

L^AT_EX的简单应用*

王广民[†]

武汉大学系统工程研究所

December 4, 2006

摘 要

本文目的在于简单介绍L^AT_EX，对于L^AT_EX的重要命令有所了解，基本上能够使用L^AT_EX。

1 L^AT_EX的简单介绍

首先打开WinEdt，在新建的文档中输入源程序

```
\documentclass{article}
\begin{document}
\end{document}
```

保存为扩展名为tex的文件。在工具栏（如下图所示）中，按钮L^AT_EX用来编译此源文件，编译成功



后，我们可以点按钮DVI来看DVI格式的编译结果。但是此例子是空的，仅仅有L^AT_EX所必须具有的语句。下面我们再介绍工具栏中几个有用的按钮。Dvi2pdf和dvi2pdf分别把dvi 文件转化为pdf文件和ps，而ps2pdf是把ps文件转化为pdf文件。 Σ 和 \tilde{E} 用来打开或隐藏数学字符集和扩充字符集。

2 L^AT_EX字符

2.1 文字字符集

文字字符是指不需要进入数学状态就能排印出来的字符，它包括两大类：键盘字符和扩充字符。键盘字符包括键盘上除\$, &, ~, \, %, {, }, #以外的所有字符；扩充字符是指L^AT_EX定义的字符，如重音字符、希腊字符、特殊字符以及上述的例外键盘字符等。

*这是一个关于L^AT_EX的简单讲义（版本1.1）

[†]Email:wgm97@163.com

2.2 数学字符集

数学字符集是指要进入数学状态才能排印出来的字符，它主要用于数学公式的排印。

2.3 特殊字符与L^AT_EX度量单位

一、特殊字符及意义

特殊字符是用作控制符。它们的意义如下：

1、\$字符

此符号表示进入或退出数学状态。

2、&字符

此符号用作制表(tabular)和阵列(array)环境中的列分隔符。

3、~字符

此符号表用于连接两个不允许分开的字符或字，并产生一个空格。例如人名与姓是不允许分开排在两列上的，但有时恰好在人名与姓之间换行，为了避免这种情况的发生，通常使用该符号把人名与姓连接起来，告诉L^AT_EX被连接的两个字是不可分的。当遇到恰好要被该符号连接的两字间换行时，L^AT_EX就会将两字一起换到下一行，而适当调节上一行的字间距，使行充满。

4、^字符

此字符为上角标控制符。

5、_字符

此字符为下角标控制符。

6、\字符

此字符为L^AT_EX命令引导符，所有L^AT_EX命令均以\字符开始。

7、{、}字符

此字符为L^AT_EX命令作用域的起始符和结束符。

8、%字符

此字符为注释标志符，跟随其后的文字为注释内容。

二、L^AT_EX度量单位

L^AT_EX的常用尺寸度量单位有英寸(inch)、厘米(cm)、毫米(mm)和点(pt)等，点定义为1/72.72英寸。

3 L^AT_EX命令集

一、文献的形式命令

此命令位于源文件的第一行，它的格式为：`\documentclass[options]{style}`

options选项用于选择文献的字号、版式和数学公式的编号及对齐方式等。options的可选参数如下：

$$options = \left\{ \begin{array}{ll} 11pt, 12pt, & \text{(字号选择)} \\ twoside, twocolumn, titlepage, proc, & \text{(版式选择)} \\ leqno, fleqn, & \text{(公式及其编号对齐方式选择)} \end{array} \right.$$

style参数用于文献形式选择，L^AT_EX包括四种标准文献形式，即文章(article)、报告(report)、书籍(book)、信件(letter)和幻灯片(slide)。

这里我们都选择默认的选项，即字号为10pt，版式为单列单面式，数学公式版式为居中，编号在公式右边并右对齐。文献形式为article。

二、字体和字号命令

字体命令用于定义字体，在 \LaTeX 中用的字体有：罗马字体（ \rm ）、黑体（ \bf ）、意大利字体（ \it ）、小号大写字体（ \sc ）、斜体（ \sl ）、打字机字体（ \tt ）和强调型字体（ \em ）。其中强调型字体不是一种新的的字体，而仍是上述六中之一。 \LaTeX 规定如果当前字体为罗马字体，则强调型为意大利字体。如过当前字体为非罗马字体，则强调型字体为罗马字体。例如： \sl 这是斜体字，结果为：这是斜体字。

字号命令用于定义字符的大小。字号命令有： \tiny 、 \scriptsize 、 \footnotesize 、 \small 、 \normalsize 、 \large 、 \Large 、 \LARGE 、 \huge 、 \Huge 。

字号命令的引用格式为：{命令 内容}。例如：这是小写字号

三、正文版式命令

正文版式命令用于设置正文的各种参数，这些命令均位于序言区。

1. 正文宽度命令 \textwidth 和高度命令 \textheight 此命令定义正文版面的宽度和高度，引用格式为： $\text{\textwidth}(\text{\textheight})$ size
2. 段首退格命令 \indent 此命令要求每一新段的首行均要退格， \LaTeX 默认退格数为两字符，约20pt
3. 非退格命令 \nonindent 此命令的功能与 \indent 命令刚好相反，取消首行的退格。
4. 段首行退格设置命令 \parindent 此命令设置段首行后退距离，，是一个比 \indent 命令更一般的命令。 \indent 命令的后退距离是默认的，而该命令的后退距离是可设置的，命令格式为： \parindent size.size即为段首行后退距离，其默认值为20pt.
5. 段间距命令 \parskip 和行间距命令 \baselineskip 这两个命令分别设置段之间及行之间的距离。在本文中我们采用默认值。

四、文献头命令

任何一篇文献都有一个头，其中包括文献标题、作者、地址、时间， \LaTeX 提供了自动排列这些信息的命令。我们就以该文的源文件来说明文献头命令的用法。下面是本文文献头的源文件：

```
\documentclass[10pt]{article}
\textwidth=160mm \textheight = 225mm
\parindent=20pt
\parskip=0pt \baselineskip=12pt
\oddsidemargin=0pt \evensidemargin=0pt
\pagestyle{plain}
\usepackage{CJK}
\usepackage{\LaTeX sym}
\usepackage{graphicx}
\begin{CJK*}{GBK}{song}
\title{\LaTeX 的简单应用\thanks{这是一个关于\LaTeX 的简单讲义}}
\author{王广民\thanks{Email:wgm97@163.com}}\ 武汉大学系统工程研究所}
\begin{document}
\maketitle
\begin{abstract}
本文目的在于简单介绍\LaTeX，对于\LaTeX
```

的重要命令有所了解，基本上能够使用 \LaTeX 。

```
\end{abstract}
\section{\LaTeX 字符}
\subsection{文字字符集}
文章内容
\end{CJK*}
\end{document}
```

文献头命令包括：标题命令（ $\backslash title$ ）、作者命令（ $\backslash author$ ）、年月日命令（ $\backslash date$ ）、命令头命令（ $\backslash maketitle$ ）和注释命令（ $\backslash thanks$ ）。命令头命令是产生文献头。而且文献头可以包括摘要。它们的引用格式可以看上面的源文件。如果没有作者则不能不用作者命令，而应该用作 $\backslash author\{ \}$ ，而对于年与日命令，如果没有该命令则由系统自动生成时间，否则只能像作者命令一样用作 $\backslash date\{ \}$ 。

五、章节命令和页注命令

1. 节命令 $\backslash section$ 该命令适合于除letter以外的另外三种形式文献，用于产生节标题，其格式为： $\backslash section\{节标题\}$ 。节标题自动编号，在article中，编号为1,2,...，若节号不需编号，可输入 $\backslash section*\{节标题\}$ 。节标题单独占一行，并与上下文之间留有适当的间隔。
2. 子节命令 $\backslash subsection$ 和段命令 $\backslash paragraph$ 的用法和节命令相同。
3. 部分 $\backslash part$ 和章命令 $\backslash chapter$ 这两个命令都主要用于book和report形式文献。命令格式为：命令 $\{部分或者章标题\}$ 。排版结果单独占一页，并且自动编号。例如： $\backslash chapter\{基础知识\}$ ，排版结果为：**Chapter 1 基础知识**。如果不需要自动编号，可以输入 $\backslash chapter*\{基础知识\}$ 。
4. 页注命令 页注命令用于控制头注和脚注的有无及排版方式，页号的排印以及控制边注的有关参数。这里只说明页式命令 $\backslash pagestyle$ 。该命令控制头注于脚注的有无及排印方式，页码的排版方式，其格式为： $\backslash pagestyle\{options\}$ 。
参数options为选项，其可选参数及其意义如下：
plain:空头注，页码写在脚注区，这是 \LaTeX 的默认选项。
empty:空头注，空脚注。
headings:头注格式由文献形式确定。
myheadings:头注内容自定义
这里一般使用plain或者empty。

六、数学模式命令

数学模式命令是指在数学模式下才有效的命令，下面描述的命令均要在数学模式中使用。

1. 下横省略号命令 $\backslash ldots$ 此命令产生省略号：...，符号居于行的下部。
2. 增字间距命令 $\backslash !$ 、 $\backslash ,$ 、 $\backslash :$ 、 $\backslash ;$ 、 $\backslash over$ \LaTeX 自动处理的字间距一般为一个字符，若要特别加大某处的字间距，可用这五种命令。这五个命令产生的字间距均略大于标准字间距。将这些命令插入需要加大字间距的地方，该处即产生较大的字间距。上面五个命令产生的字间距依次增大，其中 $\backslash ,$ 命令也可用于文字状态。
3. 下画线命令 $\backslash underline$ 为了强调，除了使用不同的字体外，还可使用下画线，利用此命令即可在强调内容下部产生下画线，命令格式为： $\backslash underline\{强调内容\}$ 。此命令也可用在非数学状态。

注意后者在`\hspace`、`\hskip\hfill`、`\vspace`、`\vskip\vfill`等命令中的使用，此命令产生的空格可作为这些命令中的内容。例如：

`\hspace{2in}`这是一个例子。它的排版结果为：

这是一个例子。它的排版结果并不后退2in，而

`\mbox{}\hspace{2in}`这是一个例子。它的排版结果为：

这是一个例子。它的排版结果确是后退了2in。

本节描述了 \LaTeX 系统的常用基本命令，并给出了一些简单的应用实例。综合应用这些命令排版复杂的数学公式、表格等，可参阅第五节的综合实例。对于一些不常用的特殊命令，如定义宏命令、参考文献的交叉引用，图表的交叉应用和公式的交叉引用参看第六节。对于单列和双列的互相转换，作书的目录、索引，定义新的环境、命令和定理以及等，在本文中并没有进行说明。

4 环境集

\LaTeX 环境包括文字版式环境、图表环境和数学版式环境。环境引用的格式为：`\begin{环境名}[选项]...内容...\end{环境名}`。环境的作用范围为`\begin{环境名}`与`\end{环境名}`之间的内容。`\begin{环境名}`与`\end{环境名}`必须成对使用，否则编译会出错，环境可以相互嵌套。这里我们仅列出文字版式环境中的命令，并说明几种比较常用的命令，对于图表环境和数学版式环境我们在后面给予说明。

文字版式环境主要有：`center`环境、`flushleft`环境、`flushright`环境、`itemize`环境、`enumerate`环境、`description`环境、`minipage`环境、`quote`和`quotation`环境、`verb`和`verbatim`环境和`verse`环境等。下面我们对一些常用的环境给予说明。

- `center`、`flushleft`、`flushright`环境

1. `center`环境将使位于`\begin{center}`与`\end{center}`之间的内容居中排列，引用格式中无选项。

2. `flushleft`环境将使位于`\begin{flushleft}`与`\end{flushleft}`之间的内容靠左边界对齐排版，而右边界则不一定对齐，引用格式中无选项。

3. `flushright`环境将使位于`\begin{flushright}`与`\end{flushright}`之间的内容靠右边界对齐排版，而左边界则不一定对齐，引用格式中无选项。

上面的条款1，2，3就是我们采用`enumerate`环境产生的，该环境引用格式中无选项。而每一个条款的效果是我们分别采用`center`、`flushleft`和`flushright`环境生成的。

- `minipage`环境

此环境定义局部段落的宽度小于标准正文宽度。如论文的摘要宽度通常比正文窄，则可利用此环境命令设置它的宽度。该环境的引用中有选项，格式为：`\begin{minipage}{size},...,内容,...\end{minipage}`。

- `verb`和`verbatim`环境

这两个环境均用于取消 \LaTeX 排版，即位于这两个环境中的内容将按源文件中的输入格式输出， \LaTeX 不对它重新排版。`\verb`的引用格式为：`\verb|内容|`，这是取消单行 \LaTeX 排版的命令格式。如果要取消多行 \LaTeX 排版，则用`verbatim`环境，格式为：

`\begin{verbatim}...内容,...\end{verbatim}`

上面由实心圆标记的条款列举就是我们用itemize环境生成的，该环境引用格式中无选项。itemize环境和enumerate环境的引用格式为：

```
\begin{ itemize(enumerate) }
\item...,内容, ...
:
\item...,内容, ...
\end{itemize(enumerate)}
```

5 数学公式和表格的排版

5.1 数学公式排印功能

在该节中我们介绍数学公式的排版。数学状态有两种：一是与正文混合排印，另一种是单独排印。混合排印是用 \dots 或者 (\dots) 或者 $\begin{math}\dots\end{math}$ 将数学公式括起来。单独排印时用 $\begin{displaymath}\dots\end{displaymath}$, $[\dots]$ 以及equation,array,eqnarray等环境将公式括起来。数学环境主要用于数学公式的排版、数学符号的产生和数学公式的自动编号等。equation环境和eqnarray环境排版的数学公式可以自动编号，并且环境引用中无选项，例如：

```
\begin{eqnarray}
&\min F(x,y)=a^T x+b^T y\\
&\min f(x,y)=c^T x+d^T y\\
&s.t.\\
&Ax+By\leq r \\
&x,y\geq 0
\end{eqnarray}
```

其结果为

$$\min F(x,y) = a^T x + b^T y \quad (1)$$

$$\min f(x,y) = c^T x + d^T y \quad (2)$$

$$s.t. \quad (3)$$

$$Ax + By \leq r \quad (4)$$

$$x, y \geq 0 \quad (5)$$

在eqnarray环境中如果不需要编号，则输入

```
\begin{eqnarray*}
公式源代码
\end{eqnarray*}
```

结果为：

$$\min F(x,y) = a^T x + b^T y$$

$$\min f(x,y) = c^T x + d^T y$$

$$s.t.$$

$$Ax + By \leq r$$

$$x, y \geq 0$$

. 图表环境有: tabbing环境、tabular环境、table和figure环境和picture环境等。下面介绍一下tabular 环境。该环境可以很容易制作各种复杂的表格, 引用中有选项。命令格式为:

`\begin{tabular}`对齐方式选择...内容...`\end{tabular}` 对齐方式包括左对齐(l)、右对齐(r)和居中(c).具体应用见下节。

5.2 数学公式和表格的排印实例

$$\frac{X_i^j}{Y_i Z^j} \quad \arctg \sqrt[n]{e^{\log_e x}}$$

$$\sum_{i=0}^n \int_0^{+\infty} f_i(x) dx = \sum_{i=0}^n \int_0^{+\infty} f_i(x) dx$$

$$\sum_{i=0}^n \int_0^{+\infty} f_i(x) dx = \sum_{i=0}^n \int_0^{+\infty} f_i(x) dx$$

$$\prod_{i=1}^n T_i^j = \prod_{i=1}^n T_i^j$$

$$\frac{dy}{dt} = \dot{y} = y' = y'' \neq y'''$$

$$\frac{dy}{dt} = \dot{y} = y' = y'' \neq y'''$$

$$T(a) = \begin{cases} 1, & \text{if } a < 0, \\ 0, & \text{if } a = 0, \\ -1, & \text{if } a > 0. \end{cases}$$

$$T(a) = \begin{cases} 1, & \text{if } a < 0, \\ 0, & \text{if } a = 0, \\ -1, & \text{if } a > 0. \end{cases}$$

$$x_0, x_1, x_2, \dots, x_n, x_{n+1}$$


```

\vdots&\vdots&\vdots&\ddots&\vdots\\
x_{n}&x_{n+1}&x_{n+2}&\cdots&x_{2n}\\
\end{array}
\right|
\\

```

$$\begin{vmatrix} x_0 & x_1 & x_2 & \cdots & x_n \\ x_1 & x_2 & x_3 & \cdots & x_{n+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_n & x_{n+1} & x_{n+2} & \cdots & x_{2n} \end{vmatrix}$$

```

\begin{tabular}{|c|c|c|c|c|c|c|}
\hline { \mbox{Prob.}}&\multicolumn{3}{c|}{ \mbox{result in the paper}}&\hline
\cline{2-4}
No.&$(x,y)$&$F(x,y)$&&$f(x,y)$&\hline
\hline $1$&$(15, 7.5, 10, 7.5)$&$331.25$&&$25$&\hline
\hline $2$&$(0.8462, 0.7692, 0)$&$-2.0771$&&$-0.5918$&\hline
\hline $3$&$(0.611,0.389,0,0,1.833)$&$0.6389$&&$1.6806$&\hline
\hline
\end{tabular}
\end{center}

```

Prob.	result in the paper		
No.	(x, y)	$F(x, y)$	$f(x, y)$
1	(15, 7.5, 10, 7.5)	331.25	25
2	(0.8462, 0.7692, 0)	-2.0771	-0.5918
3	(0.611, 0.389, 0, 0, 1.833)	0.6389	1.6806

	无编号	自动编号
单行公式	$\begin{array}{c} \text{(内容)} \end{array}$	
	$\begin{array}{c} \text{(内容)} \end{array}$	

6 特殊技巧

6.1 宏命令

宏命令就是一些命令的集合，如果这样的原命令组合在文件中要反复经常使用，用宏命令就可以简化输入。宏命令的基本形式为：

```
\newcommand{} [] {}
```

第一个花括号是新的宏命令名，方括号中是形式参数的个数，后面的花括号是原命令的组合。例如：文章中经常出现混合偏导数

$$\frac{\partial^3 u}{\partial t \partial x^2}$$

。

在文本中要输入命令

```
\frac{\partial^3 u}{\partial t \partial x^2}
```

如果定义了宏命令就可以减少命令，不必每次重复这些命令，只要定义：

```
\newcommand{\prdd}[3]{\frac{\partial^3 #1}{\partial #2 \partial #3^2}}
```

这样在文本中只要输入`\prdd u {t}{x}`

就能得到所要的偏导数。宏命令`[3]`表示宏命令中的形式参数的个数，这个数最大为9。`#1`，`#2`，`#3`代表不同的形式参数。

与`\newcommand`有关的另一个命令是`\renewcommand{}{}`，这个命令是修改原来`\newcommand`命令。第一个花括号是要被修改的命令，第二个命令是修改后的命令。例如，在用CJK宏包打中文的article时，参考文献为“References”，如果我们想让他为“参考文献”，则用宏命令

```
\renewcommand{\refname}{参考文献}即可。
```

L^AT_EX系统不仅允许用户定义宏命令，而且允许用户建立新的环境命令，新环境命令的基本形式是：

```
\newenvironment{}{}{}
```

第一个花括号是定义后的环境名，第二和第三个花括号是原先的两个环境命令，新环境可以像宏命令一样有替换命令。例如定义

```
\newenvironment{www}{\begin{document}}{\end{document}}
```

则原先的两个命令可用

```
\begin{www}
```

内容

```
\end{www}
```

代替，这样就很方便。另外，对于原理、定理、定义或引理等，为了表示这些条款，系统提供了环境命令`\newtheorem`，举例如下：

```
\newtheorem{mythname}{Theorem}
```

```
\begin{mythname}
```

定理内容

```
\end{mythname}
```

输出为：

Theorem 1 定理内容

如有第二条定理，则系统自动编号

Theorem 2 定理的内容

另外，`\newtheorem`命令可以有数值变量。例如：

```
\newtheorem{myprop}{Propositon}[section]
```

```
\begin{prop}
```

这是一个命题

```
\end{prop}
```

我们在这一节中输入上一段，则输出的结果如下：

Propositon 6.1 这是一个命题

编号是系统自己排好的。

6.2 文献和图表的引用

无论在写文章还是写书都会遇到对文献图表的引用，在写短文章时对文献的引用可以采用简单的上标法就行了，然而当有大量的文件和图表要引用时，这种方法就显得不方便了，因为要记忆引用的文献图表章节等的编号， \LaTeX 提供了交叉引用的命令，是你在输入文件是不必记住要引用的文献图表章节的具体编号，只要你在引用处做一个标示符，最后系统在编译时能自动编号，具体方法是在你要引用的部分进行标记，标记方法分别如下：

1. 在书籍中对章节进行引用时，首先对要引用的章节进行标记，例如：

```
\chapter {First Miss}\label{FM}
```

FM是作者自己给First Miss这一章的记号，在文章中引用时可写成：

...what we talk about can be found in Chapter $\text{\ref{FM}}$...

在文件最后编译时，章的号码被自动编排。

2. 在对方程引用时，被引用方程的标记方法如下：

```
\begin{equation}\label{eq1}      标识的记号有自己定  
公式内容  
\end{equation}
```

3. 对图表的标识方法：

```
\begin{figure}\label{fig1}      标识的记号有自己定  
图表  
\end{figure}
```

当在文章中引用时可写为：

The Eq. $\text{\ref{eq1}}$ indicates a result as shown in Fig. $\text{\ref{fig1}}$...

这样在系统最后编译时自动把方程和图表的号码编好了。这种引用的用处是在引用时不必涉及被引用部分的序号，也方程和图表的序号是自动的，在你输入源文件时不必预先知道，这样就很简便而且不易出错。

4. 对参考文献的引用，标识得命令是 $\text{\cite{},{}}$ 中的标记，可自行定义，但引用的记号必须与参考文献一致。例如我们要对本文的文献一就行引用，在文章中就可以用：

这是一个对参考文献交叉引用的例子 $\text{\cite{s1}}$

结果为：这是一个对参考文献交叉引用的例子[1]

在又交叉引用时，必须用 \LaTeX 编译两次才能产生编号，系统地一次编译时产生一个辅助文件(.aux)用于存放交叉引用的信息，在有了辅助文件之后再编译一次就能产生正确的编号。

7 参考文献的使用

参考文献的引用格式为:

```
\begin{thebibliography}{99}
\bibitem{label}{}
:
\bibitem{label}{}
\end{thebibliography}
```

在`\begin{thebibliography}{99}`中的数字表示打印的宽度, 这个宽度按最宽的项计算(以多少个字符为单位), 这个变量是必需的, 否则会出错。一般使用99, 也就是说最大可以包含99个字符。例如本文的参考文献的形式为:

```
\begin{thebibliography}{99}
\bibitem{s1} 作者名. \emph{文章名. 杂志名,}出版年份. {\bf 卷(期)}: 起止页码.
\bibitem{s3} 作者名. \emph{书名.}出版地:出版年份.起止页码.
\end{thebibliography}
```

参考文献

[1] 作者名. 文章名. 杂志名,出版年份. **卷(期)**: 起止页码.

[2] 作者名. 书名.出版地:出版年份.起止页码.