

L^AT_EX排版心得

李东风

2005年12月24日

目录

1	CJK汉字系统和Emacs的使用	1
1.1	CJK的优点	1
1.2	CTEX安装设置	1
1.3	用Emacs和AucTeX编辑L ^A T _E X文件	2
1.3.1	安装与设置	2
1.3.2	Emacs的基本编辑功能	3
1.3.3	L ^A T _E X编辑编译	4
1.3.4	AUC _T E _X 设置	6
1.3.5	用reftex作引用	6
1.4	CJK T _E X源文件示例	6
2	L^AT_EX高级应用	9
2.1	PS和PDF文件的生成	9
2.2	带书签的PDF文件	10
2.3	版式设置	10
2.4	大文件处理	12
2.5	如何同时应用CJK和CCT	12
2.6	插入图形	13
2.7	源程序排版	16
3	幻灯片演示制作	18
3.1	用Beamer制作幻灯片	18
3.1.1	基本的beamer幻灯片	18
3.1.2	同时生成幻灯片和打印版本	19
3.2	用PDFSlide制作幻灯片	20
3.2.1	PPower4软件	21
4	其他技巧	22
4.1	Linux中L ^A T _E X和中文的使用	22
4.1.1	Linux下CJK和中文字体的安装	22
4.1.2	Redhat Linux 9的中文双拼输入问题	23
4.2	其他问题	23
4.2.1	关于英文系统和中文系统的兼容性问题	23
4.2.2	一些小的问题	23
4.2.3	如何临时增加一页高度	24
4.2.4	如何避免浮动图表不必要地占据整页	24

第一章 CJK汉字系统和Emacs的使用

1.1 CJK的优点

本文保存了用 \LaTeX 和 \PDFLaTeX 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 \LaTeX 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 \LaTeX 源程序。

我们在 \LaTeX 下写中文文章或书用的汉字支持主要有两种：一种是CCT，使用方便，排版符合中文排版习惯，利用CJK选项也可以支持现在的标准 \TeX 和 \LaTeX 工具，但是CCT只提供了几个自定义的版式(class)；另一种是CJK，它支持标准的 \TeX 和 \LaTeX 工具，可以生成高质量的PS文件和PDF文件，生成的PDF还支持书签导航。适当设置后用CJK生成的PS文件和PDF文件可以任意放大而不失真。

1.2 CTEX安装设置

我用过的是<http://www.ctex.org/>下载的CTEX2.4套装和配套的PS Type 1字体。其中包含了MikTeX 2.4和CJK, CCT, 天元 \TeX 。

注意：如果需要用Emacs作编辑、编译环境，应该先安装Emacs。

为安装CTEX 2.4，只要先下载CTEX套装和CTeX-Fonts套装，依次运行安装就可以了。假设安装在了C:\CTeX目录中。这个版本开始可以进行宏包按需下载安装，所以可以先安装一个比较小的宏包集。安装了CTEX套装和字体套装后，应该看一看安装后的 $\backslash\text{CTeX}\backslash\text{localtexmf}\backslash\text{dvips}\backslash\text{config}$ 或 $\backslash\text{CTeX}\backslash\text{texmf}\backslash\text{dvips}\backslash\text{config}$ 中的config.ps文件，其中应该有一行

```
p +psfonts_t1.map
```

这样保证生成的PS文件质量比较高。

CTEX套装所带的WinEdt功能很强，比如插入各种符号、编译按钮、彩色加亮显示等。但是WinEdt在显示汉字时很难看，所以我自己原来是用UltraEdit来编辑和编译。UltraEdit比WinEdt的 \LaTeX 功能稍弱，但是它也支持彩色加亮显示，并支持不同程序语言，可以自定义工具按钮和快捷键。在得到新的Emacs 21.2后我用Emacs和AucTeX来编辑 \LaTeX 文件，Emacs功能除可以彩色加亮显示、集成编译、从DVI反查源文件外还有其他强大功能。现在我用的是Emacs 21.3版本。

用CJK写的文件用`latex`命令就可以生成标准的DVI文件，用CTeX套装中的Yap程序可以查看DVI文件。用`dvips`命令可以生成PS文件，用`pdflatex`命令可以生成PDF文件，用`dvipdfm`命令可以从DVI生成PDF文件，用`dvipdfmx`命令可以从DVI生成PDF文件且可以从PDF中复制文本内容。

1.3 用Emacs和AucTeX编辑L^AT_EX文件

在Linux及其他Unix类操作系统上广泛使用的Emacs编辑器是一个功能极为强大的集成工作环境，在安装了AucTeX包后可以作为一个集成的L^AT_EX编辑、编译、浏览、打印系统(需要MikTeX或fpTeX作为后台支持)。在编辑L^AT_EX文件时有很好的语法彩色加亮功能，也支持源文件位置到DVI位置的正、反向对应。原来的Emacs移植到MS Windows上的版本对中文输入的支持较困难，但是新的版本(21.2和21.3)在MS Windows下可以直接使用Windows自己的输入法并使用Windows的字体。

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。

1.3.1 安装与设置

1. 从网上下载`emacs-21.3-fullbin-i386.tar.gz`然后把它解压到一个目录(注意压缩包中原来有目录，可以在解压后对目录改名为`emacs`)，比如C:\emacs。然后运行其中的`addpm.exe`可以在Windows的开始菜单增加Emacs的程序项。
2. 然后，需要下载一个`gnuserv.zip`文件，将其中的`gnuserv.el`展开到Emacs的`site-lisp`子目录中。把其中的四个EXE文件移动到Emacs的`bin`子目录中。
3. 为了更好编辑L^AT_EX文件需要从网上下载一个“`auctex-11.14.zip`”文件，把它解压到emacs安装目录的`site-lisp\auCTEX`下面(目录名不要有版本号)，
4. 在emacs的`site-lisp`目录中增加一个`site-start.el`文件，其中包含行

```
(load "tex-site")
(require 'gnuserv)
(gnuserv-start)
(setq gnuserv-frame (selected-frame))
(add-hook 'LaTeX-mode-hook 'turn-on-reftex)
(setq TeX-command-list
  (list (list "LaTeX" "%l \\nonstopmode\\input{%t}"
             'TeX-run-LaTeX nil t)
        (list "View" "%v" 'TeX-run-discard nil nil)
        (list "DVI->PS" "dvips %d -o %f "
             'TeX-run-command t nil)
        (list "View PS" "start %f"
             'TeX-run-command nil nil)
        (list "LaTeX PDF"
             "pdflatex \\nonstopmode\\input{%t}"
             'TeX-run-LaTeX nil t)
        (list "DVI->PDF" "dvipdfmx %d"
             'TeX-run-command t nil)
        (list "View PDF" "start %s.pdf"
```

```

      'TeX-run-command nil nil)
    (list "BibTeX" "bibtex %s"
      'TeX-run-BibTeX nil nil)
    (list "Index" "makeindex %s"
      'TeX-run-command nil t)
    (list "Other" "" 'TeX-run-command t 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 'chinese-iso-8bit-dos)
(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L^AT_EX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。

5. 运行CTEX中的DVI显示程序Yap，选“View — Options”菜单再选“Inverse search”部分，选Program为“Gnu Emacs (Single instance)”，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，可以直接在Command栏中输入以上命令，当然要写对路径。

1.3.2 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。

表 1.1: 光标移动及删除

操作	光标键	组合键	说明
右	右	C-f	f orward
左	左	C-b	b ackward
下	下	C-n	n ext
上	上	C-p	p revious
行首	Home	C-a	
行尾	End	C-e	e 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	k ill
取消刚才的删除	无	C-y	y ank
粘贴以前的删除	无	M-y 若干次	
恢复	无	C-x u	u ndo

Emacs的编辑很有特点，表1.1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中选中一块文本，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要编辑的文件。

1.3.3 L^AT_EX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^AT_EX编辑状态。增加了菜单LaTeX和Command，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L^AT_EX文件后用Command菜单中的Latex命令编译，如果有错误可以用“C-c ‘”键组合查看。用Command菜单中的View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。

LaTeX菜单和Command菜单中的功能大多可以用快捷键来实现。下面列出一些常用的功能。

编译 除了可以用Command菜单中的LaTeX命令或LaTeX PDF命令编译外还可以用“C-c C-c”键运行编译命令，在mini-buffer处提示要运行的程序，如果提示的缺省命令不合要求可以自己输入。编译过程中可以用“C-c C-l”查看命令的输出过程。如果编译有错误，可以用“C-c ‘”键翻看下一错误的源文件位置及错误信息。在用View命令打开Yap查看时，当前编译位置会在DVI中用一个小圈表示，在DVI中双击可以返回源文件中对应位置。

为了定义缺省编译命令，可以在文件末尾加上类似如下的注释，这样是用PDFLaTeX编译：

```
% Local Variables:
% TeX-command-default: "LaTeX PDF"
% End:
```

“C-c C-r”命令可以编译当前区域(region)。编译完后改用View命令显示。可以用这个命令测试公式是否正确，当前区域最好不要有汉字。

输入辅助 ● “C-c C-m”可以输入L^AT_EX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L^AT_EX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 $\texttt{\textbf{}}$ 。
- “C-c C-f C-e”产生 $\texttt{\emph{}}$ 。
- “C-c C-f C-t”产生 $\texttt{\texttt{}}$ 。
- “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。

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。

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：

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

大文件辅助 AucTeX+Emacs对分成几部分的L^AT_EX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:
```

1.3.4 AUCT_EX设置

可以在自己的C:\.emacs文件中加入：

```
(setq TeX-parse-self t) ;; Enable parse on load
(setq TeX-auto-save t) ;; Enable save of parsed result
```

1.3.5 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。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。

1.4 CJK T_EX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
\begin{document}
\begin{CJK*}{GBK}{song}
\CJKtilde

\include{mypap-inc}

\end{CJK*}
\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。

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。

但是，这个简单的模板有一些问题：我们习惯用汉字称呼的章节名、图表等都用英文标签。为此我们需要作一些调整。在进入 CJK 环境后（`\begin{CJK*}`）和 `CJKTilde` 后面）插入如下的修改：

```
\renewcommand\contentsname{目录}
\renewcommand\listfigurename{插图目录}
\renewcommand\listtablename{表格目录}
\renewcommand\refname{文献}
\renewcommand\indexname{索引}
\renewcommand\figurename{图}
\renewcommand\tablename{表}
\renewcommand\abstractname{摘要}
\renewcommand\partname{部分}
\renewcommand\appendixname{附录}
\renewcommand\today{\number\year年\number\month月\number\day日}
\providecommand{\CJKnumber}[1]{\ifcase#1\or{一}\or{二}\or{三}%
\or{四}\or{五}\or{六}\or{七}\or{八}\or{九}\or{十}\fi}
```

这样对于 article 版式我们可以得到基本正常的中文文章。如果用 book 版式，就需要重新定义章的显示：

```
\makeatletter % '@ is now a normal "letter" for TeX
\def\chaptermark#1{%
  \markboth {\MakeUppercase{%
    \ifnum \c@secnumdepth >\m@ne
      \ifmainmatter
        第\CJKnumber{\thechapter}章 \ %
      \fi
    \fi
  #1}}{}}%
\def\sectionmark#1{%
  \markright {\MakeUppercase{%
    \ifnum \c@secnumdepth >\z@
      \S\thesection \ %
    \fi
  #1}}}%
\def\@makechapterhead#1{%
  \vspace*{50\p@}%
  {\parindent \z@ \raggedright \normalfont
    \ifnum \c@secnumdepth >\m@ne
```

```

\if@mainmatter
\huge\bfseries 第\CJKnumber{\thechapter}章\ \
%\par\nobreak
%\vskip 20\p@
\fi
\fi
%\interlinepenalty\@M
\Huge\bfseries #1\par\nobreak
\vskip 40\p@
}}
\makeatother % '@' is restored as a "non-letter"
% character for TeX

```

这样显示的章如：“第一章 SAS入门”，节如“1.1 介绍”。这个小修改还存在问题，与hyperref包不太兼容。另外还要修改文献的名字，并去掉摘要：

```

\renewcommand\bibname{文献}
%\renewcommand\abstractname{摘要}

```

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\slippy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。

LaTeX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。

第二章 L^AT_EX 高级应用

2.1 PS和PDF文件的生成

对于L^AT_EX文件，包括使用了CJK的文件，我们要生成的结果一般是PS(PostScript)文件和PDF文件，这两种文件都可以在全世界自由地交换，不依赖于具体的操作系统和语言环境。

假设L^AT_EX源文件为art.tex，生成PS的步骤为：先用

```
latex art
```

命令编译为art.dvi。这个DVI文件可以用

```
yap art
```

来查看。从DVI 可以直接打印，但是我们要交流的话一般还是应该生成PS或PDF。

从DVI文件可以生成PS也可以生成PDF。为了从art.dvi生成PS文件，用

```
dvips art
```

这样得到art.ps。为了查看或打印PS文件系统中需要安装有gsview程序。事实上，PS文件可以直接打印到很多支持PS的激光打印机上。为了从art.dvi生成PDF，用

```
dvipdfm art
```

这样得到art.pdf。注意这一功能需要较新的CTEX或ChinaTeX，因为要把EPS格式的图形插入到PDF中需要一些辅助性的程序把EPS图形自动转换为PDF格式。如果把dvipdfm替换成dvipdfmx程序还可以生成能从中复制文本的PDF文件。

从PS文件也可以生成PDF，新版的gsview中提供了这样一个转换功能，转换结果可能比较粗糙。

要生成PDF也可以直接用pdflatex编译生成而不经DVI到PDF的转换。要把L^AT_EX源文件art.tex编译为PDF，只要用

```
pdflatex art
```

就可以得到art.pdf。

以上几种办法得到的PS文件和PDF文件在正确设置PS Type 1字体的条件下字体都很光滑，放大后没有锯齿。dvipdfmx缺省用的是True Type字体，效果也很好。

PDF和PS文件有什么差别呢？PDF实际是PS的一个子集，查看PS的程序也能查看PDF文件。但是PDF显示速度快，而且可以在其中方便地导航，交流起来和PS一样方便，所以我们可以尽量使用PDF文件。PS文件和PDF文件在对插入图形的支持方面有些差别，下文中将详细介绍。

2.2 带书签的PDF文件

用L^AT_EX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。

2.3 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 $\text{lin}+2.46\text{cm}=5\text{cm}$ ，外侧边界为 $\text{lin}+1.36\text{cm}=3.9\text{cm}$ ，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,R0]{\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[R0]{\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 \LaTeX 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。

2.4 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。
- 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。

2.5 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T_EX和L^AT_EX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[pdftex,CJKbookmarks=true,colorlinks=true]{hyperref}

\begin{document}
\include{sec01}
\include{sec02}
\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控文件中而不应该直接使用CJK的字体命令。在CJK主控文件中我们定义

```
\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。

2.6 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`环境内时一般用一个`\centering`命令把图形居中。

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 \LaTeX 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宽度超过行宽，否则会产生很多多余的空白。

为了适应同时用 \LaTeX 和 \LaTeX 编译的需要， \LaTeX 插入EPS图形而 \LaTeX 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
```

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
```

然后为了在 \LaTeX 编译时用EPS图，在 \LaTeX 编译时用PDF图，在`maind.tex`文件头作了如下定义：

```
\providecommand{\img}[2][]{%
  \includegraphics[#1]{\figureprefix #2.eps}
}
```

而在`mainp.tex`文件头作了如下定义：

```
\providecommand{\img}[2][]{
  \includegraphics[#1]{\figureprefix #2.pdf}
}
```

其中的#1参数是插图选项如width=0.5\linewidth, #2参数是插入的图形。命令\figureprefix是自己定义的一个图形文件名前缀, 如“高级技巧”一章所有图形用“img-tricks-018.pdf”的格式, 就可以定义

```
\providecommand{\figureprefix}{img-tricks-}
```

用\img插图例如

```
\img[width=0.5\linewidth]{018}
```

这时对maind.tex插入img-tricks-018.eps 而对mainp.tex插入img-tricks-018.pdf。分别情况仿照\img定义其他命令可以做到在maind.tex中插入EPS图而在PDF中插入PNG, JPEG或GIF图形。

另外, 如果不需要生成带导航窗格的PDF, 也可以用dvi2pdf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。

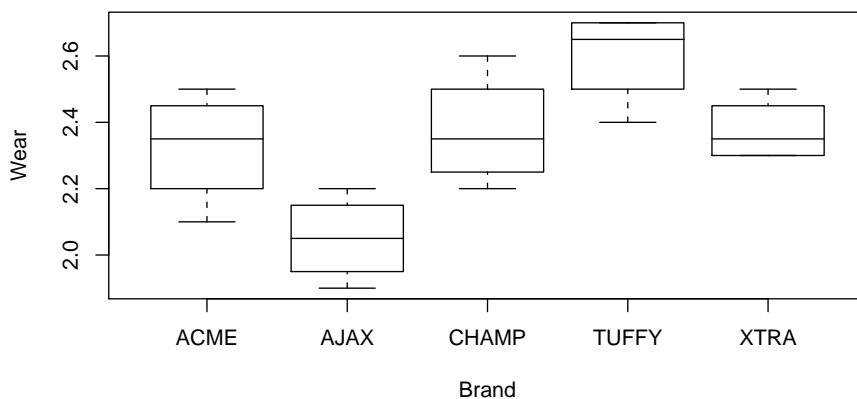


图 2.1: 插入单个浮动图的例子

图2.1是插入单个浮动图形的例子。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。用到的命令如下:

```
\begin{figure}[htbp]
\centering
\img[width=\textwidth]{018}
\caption{插入单个浮动图的例子\label{fig:figfloat1}}
\end{figure}
```


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2.2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：

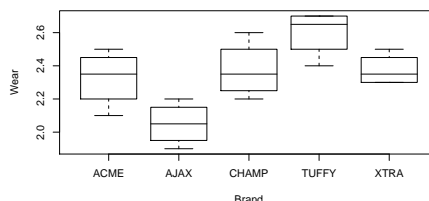


图 2.2: 用picins包插入的靠边图

```
\piccaption{用picins包插入的靠边图\label{fig:parpic}}
\parpic[r]{
\pichskip{6em}
\imgp[width=0.5\textwidth]{018}}
```

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2.3和图2.4:

```
\begin{figure}[htbp]
\begin{minipage}[t]{0.48\linewidth}
\centering
\imgp[width=\textwidth]{018}
\caption{这是第一个图\label{fig:float2-1}}
\end{minipage}
\hfill
\begin{minipage}[t]{0.48\linewidth}
\centering
\imgp[width=\textwidth]{018}
\caption{这是第二个图\label{fig:float2-2}}
\end{minipage}
\end{figure}
```

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)：

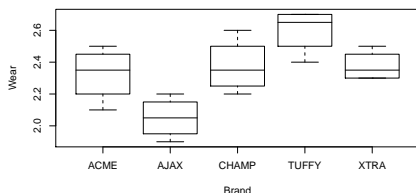


图 2.3: 这是第一个图

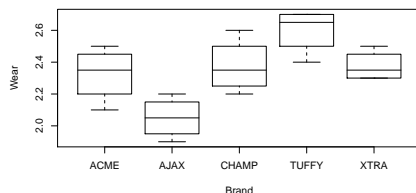


图 2.4: 这是第二个图

```
postscript("hor.eps", paper="special",
           width=12, height=8, horizontal=F)
```

也可以直接生成PDF图形:

```
pdf("hor.pdf",
    width=12, height=8, horizontal=F)
```

这样得到的图在插入L^AT_EX时不需要旋转。

如果我们有大量的图, 不希望和L^AT_EX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。

2.7 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:

```
\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]`。

为了排一些程序模板, 用了tabular环境对齐。如下例:

```
\begin{tabular}{l111}
&&\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 & \cdots & & \backslash
\quad & OTHERWISE & \emph{语句}; & \backslash
\multicolumn{3}{l}{END;} & \backslash
&&
\end{tabular}
```

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

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) \{ \\\
\\> 表达式组\ldots \ldots \\\
\\> \} else \{ \\\
\\> 表达式组\ldots \ldots \\\
\\> \} \\\
\end{tabbing}
\end{minipage}
```

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

第三章 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 \LaTeX 打的那么用 \LaTeX 做幻灯片就是最合理的选择。用PDF \LaTeX 宏包或beamer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。

3.1 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。

3.1.1 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：

```
\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}

```

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。

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。

3.1.2 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<presentation>`。

3.2 用PDFSlide制作幻灯片

L^AT_EX制作幻灯用到的是一个叫做pdfslide的宏包，CTEX和ChinaTeX中已经包含了。只要在文件头用`\usepackage[ams]{pdfslide}`调用这个包，用PDFL^AT_EX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}{\text{\bullet}}}
\renewcommand{\labelitemii}{\textcolor{yellow}{\text{\star}}}
\renewcommand{\labelitemiii}{\textcolor{magenta}{\text{\ast}}}
\renewcommand{\labelitemiv}{\textcolor{cyan}{\text{\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}
```

3.2.1 PPower4软件

为了实现逐步显示一页的各行效果需要安装一个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 \LaTeX 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 \LaTeX 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。

第四章 其他技巧

4.1 Linux中 \LaTeX 和中文的使用

4.1.1 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 \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 \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 \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.

4.1.2 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.

4.2 其他问题

4.2.1 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 \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.

4.2.2 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 \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 \mathscr{F} 。

如何生成如 \mathbb{R} 这样的空心字

只要引入了`amsfonts`包或`amssymb`包就可以用 \mathbb{R} 生成。

如何生成如 $\mathds{1}$ 这样的空心数字

可以在引入`dsfont`包:

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

后用 $\mathds{1}$ 实现。

4.2.3 如何临时增加一页高度

有时 \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。

4.2.4 如何避免浮动图表不必要地占据整页

\LaTeX 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页尾)，floatpagefraction的设置要求图高大于90%时才单独放在一页。