

# L<sup>A</sup>T<sub>E</sub>X 入门：

## 学习排版科技类文档

桂义林\*

2014 年 11 月 8 日

### 摘要

L<sup>A</sup>T<sub>E</sub>X 是专为学术、科技写作开发的语言和程序，拥有强大的 package 支持，使用 L<sup>A</sup>T<sub>E</sub>X 写作可以避免使用 Word 带来的一些令人头疼的排版问题，如手动调整页边距、引用格式之类。本文旨在简单介绍 L<sup>A</sup>T<sub>E</sub>X 入门级别的使用方法，使不了解 L<sup>A</sup>T<sub>E</sub>X 的人阅读后能够利用 L<sup>A</sup>T<sub>E</sub>X 进行简单的科技类文档写作。本文的内容包括：L<sup>A</sup>T<sub>E</sub>X 排版命令基础、数学公式排版、表格与插图、使用 BibTeX 管理文献和 Beamer 包制作演示文稿简介。

## 1 L<sup>A</sup>T<sub>E</sub>X 的安装

### 1.1 L<sup>A</sup>T<sub>E</sub>X 发行版

一般选择直接安装 L<sup>A</sup>T<sub>E</sub>X 发行版，常见的有跨平台的 TeX Live，Windows 下的 MikTeX、CTeX，Mac OS X 下的 MacTeX。Linux 下推荐安装 TeX Live(<http://tug.org/texlive/>)，最新版本为 TeX Live 2014。

### 1.2 关于中文支持

如果使用 Windows 下的 CTeX，使用其集成的 CJK 宏包可以方便地支持中文。但在 Linux 下使用 L<sup>A</sup>T<sub>E</sub>X 需要对中文支持进行额外的配置，一般可以选择使用 CTeX 宏包与 xeCJK 宏包组合。具体配置方法不在本文的讨论范围之内，请自行搜索互联网解决。

### 1.3 线上编辑 L<sup>A</sup>T<sub>E</sub>X 文档

[www.sharelatex.com](http://www.sharelatex.com) 和 [www.writelatex.com](http://www.writelatex.com) 这两个网站提供了线上编辑 L<sup>A</sup>T<sub>E</sub>X 文档的功能，可以自动保存工作进度，并支持多人协作。

---

\*Email: [yilin.gui@gmail.com](mailto:yilin.gui@gmail.com)

## 2 L<sup>A</sup>T<sub>E</sub>X 排版命令基础

### 2.1 L<sup>A</sup>T<sub>E</sub>X 文档结构

先看一个典型的文档头声明：

```
\documentclass[a4paper]{article}
\usepackage{amssymb,amsmath}
\usepackage{graphicx}
```

上面代码的第一行表明文档类型为一般期刊文章，下面两行是宏包的引用。

L<sup>A</sup>T<sub>E</sub>X 命令的基本形式为：

```
\command[options]{arguments}
```

一篇完整文档的结构示例如下：

```
\documentclass[a4paper]{article}
\usepackage{amssymb,amsmath}
\usepackage{graphicx}

\begin{document}

\end{document}
```

文档类型包括但不限于：

- article: 一般的期刊文章
- book: 书籍
- report: 研究报告

### 2.2 字体、字号相关

L<sup>A</sup>T<sub>E</sub>X 中通过命令对字体、字号、文字颜色等进行控制，并通过预定义的命令生成希腊字母等特殊符号。

下面看一个例子：

```
{\normalsize normal size text}\\
{\huge big size text}\\
{\color{red} red text}\\
\textbf{bold text}
```

上面代码的输出结果为：

normal size text

big size text

red text

bold text

尺寸	25	20	17	14	12
Word	一号	二号	三号	四号	小四
LaTeX	<code>\Huge</code>	<code>\huge</code>	<code>\LARGE</code>	<code>\Large</code>	<code>\large</code>
尺寸	10	9	8	7	5
Word	五号	小五	六号	小六	七号
LaTeX	<code>\normalsize</code>	<code>\small</code>	<code>\footnotesize</code>	<code>\scriptsize</code>	<code>\tiny</code>

图 1:  $\text{\LaTeX}$  中的字号

<i>Greek Letters</i>							
$\alpha$	<code>\alpha</code>	$\theta$	<code>\theta</code>	$o$	<code>o</code>	$\tau$	<code>\tau</code>
$\beta$	<code>\beta</code>	$\vartheta$	<code>\vartheta</code>	$\pi$	<code>\pi</code>	$\upsilon$	<code>\upsilon</code>
$\gamma$	<code>\gamma</code>	$\iota$	<code>\iota</code>	$\varpi$	<code>\varpi</code>	$\phi$	<code>\phi</code>
$\delta$	<code>\delta</code>	$\kappa$	<code>\kappa</code>	$\rho$	<code>\rho</code>	$\varphi$	<code>\varphi</code>
$\epsilon$	<code>\epsilon</code>	$\lambda$	<code>\lambda</code>	$\varrho$	<code>\varrho</code>	$\chi$	<code>\chi</code>
$\varepsilon$	<code>\varepsilon</code>	$\mu$	<code>\mu</code>	$\sigma$	<code>\sigma</code>	$\psi$	<code>\psi</code>
$\zeta$	<code>\zeta</code>	$\nu$	<code>\nu</code>	$\varsigma$	<code>\varsigma</code>	$\omega$	<code>\omega</code>
$\eta$	<code>\eta</code>	$\xi$	<code>\xi</code>				
$\Gamma$	<code>\Gamma</code>	$\Lambda$	<code>\Lambda</code>	$\Sigma$	<code>\Sigma</code>	$\Psi$	<code>\Psi</code>
$\Delta$	<code>\Delta</code>	$\Xi$	<code>\Xi</code>	$\Upsilon$	<code>\Upsilon</code>	$\Omega$	<code>\Omega</code>
$\Theta$	<code>\Theta</code>	$\Pi$	<code>\Pi</code>	$\Phi$	<code>\Phi</code>		

图 2:  $\text{\LaTeX}$  特殊符号：希腊字母

2.3 章节环境

- 常用的章节环境有：
- part
  - chapter
  - section
  - subsection

使用章节环境会生成相应章节标题，帮助我们确定文章的逻辑结构。

用法：

```
\section{Introduction}

\section{Developing a Morphing Program}
\subsection{The Morphing Process}
\subsection{The User Interface}
```

## 2.4 其他排版命令

L<sup>A</sup>T<sub>E</sub>X 排版命令有很多，不要期望一下记住所有命令的使用方法，应该在使用中学习和积累。下面再介绍一些常用的命令。

条目列表：

```
\begin{itemize}
\item apple
\item banana
\item cherry
\end{itemize}
```

- apple
- banana
- cherry

枚举列表：

```
\begin{enumerate}
\item apple
\item banana
\item cherry
\end{enumerate}
```

1. apple
2. banana
3. cherry

使用 `\title`、`\author` 声明标题和作者，使用 `\maketitle` 生成标题和作者。

### 3 数学公式排版

$\text{T}_{\text{E}}\text{X}$  的发明就是为了解决数学公式的排版,  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  是目前公认的对数学公式排版最强的系统。

$\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  中的公式分为行间公式和行内公式, 行间公式单独成行, 行内公式就是在文字行内的公式。

行内公式示例:

```
Euler's formula: $e^{\mathrm{i}x}=\cos{x}+\mathrm{i}\sin{x}$.
```

```
Not very complex:
```

```
\begin{math}
```

```
\Gamma_{ij}^k
```

```
=\frac{1}{2}(\frac{\partial g_{il}}{\partial u^j})(\frac{\partial g_{jl}}{\partial u^i})
```

```
+\frac{\partial g_{il}}{\partial u^j}(\frac{\partial g_{jl}}{\partial u^i})
```

```
-\frac{\partial g_{il}}{\partial u^j}(\frac{\partial g_{jl}}{\partial u^i})
```

```
\end{math}
```

Euler's formula:  $e^{ix} = \cos x + i \sin x$ . Not very complex:  $\Gamma_{ij}^k = \frac{1}{2}(\frac{\partial g_{il}}{\partial u^j} + \frac{\partial g_{jl}}{\partial u^i} - \frac{\partial g_{ij}}{\partial u^l})$

行间公式示例:

```
\[ \iiint_{\Omega} f(x,y,z) dx dy dz \]
```

```
\begin{equation}
```

```
\left| \begin{array}{ccc}
```

```
1 & 6 & 9 \\
```

```
7 & 90 & f(x) \\
```

```
9 & \psi(x) & g(x)
```

```
\end{array} \right|
```

```
\end{equation}
```

$$\iiint_{\Omega} f(x,y,z) dx dy dz$$

$$\left| \begin{array}{ccc} 1 & 6 & 9 \\ 7 & 90 & f(x) \\ 9 & \psi(x) & g(x) \end{array} \right| \quad (1)$$

使用  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  生成的数学公式就是如此简洁、高效! 目前一些论坛、博客也支持  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  格式的数学公式输入, 使得互联网上对数学问题的讨论、交流更加方便。

## 4 表格与插图

### 4.1 生成表格

使用 tabular 环境可以生成表格，我们来看一个例子：

```
\begin{table}[hbt]
\caption{This table is an example}
\begin{center}
\begin{tabular}{c|cc}
First row, first column &
  First row, second column &
  First row, third column \\ \hline
Second row, first column &
  Second row, second column &
  Second row, third column \\
Third row, first column &
  Third row, second column &
  Third row, third column \\
\multicolumn{3}{c}{...}
\end{tabular}
\end{center}
\end{table}
```

表 1: This table is an example

First row, first column	First row, second column	First row, third column
Second row, first column	Second row, second column	Second row, third column
Third row, first column	Third row, second column	Third row, third column
...		

## 4.2 插入图形

可以使用 `graphicx` 宏包支持图形的插入，在文档头部加入 `\usepackage{graphicx}`，在需要插图的地方使用 `\includegraphics` 命令。

需要注意  $\text{\LaTeX}$  本身仅支持 `eps`（矢量图形格式）和 `pdf` 两种格式的图片（使用  $\text{\LaTeX}$  命令编译），使用 `PDFLaTeX` 编译可以支持 `png` 和 `jpg` 格式的图片。一般来说正式的论文中是不允许使用 `jpg` 或 `png` 格式的图片的。

例子：

```
\begin{figure}[hbt]
\begin{center}
\includegraphics[scale=0.3]{TUGlog.pdf}
\end{center}
\caption{TeX User Group Logo}
\end{figure}
```



图 3: TeX User Group Logo

值得一提的是， $\text{\LaTeX}$  本身拥有绘图工具包（`pgfplots`），因此不借助第三方工具仅仅靠  $\text{\LaTeX}$  生成高质量的论文插图也是可行的！

下面看一个 `pgfplots` 绘图的例子：

```
\begin{figure}[hbt]
\begin{center}
\begin{tikzpicture}
\begin{axis}
\addplot [dash pattern=on 10 off 5 on 100 off 5,
domain=0:10, samples=100, very thick, blue] {sin(deg(x))};
\end{axis}
\end{tikzpicture}
\end{center}
\caption{This is generated by pgfplots!}
\end{figure}
```

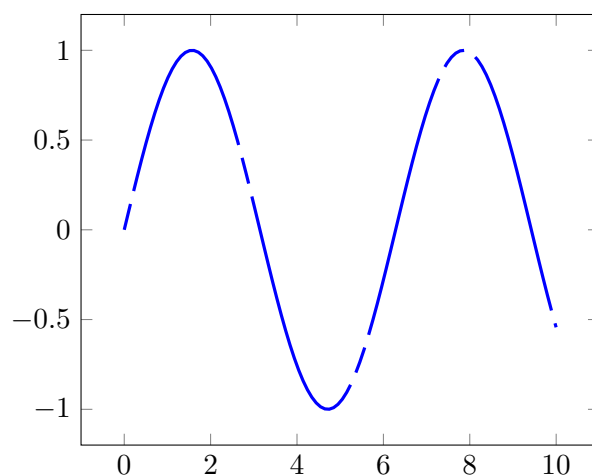


图 4: This is generated by pgfplots!

## 5 使用 BibTeX 管理文献

BibTeX 是一个使用数据库的方式来管理参考文献程序, 用于协调 LaTeX 的参考文献处理。BibTeX 文件的后缀名为.bib, 在你的整个研究生涯中, 可以只维护这样一个.bib 文件作为你的文献数据库, 每一篇文献由一个唯一的 ID 标识, 当你希望引用相关文献时, 使用 `\cite{}` 命令即可进行引用, 并可以在你的文章末尾自动生成该篇文章引用的参考文献列表。

.bib 文件中存放参考文献的条目形如:

```
@inproceedings{perez2003poisson,
  title={Poisson image editing},
  author={P{\'e}rez, Patrick and Gangnet, Michel and Blake, Andrew},
  booktitle={ACM Transactions on Graphics (TOG)},
  volume={22},
  number={3},
  pages={313--318},
  year={2003},
  organization={ACM}
}
```

为了在 LaTeX 中使用 BibTeX, 必须做下面三件事情:

1. 设置参考文献类型:

```
\bibliographystyle{plain}
```



2. 标记引用，当你在文档中使用引用时，插入  $\text{\LaTeX}$  命令：

$$\text{\cite{paper-name}}$$

3. 生成参考文献列表，在文档结束前输入：

$$\text{\bibliography{bibfile}}$$

通过 Google Scholar 可以轻松获取文献的 BibTeX 信息，如图 5 所示。

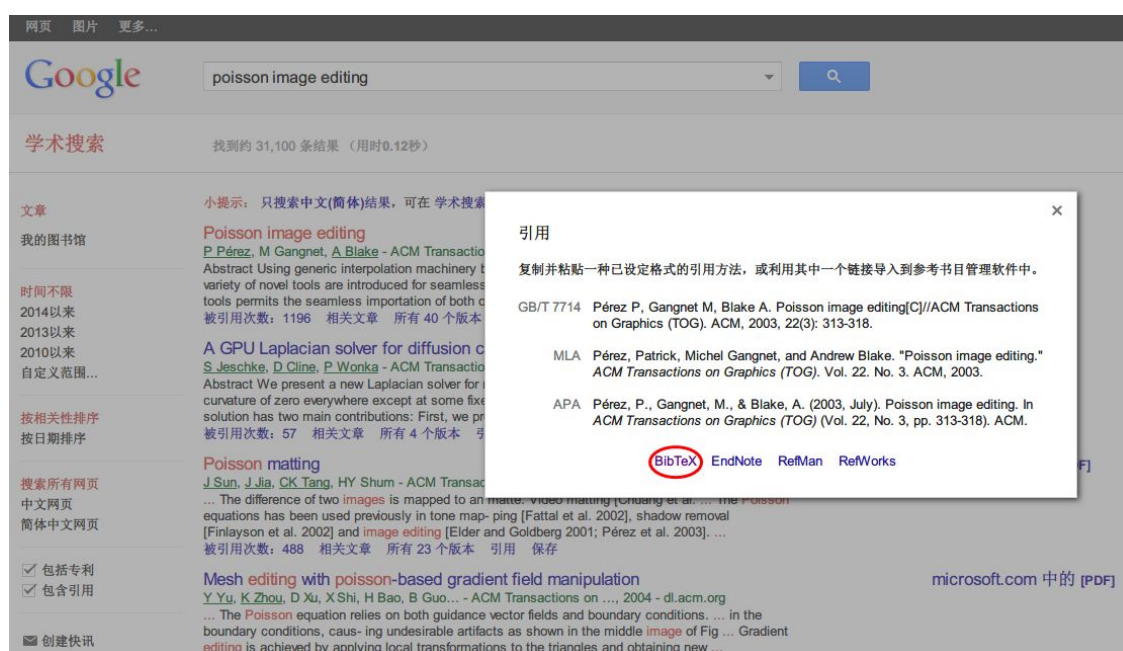


图 5: Google Scholar 获取 BibTeX 信息

## 6 使用 Beamer 包制作演示文稿

Slides，即演示文稿、幻灯片的意思，用于帮助演讲者向听众传达文字、图片甚至动画和声音等信息。大部分人只接触过微软的 Powerpoint，因此常用 ppt 代指 slides，这是一种不专业的说法，Powerpoint 只是一种可见即可得的 slides 实现方式。创建 slides 的方式有很多，除了使用 Powerpoint，还有效果非常炫的 Prezi，R 语言的 Slidify 工具可以创建支持嵌入 R 程序的 slides，你也可以用 Javascript+HTML5 自己定制基于网页的 slides。这里要介绍的是  $\text{\LaTeX}$  下的 slides 实现方案。

$\text{\LaTeX}$  上制作演示文稿 (slides) 的宏包非常多，有 pdfscreen、prosper、context 和 beamer 等等。目前使用的比较多的宏包是 beamer，因为它的语法跟标准  $\text{\LaTeX}$  几乎一样，通过 latex 和

pdf<sub>l</sub>atex 命令均可以编译, 而且生成的 slides 简洁、美观, 对于科技、学术类的演示, 其效果已经足够。

使用 beamer 创建 slides 的一般步骤为:

- (1) 将 L<sup>A</sup>T<sub>E</sub>X 的文档类型从 article 改为 beamer
- (2) 使用 section 和 subsection 命令组织逻辑结构
- (3) 使用 frame 命令添加独立的 slide
- (4) 使用 pdf<sub>l</sub>atex 命令编译两次 tex 文件

下面看一个使用 beamer 的例子:

```
\documentclass{beamer}
\usetheme{Warsaw}
\title{Example Presentation Created with the Beamer Package}
\author{Till Tantau}
\date{\today}

\begin{document}
\frame{\titlepage}

\section*{Outline}
\frame{\tableofcontents}

\section{Introduction}
\subsection{Overview of the Beamer Class}
\frame {
  \frametitle{\subsecname}
  \begin{itemize}
    \item<1-> Normal LaTeX class.
    \item<2-> Easy overlays.
    \item<3-> No external programs needed.
  \end{itemize}
}
\end{document}
```

使用 Beamer 创建 slides 的好处是实现了逻辑和内容的独立, 使用 section、subsection 控制逻辑结构, 使用 frame 创建内容。同时实现了内容和显示效果的独立, 使用 theme 可以在不影响内容的情况下轻松的改变显示风格。

## 7 进一步学习建议

- 找一本参考资料，以备查阅，在使用中学习，不要陷入每个命令的细节中
- 修改现成的模板，积累模板，打造自己的模板
- Tons of information on the web
- [tex.stackexchange.com](http://tex.stackexchange.com)

## 参考资料

- [1] CTeX 论坛翻译，一份不太简短的 LATEX 介绍
- [2] 胡贤良，浙江大学数学系《数学软件》课程讲义
- [3] 金秉文，如何在论文中画出漂亮的插图，<http://www.zhihu.com/question/21664179>