);
\draw[->] (-0.2,0) -- (7,0) node[right] {$x$};
\draw[->] (0,-1.2) -- (0,7) node[above] {$f(x)$};
\draw[thick,color=red] plot[id=x]
    function{x} node[right] {$f(x)=x$};
\draw[color=blue] plot[id=sin]
    function{sin(x)} node[right] {$f(x)=\sin x$};
%-----end graphics code -----
\end{tikzpicture}

```



V. 表格制作

LaTeX 中表格的制作不同于 Word 和 Excel 的表格制作。LaTeX 是通过代码进行导致制作表格时不直观。这对初学者来说是需要逐步适应的。LaTeX 制作表格是以单元格为基本单位，对每个单元格进行一定的格式设置，然后凑成整个表格。

5.1 简单表格

tabular 环境提供了最简单的表格功能。它用 \hline 命令表示横线，| 表示竖线；用 & 来分列，用 \\ 来换行；每列可以采用居左、居中、居右等横向对齐方式，分别用 l, c, r 来表示。

```

\begin{tabular}{|l|c|r|}
\hline
操作系统 & 发行版 & 编辑器 \\
\hline
Windows & MikTeX & TeXnicCenter \\
\hline
Unix/Linux & teTeX & Kile \\
\hline
macOS & MacTeX & TeXShop \\
\hline
跨平台 & TeX Live & TeXworks \\
\hline
\end{tabular}

```

操作系统	发行版	编辑器
Windows	MikTeX	TeXnicCenter
Unix/Linux	teTeX	Kile
macOS	MacTeX	TeXShop
跨平台	TeX Live	TeXworks

在插图一章中我们介绍了一种图形浮动环境 figure；表格也有一种类似的浮动环境 table，其标题和交叉引用的用法和图形浮动环境类似。我们可以用它给表格穿件马甲，顺便把表格简化为科技文献中常用的三线表。使用 booktabs 宏包。三条横线就分别用 \toprule, \midrule, \bottomrule 等命令表示。改进后的表格如下。

```

\begin{table}[htbp]
\centering
\begin{tabular}{lll}
\toprule
操作系统 & 发行版 & 编辑器 \\
\midrule
Windows & MikTeX & TeXStudio \\
macOS & MacTeX & TeXShop \\
跨平台 & TeX Live & TeXworks \\
\bottomrule
\end{tabular}
\end{table}

```

Table 8 这是一个三线表格的例子

操作系统	发行版	编辑器
Windows	MikTeX	TeXStudio
macOS	MacTeX	TeXShop
跨平台	TeX Live	TeXworks

`tabular` 环境中的行可以采用居顶、居中、居底等纵向对齐方式，分别用 `t`，`c`，`b` 来表示，缺省的是居中对齐。列之间的分隔符也可以改用其他符号，比如用 `||` 来画双竖线。

语法：[纵向对齐]{横向对齐和分隔符}

5.2 宽度控制

有时我们需要控制某列的宽度，可以将其对齐方式参数从 `l`，`c`，`r` 改为 `p{宽度}`。这时纵向对齐方式是居顶，`t`，`c`，`b` 等参数失效。

```

\begin{tabular}{p{80pt}p{80pt}}
\toprule
操作系统 & 发行版 \\
\midrule
Windows & MikTeX \\
macOS & MacTeX \\
跨平台 & TeX Live \\
\bottomrule
\end{tabular}

```

操作系统	发行版
Windows	MikTeX
macOS	MacTeX
跨平台	TeX Live

我们可以用列前置命令 `>{}` 配合 `\centering`，`\raggedleft` 命令来把横向对齐方式改成居中或居右。列前置命令仅对紧邻其后的一列有效，其语法如下：

语法：`>{命令}`列参数

```

\begin{tabular}{p{80pt}>{\centering}p{80pt}>{\raggedleft\arraybackslash}p{80pt}}
\toprule
操作系统 & 发行版 & 编辑器 \\
\midrule
Windows & MikTeX & TeXStudio \\
macOS & MacTeX & TeXShop \\
跨平台 & TeX Live & TeXworks \\
\bottomrule
\end{tabular}

```

操作系统	发行版	编辑器
Windows	MikTeX	TeXStudio
macOS	MacTeX	TeXShop
跨平台	TeX Live	TeXworks

若控制整个宽度，可使用 `tabularx` 宏包，其语法如下，其中 `x` 参数表示某列可折行。

语法：{表格宽度}{横向对齐，分隔符，折行}

```

\begin{tabularx}{350pt}{lXlX}
\toprule
李白 & 平林漠漠烟如织，寒山一带伤心碧。暝色入高楼，有人楼上愁。  

玉阶空伫立，宿鸟归飞急。何处是归程，长亭更短亭。&
泰戈尔 & 夏天的飞鸟，飞到我的窗前唱歌，又飞去了。  

秋天的黄叶，它们没有什么可唱，只叹息一声，飞落在那里。\\
\bottomrule
\end{tabularx}

```

李白	平林漠漠烟如织，寒山一带伤心碧。暝色入高楼，有人楼上愁。玉阶空伫立，宿鸟归飞急。何处是归程，长亭更短亭。	泰戈尔	夏天的飞鸟，飞到我的窗前唱歌，又飞去了。秋天的黄叶，它们没有什么可唱，只叹息一声，飞落在那里。
----	--	-----	---

如果想把纵向对齐方式改为居中和居底，可以使用 Mittelbach 和 Carlisle 的 `array` 宏包，它提供了另两个对齐方式参数：`m{宽度}`，`b{宽度}`。

5.3 跨行跨列

有时表格某单元格需要横跨几列，可以使用 `\multicolumn` 命令，同时用 `booktabs` 宏包的 `\cmidrule` 命令来画横跨几列的横线。它们的语法如下：

语法：`\multicolumn{横跨列数}{对齐方式}{内容}`

语法：`\cmidrule{起始列-结束列}`


```

\begin{tabular}{lll}
\toprule
& \multicolumn{2}{c}{常用工具} \\
\cmidrule{2-3}
操作系统 & 发行版 & 编辑器 \\
\midrule
Windows & MikTeX & TeXStudio \\
macOS & MacTeX & TeXShop \\
跨平台 & TeX Live & TeXworks \\
\bottomrule
\end{tabular}

```

常用工具		
操作系统	发行版	编辑器
Windows	MikTeX	TeXStudio
macOS	MacTeX	TeXShop
跨平台	TeX Live	TeXworks

跨行表格可以使用 `multirow` 宏包的 `\multirow` 命令，其语法如下，
语法：`\multirow{竖跨行数}{宽度}{内容}`

```

\begin{tabular}{llc}
\toprule
操作系统 & 发行版 & 用户体验 \\
\midrule
Windows & MikTeX & \\
\multirow{3}{*}{\centering 爽} \\
Unix/Linux & TeX Live & \\
macOS & MacTeX & \\
\bottomrule
\end{tabular}

```

操作系统	发行版	用户体验
Windows	MikTeX	
Unix/Linux	TeX Live	爽
macOS	MacTeX	

5.4 长表格

有时表格太长要跨页，可以使用 Carlisle 的 `longtable` 宏包。需要做以下工作：

1. 首先用 `longtable` 环境取代 `tabular` 环境；
2. 然后在表格开始部分定义每页页首出现的通用表头，表头最后一行末尾不用 `\\` 换行，而是加一个 `\endhead` 命令；
3. 接着定义首页表头 (如果它和通用表头不同的话)，同样地最后一行用 `\endfirsthead` 命令结尾；
4. 然后是以 `\endfoot` 命令结尾的通用表尾；
5. 然后是以 `\endlastfoot` 命令结尾的末页表尾 (如果它和通用表尾不同的话)。

```
\begin{longtable}{l1}
\multicolumn{2}{r}{接上页} \\
\toprule
作者 & 作品 \\
\midrule
\endhead
\caption{长表格} \\
\toprule
作者 & 作品 \\
\midrule
\endfirsthead
\bottomrule
\multicolumn{2}{r}{接下页\ldots} \\
\endfoot
\bottomrule
\endlastfoot
白居易 & 汉皇重色思倾国，御宇多年求不得。\\
& 杨家有女初长成，养在深闺人未识。\\
& 天生丽质难自弃，一朝选在君王侧。\\
& 回眸一笑百媚生，六宫粉黛无颜色。\\
& 春寒赐浴华清池，温泉水滑洗凝脂。\\
& 侍儿扶起娇无力，始是新承恩泽时。\\
& 云鬓花颜金步摇，芙蓉帐暖度春宵。\\
& 春宵苦短日高起，从此君王不早朝。\\
& 承欢侍宴无闲暇，春从春游夜专夜。\\
& 后宫佳丽三千人，三千宠爱在一身。\\
& 金屋妆成娇侍夜，玉楼宴罢醉和春。\\
& 姊妹弟兄皆列土，可怜光彩生门户。\\
& 遂令天下父母心，不重生男重生女。\\
& 骊宫高处入青云，仙乐风飘处处闻。\\
& 缓歌慢舞凝丝竹，尽日君王看不足。\\
& 渔阳鼙鼓动地来，惊破霓裳羽衣曲。\\
& 九重城阙烟尘生，千乘万骑西南行。\\
& 翠华摇摇行复止，西出都门百余里。\\
& 六军不发无奈何，宛转蛾眉马前死。\\
& 花钿委地无人收，翠翘金雀玉搔头。\\
& 君王掩面救不得，回看血泪相和流。\\
& 黄埃散漫风萧索，云栈萦纡登剑阁。\\
& 峨眉山下少人行，旌旗无光日色薄。\\
& 蜀江水碧蜀山青，圣主朝朝暮暮情。\\
\end{longtable}
```

Table 9 长表格

作者	作品
白居易	汉皇重色思倾国，御宇多年求不得。 杨家有女初长成，养在深闺人未识。 天生丽质难自弃，一朝选在君王侧。 回眸一笑百媚生，六宫粉黛无颜色。 春寒赐浴华清池，温泉水滑洗凝脂。 侍儿扶起娇无力，始是新承恩泽时。 云鬓花颜金步摇，芙蓉帐暖度春宵。

接下页...

接上页

作者	作品
	<p>春宵苦短日高起，从此君王不早朝。 承欢侍宴无闲暇，春从春游夜专夜。 后宫佳丽三千人，三千宠爱在一身。 金屋妆成娇侍夜，玉楼宴罢醉和春。 姊妹弟兄皆列土，可怜光彩生门户。 遂令天下父母心，不重生男重生女。 骊宫高处入青云，仙乐风飘处处闻。 缓歌慢舞凝丝竹，尽日君王看不足。 渔阳鼙鼓动地来，惊破霓裳羽衣曲。 九重城阙烟尘生，千乘万骑西南行。 翠华摇摇行复止，西出都门百余里。 六军不发无奈何，宛转蛾眉马前死。 花钿委地无人收，翠翘金雀玉搔头。 君王掩面救不得，回看血泪相和流。 黄埃散漫风萧索，云栈萦纡登剑阁。 峨嵋山下少人行，旌旗无光日色薄。 蜀江水碧蜀山青，圣主朝朝暮暮情。</p>

5.5 宽表格

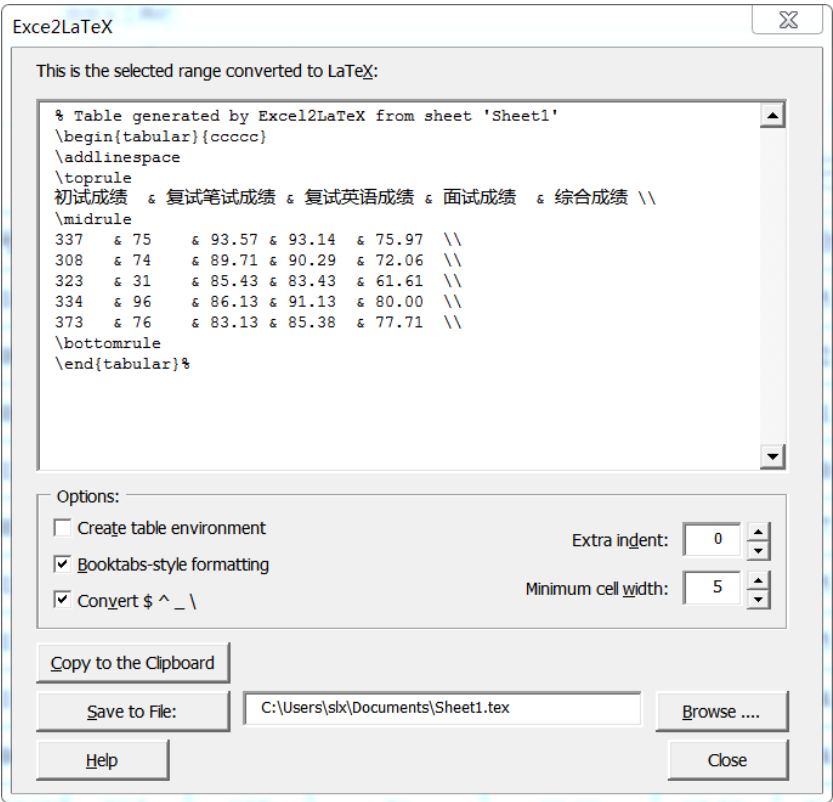
表格太宽时可以使用 Fairbairns² 等人的 `rotating` 宏包。其方法很简单,用 `sidewaystable` 环境替代 `table` 环境即可。

另外,还有一个方法: 参看这里: [LaTeX 技巧 896: LaTeX 中的浮动体: 处理超宽问题](#)

5.6 工具帮助你

在 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 里输入制作表格有点比较繁琐的,尤其是多少个列还要算好。下面这一工具会解决很多烦恼,这一工具 `Excel2LaTeX` 可以让大家在 `excle` 里输入 LaTeX 所需的表格,然后会自动生成你所需的 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 代码,然后直接 `input` 到我们的正文即可了,非常棒的工具。现在这一工具兼容了 `Excel2000-2016` (32 位和 64 位) 的版本,同时兼容 `Mac` 下的 `Excel2004,2011` 和 `2016` 版本。有这类困扰的用户可以下载试用下。

²1970 年代剑桥数学学士, 电脑硕士, 现任剑桥网管。UK FAQ 的维护者。



Excel

Item		
Animal	Description	Price (\$)
Gnat	per gram	13.65
	each	0.01
Gnu	stuffed	92.5
Emu	stuffed	33.33
Armadillo	frozen	8.99

Excel2LaTeX

Item		
Animal	Description	Price (\$)
Gnat	per gram	13.65
	each	0.01
Gnu	stuffed	92.5
Emu	stuffed	33.33
Armadillo	frozen	8.99

Formatting	Currency	Percent	\$x5	\$x^25
	\$ 3.24	40%	1.00	1.00
Bold	\$ 9.10	30%	1.50	2.25
<i>Italic</i>	\$ (8.20)	-3%	2.00	4.00
Both	\$ 3.00	100%	3.00	9.00

Formatting	Currency	Percent	x	x ²
	\$ 3.24	40%	1.00	1.00
Bold	\$ 9.10	30%	1.50	2.25
<i>Italic</i>	\$ (8.20)	-5%	2.00	4.00
Both	\$ 3.00	100%	3.00	9.00

1.0	2.0	3.0
4.0	5.0	6.0
7.0	8.0	9.0

a	b	c
1.0	2.0	3.0
4.0	5.0	6.0
7.0	8.0	9.0

下载地址在这里: <https://www.latexstudio.net/archives/6992.html>

VI. 参考文献

在学术文档中人们经常要用到参考文献 (bibliography) ，这样做既提供事实，又客观公正, 体现科学严谨。

L^AT_EX 中最原始的方法是用 `thebibliography` 环境和 `\bibitem` 命令来定义参考文献条目。在下面的环境中，第一行的参数 9 是参考文献条目编号的宽度；如果有几十个条目，可以把该参数改为 99。

```
\begin{thebibliography}{9}
\bibitem{Rowling_1997}
  Joanne K. Rowling,
  \emph{Harry Potter and the Philosopher's Stone}.
  Bloomsbury, London,
  1997.
\end{thebibliography}
```

VII. References

- [1] Joanne K. Rowling, Harry Potter and the Philosopher's Stone. Bloomsbury, London, 1997.

VIII. 附录

若是大家需要详细学习可以关注我们公众号: LaTeX 工作室, 获取电子书和免费视频, 还有大量的使用技巧.