

MCM L^AT_EX Template

Version 4.0

The usage of L^AT_EX Template for MCM

February 8, 2011

Package author:
Chinatexer



Abstract

摘要内容在美国数学建模竞赛中起到了举足轻重的作用，凡是参加比赛的同学必要认真研习英文的摘要写作。

这个模板是继承了原来几版的模板，同时对其进行了简化和增强，写成新宏包 `MCMthesis.sty`. 使得模板更简洁实用，易用。

模板支持 `latex` 和 `pdflatex` 两个方式来运行，并在 C_T_EX, MiK_T_EX, T_EXLive 几个新版系统中测试模板，因此模板能够满足不同用户需要。

最后预祝您比赛顺利。Please remember that "Decisions Are More Important Than Results, Enjoy it!"

1	模板运行	3
1.1	批处理命令	3
1.2	编辑器编译按钮	3
2	模板使用	4
2.1	模板说明	4
2.2	页眉参赛号	5
2.3	模板字体说明	5
2.4	模板宏包说明	5
3	用 L^AT_EX 写论文	6
3.1	L ^A T _E X 语句和特殊符号	6
3.2	控制空格, 分行, 分段, 分页, 章节	6
4	数学输入	7
4.1	数学公式	7
4.2	定理环境	8
5	插图	9
5.1	插图格式	9
5.2	常用插图技巧	9
5.3	怎么让图片不乱跑	10
6	表格	10
6.1	常用表格技巧	10
6.2	excel 与 L ^A T _E X 表格	10
6.3	三线表示例	10
7	枚举和列表	11
7.1	常用列表	11
7.2	距离	12
7.3	标签和计数器	13
7.4	enumerate 宏包	14
8	参考文献	14
9	其他	15
10	一些建议	16
10.1	关于分数的写法	16
10.2	标点	16
10.3	列举环境: enumerate	17
10.4	什么是 T _E X/L ^A T _E X, 我是否应该选择它	18
10.5	我该用什么编辑器?	18
10.6	我该去哪里寻找答案?	19

10.7 我应该看什么 L ^A T _E X 读物?	19
10.8 什么知识会过时? 什么不会?	19
10.9 插图格式	21
10.10 L ^A T _E X 作图	21

1 模板运行

1.1 批处理命令

新版模板支持 `latex` 和 `pdflatex` 两种运行方式。我写了两个批处理文件 `makefile.bat` 和 `makepdflatex`，双击批处理文件即可运行，查看最后运行结果。

`makefile.bat` 内容如下：

```
@echo off
latex mcmpaper
latex mcmpaper
call clean.bat
dvi2pdf mcmpaper
start mcmpaper.pdf
```

`makepdflatex.bat` 内容如下：

```
@echo off
pdflatex mcmpaper
pdflatex mcmpaper
call clean.bat

start mcmpaper.pdf
```

这里简要叙述两种运行方式之差异，

- L^AT_EX 本身只支持 eps 图片，运行得到的 pdf 文件较小，
- 而如果采用 `pdflatex` 编译，图形文件可用 pdf, png, jpg 等常见格式；如果采用 `latex` 编译，请用 eps 图形格式。

模板为了保证二者都可运行正常，即 `pdflatex` 也可以支持 eps 图片，模板已经做了这个考虑，添加了 `epstopdf` 宏包。若是 eps 图会自动转换为 pdf 图片。

因此，在比赛中，大家根据自己掌握情况来使用运行方式。`latex` 需要较多图片格式上的考量，推荐使用 `Imagemagick` 软件。

1.2 编辑器编译按钮

有用户提出 `WinEdt6.0` 界面不适应，在 `winedt5` 中编译按钮比较显著，而 `winedt6.0` 中编译按钮放在一个下拉框里，用户只需在下拉框点击所需编译方式即可。图中对于新旧界面进行了比对，并说明了两者按钮的位置。

书接上文上文所用的编译方法，按钮点击方法组合为：①+① 和 ②+②+③。

注意编译两次
是为了页眉显示
正确和文中
交叉引用显示
正常

模板在 `CTEX2.8` 和 `CTEX2.9` 环境下做了代码测试，运行正常。对于 Windows 可视化编辑器，我更推荐使用 `TeXniccenter` 编辑器，不仅开源，编译比较简单方便。随着 TeX 学习地慢慢深入，多数用户会更青睐 `emacs` 或者 `vim` 编辑器，这是后话，这里不再赘述。

另外，模板为了满足 Linux 用户的需要，我用 `TeXLive` 进行编译模板。下面是我用 `TeXLive` 编译的情况，模板兼容没有问题。

2.2 页眉参赛号

首先，如果您已经报名参加美国建模竞赛，每个小组有个对号。这个模板只需要在引入 `MCMthesis` 时候添加您们组的对应号码。例如：若是你们对号为 20111111，只需把模板的对号修改为您的对号即可。

```
\usepackage[20111111]{MCMthesis}
```

2.3 模板字体说明

这次没有沿用前面版本的字体，主要是考虑到数学字体的显示不够理想。现在所采用的字体是依照《UMAP Journal》字体来定制该模板字体。使用了如下宏包，

```
\usepackage{palatino}
```

关于字体，美国比赛没有特定要求，推荐大家使用系统默认字体即可。

2.4 模板宏包说明

模板已经预先引入了一些常用宏包，简单说明如下：

<code>xcolor</code>	颜色文本和定义颜色宏包
<code>fancyhdr, fancybox</code>	定制页眉，阴影盒子圆角盒子使用
<code>graphicx</code>	用于导入图片
<code>lastpage</code>	用于引用计算文档页数
<code>listings</code>	用于导入源文件，源代码
<code>appendix</code>	定制附录
<code>enumerate</code>	enumerate 列表定制，参看 7 节
<code>amsthm, amsfonts</code>	数学定理和数学常用字体
<code>amsmath</code>	引入 <code>amstex</code> 宏包定义了许多公式环境
<code>bm</code>	用法 <code>{\bm \alpha}</code> 加粗数学符号.
<code>amssymb, mathrsfs</code>	常见数学符号，花体数学符号
<code>latexsym</code>	常见符号
<code>subfigure</code>	用于导入子图。参看 5 节
<code>longtable</code>	产生跨页表格。参看 6 节
<code>multirow</code>	用于表格合并行。参看 6 节
<code>hhline</code>	用于表格产生双线
<code>tabularx</code>	可定制性更强的表格环境。
<code>array</code>	主要用于增加和增强列样式。
<code>slashbox</code>	表格头画斜线，参看 6 节
<code>colortbl</code>	绘制彩色表格。
<code>booktabs</code>	绘制三线格，参看 6 节

若是你使用到某些宏包，已在列表中，便可直接使用，若不在可以再导言区再次加载即可。

3 用 L^AT_EX 写论文

下面部分介绍常见的文档输入常见命令，推荐大家阅读《lnotes》来学习其基础知识。电子书文档我一并附在模板中。

3.1 L^AT_EX 语句和特殊符号

一般以\开头的都是 L^AT_EX 命令。常用的命令一般用一个右斜杠开始，以空格，其他符号或阿拉伯数字结束。有些命令需要参数，参数以大括号括起来，紧跟在命令后面。大部分 L^AT_EX 命令是相当“名符其实”的，看不懂的可以查各种参考手册。

以下这些是 L^AT_EX 中的特殊符号：

\$ % ^ & _ { } ~ \

在这些字符前加上右斜线，就可以正常输出。\\ 除外，这是一个换行命令（见下一节。）

% 代表注释，这个符号之后的本行内容都不会出现在编译结果里。

3.2 控制空格，分行，分段，分页，章节

空格 在 L^AT_EX 中，所有的文字之间的普通空格都会被自动忽略。连续的空格被认为是一个空格。行首的空格通常被忽略。按下回车产生的断行也被认为是空格。

而当为了美观需要，不得不强行插入空格时，这里有几种选择：

- ~，产生不可断行的半个空格；
- \，即\之后再加一个空格，产生 1/3 个字母 m 宽度空格；
- \hspace{1cm}，其中 1cm 可以随便替换；
- \quad，一个字母 m 的宽度；
- \qquad，两个字母 m 的宽度；

这些命令大部分不需要强行记忆，如果真的很常用的话你用几次就记住了，否则就到用的时候去查阅各种资料 /google。

分行 这里我们说的是段内分行。在 L^AT_EX 中，普通的回车会被视为空格，而分行的命令则是\\。分行与分段的区别在于下一行的段首不会缩进，并且这两行之间的间距还是普通行间距，比段间距要小。

分段 一个空行意味着一段的结束。连续的空行被当作是一个空行。也可以使用分段符：\par。

分页 使用\newpage 命令强制分页。

章节 L^AT_EX 使用章节类命令控制文章结构。章节类命令可以自动生成对应的格式，并且会自动插入目录中。

对于本科毕业论文，从大到小的章节依次为：

\chapter, \section, \subsection, \subsubsection

对章节命令的结构，可以参考任意一篇源文件，很容易看懂的。章节命令后紧跟大括号，大括号内的就是该章节的名字。

4 数学输入

4.1 数学公式

关于公式输入推荐阅读《mathmode》或《Math Into L^AT_EX》这两本足够学习 L^AT_EX 数学公式输入的知识了。另外我的博客基本把公式的输入概括了许多内容参看：

1. latex 数学宏包汇总集锦
2. 常见数学公式输入
3. 常用的数学公式字体
4. L^AT_EX 数学公式输入初级入门
5. 数学公式的单行和多行入门
6. 公式字体大小设置
7. 公式加粗方法技巧
8. 如何去掉行间公式前的空格缩进
9. 使用 align 环境输入多行公式的技巧
10. 使用 gather 输入多行公式的技巧
11. 使用 split 输入多行公式技巧
12. 使用 multiline 环境输入公式
13. 长公式的断行问题
14. 规划公式的几种输入方法
15. 如何让行间公式左对齐
16. 如何让 L^AT_EX 公式字体变小
17. 公式中的文本, amsmath 宏包
18. 如何让公式号为另一编号形式的衍生
19. 公式编号与章节关联的标准方法
20. cases 宏包编号模式示例与细节问题
21. 如何实现等号对齐

有些读者是直接使用 Mathtype 来转化出来 L^AT_EX 的公式代码，这里不提倡，因为转化中会出现不少错误和排版上的很多瑕疵。如果读者对于 L^AT_EX 较为陌生，可以作为入门的一种输入方式。如果读者较为熟悉，最好自己输入或者调整或改写 MathType 所转化生产的公式代码。参看博文：[L^AT_EX 技巧 5: MathType 转化为 L^AT_EX 公式语言](#)

L^AT_EX 的数学公式排版是它最出色最不可替代的地方。一般行内的行内数学环境用 `$.....$` 表示。数学环境内可以很容易的使用各种上下标，数学符号和希腊字母。表1是几个简单的例子。

如果你要使用更复杂的公式，建议使用 equation 环境：

`\begin{equation}...\end{equation}`

在 equation 环境内不再需要加美刀符号，而且会自动生成公式编号。一个简单的例子：

$$w(y) = (1/\rho_{max}) \int_0^{L_x} \rho(x, y) dx \quad (1)$$

代码：

Table 1: 几个数学环境的例子

显示效果	代码
x^2	<code>\$x^2\$</code>
x_1	<code>\$x_1\$</code>
π	<code>\$\$\pi\$</code>
Γ	<code>\$\$\Gamma\$</code>
$e^{\pi i} + 1 = 0$	<code>\$e^{\pi i}+1=0\$</code>
$\int_0^\infty \frac{a}{x} = \nabla^2 f$	<code>\$\$\int_0^\infty \frac{a}{x} = \nabla^2 f\$</code>

```
\begin{equation}
w(y)=(1/\rho_{\max})\int_0^{L_x}\rho(x,y)dx
\end{equation}
```

这些数学符号依然是不用强记，常用的（比如上下标，希腊字母）自然会用熟，不常用的可以点工具栏上的 Σ 键，在弹出的工具栏中找。实在不会排的可以扔到板上去求助。

4.2 定理环境

模板引入了 `amsthm` 的定理环境，我翻看了历年的论文，这个环境用得不多。关于定理样式，不建议大家自己去修改，用默认即可。

使用方法如下：

```
\begin{Theorem} \label{thm:latex}
\LaTeX 的输出是最完美的。
\end{Theorem}

先证明一个引理
\begin{Lemma} \label{thm:tex}
\TeX 文件在不同操作下的排版结果完全一致。
\end{Lemma}

\begin{proof}
这是证明。
\end{proof}
```

Theorem 4.1. *L^AT_EX* 的输出是最完美的。

先证明一个引理

Lemma 4.2. *T_EX* 文件在不同操作下的排版结果完全一致。

Proof. 这是证明。 □

5 插图

5.1 插图格式

前面提到，L^AT_EX 主要支持 EPS 格式的插图文件，而 PDFL^AT_EX 则更喜欢 PDF、PNG、JPG 格式的图形。为解决兼容性，最新版的 PDFL^AT_EX 会自动把 EPS 文件转换为 PDF 文件。因此，使用 EPS 格式的插图可能具有最广泛的通用性。

5.2 常用插图技巧

1. 在 L^AT_EX 文档中插入图片常用方法
2. L^AT_EX 插图的常用方法
3. L^AT_EX 常见图形宏包合集与示例
4. L^AT_EX 如何创建不浮动图形
5. subfigure 排版并排放置两图
6. 常见图形宏包合集与示例
7. 如何将图形与标题均置于盒子中
8. jpg 图片批量转换 eps 小工具
9. L^AT_EX 图片控制命令，位置控制
10. subfigure 竖排图形
11. L^AT_EX 如何直接插入 JPG 插图
12. L^AT_EX 如何插入段内的图片
13. 如何拷贝 PDF 和 PS 文件中的图片
14. L^AT_EX 如何定制图片表格标题字体
15. L^AT_EX 如何在图片标题里使用脚注

我们所用的编译方法支持 jpg 和 eps 等格式的图片。如果你已经有一篇 word 文档，想把其中的图全部导出，那么可以将它另存为 html 文件，这时会在同目录下找到一个文件夹，里面是所有用到的图片。

上文的一个插图的插图命令如下：

```
\begin{figure}[h]
\centering
\includegraphics[width=.65\linewidth]{t1.png}
\caption{\TeXLive 编译演示}
\end{figure}
```

对于新手，不建议手打上述命令，因为你很可能遗忘了某个参数而导致排版问题。复制粘贴，把 1.eps 换成你要的文件名，把 caption 改成图的标题，label 内的内容改成你喜欢的随便什么内容就好。width 是图片宽度，你要是觉得窄了可以弄宽点儿。label 这一命令和 ref 对应。你在这 `\label{xxx}` 了一下之后，文章内就可以用 `\ref{xxx}` 引用这个内容。引入这一功能是为了避免你以后改变图片顺序之后需要修正引用处编号的麻烦。

5.3 怎么让图片不乱跑

用惯 word 的童鞋对 L^AT_EX 最不喜欢的地方就是图片和表格乱跑。如果你使用的是我上面这一段代码的话，那么图片优先选择出现在代码位置上，在此处没有空间插图的情况下出现在下一页的页首。所以要让他不乱跑，我一般有这么几种办法：

1. 改变图片大小，即改变`width=0.5\textwidth` 这个选项。图片小了自然就好插了。
2. 改变文字结构（如插入`\newpage`），让更多的文字到后面去。

我个人觉得就易读性标准来说，出现在下一页页首也没什么不可接受的，所以一般不干预这类浮动体的位置。只要你有正确的 caption 和 label，图片出现在哪里并不是一个非常重要的问题，读者总是能方便的找到的。

6 表格

6.1 常用表格技巧

关于表格，这里不再多述，我博客已有的内容基本涵盖了绘制表格的常见问题。

1. L^AT_EX 编辑表格的常用宏包与示例
2. 精致的表格教程
3. L^AT_EX 如何绘制多斜线表格
4. L^AT_EX 如何在表格中竖排文本
5. 如何调整表格的上下高度
6. 如何生成固定宽度的表格？
7. 使用 `slashbox` 宏包绘制表格斜线
8. L^AT_EX 中的表格
9. 跨页（多页）表格制作技巧
10. L^AT_EX 绘制三线表格
11. 表格合并行的绘制技巧
12. 表格合并列的细节问题
13. L^AT_EX 中文字、表格混编的解决小窍门
14. L^AT_EX 表格绘制
15. 定义表格的线宽，加粗的表格线示例

6.2 excel 与 L^AT_EX 表格

若是大家对于表格输入比较有困难，可以借助 excel 软件来输入，参看 [Excel2LaTeX 表格处理软件图解教程](#)，插件下载地址为：[点击这里下载](#)。

6.3 三线表示例

我在示例文档中使用的这张表格
代码

Table 2: 表格示例三

一列	两列	三列	四列
1	2	3	4
2	3	4	5
3	4	5	6

```

\begin{table}[htb]
\setlength{\abovecaptionskip}{0pt}
\setlength{\belowcaptionskip}{10pt}
\label{tab:table3}
\caption{表格示例三}
\centering
\begin{tabular}{c c c c}
\toprule
一列 & 两列 & 三列 & 四列 \\
\midrule
1 & 2 & 3 & 4 \\
2 & 3 & 4 & 5 \\
3 & 4 & 5 & 6 \\
\bottomrule
\end{tabular}
\end{table}

```

在`\begin{tabular}`的`{c c c c}`，是控制有多少列，每列怎么对齐。`cccc`就是4列，每列都居中对齐。又比如`lcr`，就是一共三列，第一列左对齐，第二列居中，第三列右对齐。表格中的数字就是内容，实际使用的时候用自己的内容替换就好。代码中的`&`表示分格子，`\\`表示分行。

7 枚举和列表

7.1 常用列表

```

begin{list_type}
\item The first item
\item The second item
\item The third etc \ldots
end{list_type}

```

默认的 list_type 有 enumerate、itemize 和 description，区别主要是列项标签的不同。enumerate 是有序的列表；itemize 用圆点；description 需要 \item[label] 指定标签。各种列表环境可以互相嵌套，此时，enuemrate 的标签会自动变化以体现分级层次。

前文基础部分便是使用列表来实现的。代码如下：

```
\begin{description}
\item{空格} 在 \LaTeX ....
\begin{itemize}
\item{\verb|~|, 产生不可断行的半个空格;
.....
\end{itemize}
.....
\item{分页}
使用 \verb|\newpage| 命令强制分页。
.....
\end{description}
```

如果不想让列项第一行和标签同行，加一个 \hfill。

```
\begin{description}
\item[First] \hfill \\\
The first item
\item[Second] \hfill \\\
The second item
\item[Third] \hfill \\\
The third etc \ldots
\end{description}
```

First

The first item

Second

The second item

Third

The third etc ...

7.2 距离

在列表的内部，很容易改变一些距离

```
\begin{itemize}
\setlength{\itemsep}{1pt}
\setlength{\parskip}{0pt}
\setlength{\parsep}{0pt}
```

```
\item first item
\item second item
\end{itemize}
```

- first item
- second item

更详细的参数需要看下面 list 环境。

7.3 标签和计数器

枚举的列表计数可以通过其计数器来改变。enumerate 提供了四个计数器 enumi, enumii, enumiii, enumiv 对应不同层次的枚举。

```
\begin{enumerate}
\setcounter{enumi}{4}
\item fifth element
\end{enumerate}
```

会产生

5. fifth element

L^AT_EX 可以用其他格式打印这些计数器的值，

命令	例子
<code>\arabic</code>	1, 2, 3 ...
<code>\alph</code>	a, b, c ...
<code>\Alph</code>	A, B, C ...
<code>\roman</code>	i, ii, iii ...
<code>\Roman</code>	I, II, III ...
<code>\fnsymbol</code>	星号, 单剑号, 双剑号等

```
% 重定义第一级计数显示
\renewcommand{\theenumi}{\Roman{enumi}}
\renewcommand{\labelenumi}{\theenumi}

% 重定义第二级计数显示
\renewcommand{\theenumii}{\Alph{enumii}}
\renewcommand{\labelenumii}{(\theenumii)}
```

这样看起来有些麻烦。有时候直接写

```
\renewcommand{\labelenumi}{\Roman{enumi}}
```

但是这样有一个缺点：如果在列表内部引用这些计数器，就会按原来的格式打印。

对列举 `itemize` 而言，无须计数。标签的形式可以通过 `\item[label]` 来改变。或者通过各级 `\labelitemi`, `\labelitemii`, `\labelitemiii`, `\labelitemiv`。所以

```
\item[ $\star$ ]
```

等价于

```
\renewcommand{\labelitemi}{ $\star$ }
```

7.4 enumerate 宏包

`enumerate` 宏包为 `enumerate` 提供了更灵活的标签。可以使用 `A`, `a`, `I`, `i`, `1` 作为可选项产生 `\Alph`, `\alph`, `\Roman`, `\roman`, `\arabic` 的效果。

```
\usepackage{enumerate}
.....
\begin{enumerate}[(i)]
\item The first item
\item The second item
\item The third etc \ldots
\end{enumerate}
```

8 参考文献

用 L^AT_EX 处理文档，经常就要书写参考文献，本篇就是介绍如何在 L^AT_EX 中使用参考文献，注意这里讲的是 L^AT_EX 默认的 `thebibliography` 环境，今后大家最好用更高级的参考文献管理工具 `BibTEX`，初学者推荐大家使用 `thebibliography`。参看：[BibT_EX 使用介绍](#)。

参考文献样式，模板使用的是系统默认的参考文献形式，若是大家使用类似《UMAP Journal》参考文献样式，参看我的博客文章：[L^AT_EX 技巧 472](#) 把代码放在宏包内即可。

参考文献格式：

```
\begin{thebibliography}{编号样本}
\bbitem[记号]{引用标志} 文献条目 1
\bbitem[记号]{引用标志} 文献条目 2
.....
\end{thebibliography}
```

其中文献条目包括：作者，题目，出版社，年代，版本，页码等。引用时候要可以采用：`\cite{引用标志 1, 引用标志 2, ...}`

下面来看一个具体的例子

```

\documentclass{article}
\usepackage{CJK}
\begin{document}
\begin{CJK}{GBK}{kai}
引用一篇文章 \cite{article1} \\
引用一本书 \cite{book1}\\
等等
\begin{thebibliography}{99}
\bibitem{article1} 文章标题 作者 期刊 年代 页码
\bibitem{book1} 书标题 作者 出版社 年代
\end{thebibliography}
\end{CJK}
\end{document}

```

注意需编译两次才可正常显示引用，其中`{thebibliography}{99}`，这个数字 99 指的是参考文献的项目按照数字进行编号，并且最多为 99 个，如果你有更多的项目，把这个数字改大一点就行了。如何你想用对参考文献的项目进行其他类型的编号，你可以将上面例子中的 `thebibliography` 环境替换成

```

\begin{thebibliography}{ABCDEFGHIJK}
\bibitem[authour1]{article1} 文章标题 作者 期刊 年代 页码
\bibitem[authour2]{book1} 书标题 作者 出版社 年代
\end{thebibliography}

```

这个时候的引用标记将变成 `\bibitem` 后面括号 `[]` 里面的内容了。

9 其他

- `listings` 用于导入源文件，这部分原模板有实例，这一版本我修改了样式。较少论文会用到这一功能。
使用方法参看：
 1. [LaTeX 程序代码排版，Listings 宏包](#)
 2. [Listings 宏包定制 python 源代码输出](#)
 3. [如何利用 listing 宏包制作专业的 C++ 输出代码？](#)
- 关于附录，不再详述，参看：[LaTeX 技巧 212: LaTeX 附录标题的制作和使用技巧](#)

10 一些建议

10.1 关于分数的写法

L^AT_EX 提供宏命令 `\frac`, 用以打印分数. 为使得版面整齐, 该命令的使用应遵循以下原则:

1. 仅在分行表达式中使用,
2. 不嵌套使用,
3. 不在上下标中使用.

也就是说, 行内表达式和上下标中出现分数时一律用 a/b 表示, 如 $(x+2)/((3x^2+4)(7+y))$. 下面是居中表达式:

$$x^2 = y^{1/2} + 3.$$

多行表达式: 尽量在加、减、乘、等号前换行. 在乘号前换行时, 下一行首用 `\times`:

10.2 标点

科技文献中一般用半角标点, 请参考《中国科学》发表的论文.

如果使用全角标点, 可以使用

`\punctstyle{style}`

选择标点样式, 有效值为

quanjiao (所有标点符号占一个汉字宽度,
相邻两个标点占一个半汉字宽度)

banjiao (所有标点符号占半个汉字宽度)

hangmobanjiao (所有标点符号占一个汉字宽度, 行末行首半角)

kaiming (句号、叹号、问号占一个汉字宽度, 其他标点占半个汉字宽度)

缺省为全角式. 注意: 不论选择哪种样式, 都提供行末对齐 (margin kerning) 功能.

Theorem 10.1. *L^AT_EX 的输出是最完美的.*

先证明一个引理

Lemma 10.2. *T_EX 文件在不同操作下的排版结果完全一致.*

Proof. 这是证明. □

定理 10.1 的证明. 显然是错的. □

单个带编号的表达式

$$x = y + z \tag{2}$$

单个不带编号的表达式

$$y = x - z.$$

不带编号的多行表达式

$$\begin{aligned} x &= y + z \\ &= z - s \\ &< 3. \\ &\text{一些注释} \end{aligned}$$

带编号的多行表达式

$$x = y - z, \tag{3}$$

$$y = x + z, \tag{4}$$

$$z = y - x. \tag{4}$$

引用: 定理10.1的推论是什么呢? 方程式编号: 由 (2)(4) 式.

10.3 列举环境: enumerate

环境 `enumerate` 已经被改写, 增加了一个可选参数 [字符串], 用以控制所进。例如,

```
\begin{enumerate}
\item This is an example.
\item This is an example.
\begin{enumerate}
\item This is an example.
\item This is an example.
\end{enumerate}
\end{enumerate}
```

的输出为

1. This is an example.
2. This is an example.
 - (a) This is an example.
 - (b) This is an example.

以下内容是 milksea@bbs.ctex.org 撰写的关于 T_EX 的简单介绍。注意这不是一个入门教程，不讲 T_EX 系统的配置安装，也不讲具体的 L^AT_EX 代码。这里仅仅试图以一些只言片语来解释：进入这个门槛之前新手应该知道的注意事项，以及遇到问题以后该去如何解决问题。

10.4 什么是 T_EX/L^AT_EX，我是否应该选择它

T_EX 是最早由高德纳 (Donald Knuth) 教授创建的一门标记式宏语言，用来排版科技文章，尤其擅长处理复杂的数学公式。T_EX 同时也是处理这一语言的排版软件。L^AT_EX 是 Leslie Lamport 在 T_EX 基础上按内容 / 格式分离和模块化等思想建立的一集 T_EX 上的格式。

T_EX 本身的领域是专业排版 (即方正书版、InDesign 的领域)，但现在 TeX/LaTeX 也被广泛用于生成电子文档甚至幻灯片等，T_EX 语言的数学部分偶尔也在其他一些地方使用。但注意 T_EX 并不适用于文书处理 (MS Office 的领域，以前和现在都不是)。

选择使用 T_EX/L^AT_EX 的理由包括：

- 免费软件；
- 专业的排版效果；
- 是事实上的专业数学排版标准；
- 广泛的西文期刊接收甚或只接收 LaTeX 格式的投稿；
-

不选择使用 T_EX/L^AT_EX 的理由包括：

- 需要相当精力学习；
- 图文混合排版能力弱；
- 仅流行于数学、物理、计算机等领域；
- 中文期刊的支持较差；
-

请尽量清醒看待网上经常见到的关于 T_EX 与其他软件的优劣比较和口水战。在选择使用或离开之前，请先考虑 T_EX 的应用领域，想想它是否适合你的需要。

10.5 我该用什么编辑器？

编辑器功能有简有繁，特色不一，从简单的纯文本编辑器到繁复的 Emacs，因人而异。基本功能有语法高亮、方便编译预览就很好了，扩充功能和定制有无限的可能。初学者可以使用功能简单、使用方便的专用编辑器，如 TeXWorks、Kile、WinEdt 等，或者类似所见即所得功能的 LyX；熟悉的人可以使用定制性更强的 Notepad++、SciTE、Vim、Emacs 等。这方面的介绍很多，一开始不妨多试几种，找到最适合自己的才是最好的。

另外提醒一句，编辑器只是工作的助力，不必把它看得太重。一些编辑器有极为繁杂的功能，一些编辑器常常会引来黑客们的论战 (如 Emacs 与 Vim)。为工作，别为这些浪费太多精力，适用即可。

10.6 我该去哪里寻找答案？

- 0、绝对的新手，先读完一本入门读物，了解基本的知识。
- 1、无论如何，先读文档！绝大部分问题都是文档可以解决的。
- 2、再利用 Google 搜索，利用 (bbs.ctex.org) 版面搜索。
- 3、清楚、聪明地提出你的问题。

10.7 我应该看什么 L^AT_EX 读物？

这不是一个容易回答的问题，因为有许多选择，也同样有许多不合适的选择。这里只是选出一个比较好的答案。更多更详细的介绍可以在版面和网上寻找（注意时效）。

近两年 T_EX 的中文处理发展很快，目前没有哪本书在中文处理方面给出一个最新进展的合适综述，因而下面的介绍也不主要考虑中文处理。

1. 我可以阅读英文
 - (a) 我要迅速入门：ltxprimer.pdf (LaTeX Tutorials: A Primer, India TUG)
 - (b) 我要系统学习：A Guide to LaTeX, 4th Edition, Addison-Wesley 有机械工业出版社的影印版（《LaTeX 实用教程》）
 - (c) 我要深入学习：要读许多书和文档，TeXbook 是必读的
 - (d) 还有呢？去读你使用的每一个宏包的说明文档
 - (e) 还有许多专题文档，如讲数学公式、图形、表格、字体等
2. 我更愿意阅读中文
 - (a) 我要迅速入门：lnotes.pdf (LaTeX Notes, 1.20, Alpha Huang)
 - (b) 我要系统学习：《LaTeX2 科技排版指南》，邓建松（电子版）如果不好找，看《LaTeX 入门与提高》第二版，陈志杰等
 - (c) 我要深入学习：TeXbook0.pdf (特可爱原本，TeXbook 的中译，xianxian)
 - (d) 还有呢？英语，绝大多数 TeX 资料还是英文的

10.8 什么知识会过时？什么不会？

T_EX 是排版语言，也是广泛使用的软件，并且不断在发展中；因此，总有一些东西会很快过时。作为学习 T_EX 的人，免不了要看各种各样的书籍、电子文档和网络论坛上的只言片语，因此了解什么知识会迅速过时，什么知识不会是十分重要的。

最稳定的是关于 Primitive T_EX 和 Plain T_EX 的知识，也就是 Knuth 在他的《The TeXbook》中介绍的内容。因为 T_EX 系统开发的初衷就是稳定性，要求今天的文档到很久以后仍可以得到完全相同的结果，因此 Knuth 限定了他的 T_EX 语言和相关实现的命令、语法。这些内容许多年来就没有多少变化，在未来的一些年里也不会有什么变化。Primitive T_EX 和 Plain T_EX 的知识主要包括 T_EX 排版的基本算法和原理，盒子的原理，底层的 T_EX 命令等。其中技巧性的东西大多在宏包设计中，初学者一般不

会接触到很多；而基本原理则是常常被提到的，譬如，T_EX 把一切排版内容作为盒子 (box) 处理。

相对稳定的是关于基本 L^AT_EX 2_ε 的知识，也包括围绕 L^AT_EX 2_ε 的一些核心宏包的知识。L^AT_EX 2_ε 是自 1993 年以来的一个稳定的 L^AT_EX 版本，直到最近的一次修订 (2005 年) 都没有大的变动。L^AT_EX 的下一个计划中的版本 L^AT_EX 3 遥遥无期，在可预见的将来，L^AT_EX 2_ε 不会过时。L^AT_EX 2_ε 的知识是目前大部分 L^AT_EX 书籍的主体内容。关于 L^AT_EX 的标准文档类 (article、report、book、letter、slide 等)，关于基本数学公式的输入，文档的章节层次，表格和矩阵，图表浮动体，LR 盒子与段落盒子……这些 L^AT_EX 的核心内容都是最常用的，相对稳定的。与 L^AT_EX 2_ε 相匹配的核心宏包，如 graphics(x)、ifthen、fontenc、doc 等，也同样是相对稳定的。还有一些被非常广泛应用的宏包，如 amsmath 系列，也可以看作是相对稳定的。

简单地说，关于基本 T_EX/L^AT_EX 的语言，都是比较稳定的。与之对应，实现或者支持 T_EX/L^AT_EX 语言的软件，包括在 T_EX/L^AT_EX 基础上建立的新的宏，都不大稳定。

容易过时的是关于第三方 L^AT_EX 宏包的知识、第三方 T_EX 工具的知识，以及新兴 T_EX 相关软件的知识等。T_EX 和 L^AT_EX 语言是追求稳定的；但无论是宏包还是工具，作为不断更新软件，它们是不稳定的。容易过时的技术很多，而且现在广泛地出现在几乎所有 L^AT_EX 文档之中，因此需要特别引起注意：宏包的过时的原因可能是宏包本身的升级换代带来了新功能或不兼容，也可能是同一功能的更新更好的宏包代替了旧的宏包。前者的典型例子比如绘图宏包 PGF/TikZ，现在的 2.00 版功能十分强大，和旧的 1.1x 版相差很大，和更旧的 0.x 版本则几乎完全不同；后者的典型例子比如 caption 宏包先是被更新的 caption2 宏包代替，后来 caption 宏包更新又使得 caption2 宏包完全过时。——安装更新的发行版可以避免使用过时的宏包；认真阅读宏包自带的文档而不是搜索得到的陈旧片断可以避免采用过时的代码。

工具过时的主要原因也是升级换代和被其他工具替换。前者的典型例子是编辑器 WinEdt 在 5.5 以后的版本支持 UTF-8 编码，而旧版本不支持；后者的典型例子是中文字体安装工具从 GBKFonts 到 xGBKFonts 到 FontsGen 不断被取代。图形插入是一个在 T_EX 实现、宏包与外围工具方面都更新很快的东西。在过去，最常用的输出格式是 PS (PostScript) 格式，因此插入的图像以 EPS 为主流。使用 Dvips 为主要输出工具，外围工具有 GhostScript、bmeps 等等，相关宏包有 graphics 等，相关文档如《LaTeX2e 插图指南》。

但凡提及“L^AT_EX 只支持 EPS 图形”的，就是这个过时的时代的产物。事实上 T_EX/L^AT_EX 并不限定任何图形格式，只不过是当时的输出格式 (PS) 和工具 (Dvips) 对 EPS 情有独钟而已。后来 PDF 格式成为主流，pdfT_EX、DVIPDFM、DVIPDFMx、X_gL^AT_EX 等工具则主要支持 PDF、PNG、JPG 格式的图形，涉及一系列工具如 ImageMagick、ebb 等。

值得特别提出注意的就是，中文处理也一起是更新迅速、容易过时的部分。而且因为中文处理一直没有一个“官方”的“标准”做法，软件、工具、文档以及网上纷繁的笔记也就显得相当混乱。从八十年代开始的

CCT 系统、天元系统，到后来的 CJK 方式，到近来的 X_EL^AT_EX、LuaTeX 方式，中文处理的原理、软件、宏包、配置方式等都在不断变化中。

10.9 插图格式

前面提到，L^AT_EX 主要支持 EPS 格式的插图文件，而 PDFL^AT_EX 则更喜欢 PDF、PNG、JPG 格式的图形。为解决兼容性，最新版的 PDFL^AT_EX 会自动把 EPS 文件转换为 PDF 文件。因此，使用 EPS 格式的插图可能具有最广泛的通用性。

10.10 L^AT_EX 作图

目前已经有很多优秀的 L^AT_EX 作图宏包，如 pgf/Tikz 和 pstricks，两者都具有强大的作图能力。