

# L<sup>A</sup>T<sub>E</sub>X排版心得

李东风

2010年4月9日

# 目 录

第一章	<b>L<sup>A</sup>T<sub>E</sub>X</b> 入门.....	1
1.1	L <sup>A</sup> T <sub>E</sub> X基本知识 .....	1
1.1.1	L <sup>A</sup> T <sub>E</sub> X历史和特点 .....	1
1.2	安装与设置 .....	2
1.3	编辑和编译 .....	2
1.3.1	在TeXWorks中编写L <sup>A</sup> T <sub>E</sub> X文件 .....	2
1.3.2	在WinEdt中编写L <sup>A</sup> T <sub>E</sub> X文件 .....	3
1.3.3	用命令行编译 .....	3
1.3.4	编译为HTML .....	4
1.4	L <sup>A</sup> T <sub>E</sub> X源文件 .....	4
1.4.1	L <sup>A</sup> T <sub>E</sub> X输入文件 .....	4
1.4.2	特殊字符 .....	4
1.4.3	L <sup>A</sup> T <sub>E</sub> X命令 .....	5
1.4.4	输入文件的结构 .....	6
1.5	CJK L <sup>A</sup> T <sub>E</sub> X源文件示例 .....	6
1.6	一般文本排版 .....	7
1.6.1	段落 .....	7
1.6.2	标题、章节 .....	7
1.6.3	其它结构 .....	7
1.6.4	环境 .....	8
1.6.5	制表环境 <code>tabular</code> .....	8
1.6.6	浮动环境 .....	8
1.7	数学公式入门 .....	8
第二章	<b>Emacs+AucTeX</b> .....	11
2.1	安装 .....	11
2.2	Emacs的基本编辑功能 .....	12
2.3	用Emacs+AucTeX进行L <sup>A</sup> T <sub>E</sub> X编辑编译 .....	13
2.3.1	输入辅助 .....	14
2.3.2	注释 .....	14
2.3.3	数学符号输入辅助 .....	14
2.3.4	大文件辅助 .....	15
2.4	用reftex作引用 .....	15
第三章	<b>L<sup>A</sup>T<sub>E</sub>X</b> 提高.....	16
3.1	编译结果 .....	16
3.2	带书签的PDF文件 .....	17
3.3	版式设置 .....	17

3.4	大文件处理	18
3.5	如何同时应用CJK和CCT	19
3.6	插入图形	19
3.7	源程序排版	22
<b>第四章</b>	<b>幻灯片演示制作</b>	<b>25</b>
4.1	用Beamer制作幻灯片	25
4.1.1	基本的beamer幻灯片	25
4.1.2	编译速度	26
4.1.3	同时生成幻灯片和打印版本	27
4.2	用PDFSlide制作幻灯片	27
4.2.1	PPower4软件	29
<b>第五章</b>	<b>其他技巧</b>	<b>30</b>
5.1	Linux中 $\text{\LaTeX}$ 和中文的使用	30
5.1.1	Linux下CJK和中文字体的安装	30
5.1.2	Redhat Linux 9的中文双拼输入问题	31
5.2	其他问题	31
5.2.1	关于英文系统和中文系统的兼容性问题	31
5.2.2	一些小的问题	31



# 第一章

## L<sup>A</sup>T<sub>E</sub>X入门

本文先简单介绍T<sub>E</sub>X和L<sup>A</sup>T<sub>E</sub>X系统，然后是一些经验总结，其中包括作者用L<sup>A</sup>T<sub>E</sub>X和PDFL<sup>A</sup>T<sub>E</sub>X打印《统计软件教程》一书的心得，编辑一些数学书籍的经验，和使用L<sup>A</sup>T<sub>E</sub>X制作课堂演示课件的一些经验。本文排版结果展示了一些特殊技巧，要知道自己如何使用这些技巧要参考本文的L<sup>A</sup>T<sub>E</sub>X源程序。

### 1.1 L<sup>A</sup>T<sub>E</sub>X基本知识

#### 1.1.1 L<sup>A</sup>T<sub>E</sub>X历史和特点

T<sub>E</sub>X是著名计算机专家Donald E. Knuth发明的一个用于文章排版的计算机软件系统，也是一种编程语言。1982年公开。以软件稳定性高著称。

L<sup>A</sup>T<sub>E</sub>X是在T<sub>E</sub>X基础上是排版更方便的一个子系统。

L<sup>A</sup>T<sub>E</sub>X和MS Word比较有如下特点：

- MS Word是桌面文档系统，不能作为高质量印刷。强调WYSIWYG。
- L<sup>A</sup>T<sub>E</sub>X不提倡作者自己规定文章样式，而使用统一样式。强调文章的逻辑结构而不是直观样式。
- 很多杂志投稿接受L<sup>A</sup>T<sub>E</sub>X电子版会提供自己的模板，只要使用他们的模板就可以完美地融入整个杂志中。
- 强项在于数学公式和特殊符号丰富易用。
- 容易生成公式图表编号、引用、交叉引用、目录。
- 用L<sup>A</sup>T<sub>E</sub>X可以容易地管理大型文档。
- L<sup>A</sup>T<sub>E</sub>X可以输出成Postscript、PDF、HTML等格式。
- 一些数学计算软件可以直接生成L<sup>A</sup>T<sub>E</sub>X表格。
- T<sub>E</sub>X和L<sup>A</sup>T<sub>E</sub>X是免费的！各操作系统都支持。支持中文。

## 1.2 安装与设置

L<sup>A</sup>T<sub>E</sub>X有许多可选的安装方式，功能各有特点。为了方便处理中文，在MS Windows系统中作者建议使用CTEX安装包。尽管有其他方法也能处理中文，但现在比较完善的还是CTEX。

从[www.ctex.org](http://www.ctex.org)中获取软件。最新版本是2.8版。安装完成后，可以使用同时安装的WinEdt或TeXWorks输入源文件和编译、查看。为了能够在TeXWorks中能使用GBK编码的中文，选菜单“编辑—选项”，选编码为“System”。

Emacs编辑器是在Linux及其他Unix类操作系统上广泛使用的一个功能极为强大的集成工作环境，有MS Windows版本。在安装了AucTeX包后可以作为一个集成的L<sup>A</sup>T<sub>E</sub>X编辑、编译、浏览、打印系统(需要MikTeX作为后台支持)。在编辑L<sup>A</sup>T<sub>E</sub>X文件时有很好的语法彩色加亮功能，支持源文件位置到DVI位置的正、反向对应，还可以预览文内的公式和图形。

注意：Emacs的使用和其他编辑器差别较大，可能需要较长的适应过程。

为了安装设置Emacs+AucTeX+CTEX，请参见第2章。

## 1.3 编辑和编译

L<sup>A</sup>T<sub>E</sub>X源文件需要经过编译才能查看和打印。一般编译为DVI、PDF、Postscript、HTML等格式。

DVI格式是一种编译结果，用专门的查看程序YAP查看，可以在写作中间过程中使用。为了提供给大众浏览还是应该变成PS或PDF格式。可以把DVI转换为PS和PDF，也可以直接把L<sup>A</sup>T<sub>E</sub>X文件用PDFL<sup>A</sup>T<sub>E</sub>X编译为PDF。

编译过程会生成一些辅助文件，在得到结果后可以删除这些辅助文件。辅助文件包括\*.log, \*.aux, \*.toc, \*.out, \*.nav, \*.vrb等文件。

### 1.3.1 在TeXWorks中编写L<sup>A</sup>T<sub>E</sub>X文件

TeXWorks是一个便于初学者使用而又具有强大功能的L<sup>A</sup>T<sub>E</sub>X编辑、编译、查看集成环境，是免费的。编译结果为PDF。它支持L<sup>A</sup>T<sub>E</sub>X的语法加亮显示，源程序和编译PDF结果左右并排放置，可以从源程序跳到结果中对应位置，也可以从PDF结果跳到源程序相应位置。可以通过L<sup>A</sup>T<sub>E</sub>X+CJK支持中文，也可以用XeTeX+Unicode直接使用Unicode编码的中文。

TeXWorks提供了几个简单模板，初学者可以用这些模板作为自己的开始样式。用户自己的模板也可以放在C:\Documents and Settings\你的用户名\TeXworks 下面的templates目录中。

为了使用GBK编码的CJK中文，可以在“编辑—选项—编辑器—编码”中选择System，这样缺省使用系统的中文编码，也可以在文件首行写特殊注释

```
% !TEX encoding = System
```

如果要使用Unicode，把此行的“System”改为“utf8”。

为了编译，只要按图标栏的“排版”菜单即可。用Ctrl+Click或右键菜单可以在源程序和对应的排版结果之间来回跳转。注意，在使用beamer包时，\maketitle命令会导致逆向查找功能失败。

编译输出显示在编辑区和状态行之间，如果编译出错，会在输出和状态行之间打开一个控制台行，在此回车可继续，输入x后回车可以停止编译。

编译会产生一些\*.aux, \*.log, \*.out等辅助文件, 可以用“文件—删除辅助文件”菜单来删除。

要注释掉选定的行, 用菜单“格式—注释掉”。命令“格式—取消注释”可以去掉注释。

TeXWorks有辅助输入功能。打入一个命令的一部分然后按TAB键可以自动完成这个命令, 比如打入`latex`然后按TAB键可以变成`\LaTeX`。打入以`b`为前缀环境名或部分环境名可以插入环境, 如果有多个环境可以匹配可以用TAB键切换。插入的环境中有圆点用来定位到某输入位置, 用`Ctrl+TAB`键可以切换这些输入位置。

如果编辑窗的字太小或字体不合适, 可以用“Format—Font...”菜单修改。

在编写英文文件时, 可以选菜单设置“编辑—拼写—en\_US”来打开拼写检查功能。

当文件比较长的时候, 可以选菜单“窗口—显示—标签”显示文件的树状结构导航。

用“Edit—Preferences...”菜单的“排版”页可以设置默认的编译命令, 如PDFLaTeX。从图标栏的“排版方式”下拉菜单可以临时改用某编译命令。如果某个文件总是用某一命令编译, 可以在文件开头加上特殊注释如

```
% !TeX program = xelatex
```

这指定此文件用`xelatex`编译。

如果使用主控文件和子文件, 在子文件的开头应加上注释行

```
% !TeX root = mypap.tex
```

其中`mypap.tex`是主控文件名。

### 1.3.2 在WinEdt中编写 $\text{\LaTeX}$ 文件

WinEdt和是一个一般性的文本文件编辑器, 是共享软件, 提供了对 $\text{\LaTeX}$ 编辑、编译的较好支持。支持 $\text{\LaTeX}$ 的语法高亮显示, 辅助输入, 编译和查看结果的快捷方式。可以编译为DVI或PDF格式。

WinEdt中有编译用的快捷图标, 可以用 $\text{\LaTeX}$ 把文件编译为DVI格式, 用 $\text{Xe}\text{\LaTeX}$ 把文件编译为DVI格式, 用 $\text{PDF}\text{\LaTeX}$ 把文件编译为PDF格式, 可以把DVI格式转换为Postscript格式(\*.ps)或PDF格式, 把Postscript格式转换为PDF格式。可以查看编译的DVI或PS、PDF文件。在查看DVI时双击某处可以反向查找源程序中对应位置(TeXWorks和Emacs+AucTeX也具有这样的功能)。

### 1.3.3 用命令行编译

如果设置了程序路径, 也可以在一个命令行窗口进行编译, 如:

```
latex latex02
yap latex02
dvips latex02
dvi2pdf latex02
```

### 1.3.4 编译为HTML

为了能够把公式显示在HTML结果中，需要预先安装好ImageMagic软件，确保路径中能找到。TeX2ht软件可以把L<sup>A</sup>T<sub>E</sub>X源程序编译为HTML格式，用htlatex命令编译。如在命令行运行

```
htlatex latex02
```

应该预先把待编译的文件和图形放在一个单独的目录中。编译后可能会生成一些图形文件，这些图形文件要伴随生成的HTML文件。

## 1.4 L<sup>A</sup>T<sub>E</sub>X源文件

### 1.4.1 L<sup>A</sup>T<sub>E</sub>X输入文件

L<sup>A</sup>T<sub>E</sub>X输入文件是一种计算机源程序，但是简单任务只需要很少的命令。多个空格和一个空格不区分，所以不能像Word那样用增加空格对齐。空行代表不同段落。

图1.1是最简单的一个输入文件latex01.tex。图1.2是最简单的一个中文输入文件latex02.tex。

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

Paragraph 1.
The input for LATEX is a plain {\LARGE ASCII} text file.
You can create it with any
text editor. It contains the text of the document,
as well as the commands
that tell LATEX how to typeset the text.

Paragraph 2.
‘‘Whitespace’’ characters, such as blank or tab,
are treated uniformly as
‘‘space’’ by LATEX. Several consecutive whitespace
characters are treated
as one ‘‘space.’’ Whitespace at the start
of a line is generally ignored, and a
single line break is treated as ‘‘whitespace.’’.

\end{document}
```

图 1.1: 最简单的英文L<sup>A</sup>T<sub>E</sub>X输入文件

### 1.4.2 特殊字符

- L<sup>A</sup>T<sub>E</sub>X命令中要用到一些特殊字符，所以实际想印出这些特殊字符时必须使用替代命令。



```
\documentclass{article}
\usepackage{CJK}
```

```
\begin{document}
\begin{CJK*}{GBK}{song}
```

L<sup>A</sup>T<sub>E</sub>X不提倡作者自己规定文章样式，而使用统一样式。

强调文章的逻辑结构而不是直观样式。

很多杂志投稿接受L<sup>A</sup>T<sub>E</sub>X电子版会提供自己的模板，

只要使用他们的模板就可以完美地融入整个杂志中。

强项在于数学公式和特殊符号丰富易用。

容易生成公式图表编号、引用、交叉引用、目录。

“用L<sup>A</sup>T<sub>E</sub>X可以容易地管理大型文档。”

L<sup>A</sup>T<sub>E</sub>X可以输出成Postscript、PDF、 $\LaTeX$ ，HTML等格式。

一些数学计算软件可以直接生成L<sup>A</sup>T<sub>E</sub>X表格。T<sub>E</sub>X和L<sup>A</sup>T<sub>E</sub>X是免费的！

各操作系统都支持。支持中文。

```
\end{CJK*}
\end{document}
```

图 1.2: 最简单的中文L<sup>A</sup>T<sub>E</sub>X输入文件

- # \$ % ^ & \_ { } ~ \.
- \# \\$ \% \^{} \& \\_ \$\{ \\$ \\$\} \\$ \~{} \\$\backslash\$.
- 双引号左边用两个反单引号，右边用两个单引号。
- 连字符用--代替: my-file。
- 破折号用---代替: Hero—who?。

### 1.4.3 L<sup>A</sup>T<sub>E</sub>X命令

- 命令以反斜杠开头，后面英文字母组成的命令。如\LaTeX。
- 命令后面的空格被忽略。
- 命令可以有参数，写在大括号内，如\texttt{name} 显示为name。
- 为了限制一个命令的作用范围，把希望有作用的范围包在{\dots}内。  
如{\Large大字体}结果为**大字体**。
- %后面是注释。
- 注释中的空格、换行不被解释，所以一个长单词如果用注释在结尾拆开最后结果是相连的。（见latex03.tex）

### 1.4.4 输入文件的结构

- `\documentclass`命令规定使用的文档模板，是必须的。常用的有`article`、`book`、`report`、`beamer`等。
- 文档内容写在`\begin{document}`和`\end{document}`之间。
- `\documentclass`和`\begin{document}`之间的部分叫做序言(preamble)，主要是引入一些额外的宏包(macro package)和作一些用户定制。
- 引入宏包如`\usepackage{CJK}`。这是支持中文输入文件排版的宏包。
- 另外常用的宏包有：
  - `graphicx`: 插图用。
  - `amsmath`: 一些方便的公式、符号命令。

## 1.5 CJK L<sup>A</sup>T<sub>E</sub>X源文件示例

下面是一个基本的使用CJK的框架:

```
\documentclass[a4paper]{article}
\usepackage{CJK}           % 用CJK支持汉字
\usepackage{CJKpunct}      % 支持汉字标点
\AtBeginDocument{\begin{CJK*}{GBK}{song}\CJKtilde\CJKcaption{GB}}
\AtEndDocument{\end{CJK*}}
\begin{document}
\include{mypap-inc}
\end{document}
```

其中的内容部分写在一个单独的文件 `mypap-inc.tex`中，用`\include{mypap-inc}`可以把文件内容整体插入。这样拆成主文件、子文件的好处是可以写几个主文件，比如一个主文件用于生成 DVI，编译速度快；一个主文件用于生成 PDF，编译较慢但是结果交流容易；一个主文件用于生成课堂演示幻灯片；一个主文件用于 CCT，等等。这些主文件共享一个内容子文件。因为生成 DVI 和 PDF、幻灯片、使用 CCT 等所需的文件头不同(采用的宏包不同，插图方式不同)所以用几个主文件来调用相同的内容文件是比较好的办法。这种使用 CJK 的方法适用于其他 class 如幻灯类 beamer，但有可能需要加上 CJK 选项。

要注意的是使用`\include`命令会在插入内容的前后换页。

用类似`\CJKfamily{fs}`的命令可以改变现在所用字体，其中 `fs` 是仿宋体，`kai` 是楷体，`hei` 是黑体。

使用`\CJKcaption{GB}`要用到 CJK 包中的 `GB.cpx`，系统中的文件有错误，我们这里提供可用版本放在与文件相同的子目录下。

CJK 对夹杂中英文的段落换行有时不合适产生太长的行或太稀的间隔，为此可以使用`\sloppy`命令放松拆行的标准。更严密的解决办法是在适当地方加上单词拆词的提示`\-`和小的空白`\,`，及弹性空白`\~`允许在该处拆行。

L<sup>A</sup>T<sub>E</sub>X 习惯在章节开始段首不缩进，为了使得开始段也段首缩进可以引入 `indentfirst` 宏包。

## 1.6 一般文本排版

### 1.6.1 段落

- L<sup>A</sup>T<sub>E</sub>X只需规定文章的结构: 段落、小节、节、章、标题、作者、日期等。
- 用空行分开的部分为段落。源文件中的段落内换行不影响最后的排版结果。
- 一定不要在无意识中新开一个段落。比如, 公式后面如果有空行则为新段落。
- 段落由句子组成, L<sup>A</sup>T<sub>E</sub>X试图根据标点分开句子。
- L<sup>A</sup>T<sub>E</sub>X可以在行尾拆分单词以使得行内的空白宽度适中。
- 用`\`强制在段内换行。
- 用`\newpage`强制换页。
- 如果某些行难以拆分, 会造成长行, 这时可以在前面用一个`\sloppy`命令, 可以在较前面拆分, 造成较多行内空白, 过后用`\fussy`命令恢复。
- 可以用`\-`加入一个单词中提示可以在该位置换行。(见`latex03.tex`)

### 1.6.2 标题、章节

- 用`\title`命令指定标题, `\author`命令指定作者, `\date`命令指定日期(见`latex04.tex`)。 `\maketitle`制作标题。
- 在`article`文档内, 用`\section{...}`, `\subsection{...}`, `\subsubsection{...}`, `\paragraph{...}`, `\subparagraph{...}`指定文档结构。
- 见`latex04.tex`。这也是使用CCT的例子。
- 在`article`文档内还可以用`\part{...}`命令把文档分成几个部分, 各部分章节仍统一编排。
- 在`book`文档内有更高级别的`\chapter{...}`。
- 用`\appendix`命令指定后面章节为附录, 编号变成A, B, C等等。
- 用`\tableofcontents`命令制作目录。

### 1.6.3 其它结构

- 用`\emph{...}`强调内容。
- 用`\footnote{...}`作脚注。
- 在某个章节用`\label{sec:intro}`之类作标签, 可以在其它地方用`\ref{sec:intro}`引用这个章节。可以在其它地方用`\pageref{sec:intro}`引用该处的页码。

### 1.6.4 环境

- 用`\begin{环境名}`开始, 用`\end{环境名}`结尾。
- 中间包含允许的内容。
- 如: 列表`itemize`环境。用`\`命令开始一个条目。(见`latex04.tex`)。
- 如: 有序列表`enumerate`环境。(见`latex04.tex`)。
- 如: 居中环境`center`环境。(见`latex04.tex`)。
- 原样输出环境`verbatim`, 行内可以用`\———命令`。(见`latex04.tex`)。

### 1.6.5 制表环境tabular

- 开始环境有参数`\begin{tabular} {...}`, 参数如`ccc`, 表示有三列, 都居中对齐。`l`表示左对齐, `r`表示右对齐。`|`表示增加一条竖线。
- 表格行用`\\`结尾; 用`&`分开表格单元。
- 表格行之间用`\hline`画线。见`latex04.tex`。

### 1.6.6 浮动环境

- 长表格占地多, 可能无法在本地排开只好到下一页, 造成前页多余空白。
- L<sup>A</sup>T<sub>E</sub>X把表格包在`table`环境中, 则表格可以自动排到合适的位置。
- 在`table`环境中, 用`\label{tab:person}`之类定义表格标签, 然后在其它地方可以用`\ref{tab:person}`引用表格号码。
- 图形用`figure`浮动环境。

## 1.7 数学公式入门

数学公式是L<sup>A</sup>T<sub>E</sub>X的强项。数学中所有符号基本都可以在L<sup>A</sup>T<sub>E</sub>X中找到。(参见: `symbols-a4.pdf`.)

行内公式包在`$`和`$`之间, 或包在`\(`和`\)`之间。(见图1.3的`latex05.tex`.)

独立公式(displayed formula)包在`$$`和`$$`之间, 或包在`\[`和`\]`之间。注意独立公式前后不要留有空行, 除非另起一段。

行内公式和独立公式风格有所不同。

数学公式中的多个空格只能作为一个, 不能空行, 每个字母都认为是一个变量, 需要一些普通文字可以用`\mbox{...}`命令。

一些基本公式:

- `^`后为上标, 如`a^2`结果为 $a^2$ , `a^{(2)}`结果为 $a^{(2)}$ 。
- `_`后为下标。如`a_1`结果为 $a_1$ 。
- `a_1^2`结果为 $a_1^2$ 。注意含多个字符的上下标需要用大括号包围, 如 $e^{ij\lambda}$ 应输入为`$e^{\mathit{ij}\lambda}$`。
- 分式: `\frac{分子}{分母}`格式。如 $\frac{a}{b}$ 。

- 求和: `\sum`。如 $\sum_{i=1}^n \frac{1}{i^2}$ 。
- 积分: `\int`。如 $\int_0^1 f(x)dx$ 。

用`equation`环境保存自动编号的公式。在环境内用如`\label{eq:abc1}`之类定义公式标签，在别处用(`\ref{eq:abc1}`)来引用公式号。

用`eqnarray`环境保存自动编号的多行公式，不同行用`\\`分隔，中间的等号改成`& = &`表示则可以对齐。没有等号的行则在需要与其它行等号对齐的位置加上`&&`。`eqnarray`环境改成`eqnarray*`环境则不编号。

复杂公式和符号参见`latexbook.pdf`，`symbols-a4.pdf`。

```

\documentclass{article}
\usepackage{CJK}
\usepackage{CJKpunct}
\AtBeginDocument{\begin{CJK*}{GBK}{song}\CJKtilde\CJKcaption{GB}}
\AtEndDocument{\end{CJK*}}

\begin{document}

行内公式: \(\int_0^1 \sum_{i=1}^n f(x_i, \theta) d\theta\).
$a^{(2)}$正确。$a^{(2)}$错。分式$\frac{a}{b}$。
独立公式:
$$
\int_0^1 \sum_{i=1}^n f(x_i, \theta) d\theta
$$
或
\[
\text{\mbox{rain}}^2 + \text{\mbox{wind}}^2 = c^2.
\]
注意独立公式前后不要留有空行，除非另起一段。

\begin{equation}
a + b = c^2 \label{eq:abc2}
\end{equation}

编号公式:
\begin{equation}
a + b = c \label{eq:abc1}
\end{equation}
引用如公式(\ref{eq:abc1})。

编号多行公式:
\begin{eqnarray}
a + b &=& c \label{eq:abc11} \\
a^2 + b^2 &=& c^2 \label{eq:abc21}
\end{eqnarray}

不编号多行公式:
\begin{eqnarray*}
a + b &=& c \\
a^2 + b^2 &=& c^2
\end{eqnarray*}

\end{document}

```

图 1.3: 包含公式的L<sup>A</sup>T<sub>E</sub>X输入文件

## 第二章

# Emacs+AucTeX

对于熟悉Emacs的用户，用Emacs+AucTeX可以实现 $\text{\LaTeX}$ 的编写、编译、查看。其功能比TeXWorks和WinEdt更强，但是其使用不同于通常的MS Windows编辑程序，所以不建议一般MS Windows用户使用。

Emacs+AucTeX特点为：

- 语法高亮显示；
- 预览功能，即在编辑器内部直接显示公式编译结果和图形；
- 源文件和DVI结果正向和反向互查；
- 命令、环境、符号辅助输入；
- 主控文件和子文件支持；
- 公式、图标、文献等的索引支持；
- 编译菜单和快捷方式。

### 2.1 安装

先安装MS Windows用的emacs-23.1.zip文件，只要把它释放在合适的目录就可以了，比如安装在c:\emacs中。找到一个gnuserv.zip软件包，把其中的四个.EXE程序文件放在emacs安装目录的bin目录中。

然后下载安装auctex-11.85-e23.1-msw.zip，只要把它释放到emacs安装目录就可以了，注意目录结构是以emacs安装目录为根的。

在emacs的site-lisp目录中创建或修改site-start.el文件，使其中包含行

```
;; for LaTeX Yap reverse search
(require 'gnuserv)
(gnuserv-start)
(setq gnuserv-frame (selected-frame))
(require 'tex-mik)
(require 'preview)
;; For LaTeX AucTeX mode accompanied by reftex mode.
(add-hook 'LaTeX-mode-hook 'turn-on-reftex)
(setq reftex-plug-into-AUCTeX t)
```

```
;; for selection using Ctrl-2
(global-set-key (quote [67108914]) (quote set-mark-command))

;; for Chinese environment
(set-language-environment 'Chinese-GB)
(set-keyboard-coding-system 'euc-cn)
(set-clipboard-coding-system 'euc-cn)
(set-terminal-coding-system 'euc-cn)
(set-buffer-file-coding-system 'euc-cn)
(set-selection-coding-system 'euc-cn)
(modify-coding-system-alist 'process "*" 'euc-cn)
(setq default-process-coding-system
      '(euc-cn . euc-cn))
(setq-default pathname-coding-system 'euc-cn)

;; for global font lock mode
(global-font-lock-mode t)

;; transient mark mode
(setq transient-mark-mode t)

;; show paren mode
(show-paren-mode t)
```

这样可以在每次启动Emacs时提供 $\text{\LaTeX}$ 编辑支持。Emacs还会生成一个C:\.emacs文件用来保存一些用户设置。

装完Emacs和AucTeX后再安装CTEX 2.8。这样可以省去一些麻烦。运行CTEX中的DVI显示程序Yap，选“View — Options”菜单再选“Inverse search”部分，如果其中Program有可选项“Gnu Emacs (Single instance)”，则选中这一项。这时对应的Command栏应显示

```
"C:\.emacs\bin\gnuclientw.exe" -F +%l "%f"
```

这样可以实现用Emacs+AucTeX编译的DVI可以正反向位置对应的功能。如果Yap反查设置的对话框中不能出现Emacs的选项，可以增加新命令，在Command栏中输入以上命令，当然要写对路径。

这个版本的Yap有些问题，不能修改反向查找命令，可以用regedit来修改，查找\CTEX\WinEdt\winedt.exe "[Open(|%f|);SelPar(%l,8)]" 然后改成上面的设置即可。

## 2.2 Emacs的基本编辑功能

在Emacs中用“Ctrl-x d”(按住Ctrl键按x键，放开Ctrl后按d键，此操作可简写为“C-x d”)可以输入一个目录回车然后在窗口中管理这个目录，比如光标在某文件名上回车可以打开这个文件。命令“Ctrl-x Ctrl-f”可以打开一个旧文件或建立一个新文件。

Emacs的编辑很有特点，表2.1列出了常用快捷键。注意用“C-x”表示按住Ctrl键再按x键，用“M-x”表示按住Alt键再按x键。Emacs窗口最下面状态



表 2.1: 光标移动及删除

操作	光标键	组合键	说明
右	右	C-f	<b>f</b> orward
左	左	C-b	<b>b</b> ackward
下	下	C-n	<b>n</b> ext
上	上	C-p	<b>p</b> revious
行首	Home	C-a	
行尾	End	C-e	<b>e</b> nd
下页	PageDown	C-v	
上页	PageUp	M-v	
文首	Ctrl-Home	M-<	
文尾	Ctrl-End	M->	
本行居中	无	C-l	
删除当前字符	Delete	C-d	
删除前一字符	Backspace	无	
从光标位置删除到行尾	无	C-k	<b>k</b> ill
取消刚才的删除	无	C-y	<b>y</b> ank
粘贴以前的删除	无	M-y 若干次	
恢复	无	C-x u	<b>u</b> ndo

行下的地方叫做mini-buffer，经常利用这里输入一些命令参数，如果要取消可以用“Ctrl-g”组合键。

注意“M-<”可能需要同时按住Alt和Shift键

复制、粘贴：在Emacs中选一块文本，可以用鼠标拖动，也可以按“Ctrl-2”组合键标记开始，把光标移动到另一位置标记结尾(标记的文本可能不加亮显示，可在Options菜单中打开Active Region Highlighting，这样在开始标记后可以加亮显示) 注意开始标记的命令本来是“Ctrl-@”，但是那样需要同时按Ctrl和Shift，所以我们在site-start.el文件中增加一行

```
(global-set-key (quote [67108914]) (quote set-mark-command))
```

就可以定义使用Ctrl-2来开始标记。标记后，可以用“Ctrl-w”命令剪切或用“Alt-w”命令复制，用“Ctrl-y”命令粘贴。用“Ctrl-x Ctrl-x”可以切换标记区的头尾。

在Emacs中保存了最近的若干次复制或剪切内容，先用“Ctrl-y”粘贴后可以用“Alt-y”替换成以前的内容。Emacs中还记住了最近几次开始标记的地点，可以用“Ctrl-u Ctrl-2”命令返回到以前的标记点。

Emacs是一个多窗口程序，可以同时编辑多个文件。Emacs中的一个子窗口叫做一个buffer(缓冲区)。状态行中显示了文件名，用鼠标左键或右键单击文件名可以切换到要编辑的文件。

## 2.3 用Emacs+AucTex进行L<sup>A</sup>T<sub>E</sub>X编辑编译

如果打开或新建的文件名以“.tex”结尾则自动进入L<sup>A</sup>T<sub>E</sub>X编辑状态。增加了菜单Preview、LaTeX和Command，Preview菜单控制在Emacs内部的预览。LaTeX菜单是辅助编辑命令，比如输入环境、命令、命令自动完成、查看编译错误等，Command命令用于编译、浏览、转换(PS, PDF)等。

打开 $\text{\LaTeX}$ 文件后，如果要编译为DVI并可反查，不要选Command菜单中的“TeXing Options”的“PDF Mode”，选中Command菜单中的“TeXing Options”的“Source Specials”，用图标栏的Run  $\text{\LaTeX}$ 命令编译。如果有错误可以用“C-c ’”键组合查看。用图标栏的Run Viewer命令可以打开MikTeX中的Yap程序对DVI进行浏览。Yap中当前编辑位置用一个小圈显示出来。在DVI中移动到某一位置然后双击可以返回到Emacs中源文件的对应位置。

如果要编译为PDF，则应该选中Command菜单中的“TeXing Options”的“PDF Mode”，然后用图标栏的Run  $\text{\LaTeX}$ 命令编译为PDF，用图标栏的Run Viewer命令查看编译的PDF。

在Emacs中用“Preview—Generate previews for document”可以为整篇文章打开预览。用“Remov previews ...”关闭预览。可以预览选定部分、所在环境等。

### 2.3.1 输入辅助

- “C-c C-m”可以输入 $\text{\LaTeX}$ 命令(macro)。在输入时可以用TAB键完成剩余部分。
- “C-c C-e”可以输入 $\text{\LaTeX}$ 环境，在输入过程中还提示需要的选项。
- “C-c C-s”可以输入章节命令。
- “M-RETURN”可以输入列表环境的下一item。
- “C-c C-f C-b”产生 $\text{\texttt{\textbf{}}}$ 。
- “C-c C-f C-e”产生 $\text{\texttt{\emph{}}}$ 。
- “C-c C-f C-t”产生 $\text{\texttt{\texttt{}}}$ 。
- “C-c {”产生一对大括号。
- “C-c ]”关闭当前环境。
- “C-c C-q C-r”排列当前region的格式(indent)。“C-c C-q C-e”排列当前环境。

### 2.3.2 注释

用“C-c ;”可以注释当前选定区域，用“C-u - C-c ;”可以取消当前区域的注释。

### 2.3.3 数学符号输入辅助

“C-c ~”进入数学符号辅助输入状态，出现Math菜单可以输入常见数学符号，也可以以‘为前缀加上一个单键输入数学符号。各希腊字母都可以用‘加字母键输入。其他数学符号对应关系如下：

‘< $\rightarrow$ ≤	‘> $\rightarrow$ ≥	‘I $\rightarrow$ ∞	‘A $\rightarrow$ ∀	‘E $\rightarrow$ ∃	‘i $\rightarrow$ ∈
‘* $\rightarrow$ ×	‘. $\rightarrow$ ·	‘{ $\rightarrow$ ⊂	‘} $\rightarrow$ ⊃	‘[ $\rightarrow$ ⊆	‘] $\rightarrow$ ⊇
‘0 $\rightarrow$ ∅	‘+ $\rightarrow$ ∩	‘C-e $\rightarrow$ exp	‘c 字母键 $\rightarrow$ 花体字母		

### 2.3.4 大文件辅助

AucTeX+Emacs对分成几部分的L<sup>A</sup>T<sub>E</sub>X源文件支持很好，可以把文件头尾作为一个文件，如mypap.tex，而其中的内容作为若干个文件，如mypap1-inc.tex，mypap2-inc.tex等。在主文件(master file) mypap.tex中内容的位置写命令`\include{mypap1-inc}`等可以包含入实际内容。一定注意`\include`命令里面的文件名要省略.tex后缀。

AucTeX的好处在于：

- 即使内容处于不同文件中，编译查错仍可自动找到错误位置。
- Yap的DVI反向查找可以正确找到子文件中相应位置。
- 用“C-c C-c”命令编译时即使当前文件是子文件实际编译的仍是主文件。这有可能需要自己在子文件末尾的地方加上类似如下的注释：

```
%%% Local Variables:
%%% mode: latex
%%% TeX-master: "mypap"
%%% End:
```

## 2.4 用reftex作引用

Emacs对.tex文件可以自动启用reftex支持，reftex用来帮助查看文章的章节结果和辅助管理引用，如公式、图、表等。如果编辑在编辑.tex文件时，按C-=可以显示当前的目录结构，把光标移动到目录中任一位置可以移动到源文件中该章节位置。在目录buffer中用l命令可以显示当前定义的引用，r命令可以重新扫描目录结构。在源文件中用C-)可以提示输入引用(`\ref{...}`)，选e可以提示公式的引用，在目录中选取一个引用，选哪个公式的引用时该公式的源程序会显示在状态行。

## 第三章

# L<sup>A</sup>T<sub>E</sub>X提高

### 3.1 编译结果

L<sup>A</sup>T<sub>E</sub>X编译的结果有:

- PDF, 是一种不依赖于操作系统和语言环境的文档格式, 一般不可编辑, 在任何操作系统和语言环境都可以不变样显示和打印。
- Postscript(PS), 和PDF类似, 因为现在PDF的浏览程序更好用所以PS格式逐渐被PDF格式取代。
- DVI格式, 是T<sub>E</sub>X特有的不依赖于显示系统的输出格式, 编译为DVI比编译为PDF要快一些, 能与源文件正向、反向互查位置, 所以在写草稿时可以用。最终稿一般输出为PDF。
- HTML格式, 这是网上的文章用的格式, 显示于网络浏览器中。

编辑器TeXWorks编译产生PDF文件, 可以在源文件和结果之间互相查找。  
WinEdt提供了快捷图标可以:

- 编译为DVI, 可以在源文件和结果之间互相查找;
- 直接编译为PDF(用PDFL<sup>A</sup>T<sub>E</sub>X);
- DVI转换为PS;
- DVI转换为PDF;
- PS转换为PDF。

Emacs+AucTeX提供了菜单和快捷图标可以:

- 选择编译为DVI还是PDF;
- 编译, 根据选择结果编译成DVI或PDF;
- 清除辅助文件;
- 用“Command—Other”菜单, 可以运行dvips 文件名命令, 把DVI转换为PS; 运行dvi<sub>ps</sub>dfm 文件名命令, 把DVI转换为PDF; 把L<sup>A</sup>T<sub>E</sub>X源文件转换为HTML文件。

## 3.2 带书签的PDF文件

用L<sup>A</sup>T<sub>E</sub>X的Hyperref包可以生成带书签和超链接的PDF文件。这样的PDF文件在浏览时可以在左窗格显示文件的目录结构，单击某一章节可以迅速跳到该处。文中的交叉引用如图表、章节、文献等都可以通过链接直接跳到定义处。

使用Hyperref包等功能十分容易，只要在引言(preamble)部分加上

```
\usepackage[pdftex]{hyperref}
```

则用PDFLaTeX生成的PDF文件自动带有书签和超链接。但是，如果我们使用CJK中文，则需要加不同的选项：

```
\usepackage[pdftex,CJKbookmarks=true,colorlinks=true]{hyperref}
```

即便如此生成的PDF中导航栏仍有可能显示乱码，为此，可以用如下的编译步骤：

```
pdflatex mypap
pdflatex mypap
gbk2uni mypap.out
pdflatex mypap
```

其中gbk2uni把书签编码转换为UTF-8 Unicode。在文件首次编译时可能有提示错误，只要回答以Q命令让其继续编译，然后再编译一遍就可以了。

Hyperref包提供的`\hypertarget{name}{text}`命令可以定义一个文内超链接，可以用`\hyperlink{name}{text}`来引用。

## 3.3 版式设置

用

```
\pagestyle{plain}
```

产生只有页号的页。用

```
\pagestyle{headings}
```

产生有页眉且页码在页眉的页面。

为了设置版心宽度和左右边空，用如

```
\setlength{\textwidth}{12.1cm}
\setlength{\oddsidemargin}{2.46cm}
\setlength{\evensidemargin}{1.36cm}
```

设置了版心宽为12.1厘米，纸张内侧边界为 $1\text{in}+2.46\text{cm}=5\text{cm}$ ，外侧边界为 $1\text{in}+1.36\text{cm}=3.9\text{cm}$ ，总共宽21.0cm(A4纸宽度)。用这样的命令设置边空要加上缺省的1英寸(2.54厘米)。A4纸高度为29.7厘米，可以用`\textheight`和`\topmargin`设置版心高度，注意上边空也要加上缺省的1英寸。

如果要设置自己定义的页眉和页脚，可以使用fancyhdr包。

```
\usepackage{fancyhdr}
\pagestyle{fancy}
```

引入该包并声明使用其版式。然后，在后面的引言或正文中(如使用CJK需要在进入CJK环境后)用\fancyhead定义页眉，用\fancyfoot定义页脚，重定义\headrulewidth和\footrulewidth可以要求分隔线。例如：

```
\fancyhf{} % 清除原有设置
\fancyhead[RE]{\footnotesize\leftmark} % 左内侧
\fancyhead[LO]{\footnotesize\rightmark} % 右内侧
\fancyhead[LE,RO]{\thepage} % 左、右外侧
\fancyfoot[LE]{\footnotesize\CJKfamily{kai}\LaTeX排版心得} % 左外侧
\fancyfoot[LO]{\footnotesize\CJKfamily{fs}编著：李东风} % 右内侧
\fancyfoot[RO]{\scriptsize %
http://www.math.pku.edu.cn/teachers/lidf/index.html} % 右外侧
\renewcommand\headrulewidth{0.4pt} % 页眉分隔线
\renewcommand\footrulewidth{0pt} % 页脚分隔线
```

fancyhdr的页眉和页脚都分为左(L)中(C)右(R)三部分，并且左侧页(E)和右侧页可以分别定义。命令\leftmark标记章，\rightmark标记节(对article样式左边是节右边是小节)，由L<sup>A</sup>T<sub>E</sub>X自动生成，也可以自己用\markboth定义，如

```
\chapter*{附录\markboth{附录}{}}
```

\的第一个参数是左页眉要标的内容，第二个参数是右页眉要标的。因为用星号格式的\chapter和\section命令定义的章节不自动定义页眉所以需要时要自己调用\markboth。

有时我们需要比12pt更大的字体。这可以用scalefont宏包实现，只要引用此宏包然后在正文内使用\scalefont{倍数}就可以改变后面字体的倍数，如\scalefont{1.21}放大1.21倍，这时标准宽度文章行宽为23个汉字。放大1.44有19个汉字，放大1.73有16个汉字，放大2.07有13个汉字。也可以要求缩小。这样放大不能保证章节标题大小也相应放大。

也可以使用CJKfont宏包指定汉字字体大小，比如用\zihao[5]设定五号字体。

### 3.4 大文件处理

有一些技巧可以处理大的文件。

- 可以把文件头、尾写成一个主控文档main.tex，然后在文档内部使用\include{文件名}的方法插入真正的内容文件。注意文件名不要包括.tex扩展名。 \include{会在插入内容前后换页。
- 如果有多个文件插入，还可以在前言(preamble)部分加一个\includeonly{文件清单}命令要求只插入列出的部分文件。如果没有\includeonly命令则正文中所有用\include命令插入的文件都得到处理，如果有\includeonly命令则只处理指定的文件，但是页码、索引等仍可以用到全文的信息。AucTeX+Emacs可以很好地支持多文件源程序，可以正确显示错误位置、Yap DVI反查。

- 主文件可以有几个，共享相同的内容子文件，可以适应不同的设置或输出需要。这样把内容和头尾分开有明显的好处：内容只需要一份，可以与几个不同的主控文件配合，比如生成DVI用的主控文件，生成PDF用的主控文件，生成课堂演示PDF的主控文件，CCT用的主控文件（是的，这种方法可以同时使用CJK和CCT处理同一文章）。
- 为了临时注释掉一部分内容，可以用`\iffalse`开头用`\fi`结尾把该部分包裹起来。为了容易找到这些注释标志可以在其附近加上若干个分号注释符。

### 3.5 如何同时应用CJK和CCT

CCT排版结果更符合中文排版习惯，而且有些刊物投稿要求使用CCT，所以我们经常需要把文章写成CCT格式。新版(0.6.0以上)的CCT已经支持CJK，生成的DVI不再需要补丁(patchdvi)，可以兼容标准的 $\TeX$ 和 $\LaTeX$ 包，也能生成高精度的PS和PDF结果。

CCT的一个大致的框架为：

```
\documentclass[a4paper,CJK]{cctbook}
%% 需要的包用\usepackage调入
\usepackage[CJKbookmarks=true,colorlinks=true]{hyperref}

\begin{document}
\include{ch01}
\include{ch02}
\end{document}
```

另一种常用的版式是cctart，即article的CCT版本。

为了同一内容文件兼容于CJK和CCT之间，只要写一个CJK的主控文件和一个CCT的主控文件，而内容放在单独的文件中用`\include{文件名}`调入。

要注意内容文件中改变字体的命令应该定义在主导文件中而不应该直接使用CJK的字体命令。在CJK主导文件中我们定义

```
\newcommand{\songti}{\CJKfamily{song}}
\newcommand{\fangsong}{\CJKfamily{fs}}
\newcommand{\kaishu}{\CJKfamily{kai}}
\newcommand{\heiti}{\CJKfamily{hei}}
```

这样我们就可以在内容文件中用`\heiti`等命令来改变汉字字体了。另外内容文件中也不要直接用CCT的字号命令(`\zihao{5}`之类) 而应该使用 $\LaTeX$ 标准的`\large`命令等。

### 3.6 插入图形

$\LaTeX$ 插图主要用`graphicx`包，它可以用`\includegraphics`命令插入一个图形。可以直接插入到段落中相当于一个字符或段内公式，也可以放在 $\LaTeX$ 的`figure`环境内作为浮动图形。放在`figure`环境内时一般用`center`环境把图形居中。

编译成DVI时可以插入的图形主要是EPS图形。一般我们生成的单页的PS图形都可以看成EPS图形，对某些不能正常计算边框的PS图形可以在新版

的gsview中打开然后使用其中的“PS to EPS”菜单。要把其它格式的图形转换为EPS图形有多种方法：可以用GIMP、Photoshop等图象处理图形进行转换，也可以安装一个支持PS的打印机驱动程序利用其打印到文件的功能把图形打印成PS文件，等等。

用PDFLaTeX编译时可以支持PDF、PNG、JPEG、GIF等格式的图形。为了把PS或EPS文件转换为PDF可以用新版的gsview中的转换功能。

注意我们插图比较理想的格式是PDF和EPS，因为这两种格式支持矢量图形，在放大时不会失真。如果原来的图形就是点阵格式如PNG、JPEG、GIF就不必转换为PDF了。

`\includegraphics`命令的可选参数主要是`width=`和`height=`，如果希望保持原图的高宽比就只要指定`width`和`height`之一。注意比较大的插图一定不要设定超过行宽的宽度，否则会产生很多多余的空白。

为了适应同时用L<sup>A</sup>T<sub>E</sub>X和PDFLaTeX编译的需要，L<sup>A</sup>T<sub>E</sub>X插入EPS图形而PDFLaTeX插入PDF图形，可以使用两个主文件`maind.tex`和`mainp.tex`分别用来生成DVI和PDF，使用`\include`命令调用相同的内容子文件。

在`maind.tex`的文件头引入

```
\usepackage{graphicx} % 用来支持DVI中插图
\usepackage{picins}    % 插入靠边的图
\usepackage[CJKbookmarks=true,%
  colorlinks=true]{hyperref} % 加超链接
\DeclareGraphicsExtensions{.eps}
```

在`mainp.tex`的文件头引入

```
\usepackage[pdftex]{graphicx} % 用来支持PDF中插图
\usepackage{picins}            % 插入靠边的图
\usepackage[pdftex,CJKbookmarks=true,%
  colorlinks=true]{hyperref} % 加超链接
\DeclareGraphicsExtensions{.pdf,.jpg,.png}
```

然后为了在L<sup>A</sup>T<sub>E</sub>X编译时用EPS图，在PDFLaTeX编译时用PDF、PNG、JPG图，只要文件名相同而扩展名不同，在`\includegraphics`命令中只给出文件名而不指定扩展名即可。

如果我们有大量的图，不希望和L<sup>A</sup>T<sub>E</sub>X源文件放在同一目录，可以把图放在一个`figs`子目录，假设还有一个`grs`子目录，只要在主控文件前言(preamble)部分加上

```
\graphicspath{{figs/}{grs/}}
```

就可以自动从这两个目录中查找图形来插入。

如果不需要生成带导航窗格的PDF，也可以用`dvipdfm`命令从DVI直接生成PDF，这样图形只要有EPS图形就够了，不需要同时拥有EPS和PDF两种格式的图形。

图3.1是插入单个浮动图形的例子。用到的命令如下：

```
\begin{figure}[htbp]
\centering
\includegraphics[width=\textwidth]{img01}
\caption{插入单个浮动图的例子\label{fig:figfloat1}}
\end{figure}
```



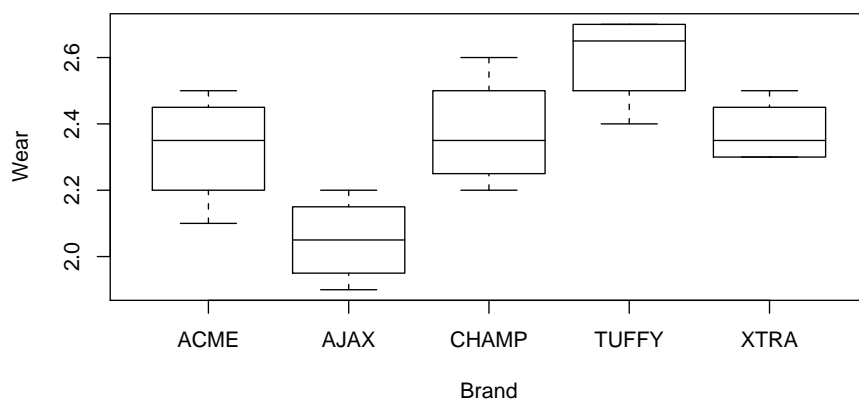


图 3.1: 插入单个浮动图的例子

浮动图形环境的选项h表示当前位置, t为top, b为bottom, p为单独只有图形的一页。注意\label放在\caption内部比较保险。

如果浮动图形位置不满意可以把插图命令在正文中前后调整。

图3.2为边上插入图形的一个例子。用了特殊的picins包, 这个包主要提供了不是浮动而是在当前段左边或右边插图的能力, 图形还可以有各种边框。插图命令为\parpic, 可选参数[r]说明放在段右侧, 后面的必要参数为实际的插图, 我们用自定义的\img命令插入了图形。这样插图的说明标题要用\piccaption 放在实际插图命令\parpic前面才行。这个例子中的\pichskip指定图横向离开文本段多宽, 不是必须的。源程序如下:

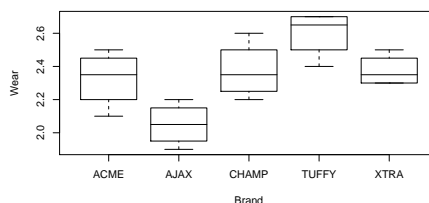


图 3.2: 用picins包插入的靠边图

```
\piccaption{用picins包插入的靠边图\label{fig:parpic}}
\parpic[r]{
\pichskip{6em}
\includegraphics[width=0.5\textwidth]{img01}}
```

这个插图方法还不太完善, 有时得到不好的结果。

另外一种插入边图的办法是使用floatflt宏包, 有时会覆盖在后面段落, 不理想。

对于比较小的图形我们希望把两个并排放在一起浮动, 但是有各自的图形标题, 可以用下例的方法, 结果见图3.3和图3.4:

```
\begin{figure}[htbp]
\begin{minipage}[t]{0.48\linewidth}
```

```

\centering
\includegraphics[width=\textwidth]{img01}
\caption{这是第一个图\label{fig:float2-1}}
\end{minipage}
\hfill
\begin{minipage}[t]{0.48\linewidth}
\centering
\includegraphics[width=\textwidth]{img02}
\caption{这是第二个图\label{fig:float2-2}}
\end{minipage}
\end{figure}

```

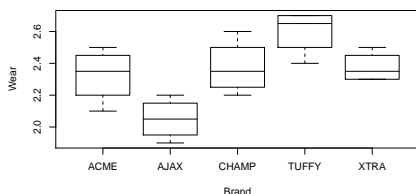


图 3.3: 这是第一个图

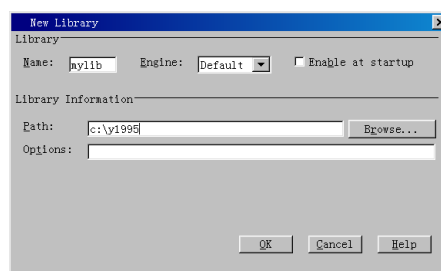


图 3.4: 这是第二个图

为了在Splus中生成横向的图形，可以用如下的postscript图形设备调用：

```
postscript("hor.eps", region=c(18.00, 7.92, 784.32, 593.76))
```

其中的region参数用于指定绘图区域的大小，其值为左下角和右上角坐标，如果需要可以放大此区域(比如过多图形元素重合以至于分辨不清时可以放大此区域)。

在R中用postscript设备生成EPS文件时可以用一个paper="special"选项直接指定纸张大小，单位是英寸(1in=2.54cm)：

```

postscript("hor.eps", paper="special",
           horizontal=FALSE, onefile=FALSE,
           width=20/2.54, height=16/2.54, horizontal=F)

```

也可以直接生成PDF图形：

```

pdf("hor.pdf",
    width=20/2.54, height=16/2.54, horizontal=FALSE)

```

这样得到的图在插入L<sup>A</sup>T<sub>E</sub>X时不需要旋转。

### 3.7 源程序排版

源程序排版使用fancyvrb宏包。用如下命令可以设置一些统一的选项：

L<sup>A</sup>T<sub>E</sub>X 排版心得

```
\RecustomVerbatimEnvironment{Verbatim}{Verbatim}{%
frame=single,framesep=5pt,xleftmargin=1cm,fontsize=\normalsize}
```

这样要求外边框(frame), 框内左边空5pt, 左外边界1cm, 字体用正常大小。Verbatim环境也可以单独加选项, 如`\begin{Verbatim}[fontsize=\small]`。

可以用`\VerbatimInput{file.ext}`导入一个外部文件内容, 显示为原样显示。此命令可以加选项`[frame=single]`要求画边框。

为了排一些程序模板, 用了tabular环境对齐。如下例:

```
\begin{tabular}{l1l}
&&\backslash
\multicolumn{3}{l}{SELECT (\emph{选择表达式});} \backslash
\quad & WHEN(\emph{值列表}) & \emph{语句}; & \backslash
\quad & WHEN(\emph{值列表}) & \emph{语句}; & \backslash
\quad & ..... & & \backslash
\quad & OTHERWISE & \emph{语句}; & \backslash
\multicolumn{3}{l}{END;} \backslash
&&
\end{tabular}
```

结果如下(非浮动):

```
SELECT (选择表达式);
  WHEN(值列表) 语句;
  WHEN(值列表) 语句;
  .....
  OTHERWISE   语句;
END;
```

其中在tabular环境选项的两列之间用了`@{ }`来取消表格环境原来比较宽的列间距。表格前后各多加了一个空行以区分上下文。不想对齐的部分用`\multicolumn`命令合并即可。

还可以使用tabbing环境进行对齐。如

```
\begin{minipage}{0.8\textwidth}
\begin{tabbing}
\quad \= if \= \kill

\> if(条件) \{ \backslash
\>\> 表达式组\ldots \ldots \backslash

\> \} else \{ \backslash
\>\> 表达式组\ldots \ldots \backslash

\> \} \backslash
\end{tabbing}
\end{minipage}
```

结果为:

```
if(条件) {  
    表达式组.....  
} else {  
    表达式组.....  
}
```

## 第四章

# 幻灯片演示制作

现在人们一般用Powerpoint作演示，但是如果公式比较多就不方便。另外，如果你原来的文章是用 $\text{\LaTeX}$ 打的那么用 $\text{\LaTeX}$ 做幻灯片就是最合理的选择。用beamer宏包或PDFSlides宏包可以制作课堂演示用的幻灯片。Beamer制作的幻灯片比较漂亮。

### 4.1 用Beamer制作幻灯片

beamer是一个制作PDF幻灯片的包。其优点包括：

- 自动生成导航菜单；
- 支持自动的分步显示(列表环境)和定制的分步显示；
- 带有不同的色彩、样式；
- 除支持生成屏幕显示的幻灯片PDF外，用相同的输入可以得到普通打印版PDF，且可以选择那些内容进入幻灯片，那些内容进入打印版；
- 支持CJK中文。

#### 4.1.1 基本的beamer幻灯片

安装好beamer包后，用beamer制作幻灯片PDF的基本框架为(文件slides.tex)：

```
\documentclass[CJK]{beamer}
\usepackage{CJK}
\usetheme{Warsaw}
\setbeamercovered{transparent}

\begin{document}
\begin{CJK*}{GBK}{song}

\title{演讲题目}
\subtitle{
小标题一\\
小标题二
```

```

}
\author{作者名\\
单位名}
\date{日期}
\begin{frame}
\maketitle
\end{frame}

\include{mypres}

\end{CJK*}
\end{document}

```

注意有CJK中文内容时要加CJK选项。其中，\usetheme指定一种样式，包括Antibes, Bergen, Berkeley, Berlin, Boadilla, Copenhagen, Darmstadt, Dresden, Frankfurt, Goettingen, Hannover, Ilmenau, JuanLesPins, Luebeck, Madrid, Malmoe, Marburg, Montpellier, PaloAlto, Pittsburgh, Rochester, Singapore, Szeged, Warsaw等，比如Warsaw生成在上面的菜单，Berkeley生成在左面的菜单，Goettingen生成在右面的菜单，Berlin生成可显示每小节页面数的菜单，Rochester生成没有菜单的样式。不同的样式有不同的颜色、字体、导航。

另外，beamer类与\AtBeginDocument命令冲突，所以只好在\begin{document}后面写\begin{CJK\*}{GBK}{song}\CJKtilde\CJKcaption{GB}，在\end{document}前面写\end{CJK\*}。

beamer中一面幻灯片用一个frame环境实现。在一个frame中，可以用\pause命令定制需要暂停显示的地方。对于itemize环境，还可以自动每个条目暂停，如：

```

\begin{frame}
\frametitle{内容标题}
\begin{itemize}[<+>]
\item 条目一
\item 条目二
\end{itemize}
\end{frame}

```

其中\frametitle加入屏幕标题。

虽然beamer支持把长的frame自动分成多个屏幕显示，但是拆分会造成一些问题，最好手工拆分长的frame。

### 4.1.2 编译速度

用PDFL<sup>A</sup>T<sub>E</sub>X编译一个长的演讲稿每次比较慢。为此，建议使用TeXWorks来编辑和编译，可以准备把主控文件slides.tex复制为drafts.tex，引入类beamer时加入draft选项，如

```
\documentclass[CJK,draft]{beamer}
```

然后，对正在编写的屏幕(frame)加[label=current]选项，如\begin{frame}[label=current]，则对该屏幕定义了一个标签current，然后在序言中用

```
\includeonlyframes{current}
```

只编译此屏幕。当一个屏幕调试成功后再注释此句以编译全文档。

### 4.1.3 同时生成幻灯片和打印版本

只要我们把主要内容放在单独的文件内，为幻灯片版本和可打印版本单独写主控文件就可以。beamer对可打印版的生成支持很好，在打印版中\frame等命令自动被转换含义。

打印版本的主要框架为：

```
\documentclass[a4paper,CJK]{cctbook}
\usepackage{beamerarticle}
\usepackage[pdftex,CJKbookmarks=true,colorlinks=true]{hyperref}

\begin{document}
\include{mypres}
\end{document}
```

也可以用cctart类或book、article版式。beamerarticle包的调入解决了正文中frame, pause等命令的解释问题。

如果有些内容只希望出现在幻灯片中，可以用

```
\only<presentation>{
只出现在幻灯片中。
}
```

如果希望只出现在可打印版本中只要用\only<article>。

## 4.2 用PDFSLIDE制作幻灯片

L<sup>A</sup>T<sub>E</sub>X另一种制作幻灯的方法是pdfslide宏包。只要在文件头用\usepackage[ams]{pdfslide}调用这个包，用PDFL<sup>A</sup>T<sub>E</sub>X编译就可以生成适合屏幕演示的PDF文件。源文件最好使用article类。生成的演示用回车翻页，带有导航图标。因为演示是基于屏幕的，在不同屏之间可以用\clearpage命令换页。

文章中可以用\color{颜色名}改变字体颜色。用\overlay{metablue.pdf}命令可以给屏幕加一个背景，这个背景可以用自己的PDF格式或JPEG格式图形代替。用\pagedissolve{Wipe /D 1}命令可以定义各页之间的转场效果。其中/D 1表示延迟1秒。

可用的转场效果为：

- \pagedissolve{Replace} 无转场效果。这是缺省取值。
- \pagedissolve{Split /D 1 /Dm /V /M /I} 垂直线向内拉幕
- \pagedissolve{Split /D 1 /Dm /V /M /O} 垂直线向外拉幕
- \pagedissolve{Split /D 1 /Dm /V /M /I} 水平线向内拉幕

- `\pagedissolve{Split /D 1 /Dm /H /M /O}` 水平线向外拉幕
- `\pagedissolve{Blinds /D 1 /Dm /H}` 多垂直线拉幕
- `\pagedissolve{Blinds /D 1 /Dm /H}` 多水平线拉幕
- `\pagedissolve{Box /D 1 /M /I}` 向内盒子
- `\pagedissolve{Box /D 1 /M /O}` 向外盒子
- `\pagedissolve{Wipe /D 1}`, `\pagedissolve{Wipe /D 1 /Di 90}` 向上推拉
- `\pagedissolve{Wipe /D 1 /Di 180}` 向左推拉
- `\pagedissolve{Wipe /D 1 /Di 270}` 向下推拉
- `\pagedissolve{Dissolve /D 1}` 渐化
- `\pagedissolve{Glitter /D 1 /Di 0}` 推拉渐化

下面是背景和转场效果的一个实例：

```
\overlay{bg151.jpg}
\pagedissolve{Wipe /D 0.5} % 向右推拉
```

其中bg151.jpg是我自己在图形程序中作的一个背景图片。注意背景不要喧宾夺主，否则对教学效果会起反面作用。

在文件头加上如下定义可以得到较悦目的项目列表：

```
\renewcommand{\labelitemi}{\textcolor{red}{\bullet}}
\renewcommand{\labelitemii}{\textcolor{yellow}{\star}}
\renewcommand{\labelitemiii}{\textcolor{magenta}{\ast}}
\renewcommand{\labelitemiv}{\textcolor{cyan}{\circ}}
```

如果必要的话还可以下载安装PPower4软件以生成逐步显现的页。

用PDFSlide包作的课堂演示文件没有目录导航功能。我们可以用Hyperef包提供的两个命令自己建立文内的超链接：`\hypertarget{链接名}{显示内容}`可以定义一个链接目的地，用“链接名”来访问，为了显示这个超链接，用`\hyperlink{链接名}{显示内容}`显示。

演示文件是以屏幕为单位的，需要自己分页。在生成课堂演示的主控文件中我们定义了：

```
\providecommand{\cls}{\clearpage}
```

而在另外一个主控文件中我们则定义

```
\providecommand{\cls}{\vskip 1cm}
```

这个主控文件可以生成课下参考用的讲义。

因为PDFLaTeX编译比较慢所以我们先编译成DVI成功后再编译成PDF，这样需要一个单独的生成DVI的主控文件。为了和结果屏幕大小一致，我们定义文本高度：

```
\setlength{\textheight}{10.6cm}
```



### 4.2.1 PPower4软件

为了实现逐步显示一页的各行的效果需要安装一个PPower4软件包。这是一个Java程序，所以需要先安装一个Java运行环境，如Java JRE 1.2。下载PPower4后把其中的 pp4sty.zip展开到 C:\CTEX\MikTeX\LocalTeXMF\tex\latex\ppower4中，把其中的 pp4p.jar复制到上述目录中，然后生成如下的批命令 PPower4.bat:

```
set basedir="C:\Program Files\JavaSoft\jre\1.2\lib"
set javaccommand="C:\Program Files\JavaSoft\jre\1.2\bin\java"
%javaccommand% -jar C:\CTEX\MikTeX\LocalTeXMF\tex\latex\
                    ppower4\pp4p.jar %1 %2 %3 %4 %5
```

把它放在 C:\CTEX\MikTeX\TeXMF\miktex\bin中。注意批命令中需要代以你自己的Java运行环境的路径和程序名。展开 pp4sty.zip中的文件到 CTEX目录中后要更新 MikTeX的文件名数据库。

PPower4主要是提供了一个pause.sty包和\pause命令。为使用其功能需要在L<sup>A</sup>T<sub>E</sub>X文件头引入pause包。在需要等待回车继续显示的地方加上\pause命令。然后对L<sup>A</sup>T<sub>E</sub>X源文件(如test.tex)用PDFLATEX命令编译，然后运行PPower4:

```
ppower4 test.pdf test-scr.pdf
```

则test-scr.pdf为生成的结果。

PPower4除了提供分步显示支持还提供了单色、渐变色背景，转场动画效果等。

## 第五章

# 其他技巧

### 5.1 Linux中 $\text{\LaTeX}$ 和中文的使用

#### 5.1.1 Linux下CJK和中文字体的安装

在Linux操作系统中一般已经安装了tetex系统,这是一套很完整的 $\text{\TeX}$ 系统.如果你用的是国内的Linux套装则CJK中文支持可能也已经安装好,比如我用过的Turbo Linux 6.0;如果你用的是Redhat等国外产品,则可能需要自己安装CJK包和中文Type1字库.

我在Redhat 9中采用的办法是利用已经安装好的Linux tetex和MS Windows下的CTEX中文套装.

tetex的主TEXMF目录安装在了/usr/share/texmf中而LOCALTEXMF目录安装在了/usr/local/share/texmf中.这两个位置可以用如下命令查询:

```
kpsewhich -expand-var \[extract_itex]TEXMF
kpsewhich -expand-var \[extract_itex]TEXMFLOCAL
```

为安装中文支持,把CTEX中的LocalTeXMF目录复制为/usr/local/share/texmf.复制的办法之一是如下命令:

```
su
(input root password here)
tar cvf ~/localtexmf.tar /mnt/C/CTEX/LocalTeXMF
cd /usr/local/share
tar xvf ~/localtexmf.tar
mv LocalTeXMF texmf
```

CTEX套装中的CJK.sty, CJK.enc文件名大小写有问题,可以把原来小写的文件名复制为大写.某些CTEX或ChinaTeX系统中的dvips/config/config.ps中有一行

```
p +lw35extra.mapfile
```

但在 $\text{\TeX}$ 中会找不到这个文件而在xdvi步骤出错,可以用百分号注释掉这一行,或者从/usr/share/texmf/dvips/config中复制一个lw35extra-adobe-bi.map为这个文件名.

最后用

```
texhash
```

命令更新文件名数据库。这时CJK汉字就可以用了, CTEX的字体也自动得到支持。dvips, dvipdf, pdflatex等也可以正常使用, 不过编译PS和PDF时可能有一些字体找不到的警告, 但是编译结果并无问题。

在Linux中编辑 $\text{\LaTeX}$ 最好使用Emacs编辑器, 可能需要自己在`/usr/share/emacs/site-lisp/`中安装AucTeX包并对`site-start.el`文件中加上

```
(load "tex-site")
```

一行启动AucTeX。在编辑.tex文件时Emacs会有一个用于编辑和编译的LaTeX菜单和Command菜单。

### 5.1.2 Redhat Linux 9的中文双拼输入问题

Linux 9中已经安装了miniChinput输入法, 包括智能拼音、五笔等输入法。但是其中的双拼不能自定义双拼方案, 所以我查找网上找到了小企鹅中文输入法(<http://www.fcitx.org/>)。这个输入法支持五笔和拼音, 双拼可以自定义方案, 没有智能拼音。

参照小企鹅输入网站的指示很容易安装好此输入法。然后把双拼方案`/usr/share/fcitx/sp.dat`复制为`~/gWubi/sp.dat`, 修改此文件为自己熟悉的方案(注意: 在MS Windows下的紫光输入法3.0中有几个双拼方案文件可以参考)。修改`~/gWubi/config`和`~/gWubi/profile`选择输入法2, 选择双拼输入=1, 并适当选择自己的喜好。按`Ctrl+Alt+Backspace`后重新进入用户就可以使用双拼输入了。

## 5.2 其他问题

### 5.2.1 关于英文系统和中文系统的兼容性问题

我们常用的CTEX安装了一些关于汉字系统的设置, 在少数情况下编译一些比较复杂的纯英文文件会因为不兼容而出错。这时我们可以采用更改MikTeX的目录次序的办法: 在安装好的MikTeX程序组中找到“MikTeX Options”程序, 启动后, 有一个“File name database—Refresh Now”选项, 这个选项是在安装了新的包后应该运行的。点这个程序的Roots选项卡, 其中有两个目录, `\CTeX\localtexmf`和`\CTeX\texmf`, 前者保存了中文系统和附加的包, 后者为MikTeX安装的主要 $\text{\TeX}$ 系统。这两个目录有次序, 当`localtexmf`在前时中文系统为优先考虑, 当`texmf`在前时英文系统为有限考虑。所以为了编译不兼容的英文文件可以修改这个次序, 工作完成后再改回来。

### 5.2.2 一些小的问题

#### 如何找到一些符号

在网上搜索一个`symbols-a4.pdf`文件, 这个文件中罗列了很多符号的用法。比如, 空心的阿拉伯数字如何打, 黑体希腊字母用什么样的命令。

### 如何定义公式中一些类似于ln的命令

比如，Cov在公式中可以用

```
\providecommand{\Cov}{\ensuremath{\mathrm{Cov}}}
```

定义后用 $\backslash\Cov$ 调用。

为了生成如 $\lim_{n \rightarrow \infty}$ 这样可以带上下标的命令，可以定义如

```
\providecommand{\argmin}{\mathop{\mathrm{arg\,},\min}}
```

如

```
$$
\argmin_{x \in A} f(x)
$$
```

生成

$$\arg \min_{x \in A} f(x)$$

### 如何生成如 $\mathscr{F}$ 这样的花体字

需要引入mathrsfs包然后用如 $\backslash\mathscr{F}$ 。

### 如何生成如 $\mathbb{R}$ 这样的空心字

只要引入了amsmath包或amssymb包就可以用 $\backslash\mathbb{R}$ 生成。

### 如何生成如 $\mathbb{1}$ 这样的空心数字

可以在引入dsfont包:

```
\usepackage[sans]{dsfont}
```

后用 $\backslash\mathds{1}$ 实现。

### 如何临时增加一页高度

有时 $\text{\LaTeX}$ 会生成一页只有一两行的几乎空的页，有时插入一个较高的图表导致前面出现一个空页。为避免这种情况，可以在内容的源代码中加入

```
\enlargethispage{2\baselineskip}
```

就可以放大内容的一页使其能容纳较多内容而不至于换页。

### 如何修改enumerate环境的序号

为了自己设计enumerate环境的需要，可以在环境内部开始的地方重新定义 $\backslash\labelenumi$ ,  $\backslash\labelenumii$  等命令。比如

```
\renewcommand{\labelenumi}{\textbf{\arabic{enumi}.}}
\renewcommand{\labelenumii}{(\alph{enumi})\quad}
```

等。也可以在 $\backslash\item$ 命令的可选参数中直接给出序号，如 $\backslash\item[一、]$ 。

### 如何避免浮动图表不必要地占据整页

L<sup>A</sup>T<sub>E</sub>X关于图表等浮动体的放置可以由htbp等建议, 在`\begin{figure}[htbp]`中的h表示建议优先放在当前位置, 其次放在页首, 再其次放在页尾, 实在不行允许单独放在一页。在具体实施时, 图表的放置位置有一些默认的限制, 比如图文混排时如果文字部分过少则把图单独放在一页。这些限制可以用如下方法进行调整:

```
\renewcommand{\textfraction}{0.10}  
\renewcommand{\topfraction}{0.90}  
\renewcommand{\bottomfraction}{0.70}  
\renewcommand{\floatpagefraction}{0.90}
```

其中textfraction的设置把混排时最少文本降低到10%, topfraction的设置允许页首的图占整页的90%(图高超过这个比例时则只能单独放在一页), bottomfraction的设置允许页尾的图占整页的70% (图高超过这个比例时不允许放在也尾), floatpagefraction的设置要求图高大于90%时才单独放在一页。

### 如何制作横向表格

太宽的表格可以横过来排印, 这样的表格要单独占一页。L<sup>A</sup>T<sub>E</sub>X的lscap包可以提供横向排版功能, lscap提供了一个landscape环境, 在此环境内排版是横向的。