

# L<sup>A</sup>T<sub>E</sub>X科学文件处理软件入门 (修订版)

薛定宇

东北大学控制仿真研究中心

一九九七年五月

## 修 订 说 明

通过两年多的努力,作者欣喜地发现 L<sup>A</sup>T<sub>E</sub>X排版软件已成为控制仿真研究中心师生的最常用排版手段,这使得我们大大地提高了工作效率,并大大地改善了排版质量。在这里除了重新介绍 L<sup>A</sup>T<sub>E</sub>X的最常用排版命令以外,引入了很多新的排版细节处理的方法和技巧,诸如在文本中插入由其它软件(如 MATLAB)绘制的图形、PostScript 文件转换及打印、规范科学文献排版等,并介绍 L<sup>A</sup>T<sub>E</sub>X2e (以前为 209) 版的特色与使用,使得用户可以得出更好的、更正规的排版效果。

## 1 引 言

L<sup>A</sup>T<sub>E</sub>X的前身是 T<sub>E</sub>X,它是由美国著名的计算机专家 D.E. Knuth 教授研制成功的。早在 1977 年,当出版社要求 Knuth 教授对他六十年代的力作 The Arts of Computer Programming 出修订版时,他感到以手工的方式对原稿进行修改有种种不便,这时他就开始着手研制一个名为 T<sub>E</sub>X的高水平的计算机排版系统,这个系统的一个比较好的版本(1.3 版)于 1982 年研制成功<sup>[3]</sup>。以后 T<sub>E</sub>X软件经过多次改进,达到了较高的水平。现在多使用的版本为 2.09 和 2e 版,而 3.0 版已于 1997 年初正式推出。

由于它可以容易地作高质量的科学文献排版工作,并可以按照作者自己的要求比较容易地对之进行扩充和修改,所以它逐渐引起出版界和各国科学工作者的兴趣。世界上很多杂志鼓励作者直接用 T<sub>E</sub>X投稿,例如著名的美国数学会(AMS)就修改了 T<sub>E</sub>X所提供的格式,推出了 AMS-T<sub>E</sub>X,鼓励数学家直接用 AMS-T<sub>E</sub>X给该会所属的杂志投稿。此外著名的牛津大学出版社、Addison-Wesley 出版社等也接受 T<sub>E</sub>X处理的稿件。一些著名的国际会议(如 IFAC 的各种会议)在接受照像制版的稿件时也提供 T<sub>E</sub>X的标准版面要求。现在如果随便打开一本国际论文集,就会发现至少有大半欧美国家作者的论文是用 T<sub>E</sub>X的各种版本处理的。T<sub>E</sub>X普及与流行的一个重要原因离不开 Leslie Lamport 的贡献,他在 T<sub>E</sub>X的基础上作了大量的修改和扩充,形成了 L<sup>A</sup>T<sub>E</sub>X(Lamport 的 T<sub>E</sub>X),并于 1985 年出版了一本著名的著作“L<sup>A</sup>T<sub>E</sub>X: A Document Presentation System”<sup>[4]</sup>,在 L<sup>A</sup>T<sub>E</sub>X中定义了很多的宏命令(macros),使得当时较为复杂的很多数学公式的处理比起 T<sub>E</sub>X变得容易得多。

Knuth 教授公开了 T<sub>E</sub>X系统的全部源程序(由 Pascal 语言编写),微机版上较实用的有 PCT<sub>E</sub>X以及德国学者 Eberhard Mattes 开发的 emT<sub>E</sub>X<sup>[5]</sup>,在 Windows 版下还有一种名为 Scientific Word 的处理软件,从一定程度上实现了“所见即所得”(What You See Is What You Get,简称 WYSIWYG)的功能,从风格上讲,该软件和 Word for Windows 很类似,然而它可以用类似 Word 的方式生成 L<sup>A</sup>T<sub>E</sub>X类源文件,所以对不是很精通 L<sup>A</sup>T<sub>E</sub>X软件的用户有一定的帮助。除了基本的科学文字处理的基本特色以外,还存在很多其它的附件程序,其中较有用的有拼写错误查找程序 MicroSpell 及 AMSpell,绘图程序 T<sub>E</sub>Xcad 等, L<sup>A</sup>T<sub>E</sub>X还提供了一个曲线绘制处理程序 PICT<sub>E</sub>X。L<sup>A</sup>T<sub>E</sub>X也出现了各种文字的版本,如 emT<sub>E</sub>X同时还有德文版。

中国科学院计算数学与科学工程计算研究所的张林波等人<sup>[7]</sup>对英文版的  $\text{emT}_\text{E}^\text{X}$ 进行了汉化,使得这一优秀的排版软件可以同样容易地对中文文章作排版,取得了巨大的成功。中文版  $\text{L}^\text{A}^\text{T}_\text{E}^\text{X}$ (又名  $\text{cct}$ ) 目前最新的版本为 5.11 版,可以从网络上直接免费取得。

由于  $\text{L}^\text{A}^\text{T}_\text{E}^\text{X}$ 是国际上相当受欢迎的排版软件,所以很多其它的应用软件,如符号代数软件 Mathematica 等,都可以产生  $\text{L}^\text{A}^\text{T}_\text{E}^\text{X}$ 可以识别的数学公式, Mathematica 的  $\text{TeXForm}$  命令可以将产生的结果自动转移成  $\text{L}^\text{A}^\text{T}_\text{E}^\text{X}$ 的公式描述方式,使得用户可以方便地将该结果传送到用户自己的  $\text{L}^\text{A}^\text{T}_\text{E}^\text{X}$ 源文件之中,这样就会省去许多重新输入的麻烦,并减少很多出错的机会。另外,  $\text{L}^\text{A}^\text{T}_\text{E}^\text{X}$ 允许在文中插入 EPS (Encapsulated PostScript, 即封装的 PS) 文件,而许多实用软件,如 MATLAB, 又都可以生成 EPS 图形文件,这样在  $\text{L}^\text{A}^\text{T}_\text{E}^\text{X}$ 下就可以插入各种软件绘制的图形,彻底做到图文并茂。

和其它文字处理软件相比,  $\text{L}^\text{A}^\text{T}_\text{E}^\text{X}$ 主要有以下几个突出特点:

- **卓越的数学排版效果:** 数学公式的排版明显优于其它软件,这是世界上公认的。稍有经验的用户可以在很短的时间内轻易地得出下面两个公式的排版结果。用户可以将这样的排版效果与难易程度和常用的 Microsoft Word 等基于 Windows 的排版软件比较。

$$U_i = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \lambda_i & 1 & 0 & \cdots & 0 \\ \lambda_i^2 & 2\lambda_i & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \lambda_i^{n-1} & \frac{d}{d\lambda_i}(\lambda_i^{n-1}) & \frac{1}{2!} \frac{d^2}{d\lambda_i^2}(\lambda_i^{n-1}) & \cdots & \frac{1}{(m_i-1)!} \frac{d^{m_i-1}}{d\lambda_i^{m_i-1}}(\lambda_i^{n-1}) \end{bmatrix}$$

$$G(j\omega) = \int_0^\infty g(t)e^{-j\omega t} dt \simeq \int_0^{T_f} g(t)e^{-j\omega t} dt$$

- **灵活的自动编号功能:**  $\text{T}_\text{E}^\text{X}$ 的交叉引用功能是大多数文字处理软件所望尘莫及的,它可以对用户指定的数学公式、图、表、页码及参考文献等自动进行编号,还可以在文中对公式等的编号进行引用。在编写较长的文件(例如编写博士或硕士论文)时,这种功能无疑是极其实用和方便的。考虑这样一种情况:如果在一个文件中已经有一些编号的公式,而用户想在文件中诸多公式之间插入一个新的公式,如果使用其它软件则原来的公式序号需要用手动的方式逐一改变,而在文本中引用公式的序号时 also 需用手动的方式去修改。这对于写作者(尤其是较长文章的作者)来说是相当麻烦的,而  $\text{L}^\text{A}^\text{T}_\text{E}^\text{X}$ 的用户却可以轻松自如地完成这种插入,因为  $\text{L}^\text{A}^\text{T}_\text{E}^\text{X}$ 系统会替用户完成这种繁琐的劳动。对文件的章节、插图、表格和参考文献的编号和引用也可以自动地完成。除此之外,  $\text{L}^\text{A}^\text{T}_\text{E}^\text{X}$ 可以自动地生成目录和索引。
- **低廉的硬件资源要求:**  $\text{L}^\text{A}^\text{T}_\text{E}^\text{X}$ 的文件实际上是由纯 ASCII 码组成的,用户可以通过  $\text{L}^\text{A}^\text{T}_\text{E}^\text{X}$ 系统提供的编译和预显示功能对之进行处理,最后通过打印机输出出来。 $\text{L}^\text{A}^\text{T}_\text{E}^\text{X}$ 文件的录入可以在有较低配置的计算机(如 8086 等低档机)上完成,而编译等处理在一个 286 上就可以实现(如果速度要求得不是很高,8086 也可以完成),所以不用象 Word for Windows 那样要求具有 Windows 环境和较高的 RAM 配置,更不必象华光等排版系统那样去要求专门的硬件卡等设施。
- **广泛的国际通用描述:** 如果只使用纯英文的  $\text{L}^\text{A}^\text{T}_\text{E}^\text{X}$ ,建立的文件可以直接投送至一些国际杂志和会议。即使使用了一些中文,而由之生成的 PostScript 文件也可以直接以文件

的形式投出。对于博士、硕士生等来说，这样做有一个好处，就是可以在一套文字处理的软件下将所发表的文章中的数学公式等复制到最终的学位论文中。

此外，由于  $\text{\LaTeX}$  文本输入的是纯 ASCII 码，所以用户可以轻易地实现“盲打”，最终将提高打字速度。试想如果要输入  $\alpha$ ，则在  $\text{\LaTeX}$  排版时只需输入命令  $\text{\$}\backslash\alpha\text{\$}$ ，而在 Word 等“所见即所得”的软件下则需打开数学公式编辑程序，打开相应的图标并从众多的子图标中找到  $\alpha$  标识，再用鼠标点中的方式获取该标识方可完成，所以基本可以说不能实现盲打。

综上所述， $\text{\LaTeX}$  具有一些其它文字处理软件难以比拟的优越性，尤其对将要写作博士与硕士论文的学生来说更是如此，所以建议控制仿真研究中心的研究生一定要精通  $\text{\LaTeX}$ ，掌握了这种工具以后，就不仅可以容易地写出具有专业文字处理水准的论文来，而且可以省却许多诸如公式编号和文本中引用不匹配等的麻烦，确实提高了论文质量。

## 2 $\text{\LaTeX}$ 版面设计与使用基础

### 2.1 $\text{\LaTeX}$ 文件的结构与页面设置

$\text{\LaTeX}$  文件是由 ASCII 码组成，就是说尽管它可以用来处理复杂的数学公式，但描述这些公式的文件却不含有任何特殊符号，所有用到的符号全部可以由最普通的键盘来输入（当然不包括中文字符，在处理中文字符用的是国标码）。2.09 版的  $\text{\LaTeX}$  文件的基本格式如下：

```
\documentstyle[options]{document style}
    前置说明命令（包括版面设置等）
\begin{document}
    文件的正文部分
\end{document}
```

其中

`options` 为附加选项，在这个命令下用户可调用一些选项，如整个文件的字体大小可以由 10pt, 11pt, 12pt<sup>1</sup> 来控制，关于字号大小下面还要专门介绍。如果想插入页面描述语言 PostScript 写的图形，可以在选项处加入 `epsf` 选项，若用户想获得双列的文本时，则可以选择 `twocolumn` 选项。用户可以在一个文件中设置多个选项，每个选项都用逗号分隔。

`document style` 为文档类型， $\text{\LaTeX}$  提供的文档风格最基本的应该包括以下四种：`book`（书籍型），`article`（文章型），`report`（报告型），`letter`（书信型），每一种风格将对应于几个后缀为 `STY` 的文件。如果采用中文版  $\text{\LaTeX}$ ，则应该在每一种风格之前加一个字符 `c`，例如用 `carticle` 取代标准的 `article`。一般  $\text{\LaTeX}$  版本都对基本的风格作了扩充，例如增添了 `thesis`（论文型）<sup>2</sup>，`proc`（论文集型）等，用户可以自己去选择一种合适的风格。如果使用  $\text{\LaTeX}$  2.09 版，将使用 `sty` 文件，很多的国际重要会议，如控制界的 IFAC 世界大会、美国控制会议及决策与控制会议等都专门设计了自己的 `sty` 文件，用户可以容易地从网络上直接获得。例如写文章时可以用 `article` 风格，而写论文可以用 `book`

<sup>1</sup>pt 为文字处理专用的长度单位，换算关系为 1 英寸 = 72.27pt，如果选择 10pt，则不能在选项中写 10pt，此项应该省略。

<sup>2</sup>在国外，很多大学根据自己学校的论文版式要求设计了自己的 `thesis` 版式，一般我们没有必要去照搬。

或 report 风格。除了这些文件之外,对不同的字号和字型还将调用如 bk11.sty (它对应于 book 风格的 11pt 字号选项) 这样的文件。

而 2e 版的起始语句由 \documentclass 引导。L<sup>A</sup>T<sub>E</sub>X2e 版所用的风格文件后缀名为 cls,而字号调用文件的后缀名为 clo。

在前面叙述中用到了一些 L<sup>A</sup>T<sub>E</sub>X 命令,如 \documentstyle 命令, L<sup>A</sup>T<sub>E</sub>X 所有的命令都是由反斜杠 (\) 引导的。

在声明了文档风格之后,还可以给出一些前置定义说明,如对页面大小的说明、对页面风格的说明,还可以定义一些用户自己的命令记号。

- **页面尺寸设置**: L<sup>A</sup>T<sub>E</sub>X 的页面尺寸设置的命令主要有下面几条, \topmargin 命令用来设置页面顶部的边界宽度, \oddsidemargin 命令用来设置奇数页左边界的宽度, \evensidemargin 命令设置偶数页左边界宽度, \textwidth 命令设置正文的宽度,而 \textheight 命令用来设置正文的高度。具体设置命令的格式为:

命令名 长度

, 例如 \textwidth 160mm

其中的 mm 为毫米单位,这样用户就将排版页面宽度设置成了 160 毫米。用户还可以采用其它单位如 cm (厘米)、in (英寸) 或 pt 等来设置长度。其它各个命令也是采用这样的格式设置的。有了这些参数以后,整个页面的规格就唯一地设计出来了,用户可以根据自己的文章要求调试出合适的页面来。

- **页面风格设置**: 页面风格包括页码的显示位置、书眉显示方式等。页面设置的方法可以由 \pagestyle 命令来完成, \pagestyle 命令的基本格式为

\pagestyle{page description}

其中 page description(页面说明值) 选项可以取作以下几个值: plain(页码在页面底部居中,没有页眉)、empty(页眉和页脚均为空)、headings(页脚为空,页眉在页面上部) 与 myheadings(自定义方式)。在自定义模式下,该语句后面应该跟一条 \markboth 命令,其设置格式为

\markboth{left page}{right page}

其中 left page 和 right page 应分别填写左右页面上的页眉,这样左右页面的页眉也就可以设置起来了。

用户可以用 \parindent 设置出每一个段落第一行文字最前面的空白的大小,例如可以通过 \parindent 5mm 在每一个段落首行设置出 5 毫米的空白,在中文版 L<sup>A</sup>T<sub>E</sub>X 软件中还定义了一个实用的长度单位 \ccwd,其意是一个汉字的宽度,用户可以使用 \parindent 2\ccwd 命令在每一段落前面留出两个汉字的空白。 \parindent 的参数也可以为负值。

如果用户想改变段落之间的距离时,可以使用 \parskip 命令来设置,如果给出命令 \parskip 3mm,则在当前行间距的基础上再加 3 毫米的段落间距。

前置命令设置完成之后,就开始正文部分了。正文部分是由 \begin{document} 和 \end{document} 语句括起来的部分,它是整个用户文件的中心内容。

如果写作较常的文章(如博士论文)时,可以将整个文件分成若干个部分,比如说每一章为一个部分,每个部分是一个的 CTX 文件,在文件正文中用 `\include{}` 来包含每一个子文件,其中大括号中应改给出文件名。如果其中有几章已经成熟或暂时不做改动,则可以在编译时对这几章不做重新编译以节省时间。但由于略去的几章相应的 AUX(辅助)文件仍被调用,所以这样做并不影响对略去章节的交叉引用,也不影响现在编译章节的编号。完成这种工作可以在 `\begin{document}` 的前面用 `\includeonly{}` 命令来设置,在大括号中写入编译时包含的文件名。如果想包含多个文件,则可以列出多个文件名,各个文件名之间用逗号分隔。

和 `\include{}` 命令相似的命令 `\input{}` 在实际中也经常使用。它和 `\include{}` 命令的区别是: `\include{}` 命令开始一个新的页面,而 `\input{}` 命令仍接着前面的内容继续排版,并不附加新的页面。

## 2.2 L<sup>A</sup>T<sub>E</sub>X 的字体和字号设置

L<sup>A</sup>T<sub>E</sub>X 提供了多种字体和字号的设置,如英文字符就可以有多种不同的设置方式,这些设置的方法可以参见下表,在中文版 L<sup>A</sup>T<sub>E</sub>X 下还提供了一些常用汉字字体,也在下表中列出:

字体定义	演示	字体定义	演示
<code>\rm</code> (罗马体)	Roman Style	<code>\tt</code> (打字机体)	TypeWriter Style
<code>\bf</code> (粗体)	<b>Bold Style</b>	<code>\sl</code> (斜体)	<i>Slanted Style</i>
<code>\it</code> (意大利体)	<i>Italic Style</i>	<code>\sf</code> (等斜体)	Sans Serif Style
<code>\underline</code> (下划线体)	<u>Underlined Style</u>	<code>\biaosong</code> (标宋体)	<b>标宋体文字</b>
<code>\songti</code> (宋体)	宋体文字	<code>\heiti</code> (黑体)	<b>黑体文字</b>
<code>\fangsong</code> (仿宋体)	仿宋体文字	<code>\kaishu</code> (楷体)	楷体文字

除了上面列出的字体以外, L<sup>A</sup>T<sub>E</sub>X 还支持其它字体(总共 256 种西文字体),但需要特殊命令来设置,这里就不再介绍了。L<sup>A</sup>T<sub>E</sub>X 还允许用户对字号的大小进行设计,在下面的表中将列出一些常用的字号<sup>3</sup>。如果用 `\zihao` 命令来改变字号大小,将只改变中文字符的大小,而对英文字符不起任何作用。

字号定义	演示	字号定义	演示
<code>\tiny</code> (微型)	Tiny Size	<code>\scriptsize</code> (上下标)	Script Size
<code>\footnotesize</code> (脚注)	Footnote Size	<code>\small</code> (小型)	Small Size
<code>\normalsize</code> (正常型)	Normal Size	<code>\large</code> (大型)	large Size
<code>\Large</code> (大型)	Large Size	<code>\LARGE</code> (大型)	LARGE
<code>\huge</code> (巨型)	huge		
<code>\zihao{0}</code> (零号)	零号	<code>\zihao{1}</code> (一号)	一号
<code>\zihao{2}</code> (二号)	二号字	<code>\zihao{3}</code> (三号)	三号字
<code>\zihao{4}</code> (四号)	四号字	<code>\zihao{-4}</code> (小四号)	小四号字
<code>\zihao{5}</code> (五号)	五号字	<code>\zihao{-5}</code> (小五号)	小五号字
<code>\zihao{6}</code> (六号)	六号字	<code>\zihao{7}</code> (七号)	七号字

<sup>3</sup>这里列出的是在 10pt 下的字号大小,在 11pt 和 12pt 下字号的大小有所不同。

上面给出的字体和字号控制命令还可以合在一起来使用，例如如果用户键入了下面一段文字

```
Here {\bf Bold Font} is used in {\Large\bf Larger Bold Format}.
```

则会产生下面的打印效果:

Here **Bold Font** is used in **Larger Bold Format**.

在字体和字号定义时一定要注意，如果既需要改变字号也需要改变字体，则应该首先定义字号的大小，然后再定义字体，否则将不能得到需要的排版结果。

除了标准的英文或中文字符以外，在文章中还可能有一些特殊的符号，如 Åström 一词应该由 L<sup>A</sup>T<sub>E</sub>X 提供的命令 \AA str\ "om 来处理，其中 \AA 将给出 Å，而 \ "o 将给出 ö，注意，\AA 命令后面的空格是不可省略的，否则 L<sup>A</sup>T<sub>E</sub>X 将认为它和后面的字符串构成一个命令，从而得不到希望的排版效果。其它可以用作这些特殊符号修饰的命令可以参见附录 A。

## 2.3 特殊字型设置技巧

中文版 L<sup>A</sup>T<sub>E</sub>X 在根目录下有一个 cct.dat 文件，其中描述各种允许字型。未改动的 cct.dat 文件的内容如下

10				
8.5	8.5	0.06	0.0	-5
11.32	11.32	0.06	0.0	-4
34.0	34.0	0.06	0.0	0
26.0	26.0	0.06	0.0	1
20.0	20.0	0.06	0.0	2
15.0	15.0	0.06	0.0	3
13.0	13.0	0.06	0.0	4
9.9	9.9	0.06	0.0	5
7.5	7.5	0.06	0.0	6
5.0	5.0	0.06	0.0	7

其中，第 1 行的 10 表示该环境中定义了 10 种字型，第 2 行至第 11 行中描述各个字型的参数，其中第 1 列为该字型汉字的宽度 (单位为 pt)，第 2 列为高度，第 3,4 列分别为水平和垂直的间距，第 5 列为“字号”，它对应于 \zihao 命令。如果用户想扩充字型，则可以将第 1 行的 10 扩大，然后再在字型行按照各列的格式添加新内容，例如将第 1 行的 10 改成 11，并在后面填写第 12 行

18.5	8.5	0.06	0.0	8
------	-----	------	-----	---

并存盘，然后运行 cctinit.exe 程序确认字型库的变化，这样就相当于定义了 8 号字，其宽度为 18.5pt，高度为 8.5pt。这时若在文本中给出 {\zihao{8} 重新定义的 8 号字 }，则将显示出 重新定义的 8 号字 字样。可见利用中文版 L<sup>A</sup>T<sub>E</sub>X 可以创造出任何高宽比例的字型来。

## 2.4 行间距和字间距的用户自定义

在标准的 L<sup>A</sup>T<sub>E</sub>X 下, 由于行间距是默认的, 所以有时会显得版面过密、不美观, 这样往往需要用户自己去重新定义行间距, 以满足排版要求。L<sup>A</sup>T<sub>E</sub>X 下提供了两种定义行间距的方法, 第 1 种是直接改变 `\baselinestretch` 变量的数值, 其命令格式为

```
\renewcommand{\baselinestretch}{1.5}
```

这种方法的定义是绝对的, 它自动地将行间距定义为 1.5 倍的默认值, 当然这里的 1.5 也可以由其它任意数值取代。这里的 `\baselinestretch` 命令除了改变正文行间距之外还将改变其它各种环境的行间距, 例如插入表格的行间距等。另一种方法是定义一个新变量 `\double`, 其格式为

```
\renewcommand{\double}{\baselineskip 1.5 \baselineskip}
```

然后在想改变行间距处执行 `\double` 命令, 这样在该命令执行后行间距将自动改变。注意这种命令是局部的, 它将只改变当前正文的行间距, 而不会改变表格等环境的行间距。若两次使用 `\double` 命令则会使得行间距变成  $1.5^2$ 。另外, 当字号改变时, `\double` 命令会自动失效。同样地, 这里的 1.5 也可以由其它数值替换。在本文中使用的行间距值为 1.24。

由于上面提到的两种自定义行间距的命令各有特点, 所以在使用改变行间距命令时应选择一个合适的方法。

中文版 L<sup>A</sup>T<sub>E</sub>X 还提供了一个改变字间距的命令 `\ziju`, 该命令的调用格式为

```
\ziju{0.5}
```

该命令使得两个汉字之间空 0.5 个汉字的距离, 当然这里的 0.5 也可以由任意数值取代。举例: `{\ziju{0.5} “加大字距 ”}` 将给出 “加 大 字 距” 的排版效果, 而 `{\ziju{1} “再加大字距 ”}` 将给出 “再 加 大 字 距” 的排版效果。再考虑 `{\ziju{3.5} English Word 字距 }` 命令, 通过模拟显示可以看出该命令的排版效果为 English Word 字距, 亦即 `\ziju` 命令不影响英文单词的间距。

## 2.5 L<sup>A</sup>T<sub>E</sub>X 的基本命令格式

前面介绍过, 所有 L<sup>A</sup>T<sub>E</sub>X 命令都是由 `\` 符号引导的一个英文字符串, 在上面的介绍中还给出了几个 L<sup>A</sup>T<sub>E</sub>X 命令的例子。L<sup>A</sup>T<sub>E</sub>X 的命令分为全局命令和局部命令, 所谓全局命令就是在整个文件中一直起作用的命令, 如果不加修改, 这种命令的设置在整个文件中都将都有效。局部命令是在一个用户指定的区域中起作用的命令, 这个区域结束之后, 该命令的作用就取消了。这种命令的例子可以参照前面给出的字体设置语句, 在该语句中的 `\bf` 字体是针对大括号中的 Bold Font 来说的, 就是说将这两个单词变成粗体字, 这两个单词处理完之后, 则恢复原来的正常状态, 后面的一个命令 `\Large\bf` 也是针对大括号中的字符串来说的, 出了这个大括号, 该命令的作用也将取消, 而恢复到设置前的形式, 用户使用时一定要注意局部命令的作用区域。

由此可见, 大括号的作用是很大的, 它可以设定局部命令作用区域的边界。用大括号括起来的区域称为一个组。在一个组之内的设置只作用到该组的结尾处为止。

在读 L<sup>A</sup>T<sub>E</sub>X 源文件时往往会发现, 其中有些行是由 `%` 来引导的, 这说明该行中 `%` 符号以后的部分全部是注释, 在编译中将不起任何作用, 只是在源文件中起提示用户的作用。

既然 L<sup>A</sup>T<sub>E</sub>X 的所有命令都是由反斜杠来引导的, 那么在 L<sup>A</sup>T<sub>E</sub>X 中的反斜杠又是怎么表示的呢? L<sup>A</sup>T<sub>E</sub>X 一共保留了十个特殊符号, 反斜杠为其中一种, 其它的一些特殊符号的使用在后面还将介绍。

L<sup>A</sup>T<sub>E</sub>X 保留的特殊符号在下面的表中给出

特殊字符	\$	&	%	#	-	{	}	^	~	\
表示方法	<code>\\$</code>	<code>\&amp;</code>	<code>\%</code>	<code>\#</code>	<code>\-</code>	<code>\{</code>	<code>\}</code>	<code>\^{}</code>	<code>\~{}</code>	<code>\backslash</code>

一些 L<sup>A</sup>T<sub>E</sub>X 版本存在着问题, 有时并不能正确地处理文本状态下的 `\backslash` 命令, 所以作者建议采用数学模式下的命令 `$\backslash$` 来进行设置。L<sup>A</sup>T<sub>E</sub>X 在连字符上也有些特殊的约定, 如 - 的显示为 -, -- 的显示为 –, --- 的显示为 —。L<sup>A</sup>T<sub>E</sub>X 的一个重要的特色是它的排版和用户输入的格式 (如中间有几个没有特别意义的空格或每一行的长短等) 并没有直接关系, 它只是在逗号或句号后面按照正常排版的要求空出足够的地方, 即使在源文件中相应的地方空出了几十个空格也不例外。L<sup>A</sup>T<sub>E</sub>X 识别段落的方法是找出一个空白的行, 在空白行后面的内容被认为是下一段落的内容。

L<sup>A</sup>T<sub>E</sub>X 有自己的字间距和行间距, 如果用户不进行修改, 则一直将使用它给出的行或字间距。如果用户想在一行中强行插入一个空格, 则可以使用 `~` 或 `\u` 来完成<sup>4</sup>, 如果想插入一个 3cm 宽的空白, 则可以使用 `\hspace{3cm}` 命令来指定, 若想在两个段落之间加入一个 3cm 的间距, 则可以使用 `\vspace{3cm}` 命令来完成。如果用户想在一行的末尾强行换行, 则可以在想换行处加一个 `\\` 命令。如果用户想开始一个新的页, 则可以采用命令 `\newpage` 来设置, 这时前面的页将产生中断, 以开始后面新的页面。如果用户想将前面一页的内容充满该页, 则不能采用 `\newpage` 命令, 而必须采用 `\pagebreak` 命令。对当前行进行中断的 `\linebreak` 和 `\newline` 命令的区别也是这样。

如果用户有一段文字 (例如一段源程序) 想原封不动地排版, 则可以使用 `\begin{verbatim}` 和 `\end{verbatim}` 括起来的环境来处理了。使用中文版 L<sup>A</sup>T<sub>E</sub>X 的用户一定要注意, 这一环境中不能包含中文字符, 否则将出现不应有的错误, 因为 CCT 程序首先将包含的中文文字翻译成 L<sup>A</sup>T<sub>E</sub>X 命令, 而 `verbatim` 环境又要求照抄环境中的内容, 所以最后得到的将是用户所不希望的。

L<sup>A</sup>T<sub>E</sub>X 中允许用户使用环境的概念, 所谓环境就是一个局部的段落组, 在这个段落组中所有的命令都是局部命令。环境是由 `\begin{环境名}` 和 `\end{同一个环境名}` 对来括起来的, 其作用相当与大括号对。前面遇到的 `document` 和 `verbatim` 都是环境名。

## 2.6 L<sup>A</sup>T<sub>E</sub>X 内部参数的显示

对有经验的用户来说, 经常需要自己去修改 L<sup>A</sup>T<sub>E</sub>X 中定义的一些默认值, 或定义一些自己的新命令, 这样在实际使用中如何监视其中一些内部参数呢? L<sup>A</sup>T<sub>E</sub>X 提供了一个命令 `\showthe` 来监视内部参数, 该命令的调用格式为

`\showthe 命令名`

例如若用户想显示当前的页面宽度则可以给出 `\showthe\textwidth` 命令, 这样在 L<sup>A</sup>T<sub>E</sub>X 编译过程中该命令的位置将自动显示出当前 `\textwidth` 的数值, 并产生一个停顿。

---

<sup>4</sup>这里的 `\u` 表示一个空格。



## 2.7 L<sup>A</sup>T<sub>E</sub>X文件的编译与模拟显示

如果用户使用的是中文版 L<sup>A</sup>T<sub>E</sub>X，则应该建立一个后缀为 CTX 的文件作为源文件，然后调用 cct 命令将之翻译成后缀为 TEX 的文件。假设用户建立的文件名为 mytest.ctx，则应该采用下面的命令来处理：

```
cct mytest
```

建立起来 L<sup>A</sup>T<sub>E</sub>X源文件 (以 TEX 为后缀的文件，即 mytest.tex) 之后，则可以用 latex 命令来对之进行编译，最后形成一个以 DVI 为后缀的文件 (DVI 是 DeVice Independent 的缩写，意即与设备无关) 等。除了生成 DVI 文件以外，还会同时生成 AUX(辅助) 文件和 LOG 文件 (屏幕中间显示信息存储文件)，如果需要自动生成目录则还会产生 TOC(Table Of Contents) 文件，如果需要产生索引则还将生成 IDX 文件 (Index)。除了 DVI 文件以外，其它生成的文件都是可读的。其中辅助文件是很重要的，它存储一些交叉引用的信息，如果一个文件中的公式序号等发生变化时，则在编译结束时给出下面的信息

LaTeX Warning: Label(s) may have changed. Rerun to get cross-references right.

这要求用户重新执行编译命令来获得正确的 DVI 文件，否则在产生的 DVI 文件中，一些自动生成的错误标号将以 ?? 的形式显示出来。在用户用到了自动生成目录等功能时，为获得正确的交叉引用关系，有时需要连续三次运行编译程序。

对上面的例子来说，如果键入

```
latex mytest
```

则会生成 mytest.dvi 文件。如果用户想将 DVI 文件作预显示，则可以调用 view 命令来实现，例如

```
view mytest
```

前面给出的几个命令都是批处理文件，对不同的 L<sup>A</sup>T<sub>E</sub>X版本，这些命令大同小异，希望用户使用时的注意。

目录可以通过 L<sup>A</sup>T<sub>E</sub>X的编译命令直接生成，索引却不能，它必须通过一个索引处理程序对产生的 IDX 文件进行处理，产生一个和 T<sub>E</sub>X兼容的文件来，这时可以通过和前面一样的过程来获得它的 DVI 文件来。

如果需要打印，利用中文版 L<sup>A</sup>T<sub>E</sub>X提供的 dvi1jp.exe, dvi2jp.exe 和 dvi24p.exe 等可执行文件可以自动地将打印内容送给 HP 系列的激光打印机、DJ500 系列喷墨打印机或各种 24 针点阵打印机。在使用 24 针打印机之前还应该运行 setprt.exe 程序来选定打印机类型，该程序会自动给出一个打印机列表，允许用户从中选择合适的打印机。若打印机类型选择不当，则打印结果将出现错误。

## 3 L<sup>A</sup>T<sub>E</sub>X对文件格式的定义

### 3.1 文件版式与标题处理

L<sup>A</sup>T<sub>E</sub>X的默认版式是单列的，如果有特殊需要，也可以将版式设置成双列的 (如一些会议所要求的那样)，这样做只需在前面介绍的 \documentstyle 命令中方括号括起来的选项中添

加 `twocolumn` 即可, 这样做将使得整个版面全部是双列的。

在特殊的要求下, 除了正文部分用双列形式处理之外, 用户还需要将文章题目、作者姓名等信息按通栏的形式处理, 这样就需要借助于 `\maketitle` 命令了。例如如果用户在双列的格式下给出了下面的一些命令

```
\title{Fokker-Planck equation solutions for a class of nonlinear systems}
\author{Dingy\u XUE}
\date{}
\maketitle
```

则将给出如下的排版格式

Fokker-Planck equation solutions for a class of nonlinear systems  
Dingyü XUE

可以看出, `\maketitle` 命令是将 `\title`, `\author` 和 `\date` 等命令中的内容显示出来。`\title` 命令后面的大括号内加的是文章的标题, 而 `\author` 命令后面的是作者的信息, 如果用户想将当前日期 (计算机中储存的日期) 也显示出来, 则取消 `\date` 命令, 否则可以在 `\date` 命令的大括号中填写要显示的日期, 如果用户在 `\date` 命令后面给出一个空的大括号, 则 `\maketitle` 命令将不显示日期信息。

`\title` 和 `\author` 等命令看起来很简单, 如果用户想将更复杂的信息 (如摘要与关键词等信息) 也做通栏处理, 则需要通过巧妙地将这些信息填写到其中的一个命令 (如 `\author`) 中。

### 3.2 L<sup>A</sup>T<sub>E</sub>X对文件章节等的自动处理

L<sup>A</sup>T<sub>E</sub>X的一个突出的优点是它可以自动地为用户生成所需的文件章节, 章节生成的格式是在有关的 STY (或 2e 版本中的 cls) 文件中设定的, 一般认为这些文件中规定的章节格式是标准的, 如果用户想改变章节显示的结构, 则应该对相应的 STY 文件作一些修改, 但要注意, 一定要把将修改的 STY 文件复制到用户自己的目录下, 以免修改后对其它人的文件格式产生不应有的影响。

L<sup>A</sup>T<sub>E</sub>X的章节划分命令如下:

`\part` (编)、`\chapter` (章)、`\section` (节)、`\subsection` (小节)、`\subsubsection`、`\paragraph` 以及 `\subparagraph`。

这些划分是由大到小而自动作用的, 这类命令的格式为

`\CommandName{TitleString}`

其中 `\CommandName` 为命令名, 而 `TitleString` 为章节标题字符串。这时在允许的情况下按 STY 文件给定的格式将显示出用户想用的章节序号及题目。所谓允许的情况是这样的, L<sup>A</sup>T<sub>E</sub>X规定最多从大到小使用四级章节划分命令, 在每一种风格文件下, 由自己规定的起始章节命令, 例如在 `article` 风格下, 最高的定义是“节” (section), 如果用户使用了命令

```
\section{This is My First Section}
```

则在编译之后将产生

## 1 This is My First Section

字样的排版效果, 这里的 1 是自动生成的, 后面再使用 `\section{}` 命令时, 则节号将自增为 2, 以后随着该命令的使用章节号将自动地增加。在 `\section{}` 命令中大括号内的字符串为

该节的题目，使用时将自动地显示出来。如果用户不想显示出该节的编号，则可以相应地使用 `\section*{}` 命令来设置。

如果在 `\section{}` 命令后边再加上下面命令

```
\subsection{This is My First SubSection}
```

则将会自动地出现下面的显示方式

### 1.1 This is My First SubSection

其中的小节序号是由  $\text{\LaTeX}$  本身带有的计数器来设置的，每使用一次 `\section{}` 命令，其小节计数就自动地置为 0，从而可以无误地开始下一小节的计数。在 `\subsection{}` 命令下面还可以设置 `\subsubsection{}` 等。

在 book 或 report 风格下，最高可以使用 part(编)，下面依次为 chapter, section, subsection。例如如果用户给出了如下的命令

```
\chapter{This is My First Chapter}
```

则相应地将给出如下所示的排版结果

## Chapter 1 This is My First Chapter

如果采用了中文版  $\text{\LaTeX}$ ，则章节题目对中文也作了相应的设置。提供了章节序号自动生成的功能无疑将给用户处理较长的文件带来极大的方便。

除了章节符号自动生成之外，用户还可以采用  $\text{\LaTeX}$  提供的命令来自动地生成脚注，例如如果想在某一个词的后面加一个脚注，只需在该词的后面加上一条 `\footnote{}` 命令，其中脚注的内容在该命令中大括号括起来的部分中填写。 $\text{\LaTeX}$  将会自动为该脚注进行编号，如果可能，则  $\text{\LaTeX}$  还将自动地把该脚注的内容安排在同一页的底部显示出来，上面加一个短的横线，请参阅本文中前面的脚注显示方式。

除了自动对脚注进行编号之外， $\text{\LaTeX}$  还允许用户对脚注进行手动编号，这样的脚注定义命令的格式如下：

`\footnote[footnote symbol]{footnote}`

其中

`footnote symbol`: 为脚注编号部分，可以填写用户自定义的脚注编号。若无此选项则按照默认的方式添加脚注编号。

`footnote`: 脚注的实际内容，其中可以包含任何用  $\text{\LaTeX}$  语句写出的内容，包括公式与图表。

### 3.3 $\text{\LaTeX}$ 目录与索引的自动生成

$\text{\LaTeX}$  是通过文件的章节声明来生成目录的。目录生成是很简单的，只需用户在正文部分的最前面给出一个 `\tableofcontents` 命令就可以了，无需用户去进行过多的干预。用户在编译  $\text{\TeX}$  文件时，将会自动地产生一个 TOC 文件。一般应该注意，在编译过程结束时往往会给出消息通知用户交叉引用不正确，要求用户重新运行编译程序，这时如果不给出该信息，则 DVI 文件就编译成功了，如果仍给出该信息，则还需重新运行 (至多三次)。

索引生成与脚注的生成很相似，在需要索引的词汇后面加一条 `\index{}` 命令，命令中给出索引的词汇。除了这些工作之外，在正文部分的前面加上一条 `\makeindex` 命令，这时如果编译，则会产生一个 IDX 文件，用户只需对该文件进行处理就可以得出标准的 T<sub>E</sub>X 文件了。

### 3.4 L<sup>A</sup>T<sub>E</sub>X 的对齐处理

默认状态下，L<sup>A</sup>T<sub>E</sub>X 软件将文件处理成左右两边都对齐的形式，即使由于无法处理而产生极小的 (如 0.1pt 这样微小的数值) 超出情况，L<sup>A</sup>T<sub>E</sub>X 编译系统也会给出一个提示来提醒用户注意。所以一般情况下用户大可放心 L<sup>A</sup>T<sub>E</sub>X 的对齐排版效果。

在使用文字处理软件时，往往会遇到下面的情况，即有一段文字往往希望排在一行的中间 (即按中间对齐)，或一行文字往往希望靠右边对齐，L<sup>A</sup>T<sub>E</sub>X 提供了解决这类问题的成功方法。

如果有一段文字，用户可以用 `\centerline{}` 命令来处理，使之被排在一行的中间，其中大括号中应该给出需要居中排列的文字内容。例如

```
\centerline{This Text is Centered}
```

将给出下面的排版结果：

This Text is Centered

如果需要中间对齐处理的文字段很长，也可以采用 `\begin{center}` 和 `\end{center}` 环境将该段文字括起来，这样做的效果和使用 `\centerline{}` 命令是一样的。

如果用户想将一段文字靠右变或靠左边对齐，则可以使用 `\begin{flushright}` 和 `\end{flushright}` 及 `\begin{flushleft}` 和 `\end{flushleft}` 环境将所需处理的段落括起来就可以了。

### 3.5 L<sup>A</sup>T<sub>E</sub>X 科技文章排版框架举例

在这里将不加解释地给出一个科技文章排版的源文件，用户可以自己去编译并显示其排版效果。用户可以试着替换或更改其中的若干内容或参数来观察排版效果。在编写设计文章时，用户可以采用这一框架来添加自己的内容。当然，这里演示的内容并非 L<sup>A</sup>T<sub>E</sub>X 的全部功能，L<sup>A</sup>T<sub>E</sub>X 本身还有更强大的功能有待用户去使用和开发。

```
\documentclass[11pt]{article}
\usepackage{epsf}
\usepackage{emlines2}
\usepackage{indenrst}
\topmargin -10mm
\oddsidemargin 0mm
\evensidemargin 0mm
\textwidth 158mm
\textheight 226mm
\parindent 2\ccwd
\parskip 0mm
\hyphenpenalty=1500
\flushbottom
\newcommand{\double}{\baselineskip 1.24 \baselineskip}
\newcommand{\citeu}[1]{${\mbox{\protect \scriptsize \cite{#1}}}$}
```

```

\newcommand{\ispace}{\hspace*{2\ccwd}}
\title{A Sample Text for \LaTeX}
\author{\begin{tabular}{c} Dingy\"u Xue \\
    Northeastern University, PRC\end{tabular}}
\date{}
\begin{document}
\maketitle
\double

\abstract{This is the abstract part of the paper. You
    can add more information to it and see what the results will be.}

\section{Introduction}
\LaTeX is a useful tool in doing the scientific
documentation with results of professional standard. Creative
results can be achieved by experience users.

\section{The State of Art of \LaTeX}
Here a template of a paper is given to show the
    state of art of such a wonderful documentation system.
\subsection{Listing environment}
Try to find out what is the layout of the following commands
\begin{itemize}
\item three environments are provided, they are {\tt itemize}.
{\tt enumerate}, and {\tt description}, where {\tt enumerate} will
create a numbering system while {\tt description} comes without
numbering.
\item try to use the other two environments to replace the {\tt itemize}
environment and see what will happen.
\end{itemize}
\subsection{Mathematical formula}
Three types of math environments are allowed\cite{Lamp}.
The following statements will give different results:

$$\alpha_2^3\int_0^\infty f(s)ds$$
,

$$\alpha_2^3\int_0^\infty f(s)ds$$
 and
\begin{equation}\alpha_2^3\int_0^\infty f(s)ds\end{equation}

\appendix
\section{This the First Appendix}
The appendix is arranged automatically if an appendix
command is issued. The numbering system will be set to A, B,...
instead of 1, 2, ...
\begin{thebibliography}{99}
\bibitem{Lamp}
    Lamport, L., 1985, \LaTeX: A Document Preparation System,
    Addison-Wesley
\end{thebibliography}
\end{document}

```

## 3.6 L<sup>A</sup>T<sub>E</sub>X环境的定义与修改方法

### 3.6.1 L<sup>A</sup>T<sub>E</sub>X下用户自定义命令语句格式

为了排版方便, L<sup>A</sup>T<sub>E</sub>X允许用户自己去定义一些命令, 如前面框架中定义的 `\double` 和 `\citeu` 等。L<sup>A</sup>T<sub>E</sub>X自定义语句的格式为

```
\newcommand{\CommandName}[Number of argument]
[defaults]{command specifications}
```

其中

`\CommandName`: 自定义命令的命令名, 它可以是任何 L<sup>A</sup>T<sub>E</sub>X未保留的字符串, 一旦定义了该命令之后, 用户就可以像使用其它命令一样使用它了。若这里使用的命令名已经存在, 则在编译时会自动给出错误信息提示, 通知用户此命令已被使用。

`number of argument`: 自定义命令所带的参数个数。

`defaults`: 自定义命令中参数的默认值。

`command specifications`: 新命令内容的定义, 若此命令带有参数, 则应该在这里给出各个参数的使用, 其中 `#1` 表示带有的第 1 个参数, `#2` 表示带有的第 2 个参数, 依此类推。

下面通过例子来演示 L<sup>A</sup>T<sub>E</sub>X自定义命令的使用方法:

1). 观察前面的 `\double` 命令, 其格式为

```
\newcommand{\double}{\baselineskip 1.24 \baselineskip}
```

这里定义的 `\double` 命令并不带有任何参数, 其意义是使得两行间的间距为当前间距 (亦即 `\baselineskip` 参数定义的 Z 值) 的 1.24 倍。

2). 再考虑 `\citeu` 命令, 其格式为

```
\newcommand{\citeu}[1]{${\mbox{\protect \scriptsize \cite{#1}}}}$}
```

可以看出, 此命令带有 1 个参数, 其意义是以上标的方式显示 `\cite` 命令返回的数值, 亦即将自动交叉引用得出的参考文献编号用上标的形式在当前位置上显示出来。当然 L<sup>A</sup>T<sub>E</sub>X允许用户定义带有 2 个或 2 个以上的参数, 这里就不给出例子了。

L<sup>A</sup>T<sub>E</sub>X还允许重新定义其保留的命令, 这可以通过 `\renewcommand` 命令来完成, 该命令的格式和 `\newcommand` 是完全一致的, 当该语句执行以后, 原来的命令就被替换了。同样, 若 `\renewcommand` 命令中的 `CommandName` 参数被赋予一个原来不操作的命令名, 则也会给出错误信息。

L<sup>A</sup>T<sub>E</sub>X还提供了一个 `\providecommand` 命令, 它的格式和作用和 `\newcommand` 几乎是完全一致的。若自定义的命令已经存在, 则它将忽略自定义命令的设定, 而保留原来的命令。

### 3.6.2 L<sup>A</sup>T<sub>E</sub>X下用户自定义环境语句格式

在 L<sup>A</sup>T<sub>E</sub>X下提供了许多以 `\begin` 与 `\end` 括起来的环境, 例如 `equation`, `table` 等。除了这些标准的环境之外, 还允许用户自己去定义一些新的环境, 如定义“定理”环境、“例子

”环境等。下面以自定义的 example 环境为例说明如何建立自己的新环境。

```
\newtheorem{example}{\footnotesize 例 }
\newcounter{example}[chapter]
\def\theexample{\thechapter.\arabic{example}}
```

这里在其中第 1 个语句中用 `\newtheorem` 命令声明了一个新的环境 `example`，而其标题为小号的“例”字。第 2 个语句用 `\newcounter` 语句来对 `example` 环境计数进行定义，该计数在每个章即 `chapter`) 开始时清零。第 3 个语句定义了例子标号的格式，即将 `\theexample` 定义为章号 `\thechapter` 加点，再加数字型的例子序号，例如第 5 章的第 6 个例子编号为“例 5.6”。当然这样的定义方法只适用于 `book` 和 `report` 等文档风格，而不能用于 `article` 风格，因为在 `article` 风格下最高的级别为 `section`，而不是 `chapter`。在 `article` 风格下进行定义时，可以将上面的 `chapter` 字样替换成 `section` 即可。

用类似的方式还可以定义其它新的环境，如定理、定义等。用户还可以根据前面例子中的格式对其中某些细节进行修正，以适应自己的特殊要求。

### 3.6.3 L<sup>A</sup>T<sub>E</sub>X 环境的修改

L<sup>A</sup>T<sub>E</sub>X 的默认页号为阿拉伯数字，如果用户想用其它特定的方式来显示页号时，则需要采用再定义的方式来修改 `\thepage` 命令参数，具体的 L<sup>A</sup>T<sub>E</sub>X 命令为

```
\renewcommand{\thepage}{\Roman{page}}
\setcounter{page}{1}
```

其中 `\thepage` 是页号变量，而 `Roman` 命令将页码计数器 `page` 定义成大写罗马数字，其它可以采用的转换类型定义见下表。

类型名	<code>\Alph</code>	<code>\alph</code>	<code>\Roman</code>	<code>\roman</code>	<code>\arabic</code>	<code>\fnsympol</code>
含义	大写英文字母	小写英文字母	大写罗马数字	小写罗马数字	阿拉伯数字	希腊字符
举例	A, B, C, ...	a, b, c, ...	I, II, III, ...	i, ii, iii, ...	1, 2, 3, ...	$\alpha, \beta, \gamma, \dots$

第 2 个语句中的 `\setcounter` 命令用来设定计数器的值，例如这里将页码计数器的值设置成 1。若想在某处恢复成阿拉伯数字页码标志，则可以再使用

```
\renewcommand{\thepage}{\arabic{page}}
```

当然还可以用 `\setcounter` 命令对页码计数器重新赋值。

### 3.6.4 编号格式的重新定义

在编写大型的文档(如使用 `book` 和 `thesis` 风格文件)时，往往会发现有时标号系统也不尽人意。例如在这种风格下的排版中，公式的编号定义是按照章来设置的，例如第 5 章的第 123 式的编号方法为 (5.123)，若用户想按照节来编号，即若用户想将第 5 章，第 4 节的第 26 个公式，则应该修改风格文件。

若使用 2.09 版的 L<sup>A</sup>T<sub>E</sub>X，则打开 `book.sty` 文件，将该文件邻近结尾处的

```
\@addtoreset{chapter}{section}
\def\theequation{\thechapter.\arabic{equation}}
```

命令中的 `chapter` 字样改写成 `section` 字样即可。

这时再编译该文件，则该公式的编号将变为 (5.4.26)。如果用户使用 L<sup>A</sup>T<sub>E</sub>X2e 版，则打开相应的 CLS 文件，可以发现下面两条语句，若想采用前面的方法对公式重新编号，则同样将该语句中的 `chapter` 替换成 `section` 即可。

```
\@addtoreset{equation}{chapter}
\renewcommand\theequation{\thechapter.\@arabic\c@equation}
```

## 4 数学公式及图表的排版

高质量数学公式的排版是 L<sup>A</sup>T<sub>E</sub>X 软件的一个重要的特色。L<sup>A</sup>T<sub>E</sub>X 允许用户在正文的行中、单独的公式行中以及图表中插入数学公式。除了数学公式的排版以外，L<sup>A</sup>T<sub>E</sub>X 还提供了较丰富的绘图语句和表格制造语句，在这一节中将介绍 L<sup>A</sup>T<sub>E</sub>X 软件的这些特色。

### 4.1 L<sup>A</sup>T<sub>E</sub>X 中的数学环境设置

数学公式大致分为以下三类：文中 (in-line) 公式、独立公式和带编号的独立公式。所谓文中公式是指在不换行的情况下将一个数学公式嵌入文本行中，而独立公式一般是指单独占用一行或几行的数学公式。文件中的数学公式可以按照用户的选择进行编号处理，这种公式这里又称为带编号的数学公式。

文中公式是用单美元号 (\$) 括起来的一段文字，而独立公式是用双美元号 (\$\$) 括起来的文字。首先举出一个例子来说明文中公式的概念及设置：

In line formular is as  $y=x^2+1$ . The sum is  $\sum_{i=1}^{\infty} a_i$ .

这一段文字的显示结果如下：

In line formular is as  $y = x^2 + 1$ . The sum is  $\sum_{i=1}^{\infty} a_i$ .

这里的  $\wedge$  后面的部分为乘方表达式，这里后面只是一个字符，所以可以不使用大括号括起来，如果想把多个字符作为指数，则应该使用大括号。例如  $y=x^{\{abc\}}$  对应的排版效果为  $y = x^{abc}$ ，如果错误地写成  $y=x^abc$  对应的排版效果为  $y = x^a b c$ 。L<sup>A</sup>T<sub>E</sub>X 可以处理多级的上标表示，如  $y=x^{\{a^{\{z^{\{5\}}}\}}}$  可以得出下面的排版效果  $y = x^{a^{z^5}}$ 。前面例子中的  $_i$  标号后面的部分为下标表示，和上标表示方法类似，下标也可以作多级处理。

如果将上面例子中所有的单美元号改为双美元号时，则这一文字的显示结果为

In line formular is as

$$y = x^2 + 1.$$

The sum is

$$\sum_{i=1}^{\infty} a_i.$$

注意，在这种情况下除了每个公式都由单独的行居中显示以外，在求和公式上可以发现它和文中公式的显示格式是不同的。用户一定要注意，在公式处理中美元符号是成对出现的，前面若有一个 \$ 号开始公式，则在公式结束时切记要设置一个 \$，如果是以 \$\$ 开始的独立公式，



在公式结束时切记要给出 `$$`，否则将造成混乱。独立公式还可以由 `\begin{displaymath}` 和 `\end{displaymath}` 环境来设置，其作用与双引号括起来的完全等价。

除了前面介绍的两种公式环境以外，另外一种重要的（也是代表  $\text{\LaTeX}$  特色的）数学公式排版环境是带有编号的公式排版。这种排版方式是由 `\begin{equation}` 和 `\end{equation}` 环境来设置的，如果用户使用了这样的环境，则该公式的编号就会自动生成。例如前面的例子中 `$` 环境如果改用 `\begin{equation}` 和 `\end{equation}` 环境来取代，则相应的排版结果将为

In line formular is as

$$y = x^2 + 1. \quad (1)$$

The sum is

$$\sum_{i=1}^{\infty} a_i. \quad (2)$$

可以看出，这里的 (1) 和 (2) 标号是自动生成的，如果用户在 (2) 式前面加一个带编号的公式，则原来的 (2) 将自动地更新为 (3)，而插入的公式的编号将自动地写成 (2)。如果后面还有其它带编号的公式，则其标号会自动地变成所需的形式。

在数学模式下，则如果给出下面的命令 `$This is the Text$`，将给出如下的排版效果

*This is the Text .*

可以看出，原来的文中空格全部被取消了，这就是说在数学模式下，文中原来文本中的空格将自动被取消，如果想保留文本原来状态，则应该使用 `\mbox{}`，例如，如果上面的段落改成 `$\mbox{This is the Text}$`，则将给出如下的排版效果： *This is the Text*。可见，如果做了这种处理，则可以保留原来文本中空格的效果。

如果想把一个公式作多行多列显示，则可以使用  $\text{\LaTeX}$  的 `array` 环境。下面通过一个例子来演示这种环境的用法。如果在文本中给出了下面的命令：

```
f(x)=\left\{\begin{array}{l}1 \\ 1, \& x\geq 0 \\ -1, \& \mbox{else}\end{array}\right.\\ \right.\\
```

则会出现下面的排版结果

$$f(x) = \begin{cases} 1, & x \geq 0 \\ -1, & \text{else} \end{cases}$$

可以看出，这里的 `array` 环境设置了两列，在该环境的参数中分别有两个 `l` 字样，表示这两列都是靠左边对齐的，如果将 `l` 字样改成 `c`，则说明该列内容按中间对齐，如果该成 `r` 字样，则该项内容按照右侧对齐处理，用户可以自己去观察效果。

在 `array` 环境下每一行是由 `\\` 来分隔的，在同一行中不同的列是由 `&` 来分隔的。

在这个公式中还使用了 `\left` 和 `\right` 命令，对照前面例子的排版结果可以看出，前面的 `\left` 及后面跟的大括号 `{` 将按后面内容的高度自动生成一个合适大小的括号。按照  $\text{\LaTeX}$  的要求，每一个 `\left` 命令都要有一个 `\right` 命令来结束，在这里不需要一个 `}`，所以应该加一个 `\right.` 来做结束处理。如果将 `\left` 命令后面加的大括号还可以改变成其它的符号，例如可以变成圆括号或方括号。

从这个例子还可以看出， $\text{\LaTeX}$  可以根据后面内容实际的大小来设置括号的大小，除了自动设定括号的大小之外， $\text{\LaTeX}$  还允许用户按自己喜欢的方式来选择控制括号大小的命令，例如下面的一些命令

$$\begin{aligned} & \Bigg( \bigg( \Big( \big( ( ) \big) \Big) \bigg) \Bigg) \text{ ----} \\ & \Bigg[ \bigg[ \Big[ \big[ [ ] \big] \Big] \bigg] \Bigg] \text{ ----} \\ & \Big\{ \big\{ \Big\{ \big\{ \{ \} \big\} \Big\} \big\} \Big\} \end{aligned}$$

将给出如下的排版结果

$$\left( \left( \left( \left( ( ) \right) \right) \right) \right) \quad \left[ \left[ \left[ \left[ [ ] \right] \right] \right] \right] \quad \left\{ \left\{ \left\{ \left\{ \{ \} \right\} \right\} \right\} \right\}$$

## 4.2 常用数学公式的排版

前面曾经介绍过上下标的 L<sup>A</sup>T<sub>E</sub>X 排版命令，在这里将简单介绍一些其它数学公式形式的排版命令。

- **上下标排版:** 前面介绍过，上下标是由  $\sim$  和  $-$  引导的。
- **分式的排版:** 分式是由 `\frac{}{}` 命令来处理的，其中第一个大括号括起的是分式的分子，第二个括起来的是分母，例如

`\frac{x+y}{x-y}` 将给出  $\frac{x+y}{x-y}$

下面给出的连分式排版命令

将得到如下的排版效果

$$\frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{x}}}}$$

可以看出，如果单纯地使用 `\frac` 命令在作多级嵌套式的分式排版，则嵌套级别越深，字符就越小。如果想解决这样的问题，就可以使用 `\displaystyle` 命令了，这一命令可以使得它所修饰的内容以正常大小的字符来显示。重新考虑上面的例子

将得到如下的排版效果

$$\frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{x}}}}$$

可见，这样处理之后排版效果就美观得多了。

- **根式的排版:** 根式的排版命令格式是

`\sqrt[开方次数]{根式的内容}`

如果开方次数不填写，则认为是开平方，在排版时也就不作任何修饰了。根式也可以作多级嵌套，它也会使得后面嵌套的内容变得很小，这样也可以使用前面介绍的 `\displaystyle` 命令，例如

将得到如下的排版效果

$$\sqrt[3]{1 + \sqrt[4]{1 + \sqrt[4]{1 + \sqrt{x}}}}$$

- **求和、求积分及其它:** 对于求和、求积分以及求极限等特殊的数学表达式来说，L<sup>A</sup>T<sub>E</sub>X 提供了方便的排版命令。考虑下面的例子<sup>5</sup>

<sup>5</sup>为简单起见，这里的  $dx$  是非标准写法，正确的应该为  $dx$ ，其 L<sup>A</sup>T<sub>E</sub>X 表示为 `\rm d`  $x$ ，请参见附录 C。

$$\sum_{i=1}^{\infty} a_i, \prod_{j=1,2,\dots,n} P_j, \int_{-\infty}^{\infty} f(x)dx, \lim_{t \rightarrow \infty} e(t), \int \int \int f(x,y,z)dx dy dz$$

上面的例子将给出如下的排版效果

$$\sum_{i=1}^{\infty} a_i, \prod_{j=1,2,\dots,n} P_j, \int_{-\infty}^{\infty} f(x)dx, \lim_{t \rightarrow \infty} e(t), \int \int \int f(x,y,z)dx dy dz$$

在数学模式下  $\infty$  将给出  $\infty$ ，而  $\rightarrow$  将给出箭头  $\rightarrow$ 。其它的命令可以容易地对照上面的例子得出，详细情况请参照附录 A 中列出的各个符号集。

从上面例子可以看出，在  $\sum$  等命令下也使用了“上下标”符号  $\wedge$  和  $\_$ ，但这里的“上下标”的位置与普通的上下是不同的，这说明 L<sup>A</sup>T<sub>E</sub>X 在这种处理中做了特殊的规定。从最后一个式子中还可以看出，如果直接使用若干个  $\int$  命令，所得到的排版结果中积分号之间的距离过大，这就需要利用 L<sup>A</sup>T<sub>E</sub>X 中给出的调间距的命令来调整了。在最后一个例子中各个  $\int$  之间加一些距离控制符，则可以获得如下表中所示的排版效果。

距离符号	含义	例子
	默认距离	$\int \int \int f(x,y,z)dx dy dz$
$\,$	小距离	$\int \int \int f(x,y,z)dx dy dz$
$\:$	中距离	$\int \int \int f(x,y,z)dx dy dz$
$\;$	大距离	$\int \int \int f(x,y,z)dx dy dz$
$\!$	负距离	$\int \int \int f(x,y,z)dx dy dz$

这种距离控制命令可以混合在一起使用，例如如果将表中最后一项的改成  $\! \! \!$ ，则将得到下面的排版效果

$$\int \int \int f(x,y,z)dx dy dz$$

这时的效果要好得多。一般情况下，L<sup>A</sup>T<sub>E</sub>X 可以给出相当好的默认距离，所以，除非必须要改变距离，一般不要轻易改变默认的距离设置。

- **矩阵的表示方法:** L<sup>A</sup>T<sub>E</sub>X 提供了给矩阵排版的命令，其命令格式为  $\matrix{}$ ，其中大括号中应该将矩阵的元素写入，具体的矩阵写入方法为：矩阵同一行的元素用  $\&$  符号来分隔，换行符号为  $\cr$ 。例如下面的 L<sup>A</sup>T<sub>E</sub>X 命令可以得出右面的排版结果。

```


$$\left[ \begin{matrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{matrix} \right]$$


```

在前面的例子中, 命令 `\left[` 和 `\right]` 用来给产生的矩阵加括号, 在实际应用中, 也可以使用其它的括号表示。L<sup>A</sup>T<sub>E</sub>X 提供了各种省略符号, 请参见附录 A 中列出的一些符号。利用这些省略符号和矩阵排版功能, 就可以将下面的对称三对角矩阵表示出来。

```

 $\left[\begin{matrix} b_0 & c_0 & & \\ c_0 & b_1 & c_1 & \\ & c_1 & b_2 & c_2 \\ & & \ddots & \ddots & \ddots \\ & & & c_{n-3} & b_{n-2} & c_{n-2} \\ & & & & c_{n-2} & b_{n-1} \end{matrix}\right]$ 

```

排版效果为

$$\begin{bmatrix} b_0 & c_0 & & & \\ c_0 & b_1 & c_1 & & \\ & c_1 & b_2 & c_2 & \\ & & \ddots & \ddots & \ddots \\ & & & c_{n-3} & b_{n-2} & c_{n-2} \\ & & & & c_{n-2} & b_{n-1} \end{bmatrix}$$

- **重叠符号的处理:** 在数学模式下, L<sup>A</sup>T<sub>E</sub>X 提供了多种重叠符号的排版方法, 最常用的有 `\underline{}` 和 `\overline{}` 命令, 它们分别用于在括号中的内容加下划线和上划线, 其中 `\underline{}` 的作用和文本中的命令是相似的。除了在括号中的公式上面或下面加直线之外, 还可以使用 `\underbrace{}` 和 `\overbrace{}` 来在括号中的内容下面或上面加上花括号。这两个命令还可以分别带“下标”和“上标”, 其作用是在花括号的下面和上面注上其它的公式。例如

```


$$\overbrace{a_0 + \underbrace{a_1 + a_2 + \cdots + a_{n-1}}_{(n-1) \text{ 项}}}^{(n+1) \text{ 项}} + a_n = \overbrace{\cdots}^{(n+1) \text{ 项}}$$


```

将给出如下的排版效果:

$$\overbrace{a_0 + \underbrace{a_1 + a_2 + \cdots + a_{n-1}}_{(n-1) \text{ 项}}}^{(n+1) \text{ 项}} + a_n$$

L<sup>A</sup>T<sub>E</sub>X 还提供了 `\stackrel{a}{b}` 命令来任意叠放两个公式, 其中前一个大括号放置上层的公式, 而后一个大括号放置下层的公式。例如

```


$$A \stackrel{a'}{\rightarrow} B \stackrel{b'}{\rightarrow} C$$


```

将给出如下的排版效果

$$A \stackrel{a'}{\rightarrow} B \stackrel{b'}{\rightarrow} C$$

- **带有多个编号的公式组:** L<sup>A</sup>T<sub>E</sub>X 提供了 `eqnarray` 环境来定义一组公式, 该环境类似于 `array` 环境, 所不同的是该公式组中默认情况下每一个公式均有一个编号。若用户要求其中某一个或多个公式不编号, 则应在该公式中加入 `\nonumber` 命令。考虑下面的一个段落

```


$$\begin{eqnarray} (a+b)^4 & = & (a+b)^2 + (a+b)^2 \\ & = & (a^2+2ab+b^2)(a^2+2ab+b^2) \nonumber \\ & = & a^4+4a^3b+6a^2b^2+4ab^3+b^4 \end{eqnarray}$$


```

将给出下面的排版效果

$$(a+b)^4 = (a+b)^2 + (a+b)^2 \quad (3)$$

$$= (a^2 + 2ab + b^2)(a^2 + 2ab + b^2)$$

$$= a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + b^4 \quad (4)$$

若想使用 `eqnarray` 环境而不想对所排版的公式进行编号, 则可以由 `\begin*{eqnarray}` 和 `\end*{eqnarray}` 来调用该环境。

### 4.3 数学公式中字体的改变

在数学公式中一般英文字符的默认字体为斜体字, 用户还可以通过一些设置来改变公式中的字体, 例如如果用户用了 `\cal` 定义的话, 则字体将变成手写体, 但应该注意, 这种定义只可以用于数学模式下, 而不能用于文本模式下。后面的排版命令 `\cal ABCDEFGHIJKLMNOPQRSTUVWXYZ` 将给出如下的排版结果

*ABCDEFGHIJKLMNOPQRSTUVWXYZ*

注意, 这种定义只可以用于大写字母的字体设置。

$\text{\LaTeX}$  还允许用户在数学模式下使用希腊字符, 如 `\alpha` 将给出  $\alpha$  字符, 而 `\Gamma` 将给出  $\Gamma$  字符。其它的希腊字符可以参照附录 A。除了标准的希腊字符之外,  $\text{\LaTeX}$  还提供了大量的数学符号和其它符号, 例如 `\spadesuit` 将给出  $\spadesuit$ , 而 `\heartsuit` 将给出  $\heartsuit$ , 这样的一些符号也请参见附录 A。

在数学模式下不可以直接改变公式中的字号大小, 如果想改变整个公式的字号大小, 则可以在公式标志的外面加上 `\large` 等命令就可以了, 在公式结束后可以使用 `\normalsize` 恢复原来的字号大小。当然最好还是用局部命令来完成这样的工作, 即用大括号将要改变字号的段落括起来。如果想改变公式中某一个字符或字符串的字号大小, 则首先应该将要改变的部分通过 `\mbox{}` 命令转换成文本模式, 然后对该部分可以独立地改变字号的大小及字体的设置了。若用户使用中文版  $\text{\LaTeX}$ , 则在数学模式下也不允许直接使用中文字符, 如果想使用也需要首先通过 `\mbox{}` 命令将之转换成文本模式。下面将通过一个例子来说明这种情况, 如果用户给出了 `\$x=y\mbox{\Large $Z$}^2+\mbox{中文}`, 则将获得下面的排版效果  $x = yZ^2 + \text{中文}$ 。

除了标准的数学符号之外, 用户还可以定义自己的特殊符号, 例如用户可以由下面语句定义一个 `\realR` 命令

```
\newcommand{\realR}{I \hspace*{-1.2mm} R}
```

该符号其实由 I 和 R 两个字母组成, 其中 R 字符在排版时向左移动 1.2mm, 在数学模式下该命令给出的排版效果为  $\mathbb{R}$ 。

### 4.4 $\text{\LaTeX}$ 对图表的处理

$\text{\LaTeX}$  提供了一个绘图环境, 这一环境是由 `\begin{picture}` 和 `\end{picture}` 括起来的,  $\text{\LaTeX}$  还提供了丰富的绘图命令, 但由于它使用起来并不是特别方便, 所以出现了很多专门为  $\text{\LaTeX}$  设置的绘图软件, 现在一般很少有人还在直接使用标准的  $\text{\LaTeX}$  命令直接写出图形

来。后面将要介绍一个为 L<sup>A</sup>T<sub>E</sub>X 设计的绘图软件 T<sub>E</sub>Xcad。该软件允许用鼠标器或键盘画出所需的图形，然后可以依照 L<sup>A</sup>T<sub>E</sub>X 提供的画图命令来自动生成一个 T<sub>E</sub>X 源文件来。

建立起来图形之后，则可以用 picture 环境来直接在 L<sup>A</sup>T<sub>E</sub>X 文件中插入图形，然而一般情况下用户往往会使用 figure 环境来插入图形，这种方式在 L<sup>A</sup>T<sub>E</sub>X 下称作浮动图形，它的作用和直接使用 picture 环境是不同的，picture 环境生成的图形必须处于文本中的当前位置，由于有时并不允许这种设置，所以最好要使用 figure 环境。这一环境的调用方式为：

```
\begin{figure}[图形位置选项]
\begin{center}
\begin{picture}(图形坐标参数)
图形的实际内容
\end{picture}
\end{center}
\caption{图形的名称 }
\end{figure}
```

其中图形位置选项一般有 h (当前位置)、t (页面顶部)、b (页面底部) 和 p (单独一页) 四个选项，但在实际中有时难以保证会按照指定位置放置图形，所以一般在该选项中可以填写 [htb]，其含义为，如果可以放置在当前位置，则放置在当前位置，否则可以试验顶部位置，如果仍不能放置则再试验底部。如果不能按照要求插入图形，则 L<sup>A</sup>T<sub>E</sub>X 会自动安排一个比较合理的位置来插入图形。注意，方括号中的选项排列的是有顺序的，其作用是依照给出选项的优先顺序来试验的。如果插入的图形较大，则还可以 p 选项，即使用 [htbp] 来控制图形位置。

定义了图形的位置之后，还可以使用 center 环境将图形作居中处理，当然也可以按照用户自己的需要作其它设置。然后在这样的环境中把图形的内容插入。绘制完图形之后，用户还可以在后面加一个 \caption{} 为插入的图形自动加一个编号，并产生一个图形题目 (由大括号中给出)，最后结束图形环境。

L<sup>A</sup>T<sub>E</sub>X 还专门为用户设计了一个十分有用的制表环境 tabular，这一制表环境和数学公式中的 array 环境极其相似，例如

```
\begin{tabular}{c || l | r c | }
```

引导的制表环境定义了具有四列元素的表格，其中要求第一列的各个元素按照中间对齐 (由 c 决定)，然后在第一列与第二列中间加两条竖线 (由 || 决定)。第二列元素由于给出了 l 定义，所以应该靠左边对齐，而第二列与第三列中间应该有一条竖线分隔，第三和第四列分别按右边和中间对齐，且二列之间没有竖线来分隔。第四列之后应该有两条竖线来结束。

和 array 环境相似，每一个表行中，各个表元素之间是以 & 来分隔的，而每一行结束后应该有结束符 \\ 来表示。如果在一行的末尾加一个 \hline 命令，则将在该表中整个一行的下面加一条横线来分隔。

如果想在表格内对每行表的元素作修改，则可以使用 \multicolumn{} 命令来处理，下面给出一个例子来说明 L<sup>A</sup>T<sub>E</sub>X 制表的语句及应用。

```
\begin{tabular}{l l | c r | r} \hline
\multicolumn{2}{c | }{Left} & Center & \\
\multicolumn{2}{c | }{Right} & \\\hline
123.456 & 112 & 11132 & 12345 & 12 \\\end{tabular}
```

```
My Text & 1562 & My Tests & tttt & MMMMMM \\ \hline
\end{tabular}
```

前面的例子将给出如下所示的排版效果

Left		Center		Right
123.456	112	11132	12345	12
My Text	1562	My Tests	tttt	MMMMMM

从上面的例子中可以看出 `\multicolumn{}` 命令的使用方法, 该命令共带有三个大括号, 第一个大括号中的数值是要合并的列的个数, 第二个大括号中的参数应该为列合并后的定义 (亦即对齐方式与竖线设置方式), 第三个大括号中的内容是列合并后的显示内容。

在考虑下面一个常见的表格

Left		Center		Right
123.456	112	11132	12345	12
My Text	1562	My Tests	tttt	MMMMMM

该表格可以由下面的语句来建立。

```
\begin{tabular}{l l | c r | r} \hline
\multicolumn{2}{c |}{Left} & Center & \\
\multicolumn{2}{c}{Right} \\ \hline
123.456 & 112 & 11132 & 12345 & 12 \\ \cline{3-5}
My Text & 1562 & My Tests & tttt & MMMMMM \\ \hline
\end{tabular}
```

可见使用 `\cline` 命令可以画出一条横线, 它可以只对其中某一些列起作用。

表格的行间距也是可以修改的, 这应该通过 `\renewcommand` 命令来重新定义, 如果在表格定义的前面加入下面命令

```
\renewcommand\arraystretch{2.0}
```

则新的排版效果如下

Left		Center		Right
123.456	112	11132	12345	12
My Text	1562	My Tests	tttt	MMMMMM

可以看出, 这里用了 `\renewcommand` 命令对原来的命令 `\arraystretch` 做了修正 (变成标准值的 2.0 倍), 所以原来表的行距加大了。L<sup>A</sup>T<sub>E</sub>X 还允许用户通过重新定义 `\tabcolsep` 命令参数修改表的列与列之间距离。

表格设置环境 `table` 的作用和前面介绍的 `figure` 环境相似, 该环境称作浮动表格, 也可

以按照用户指定的位置来安排。用户也可以用下面的格式来插入一个表格

```
\begin{table}[表格位置设选项]
\caption{表格的名称 }
\begin{center}
\begin{tabular}{格式定义 }
表格的实际内容
\end{tabular}
\end{center}
\end{table}
```

其中表格位置选择项的作用和前面介绍的图形位置是极其相似的，也含有相同的四个选项。这里的 `\caption{}` 命令的位置和在图形环境的不太一致。

L<sup>A</sup>T<sub>E</sub>X 在 `tabular` 环境中还定义了一个 `p` 选项，其使用方法为 `p{宽度  $w$ }`，意即该表列为宽度为  $w$  的文本，超过  $w$  的部分将自动转到下一行，该选项可以与前面介绍的 `c`，`l`，`r` 等同样使用。考虑下面的排版语句

```
\begin{tabular}{l | p{8cm} | c}\hline
1). & In the tabular environment, a tabular column of width 8cm
is used to demonstrate the use of p option. It is noted that the
results of p is different from c, l and r &  $r=a_2^5$  \\ \hline
the second row & This is to show that the words are wrapped according
to the 8cm limit & hhhha \\ \hline
\end{tabular}
```

将给出如下的排版效果，可见其中第 2 列的整个宽度为 8 厘米，而其中的文本宽度略小于 8 厘米。

1).	In the tabular environment, a tabular column of width 8cm is used to demonstrate the use of p option. It is noted that the results of p is different from c, l and r	$r = a_2^5$
the second row	This is to show that the words are wrapped according to the 8cm limit	hhhha

#### 4.5 浮动处理的参数及修正

在默认的浮动图表处理中，L<sup>A</sup>T<sub>E</sub>X 要求每一个页面上不能超过 70% 被图表占用，或每一个页面上的图表个数不能超过 2 个。在实际排版中往往会发现这样会使得实际图表位置和应该出现的位置差好几个页面，或出现浮动图表过多，无法处理的错误。这样就应该对制约浮动图表定位的一些参数进行重新定义。下面给出若干个制约定位的参数的默认值与修正方法，以使得用户获得更好的排版效果。

- `\topfraction` 参数：其含义为页面顶部可以被浮动图表占用的百分比，其默认值为 0.7，即 70% 可以为浮动图表占用。事实上，若将该参数选得较大时会得出更好的排版效果，



建议通过命令

```
\renewcommand{\topfraction}{0.98}
```

来定义该参数，亦即 98% 的位置可以为浮动图表占用。与之相应的参数 `\textfraction` 也可以通过相应的命令设置成一个较小的值，如 0.02。

相应地，页面底部的浮动图表占用率也可以通过 `\bottomfraction` 参数来重新定义，其默认值为 0.3，当然不妨将该参数定义为较大的值。

- `\topnumber` 参数：其含义为可以在页面顶部最多定位的浮动图表个数，其默认值为 2，用户可以通过下面的命令将该值设置成 4。

```
\setcounter{\topnumber}{4}
```

当然用户还可以将允许的页面底部浮动图表个数 (`\bottomnumber` 参数) 重新定义赋值，该参数的默认值为 1。

- `\floatsep` 参数：该参数定义了浮动图表上下端留出的空白位置，该参数值和正文的字号定义有关，对 10 和 11pt 的正文来说，该参数的默认值为 12pt，对 12pt 的字号定义该参数的默认值为 14pt。其实用户可以用下面的命令来较小该参数的值

```
\setlength{\floatsep}{4pt}
```

- `\floatpagefraction` 参数：该参数定义了一个浮动页面完全被浮动图表占用的百分率，其默认值为 0.5，该参数可以用 `\renewcommand` 命令来修改。

## 5 交叉引用的处理方法

前面不止一次地提到了 L<sup>A</sup>T<sub>E</sub>X 排版软件的自动编号与交叉引用的功能，并把这一点作为 L<sup>A</sup>T<sub>E</sub>X 的主要特色加以强调。在这一节中将介绍 L<sup>A</sup>T<sub>E</sub>X 软件的交叉引用方法及其应用。

所谓交叉引用就是指在文件中引用其它地方出现的公式号、图表号、章节号以及页号等。这种工作如果可以自动地完成，无疑将给用户提供了极大的方便。

在前面介绍公式、图表中曾介绍过了它们的自动编号方法，如公式环境下如果用了 `equation` 环境，则可以对公式进行自动编号。如果在公式前面再插入一个 `equation` 环境，原来的编号将自动地变成新的编号。对图形环境 `figure` 和表格环境 `table` 来说，如果使用了 `\caption{}` 命令，则也将会自动地编号。

如果想把编号作为一个标号记录下来，则可以使用 L<sup>A</sup>T<sub>E</sub>X 提供的 `\label{}` 命令，其中大括号中的字符串就是要记录的标号。应该记住，在整个文件中不可使用相同的字符串作为标号，否则在编译中将给出重复定义错误的警告。标号命令的使用位置是很重要的，在对章节进行标号时，则 `\label{}` 命令应该写在相应的章节定义语句之后，公式进行标号时，则 `\label{}` 命令应该写在 `\begin{equation}` 命令的后面，对图表的标号则应该写在 `\caption{}` 命令的后面。一般情况下，这里所谓写在某命令的后面是指中间不能隔有其它命令。

如果想在正文中引用某一个标号，则可以使用 `\ref{}` 命令来完成。为了说明自动进行交叉引用的优越性，可以通过下面的命令来理解：

Eqn.~(\ref{myeq1}) in Section~\ref{mysec1} will be shown later.

... ..

\subsection{My Next Section} \label{mysec1}

My Next Equation is given below

\begin{equation} \label{myeq1}

\dot{x}\_i=f\_i(x\_1,\cdots,x\_n,t)+\xi\_i(t), \quad i=1,2,\cdots,n

\end{equation}

其排版效果如下:

Eqn. (5) in Section 5.1 will be shown later. ... ..

## 5.1 My Next Section

My Next Equation is given below

$$\dot{x}_i = f_i(x_1, \cdots, x_n, t) + \xi_i(t), \quad i = 1, 2, \cdots, n \quad (5)$$

这样做有一个明显的优点: 无论在该节前面加减多少章节, 无论在该公式前后加减多少其它的公式, 在正文中对它们所引用的编号都可以自动地生成, 这些编号将完全对应于它所引用的编号。

如果在某一个页中应用了 \label{} 命令, 则同时也可以将其页码进行标号。若在其它的部分想引用某个页码对应的页号时, 则可以使用 \pageref{} 命令来完成, 这一命令的使用方法和 \ref{} 命令的使用方法是完全一致的。

重新考虑前面的例子, 如果第一句改变为

Eqn.~(\ref{myeq1}) in Section~\ref{mysec1} on page~\pageref{mysec1}  
will be shown later.

时, 将得到下面的排版结果

Eqn. (5) in Section 5.1 on page 26 will be shown later.

在这里, 除了章节编号以外, 该章节所在的页码也被自动地引用了。

L<sup>A</sup>T<sub>E</sub>X还提供了特殊的命令, 允许用户按指定的格式列写参考文献的内容, 然后可以自动生成参考文献, 并允许在正文中对参考文献的编号进行引用。

L<sup>A</sup>T<sub>E</sub>X规定的参考文献自动生成环境的名称为 thebibliography, 该环境的定义格式由下面例子说明

\begin{thebibliography}{99}

\bibitem{Knu01}

Knuth, D. E., The \TeX Book, Addison-Wesley

\bibitem{Lamp01} Lampport, L., 1985, \LaTeX: A Document

Preparation System, Addison-Wesley

\end{thebibliography}

在这里数字 99 为该环境要求的参数, 是用户指定的最大的参考文件编号, 用户也可以指定为其它值。在 thebibliography 环境下, 每个参考文献项应该由 \bibitem{} 命令来引导, 在大括号中应给出该参考文献的标记, 后面可以按照任意的格式来书写参考文献的内容, 用

户可以在这一环境下用相同的格式列出所有的参考文献来, 注意总的参考文献个数不要超过给定的最大编号, 否则应重新定义最大编号。

经过适当的编译之后 (一般如果有改动, 则需要两次编译), 前面的命令将给出如下的排版结果:

## 参考文献

[1] Knuth, D. E., The  $\text{\TeX}$ Book, Addison-Wesley

[2] Lamport, L., 1985,  $\text{\LaTeX}$ : A Document Preparation System, Addison-Wesley

可以看出, 这里所列的两个参考文献的序号都自动地产生了。同时, 这段命令还给出了两个标记 `Knu01` 和 `Lamp01` 可以使用。如果在文章中想引用这些参考文献, 则可以使用  $\text{\LaTeX}$  定义的 `\cite{}` 引用命令来实现, 该命令中大括号的内容为要引用的标号。例如下面的命令

```
See Ref.~\cite{Knu01} for details.
```

将给出如下的排版效果

See Ref. [1] for details.

这里的方括号是自动加上的。 `\cite` 命令还允许带其它选项, 它的一般格式为

`\cite[extra information]{label for reference}`

其中

`extra information`: 参考文献引用时的附加信息, 在下面的例子中将演示参考文献附加信息的添加方法。

`label for reference`: 与 `\cite` 命令相对应的参考文献标号。另外, 在 `\cite` 命令中还可以列入多个标号, 各个标号是用逗号分隔的, 但应该注意, 各个标号之间不能有空格。

```
\TeX~ and \LaTeX~ has been discussed in Refs \cite{Knu01,Lamp01}, see  
Ref \cite[pages 11-50]{Knu01} for more details.
```

将给出如下的排版效果

$\text{\TeX}$  and  $\text{\LaTeX}$  has been discussed in Refs [1, 2], see Ref [1, pages 11-50] for more details.

## 6 $\text{\LaTeX}$ 的高级使用技术

### 6.1 列表环境

常用的列表环境有下面三种: `itemize`, `enumerate` 和 `description`。在最前面介绍  $\text{\LaTeX}$  突出特色时就使用了 `itemize` 环境, 可以看出, 该环境的特点是在每一个列表项的前面都由一个圆的黑点 (•, 即  $\text{\LaTeX}$  的 `$\bullet$`) 来引导, 换行后列表的内容按照实际的文字对齐, 这样就可以突出前面的黑点, 使得排版效果十分美观。

在列表环境下还可以嵌套其它的列表环境, 这样便使得  $\text{\LaTeX}$  排版效果错落有致。考虑下面一个例子

```

\begin{itemize}
\item {\bf Approximate methods}: Summarised as
\begin{itemize}
\item Random Describing function method, where the signals within
the system are assumed to be Gaussian, so that the nonlinear
element is approximated by a variance-dependant gain.
\item Smith's functional approximation method.
\end{itemize}
\item {\bf Exact method}: Normally the Fokker-Planck equations ...
\end{itemize}

```

上面一段命令的排版效果如下:

- **Approximate methods:** Summarised as
  - Random Describing function method, where the signals within the system are assumed to be Gaussian, so that the nonlinear element is approximated by a variance-dependant gain.
  - Smith's functional approximation method.
- **Exact method:** Normally the Fokker-Planck equations ...

使用列表环境, 每一个列表项都是由 `\item` 命令来引导的。在第一级列表项中,  $\text{\LaTeX}$  定义为由圆点引导的, 第二级列表项是由 `-` 引导的, 后面的依次为 `*` 和 `·` 引导的, 这些引导符号是在风格文件 (STY 文件或 CLS 文件) 中设置的, 用户可以自己根据需要进行改动。

如果将前面例子中的 `itemize` 环境全部改成 `enumerate` 环境, 则将出现如下的排版效果

- 1). **Approximate methods:** Summarised as
  - (a) Random Describing function method, where the signals within the system are assumed to be Gaussian, so that the nonlinear element is approximated by a variance-dependant gain.
  - (b) Smith's functional approximation method.
- 2). **Exact method:** Normally the Fokker-Planck equations ...

这里的序号 1 和 2 与 (a) 和 (b) 是自动生成的, 和公式的自动排序一样, 如果用户在前面再加一条列表项, 则后面的序号也将自动发生变化, 以产生和实际相吻合的结果。

注意: 在各种列表环境进行嵌套时,  $\text{\LaTeX}$  规定至多可以嵌套四级, 这一般来说是足够的了。description 环境和前面叙述的两种环境略有不同, 在这种环境的设置下将不产生列表项标记与序号。

使用列表环境有时会发现, 每个列表项之间的距离可能过宽或过窄等现象, 这有时要求用户自己去定义其中一些列表参数。各个列表参数的含义如图 1 所示, 而各个长度单位都可以通过直接赋值的方法去修改。

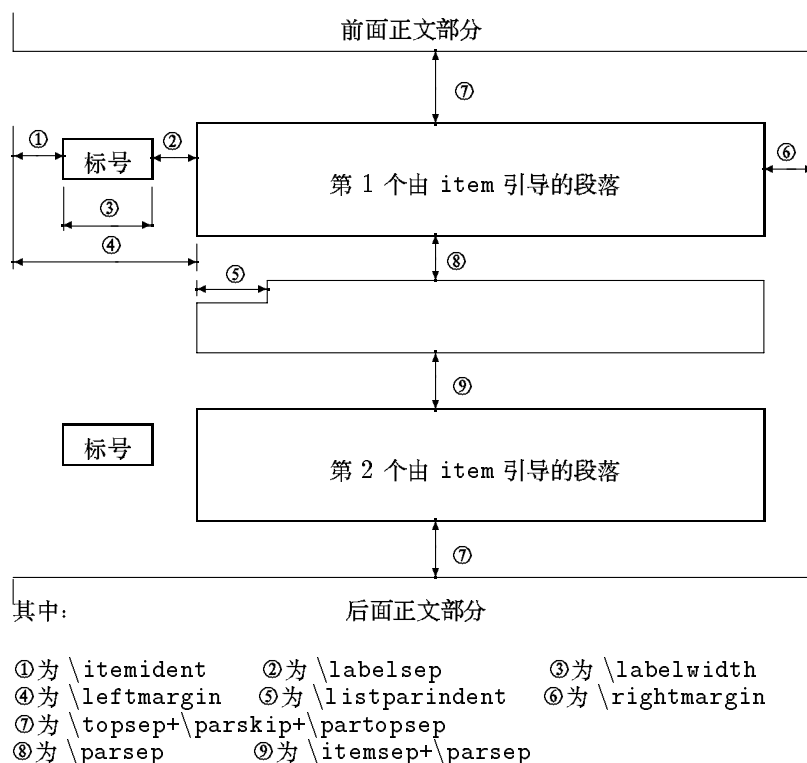


图 1 列表环境参数示意图

## 6.2 小型页环境

L<sup>A</sup>T<sub>E</sub>X提供了一种特殊意义的小型页 (minipage) 排版环境，它的作用相当于在整个排版页面下插入一个小的方框，用户可以将该方框看成一个页面，在该页面下作一些特殊的处理。

小型页面环境的设置命令为：

```
\begin{minipage}[小型页位置]{小型页宽度}
}
小型页的排版内容
\end{minipage}
```

小型页的位置和浮动图表的设置方式是相似的，但含义有所不同。小型页的位置可以使用 t, b 和 c 的记号来标明，如果使用 b 选项，则小型页的底部和其它的文字行对齐，若使用 t 选项则小型页的顶部与其它文字对齐，若使用 c 选项，则小型页按中心与其它文字对齐。页面宽度的设置方法也是很简单的，用户在该括号中给一个长度的值就可以了。小型页的高度是根据它包含的内容由 L<sup>A</sup>T<sub>E</sub>X自动产生的，用户不必在这里费太多的心思。但应该记住，当 minipage 环境跨页时，则其定义失效。小型页本身可以看成是一个“正常”的页面，它可以带有自己的脚注、公式环境、图表等，小型页本身还可以作嵌套。下面通过一个例子来说明小型页面的设置及使用。

```
The minipage Started ~~~~~
\begin{minipage}[c]{6cm}
```

```

This is the paragraphs shown in the minipage defined.

$$f(x)=2x^2+3x+3$$

\end{minipage}

```

The minipage Started      This is the paragraphs shown in the minipage defined.

$$f(x) = 2x^2 + 3x + 3$$

这里的小型页的页面宽度设置为 6cm, 该页内带有自己的数学公式 (居中处理)。因为这里使用了 c 位置设置选项, 所以该小型页的位置和前面一段文字是按照中心对齐的。

### 6.3 风格文件简介

L<sup>A</sup>T<sub>E</sub>X 的一些特殊的设置是由相应的 STY 或 CLS 文件规定的, 其中 article 文档风格是由 article.sty 和其它相应的文件指定的, 这里的其它文件是指 art10.sty (对应于 10pt 设置), art11.sty (对应于 11pt 设置), art12.sty (对应于 12pt 设置) 三个文件, 在给定的字号下, 其中只有一个文件起作用。对 2e 版本来说, 相应的文件为 article.cls, size10.clo, size11.clo 和 size12.clo。对应于 book.sty 及其字号定义文件, 2e 版中相应文件为 book.cls, bk10.clo, bk11.clo, bk12.clo 等。

对其它风格如 book 等也是一样, 它将对应于 book.sty, bk10.sty, bk11.sty, bk12.sty 文件。在 article.sty 等文件中, 将定义和字号大小无关的一些默认设置, 而其它文件将作和字号相关的一些设置。

这些文件都是可读的, 其语句结构也不是很难, 只要用户细心去读, 对其中一些默认设置进行修改还是可能的。

下面是 article.sty 文件中关于列表环境的一些默认定义语句, 这里分两个部分来定义列表环境, 前一个部分定义 enumerate 环境, 后一个部分定义 itemize 环境。先观察下面的第一个语句, 在这个语句中的 \arabic{enumi} 的含义是将第一级列表环境计数器的内容 enumi 用阿拉伯数字的形式 (arabic) 显示出来, 它后面的点表示在显示完数字之后再显示一个圆点。后面第二级的定义方式中使用了 \alph 命令, 且该命令是用圆括号括起来的, 这就是说第二级符号生成时是以圆括号括起的英文字母来标注的。

```

\def\labelenumi{\arabic{enumi}.}
\def\theenumi{\arabic{enumi}}
\def\labelenumii{\alph{enumii}}
\def\theenumii{\alph{enumii}}
\def\p@enumii{\theenumi}
\def\labelenumiii{\roman{enumiii}.}
\def\theenumiii{\roman{enumiii}}
\def\p@enumiii{\theenumi(\theenumii)}
\def\labelenumiv{\Alph{enumiv}.}
\def\theenumiv{\Alph{enumiv}}
\def\p@enumiv{\p@enumiii\theenumiii}

\def\labelitemi{$\bullet$}
\def\labelitemii{\bf --}
\def\labelitemiii{$\ast$}
\def\labelitemiv{$\cdot$}

```

在后面部分中是关于 `itemize` 环境的设置，可以看出这四级列表环境的标号分别为  $\bullet$ ,  $-$ ,  $*$  和  $\cdot$ 。如果用户不喜欢这种排列，则可以自己改变排列顺序，或加上自己喜欢的符号。

但要注意，在修改风格文件之前，一定要将要修改的风格文件复制到用户自己的当前目录下，否则如果修改了之后，将影响其它人的排版效果。

## 6.4 中文版 L<sup>A</sup>T<sub>E</sub>X 的风格定义与修正

中文版 L<sup>A</sup>T<sub>E</sub>X 提供了 2.09 版的风格文件 `carticle.sty`, `cbook.sty` 等，但没有提供 2e 版的 CLS 文件。这样用户只好自己去生成相应的 CLS 文件，例如用户可以首先将 `texinput\latex2e` 目录下的 `article.cls` 复制到当前目录下并改名为 `carticle.cls`，打开该文件并在 `\NeedsTeXFormat{LaTeX2e}` 之前应该插入如下命令

```
\input cthead.sty \relax
```

这样就可以实现中文处理了。作者生成了若干包含中文处理的 `.cls` 和 `.clo` 文件，并修正了其中的很多内容，以备调用。同时作者还修改了若干 STY 文件，使得最终排版效果更实用。

## 6.5 其它图形的插入处理

如果有一个由标准 L<sup>A</sup>T<sub>E</sub>X 命令生成的图形在另一个文件给出 (如由后面将介绍的 T<sub>E</sub>Xcad 或 PICT<sub>E</sub>X 软件生成的图形文件)，则可以简单地采用 `\input{}` 命令在图形中来引用它，例如有一个名为 `mypic.tex` 的图形文件，用户可以用下面的命令来调入之：

```
\begin{figure}[htb]
\centerline{\input{mypic}}
\caption{My imported figure} \label{myrefs}
\end{figure}
```

注意，这里文件名中缺省后缀名，L<sup>A</sup>T<sub>E</sub>X 要求插入的文件名一定要用 TEX 为后缀。

如果插入其它类型的文件时，则应该特殊进行处理。例如如果用户有一个以 EPS 为后缀的图形文件，则可以通过 `\epsffile{}` 命令来包含该文件，同时在 `\documentstyle{}` 命令的选项中包含 `epsf.sty` 文件，或在 2e 版下使用 `\usepackage` 命令来包含该文件，这在后面将介绍。EPS 文件定义的图形宽度可以容易地由 `\epsfxsize{}` 命令设置。EPS 文件要求激光打印机有一个称为 PostScript Cartridge 的特殊装置，一般价值几千元，所以国内一般使用的激光打印机都不带这种设置，故应该采用间接方法来打印带有这样包含的文件，其方法在下节中介绍。

其它类型的文件可以通过 L<sup>A</sup>T<sub>E</sub>X 提供的 `\special{}` 命令来调用。例如如果用户有一个位图文件 (以 BMF 为后缀)<sup>6</sup>，假设文件名为 `MYFIG.BMF`，则可以由下面的命令插入

```
\special{BMF=MYFIG.BMF}
```

---

<sup>6</sup>注意，这里的 BMF 文件是在 DOS 下生成的，它不是 Windows 环境产生的 BMP 文件。

## 7 常用的 L<sup>A</sup>T<sub>E</sub>X 工具包

L<sup>A</sup>T<sub>E</sub>X 的开发者和用户推出了许多可以进一步扩展其功能的工具包 (packages), 其中大多数工具包均可以从网上免费获取。在 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 版下, 可以使用 `\usepackage` 来调入相应的工具包, 典型的 2<sub>ε</sub> 版本开头部分为

```
\documentclass[11pt]{article}
\usepackage{picinpar}
\usepackage{identfst}
\usepackage{emlines2}
\usepackage{epsf}
```

而在 L<sup>A</sup>T<sub>E</sub>X 2.09 版下还可以将该 STY 文件包含在 `documentstyle` 命令中, 如可以在文件开头处加入下列命令

```
\documentstyle[11pt,emlines2,epsf]{article}
\input{picinpar.sty}
\input{identfst.sty}
```

### 7.1 图形插入与解释工具包 `emlines2` 与 `epsf`

标准版的 L<sup>A</sup>T<sub>E</sub>X 支持的绘图曲线类型是很少的, 它只支持一些简单的绘图命令。T<sub>E</sub>Xcad 和 emT<sub>E</sub>X 扩展了 L<sup>A</sup>T<sub>E</sub>X 的绘图功能, 引入了功能更强大的绘图语句。和该类语句相配合, 还编写了一个名为 `emlines2.sty` 的工具包, 在调用 T<sub>E</sub>Xcad 绘制的图形时, 必须首先调用 (包含) 该工具包, 否则其中很多绘图命令无法编译。

### 7.2 全部首行空格处理工具包 `indenrst.sty`

标准的英文科技文章排版在每一个小节的开始第 1 段均不使用首行空格 (`indent`), 而在后面的段落中使用首行空格。在中文排版中往往又要求在小节后的第 1 个段落中也采用首行空格, 这使得用户往往需要定义一个命令, 强制地留出首行空格。

L<sup>A</sup>T<sub>E</sub>X 的用户还开发了一个工具包 `indenrst.sty` 来完成这样的功能, 它只要求在文件开始时调用该工具包即可。

### 7.3 浮动环境的处理工具包

由不同的开发者编写了许多可以插入小型图形的工具包, 其中最常用的有 `floatflt.sty`, `floatfig.sty` 和 `picinpar.sty`。

`picinpar` 工具包中提供了一个基本环境 `window`, 其调用格式为

```
\begin{window}[nl,align,material,caption]
```

其中的 4 个参数的意义为

`nl` — 在 `window` 环境开始时的文本行数。

`align` — 文中 `window` 环境的对齐格式, 默认为左对齐, 选择 `l` 选项, 其余还有 `c` 和 `r` 选项, 分别为中间和右对齐。



materials — 包含的内容，可以为文本，或  $\text{\TeX}$  图形文件，也可以为 EPS 图形文件。

caption — 插入对象的说明

window 环境还派生出两个新的环境 figwindow 和 tabwindow，分别对图形和表格进行处理，它们的调用格式和前面的一致。例如可以通过下面的格式来定义 figwindow 环境

```
\begin{figwindow}[2,r,\input{fig1},{My inserted  
figure}]  
... 包含的文本部分  
\end{figwindow}
```

其中参数 2 表示在这里前 2 行作通栏处理，在以后各行中靠右面 (r 选项) 插入一个图形。当然这里的 2 和 r 选项可以由相应的参数替换。在这样的段落中插入的填写为当前目录下的 fig1.tex，而图形的说明文字为 My inserted figure，并根据上下文自动编号。

在 picinpar 工具包中提供的 tabwindow 环境的效果也是相似的。

floatfig.sty 工具包的功能类似于 picinpar.sty 工具包，在执行该工具包之前应该首先给出 \initfloatingfigure 命令作环境初始化，这样用户就可以使用 floatingfigure 环境了。该环境的调用格式为

```
\begin{floatingfigure}{8cm}  
\centerline{\input{fig1}}  
\end{floatingfigure}
```

和 picinpar 环境不同，在 floatingfigure 环境无需说明将图形靠哪一边对齐， $\text{\LaTeX}$  环境会根据奇偶页来进行自动设定。但在 floatingfigure 环境下不能指定前面有几行作通栏处理。

$\text{\LaTeX}$  提供的一些浮动环境工具包在特定场合下往往会出现这样或那样的错误，例如 picinpar 就不能跨越小节去使用，亦即该工具包在重新定义小节后会自动失效。另外如果在当前页面上若没有足够大的空间去容纳所嵌入的浮动图形时，该图形可能丢失。所以在使用这样的工具包时应注意其适用环境，以便获得最佳排版效果。

## 7.4 子图形编号处理工具包 subfigur.sty

若一个图形编号下包含有若干个子图，则可以调用 subfigur.sty 文件对各个子图进行自动编号，例如若用户绘制的两个图形存在 fig1.tex 与 fig2.tex，则下面的绘图命令将得出如图 2 (a) 和 (b) 所示的排版效果。

```
\begin{figure}[htb]  
\centering  
\mbox{  
\subfigure[the first figure]{\input{fig1}}\quad  
\subfigure[the second figure]{\input{fig2}}}  
\caption{\tt subfigur.sty 演示 }  
\end{figure}
```

## 8 常用的 $\text{\LaTeX}$ 附件程序介绍

除了  $\text{\LaTeX}$  本身提供的工具以外，还有很多学者研制出了一些为  $\text{\LaTeX}$  配套的实用软件，其

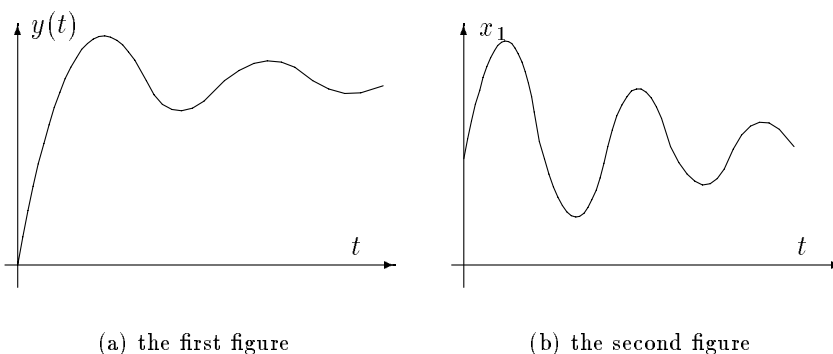


图 2 subfigur.sty 演示

中较流行的是画图软件  $\text{T}_\text{E}\text{Xcad}$ 、拼写错误检查软件 MicroSpell 及如曲线绘制软件  $\text{P}_\text{I}\text{C}\text{T}_\text{E}\text{X}$ 。中文版  $\text{L}^\text{A}\text{T}_\text{E}\text{X}$  给出了若干个应用程序, 如造字程序<sup>[7]</sup>等。在这一节中将介绍一些软件的使用。

### 8.1 用 $\text{T}_\text{E}\text{Xcad}$ 程序绘制框图

前面在介绍  $\text{L}^\text{A}\text{T}_\text{E}\text{X}$  绘图命令时曾提及  $\text{L}^\text{A}\text{T}_\text{E}\text{X}$  支持一些画图命令, 但没有给出例子, 因为直接用  $\text{L}^\text{A}\text{T}_\text{E}\text{X}$  提供的画图命令来绘制图形, 效率不是很高, 加上现在有现成的软件可以自动生成  $\text{L}^\text{A}\text{T}_\text{E}\text{X}$  可以调用的图形。在这里将介绍  $\text{T}_\text{E}\text{Xcad}$  画图软件的使用方法。

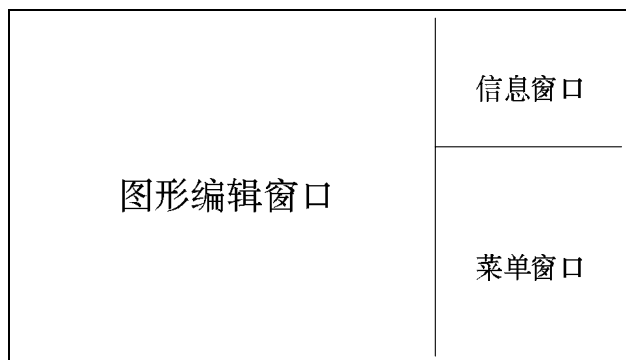


图 3  $\text{T}_\text{E}\text{Xcad}$  图形界面

如图 3 所示的图形界面, 在这一程序中提供了三个窗口, 右上角的是消息窗口, 用来显示一些产生的信息, 右下角的是菜单命令窗口, 而左边的是图形编辑窗口。

开始使用此程序时, 菜单窗口将给出程序的主菜单 (MAIN MENU), 其内容有 Draw (画图)、Edit (编辑)、Redraw (重画)、Snap (有级跳跃)、Zoom (放大)、Window (窗口)、Options (选项)、Save (存盘)、Load (读盘) 和 Quit (退出)。

如果用户点取 Snap 选项, 将给出

$\text{T}_\text{E}\text{Xcad}$  最早是德国 Goblentz 大学 W. Schiffmann 博士指导的学生设计的题目, 后来其功能逐渐扩充。 $\text{T}_\text{E}\text{Xcad}$  是用 Turbo Pascal 5.0 版本编写的, 用户可以利用鼠标器来画出图形, 然后  $\text{T}_\text{E}\text{Xcad}$  软件将该图形翻译成由  $\text{L}^\text{A}\text{T}_\text{E}\text{X}$  命令组成的源文件, 用户可以将这个源文件嵌入自己的文件之内来获得所需的图形。这里介绍的是 Georg Horn 改进的  $\text{T}_\text{E}\text{Xcad}$  3.2 (1995 年 1 月, 该软件为最终版本)。

在 DOS 下用户可以通过 `texcad` 命令来起动该软件, 这时将获得一个

一个下级菜单，中间有两个选项：On (有级) 和 Off (无级)。选择了 On，则光标的移动是按格跳动的，每一次只能跳动整数个格，这有利于精确定位，如果用户选择了 Off 选项，则光标可以平滑地移动，这会提高作图的分辨力。

如果用户选择了其中的 Draw 菜单，则将出现一个下级菜单项：Text (文字)、Framebox (文字与框)、Dashbox (虚线框)、Filled Box (充填的方框)、Line (直线)、Vector (有箭头的直线)、Circle (圆)、Filled Circle (充填的圆)、Oval (圆角矩形)、Put (加入字符串) 与 Bizier-Chain (Bezier 曲线序列) 及 Bezier-Vector (带箭头的 Bezier 曲线)。如果选择了 Text 选项，则应该用鼠标键在画图窗口上点取一个点，然后键入一个字符串。其实，由于整个文件都将用  $\text{\LaTeX}$  去处理，所以可以键入任意复杂的、由  $\text{\LaTeX}$  命令组成的字符串。然后按下回车键。这时在消息窗口中将询问文字的对齐方式，用户可以按自己的要求进行回答，最后设置出文字的位置。注意，由于  $\text{\TeXcad}$  并不带有任何排版功能，所以在画图窗口中显示文字时只有一个 T 来表示。

如果用户选择了 Framebox 绘图工具，则  $\text{\TeXcad}$  将提示用户确定方框的左下角和右上角坐标，用户可以用鼠标点中的方式来选择这些坐标，然后  $\text{\TeXcad}$  将询问用户方框中的文字，如果没有文字，则按下回车键即可，如果需要文字，则其输入的过程和前面 Text 选项一致。输入完文字后，则在方框中用 BT 字符来代替。

Dashbox 选项将允许用户画出一个虚线的方框，它将按类似于 Framebox 的方式要求用户画出一个虚线方框，然后允许用户选择虚线线段的长度，这样便可以绘制出一个虚线框了。如果用户选择了 Filled Box 选项，则  $\text{\TeXcad}$  将让用户画出一个方框，这时绘出的方框内部是充填的。

如果用户选择了 Line 选项，则  $\text{\TeXcad}$  软件将要求用鼠标键点出一个直线的起始点，然后再点中一个终止点，这样就可以绘制出一条直线。如果用户想绘制折线，则再用鼠标点中下一个点，这时，将会以前一个终止点为起点，此点为终止点再画出一条线段，用户可以采取这种方式来绘制出由多段线段构成的折线。如果用户按下了 Esc 键或右侧鼠标键，则线段绘制过程就完成了。

如果用户选择了 Vector 选项，绘制过程和前面一致，但所绘制出来的直线是带有箭头的，其箭头的方向是由起始点指向终止点。如果画折线的方式来绘制，则每一段直线都是带箭头的，注意，在  $\text{\TeXcad}$  下的表现方式略有不同。

绘制圆、圆角矩形等的命令也是很简单的，这里就不再赘述了。

徒手绘制曲线的方法也是很简单的，选择 Bezier-Chain 选项，则要求用户用鼠标左键点中起始点，再移动鼠标器，点中第 1 段曲线的终止点，然后再移动鼠标器，这时将随着鼠标器的移动出现变化的曲线。如果用户发现一个合适的形状，则可以点中鼠标左键，这时将获得一条固定的曲线。用户在移动鼠标器，并点中第 2 段曲线的终止点 (起始点为第 1 段的终止点)，再移动鼠标器，仍将得出变化的曲线，用鼠标左键选中第 2 段曲线终点，再重复上面方法画出第 3 段，第 4 段  $\cdots$ ，整个曲线绘制完成之后，点中鼠标右键确认。

选择 Bezier-Vector 则将绘制出一条带有箭头的 Bezier 曲线。

如果想对绘制出来的图形进行修改，则可以选择 Edit 菜单选项，这样将给出一个子菜单，包含下列的菜单项：Text-Edit (文字编辑)、Copy (复制)、Move (移动)、Delete (删除)、Delete all (全部删除)、Save Macro (存储宏命令)、Load Macro (读入宏命令)。

如果用户想对其中的文字加以修改,则可以选择 Edit-Text 选项,这时 T<sub>E</sub>Xcad 软件将让用户选择所要编辑的文字,如果用户用鼠标点中了一个文字标志,则在消息窗口中将询问是否点中的是要修改的,如果是则可以回答 Y, 否则回答 N。选择了要修改的文字标志,则在绘图窗口的上部显示出该标志对应的字符串,用户就可以修改该字符串了。修改完之后可以按下回车键确认。

如果用户想复制图形的某一个部分,则可以选择 Copy 选项,这时将给用户三个选择: Pick Objects (选择对象), Pick area (选择区域), Unpick (取消选择), 如果用户选了第一个选项,则允许用户用鼠标点取的方式选择一个对象(亦即图形元素),这一动作仍需要在消息窗口中给出确认信息。如果用户选中了某个对象,则该对象将变成红色,用户可以继续选择,直至选择了所有的对象之后,然后按下 Esc 键或鼠标右键来终止选择过程。这时 T<sub>E</sub>Xcad 将询问用户 First Point (第一个点) 和 Second Point (第二个点) 的坐标,用户用鼠标来回答这些问题。给出了这两个点的坐标之后,则所选择的对象就在新指定的地方复制出来了。如果用户选择了 Pick area 选项,则允许用户象前面介绍的绘制方框那样画出一个方框,在方框中所有的对象就都处于红色的状态,如果用户选中之后,则可以用前面介绍的方法进行复制。Unpick 选项将取消对原来对象的选择。

如果用户选择了 Move 选项,则和 Copy 选项的操作过程是极其相似的,所不同的是将选中的对象移动到新的位置,而不是复制到新的位置。

Delete 选项将用户选择的部分从图形窗口中删除, Delete all 选项将整个图形全部删除。T<sub>E</sub>Xcad 软件允许用户将选中的对象作为宏命令 (macros) 存储起来,以便以后使用,也可以从文件中调入宏命令,这两个工作是可以由 Save Macro 和 Load Macro 菜单项来进行的。

如果用户选择了 Zoom 菜单项,则将出现一个子菜单,其内容为: Factor (按比例缩放), Area (指定区域), All (全部), Last (上一次缩放状态)。选择了 Factor 选项,则要求用户键入一个缩放倍数,然后将把画图窗口中的内容按照给出的比例进行缩放;如果选择了 Area,则要求用户指定一个方框,然后将方框内的图形部分充满整个绘图窗口,这样就可以更精确绘制图形了;选择 All 选项将回到原来的全屏幕编辑状态,如果选择了 Last 菜单项,则恢复到上一个比例系数下进行编辑。

Options 菜单项提供了很多的默认设置,用户可以进行修改这些选项,以得到自己所需要的设置,并可以将设置存储到文件中保存起来。

绘制完图形之后,则可以用 Save 选项将图形存储成一个 L<sup>A</sup>T<sub>E</sub>X 可以识别的文件中,这样在文件中可以直接嵌入这一图形文件。具体的嵌入方法是使用 \input{} 命令来处理。

对于中文文字的处理,可以在原来西文文字输入下做一个记号,然后利用编辑软件将中文文字加到该图形文件中,并调用 CCT 程序将文件翻译成 TEX 文件,然后再用 \input{} 命令来嵌入该图形,这有 T<sub>E</sub>Xcad 软件绘制出来的图形也可以加上中文信息了。

## 8.2 英文字拼写错误的检查

由于输入文件是通过操作人员的输入而赋给计算机的,而在操作的时候难免会发生误操作从而导致一些输入错误,例如会使得个别单词的拼写和实际想输入的不一致。即使以英语为母语的熟练操作人员也无法保证在文件输入时不产生任何错误,更何况是非英语国家的操作者了。

由于这样的原因,让计算机去检查输入的文件是否正确无疑是一种相当有意义的工作,从而出现了许多英语拼写错误检查程序,如 Word for Windows 文字处理软件下就有这样的程

序。由于 L<sup>A</sup>T<sub>E</sub>X 文件具有特殊性，它允许的命令一般也是由类似英文的单词所组成，这种拼写往往不是标准的英文单词，例如 topmargin 命令其实是由两个单词组成的，在一般拼写检查程序中往往会指出错误，所以如果想完全检查一个 T<sub>E</sub>X 文件拼写是否有错，用别的软件来完成不是很有效。基于这个原因，陆续出现了很多适于 L<sup>A</sup>T<sub>E</sub>X 文件的拼写检查程序，较实用的有 AMSpell 和 MicroSpell 软件，这里着重介绍 MicroSpell 软件。

MicroSpell 是美国 Trigram Systems 公司 1991 年的产品，如果键入 SPELL，则会询问用户所要查询的文件名，打开文件之后，将自动开始搜索拼写错误，并给出一个如图 4 所示的程序界面。

MicroSpell 软件对查出的怀疑拼写错误 (Suspects) 将给出若干个建议词汇 (Suggestions)，并提供如下的菜单<sup>7</sup>：

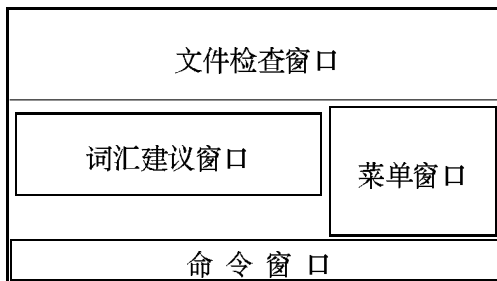


图 4 MicroSpell 程序图形界面

Accept suspect this time only (仅此一次接受该词汇)

Ignore now and throughout session (搜索过程中忽略此词汇)

Type replacement word (键入替换词汇)

Replace with suggestions 1..9 (用 1 到 9 的建议词汇进行替换)

Undo previous command (取消上一个命令)

Quit and return to old text (退出，返回原来的文本)

Exit and save corrections (退出，将改动的结果存盘)

Save in auxiliary dictionary (存入辅助字典)

Lookup word in dictionary (在字典中查询)

Edit text at suspect (对怀疑词汇进行编辑)

F1 next Help screen (按下 F1 键获得下一个帮助屏幕)

其中带有下划线的字符为该条命令的热键符号。如果用户按下了 A 键，则将执行第一菜单项 (仅此一次接受该词汇)，并继续向下搜索。如果用户按下了 I 键，则在整个查找过程中将忽略这一词汇的查错，这两个选项是在用户有把握时可以采用的选择。如果用户键入了 R，则可以在下面的命令窗口中键入一个新的词汇来替换查出的词汇。如果用户想利用此软件给出的建议，则可以按下 1 到 9 的数字键，来从建议窗口中选取合适的词汇来替换查出的词汇，并继续向下搜索。如果用户按下了 U 键，则说明用户想取消上一次的替换，恢复到替换之前的状态，注意，这一命令的使用次数是有限的。

如果用户选择了 Q 命令，则将退出 MicroSpell 软件，但对改动的部分不作存盘处理。如果选择了 X 命令，则将改动后的文件存盘，并退出 MicroSpell 软件。

如果用户选择了 S 选项，则将获得的词汇存入辅助字典中，这样用户可以根据自己的实际需要 will 将一些本专业常用的词汇存入辅助字典，以便以后再查询时可以不再去查询该词汇。如果用户按下了 L 键，则可以从字典中查询相关的词汇，例如如果用户键入 Previous\*，则将列出 Previous, Previously, Previousness 等词汇，注意，这里星号 (\*) 的用法和 DOS 下 DIR 命令中允许的文件名中的星号是一致的，除了 \* 以外，还可以采用 ?。

如果用户按下了 E 键，则可以在当前怀疑的词汇附近开一个全屏幕编辑窗口，用户在这

<sup>7</sup>菜单项后面的中文注释是作者加的，在软件中将不给出。

个窗口中可以随意编辑该文件。编辑完之后，MicroSpell 程序还将对新编辑的部分进行拼写查错处理。

如果用户按下了 F1 键，则可以得到一个新的帮助窗口，在该窗口中将给出一些热键的解释等。

查询结束之后，如果用户做了一些改动，则把原来的文件存成 QEX 文件，而改动后的文件存成源文件。

值得庆幸的是，在中文版 L<sup>A</sup>T<sub>E</sub>X 之下处理的 CTX 文件也可以通过 MicroSpell 进行查错处理。这种查错只限于对文中英文词汇的查询，不能对中文进行查错，但这样处理一般来说已经足够了。如果用户较好地掌握该软件的使用，则将会对文字处理（特别是英文文献）工作带来极大的方便。

### 8.3 CCT 提供的曲线绘制程序 PICT<sub>E</sub>X

PICT<sub>E</sub>X 是 L<sup>A</sup>T<sub>E</sub>X 提供的曲线绘制应用程序，它提供一些绘图指令，如果用户不给出，将使用 PICT<sub>E</sub>X 软件设定的默认值。PICT<sub>E</sub>X 提供的绘图指令如下：

PIC\_SIZE 和 PIC\_SIZE\*：如果用户想获得一个宽 100mm，高 70mm 的图形的话，则可以给出命令 PIC\_SIZE=100,70。注意，由于 PICT<sub>E</sub>X 使用的不是 L<sup>A</sup>T<sub>E</sub>X 环境，所以在命令不能由 \ 来引导。二者的区别在于，如果使用 PIC\_SIZE，则产生的图形实际尺寸是 100×70，再加上坐标轴等所占用的空间，实际图形的长和宽都应再加上 30mm。如果采用了 PIC\_SIZE\* 命令，整个图形需要的空间和设置应该是一致的。

WINDOW：该命令的格式为 WINDOW=[ $x_{\min}$ ,  $x_{\max}$ ,  $y_{\min}$ ,  $y_{\max}$ ]，其中给出的四个参数定义了用户窗口，如果此命令不给出，PICT<sub>E</sub>X 将自动将这四个参数取作给出数据的最大值和最小值。

ORIGIN：其命令格式为 ORIGIN= $a, b$  用来将坐标轴原点的位置定义在  $a, b$  点上，如果不给出此命令，则  $a = x_{\min}$ ,  $b = y_{\min}$ 。

LINE 和 VECTOR：命令格式为 LINE 或 VECTOR= $x_1, y_1, x_2, y_2$ ，亦即绘制一条从  $(x_1, y_1)$  到  $(x_2, y_2)$  的直线或带箭头的线段。

STRING：命令格式为 STRING= $s$ ，其中  $s$  为字符串。

GRID 和 GRAD：命令格式为 GRID 或 GRAD= $n, m$ ，其中前一个命令将图形区域绘制  $(n+1) \times (m+1)$  个网格，而后一个命令在坐标轴上打印  $(n+1) \times (m+1)$  个坐标值。

AXIS\_STATUS：命令格式为 AXIS\_STATUS= $k_1, k_2$ ，表示坐标轴状况，其中  $k_1$  对应于 X 轴设置， $k_2$  对应于 Y 轴设置。它们的可取值为 OFF（不画坐标轴）、REAL（打印实数刻度）和 INTEGER（打印整数刻度）三种。

AXIS\_TYPE：命令格式为 AXIS\_TYPE= $k_1, k_2$ ，表示坐标轴类型，其中  $k_1$  对应于 X 轴设置， $k_2$  对应于 Y 轴设置。它们的可取值为 NORMAL（正常型，亦即线性）、LOG（对数）两种。

AXIS\_NAME：命令格式为 AXIS\_NAME= $s_1, s_2$ ，表示坐标轴名称，其中  $s_1$  和  $s_2$  为字符串，分别为 X 和 Y 轴的标记。

DATA : 命令格式为  $\text{DATA}=n, x_1, y_1, \dots, x_n, y_n$ , 表示数据组, 其中  $n$  为数据组数,  $(x_1, y_1), \dots, (x_n, y_n)$  为绘图数据点, 数据点中间的部分由插值方法计算出来, 后面将介绍插值方法指定。

END : 此语句后面的内容将被忽略, 它表示数据文件到此结束。

CURVE\_TYPE : 命令格式为  $\text{CURVE\_TYPE}=k$ , 表示曲线类型, 其中  $k$  的取值为 THIN (细线)、THICK (粗线) 和 MARK (不画连线, 只画标记) 之一。

INTERPOLATION : 命令格式为  $\text{INTERPOLATION}=k$ , 表示插值方式, 其中  $k$  的取值为 LINEAR (线性插值)、SPLINE (样条插值) 之一。

MARK\_SYMBOL : 命令格式为  $\text{MARK\_SYMBOL}=s$ , 表示标记所用的符号, 其中  $s$  为一个字符串。

DASHED\_CURVE : 命令格式为  $\text{DASHED\_CURVE}=n, l_1, \dots, l_n$ , 表明曲线类型, 如果绘制实线, 则置  $\text{DASHED\_CURVE}=0$ , 否则  $l_1, \dots, l_n$  为该曲线实虚相交替的方式。

给出了下面的命令之后, 就可以绘制图形了。下面给出了一个由两条曲线构成的图形:

```
pic_size*=80,60
origin=0,0
axis_status=real,integer
axis_name='$x$', '$\sin(x)$', '\frac{\displaystyle x}{\displaystyle \pi}$'
grad=1,1
% --- First curve ---
curve_type=thick
interpolation=spline
mark_symbol='$\bullet$'
dashed_curve=4,2,0.5,0.5,0.5
DATA=33
-3.1415927e+000 -1.2246064e-016
-2.9415927e+000 -1.9866933e-001
-2.7415927e+000 -3.8941834e-001
-2.5415927e+000 -5.6464247e-001
-2.3415927e+000 -7.1735609e-001
-2.1415927e+000 -8.4147098e-001
-1.9415927e+000 -9.3203909e-001
-1.7415927e+000 -9.8544973e-001
-1.5415927e+000 -9.9957360e-001
-1.3415927e+000 -9.7384763e-001
-1.1415927e+000 -9.0929743e-001
-9.4159265e-001 -8.0849640e-001
-7.4159265e-001 -6.7546318e-001
-5.4159265e-001 -5.1550137e-001
-3.4159265e-001 -3.3498815e-001
-1.4159265e-001 -1.4112001e-001
5.8407346e-002 5.8374143e-002
2.5840735e-001 2.5554110e-001
4.5840735e-001 4.4252044e-001
6.5840735e-001 6.1185789e-001
8.5840735e-001 7.5680250e-001
1.0584073e+000 8.7157577e-001
1.2584073e+000 9.5160207e-001
```

```

1.4584073e+000  9.9369100e-001
1.6584073e+000  9.9616461e-001
1.8584073e+000  9.5892427e-001
2.0584073e+000  8.8345466e-001
2.2584073e+000  7.7276449e-001
2.4584073e+000  6.3126664e-001
2.6584073e+000  4.6460218e-001
2.8584073e+000  2.7941550e-001
3.0584073e+000  8.3089403e-002
3.1415927e+000  1.2246064e-016

% ----- Second curve -----
curve_type=thin
interpolation=linear
mark_symbol='$\circ$'
dashed_curve=0
data=2
-3.1415927e+000  -1.0000
 3.1415927e+000   1.0000

```

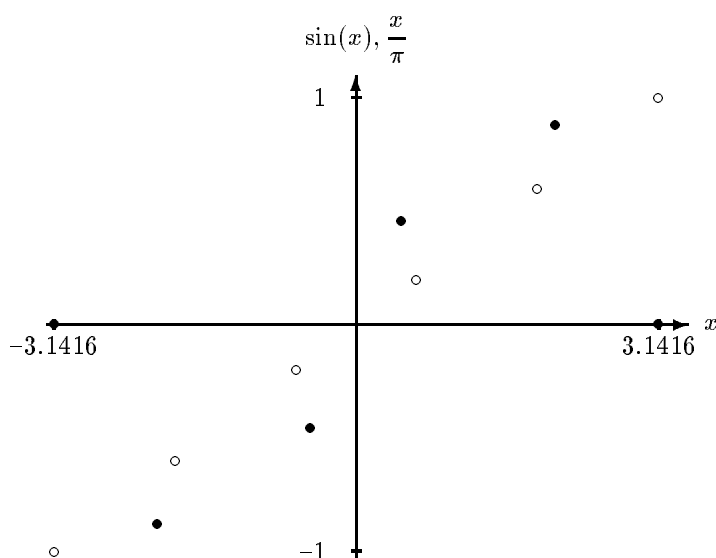


图 5 由 PICTeX 生成的图形

机, 则可以使用此分辨率, 如果用激光打印机, 则应该选择 300。由前面命令构成的图形如图 5 所示。

如果以上命令构成的文件为 MYSIN.DAT, 则用户可以调用 PICTeX 执行文件将它转换成 TEX 文件, 这样就可以在文字文件中嵌入该图形了。由于 L<sup>A</sup>T<sub>E</sub>X 绘制虚线等的命令不是很强, 所以需要借助于位图文件来处理。在 PICTeX 程序产生 TEX 文件的同时, 还产生了 PDF 文件, 用户可以根据 CCT 提供的 PDFTOBMF 程序将 PDF 文件转换成 BMF 文件。这样, 该 BMF 文件可以由产生的 TEX 文件自动调用。注意: 生成 BMF 文件时, PDFTOBMF 程序将询问用户产生 BMF 文件的分辨率, 并给出默认值 180, 如果用点阵打印

#### 8.4 EPS 图形文件的插入和打印

在实际文件排版中除了前面所述的各类对象的插入以外, 还经常遇到其它类型的图形文件的插入问题, 例如在科学界使用最广的 EPS (即 Encapsulated PostScript) 图形文件的插入就是很重要的。



#### 8.4.1 PostScript 文件概述

PostScript<sup>[1]</sup> 是著名的 Adobe 公司 20 年前开发的一种适合于激光打印机驱动的一种标准图形描述语言, 该语言是一种页面描述语言, 早就成为激光打印机绘图的标准语言, 由这种语言写的文件称为 PS 文件。很多应用软件, 如常用的 MATLAB 和 Mathematica 以及 Windows 系统均可以产生封装的 PS 文件 (Encapsulated PS, 简称 EPS 文件)。但由于其设备原因在中国并未普及起来。

该语言可以“矢量”地绘制出任意复杂的图形, 它和位图式描述是不同的, 它描述的图形元素 (例如直线等) 并不是记录图形上各个点, 而只记录直线的起点和终点, 从而从存储空间的角度来说可以大大地节省空间, 但并非其最显著的优点, 其最大的优点是它允许不走样地任意放大或缩小图形。

作为 PS 语言的延伸, EPS 文件在描述图形时是相当方便和实用的, 所谓 EPS 文件 (封装的 PS 文件), 即在原来描述整个页面的 PS 文件中提取出来一个有用的部分作为一个子图形, 在该图形下只需记录其左下角坐标及宽度和高度, 其余的部分全部封装起来, 这样用户在以后想修改图形的大小时只需修改这四个参数, 图形的内容便会自动地依照需要进行相应的放大或缩小, 而不会产生任何畸形。

PS 及 EPS 文件都是纯文本文件, 所以它可以适用于任何机型。熟悉了 PS 文件的编程语句, 则可以对之进行适当的修改。但作者不建议直接去修改 EPS 文件, 除非绝对必要。

EPS 文件是由 Bounding Box 语句来引导的 (虽然它不是第一语句), 在该语句中描述了封装 PS 的四个必要参数, 用户没有必要去这些参数。

在若干变量定义之后, 就是 PS 语言的绘图语句了。下面将举例给出几个常用的绘图语句格式:

- 直线绘制语句:  $x_1 y_1$  moveto 和  $x_2 y_2$  lineto, 其中 moveto 部分定义绘图起点位置, 而 lineto 语句定义终点。
- 文字显示语句: (字符串) show, 将括号中的字符串在当前画笔位置显示出来。
- 平移语句:  $x_1 y_1$  translate, 将该语句后面的内容在水平和垂直方向上分别平移  $x_1$  和  $y_1$  个单位。
- 局部参数保存与调入语句 gsave 和 grestore。

如果只想使用 EPS 文件, 则没有必要掌握上面的叙述。

在很多通用软件中都提供了输出 EPS 文件的方法, 例如常用的 MATLAB 语言可以由下面的步骤来生成 EPS 文件:

- 1). 调用 MATLAB 命令将所需的图形从图形窗口上绘制出来
- 2). 调用下面命令将该图形存成 EPS 文件

`print -deps 文件名`

例如 `print -deps myfile` 将在相应目录下产生 `myfile.eps` 文件, 而

```
print -deps d:\tmp\myfile
```

将在 `D:\tmp` 目录下产生 `myfile.eps` 文件。如果确实想生成彩色的 EPS 文件, 则可以将 `-deps` 选项改成 `-depsc` 即可。在 MATLAB 早期版本 (如 3.x 版) 中生成 EPS 文件并不

太方便，它需首先调用 `meta` 命令生成图元格式，然后调用 DOS 下的 `GPP.exe` 文件才可生成 EPS 文件，这里就不赘述了，好在这种版本的软件已几乎无人使用了。

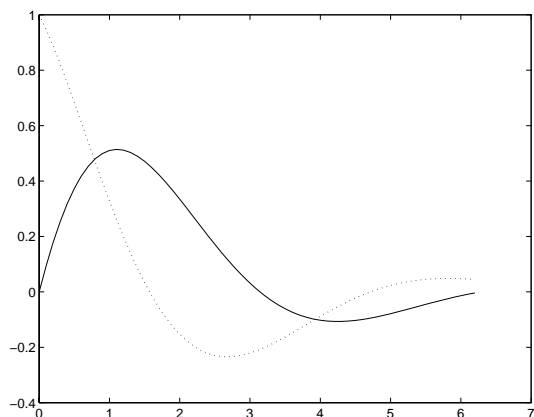
除了 MATLAB 以外，象 Word 之类 Windows 下的应用程序也可以产生 EPS 文件，其生成方法很简单，首先将打印机设置成带有 PS 卡的打印机，然后在调用 File | Print 菜单命令进行打印时选择 Print To File 选项即可。通过这样的方式可以产生所需的 PS 文件。

#### 8.4.2 EPS 文件插入举例

首先考虑在 MATLAB 下生成的图形文件 `myfig.eps`<sup>8</sup>。这样只需在  $\text{\LaTeX}$  文本中加入下面语句

```
\begin{figure}[htb]
\centerline{\epsfxsize 70mm \epsffile{myfig.eps}}
\end{figure}
```

则可以将该产生的 EPS 图形插入到该文本中，如图 6(a) 所示。注意，这样插入的图形在 view



(a) MATLAB 图形的插入



(b) PhotoShop 图形插入

图 6 EPS 文件的插入

显示命令下不会被显示出来，而只在相应的位置留出空白空间。

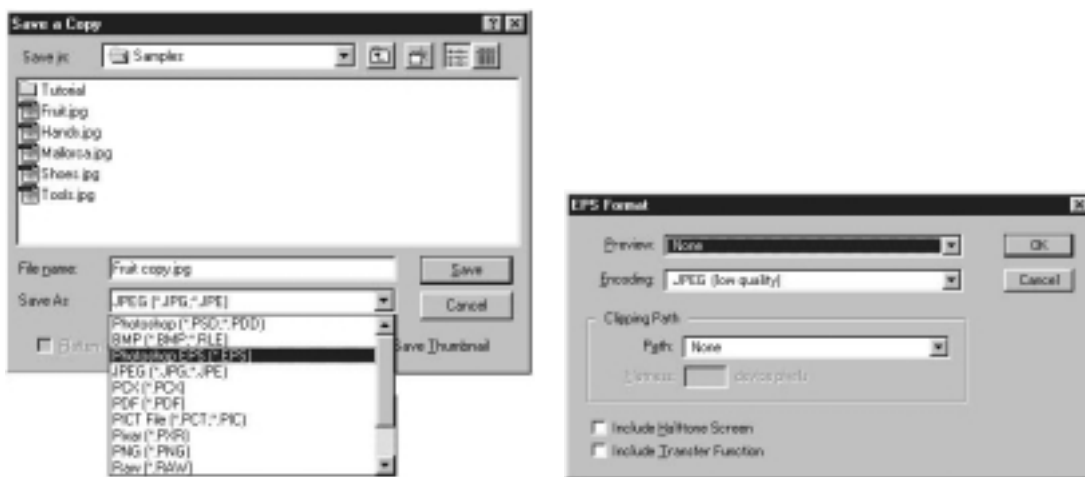
除了 MATLAB 生成的 EPS 文件之外，其它程序生成的 EPS 文件或其它格式的图形文件经转换后也能直接插入  $\text{\LaTeX}$  文本中。例如，PhotoShop 是一种照片处理的优秀软件，它可以用来将扫描得到的图形文件进行转换，最后获得 EPS 文件。我们以该软件的 `samples` 目录下的 `Fruit.jpg` 文件为例来演示如何将 jpg (或其它各种格式的图形文件) 插入  $\text{\LaTeX}$  中。

<sup>8</sup>在 MATLAB 下可以由下面各条语句生成该文件

```
>> x=0:.1:2*pi; y1=exp(-0.5*x).*sin(x);
>> y2=exp(-0.5*x).*cos(x); plot(x,y1,'-',x,y2,':')
>> print -deps myfig
```

然后将该文件移动到当前目录下即可。

采用 PhotoShop 下的 File | Save As 选项<sup>9</sup>，则可以将该软件已经调入的 jpg 文件转换成 EPS 文件，其过程是在弹出的对话框的 Save As 列表框中选择 \*.eps 选项，如图 7 (a) 所示。指定了文件名之后将给出应该如图 7 (b) 所示的对话框，用户应该在 Preview 选项中填写 None



(a) 文件名输入对话框

(b) EPS 选项对话框

图 7 PhotoShop 下文件转换对话框

字样，Encoding 选项可以从列表框中任意选择，一般来说选择 JPEG (low quality) 足以得出较满意的图形，且占用的空间较小。若想获得最高的分辨率，则应该选择其中的 ASCII 选项。这时按下 OK 按钮则将会转换出所需的 EPS 文件来。为了进一步减小生成 EPS 文件所占用的空间，用户还可以将当前图形的彩色设置改变成灰度 (grayscale) 设置。这样在存储的过程中会自动地将该图形存成 EPS 文件的格式。将生成的 fruit.eps 文件复制到当前目录下，这时在 L<sup>A</sup>T<sub>E</sub>X 文本中加入下面的语句

```
\begin{figure}[htb]
\centerline{\epsfxsize 70mm \epsffile{fruit.eps}}
\end{figure}
```

则会在 L<sup>A</sup>T<sub>E</sub>X 的文本中增加如图 6(b) 所示的图形。通过前面的例子可以看出 L<sup>A</sup>T<sub>E</sub>X 在图形插入方面的灵活性。

### 8.4.3 将基于中文的 DVI 文件转换成不依赖中文字库的 DVI 文件

中文版的 L<sup>A</sup>T<sub>E</sub>X 提供了一个可以将基于中文版生成的 DVI 文件转换成不依赖中文字库的 DVI 文件的程序 (以下将简称西文化的 DVI 文件)：PATCHDVI.exe。该程序的调用格式为

```
patchdvi -r 分辨率 原 DVI 文件名 新 DVI 文件名
```

其中分辨率的默认值为 300x300，亦即 300dpi，若想获得 600dpi 的分辨率，则应该添上 -r600x600

<sup>9</sup>在 PhotoShop 4.0 及以上版本中应该选择 Save a Copy

的选项, 这时将在硬盘上生成一个暂时的目录, 并将一些中间信息存盘。该过程完成后则会生成新的 DVI 文件。

对中文版 L<sup>A</sup>T<sub>E</sub>X 西文化的 DVI 文件进行处理, 则可以获得通用的 PS 文件, 其命令为

`dvips32 -D 分辨率 西文化的 DVI 文件名`

这里分辨率的默认值仍为 -D300x300。如果使用该命令时出现

```
c:\emtex\dvips32.exe: Font tmpxxx.dat at xxx dpi not found, characters left blank
```

的错误信息, 则说明在该分辨率下字库文件不完全, 这样就要求用户调用 makepk.exe 程序去产生该字库, 该程序的调用格式为

`makepk -R 分辨率 西文化的 DVI 文件名> tmp.bat`

然后再运行 tmp.bat 程序, 这样在该程序执行完后, 则会自动生成所需要的字库文件。这样再运行 dvips32 程序则可以生成所需的 PS 文件了。

值得指出的是, 这样生成的 PS 文件可以完全脱离中文环境, 它可以在 Windows 环境或其它平台 (如 OS 下的 OpenWindows) 上直接显示和处理。

#### 8.4.4 GhostView 软件应用

从前面的叙述中可以看出, 通过一系列变换就可以将一个含有中文的 L<sup>A</sup>T<sub>E</sub>X 文本转换为通用的 PostScript 文件, 这样就可以在带有 PostScript 驱动卡的激光打印机上直接打印出来。考虑到该卡的价格较贵, 目前国内的大多数激光打印机上都未配备这样的卡, 而 Adobe 公司的 GhostScript 程序可以将 PS 文件在非 PS 打印机 (甚至可以在点阵打印机) 上直接打印出来。

## 附录 A L<sup>A</sup>T<sub>E</sub>X 中的特殊符号及命令

L<sup>A</sup>T<sub>E</sub>X 提供了大量的符号, 由于本文正文部分篇幅有限, 不能一一解释各种特殊字符的处理, 在本附录中列出全部 L<sup>A</sup>T<sub>E</sub>X 中的数学符号, 并给出相应的命令, 以供参考。在这里也列出了一些文本模式下的特殊符号。

- 1). **注音类修饰字符:** L<sup>A</sup>T<sub>E</sub>X 提供了大量的非英文的西文类字符及其修饰方法, 汉语拼音也可以认为是其中的特殊情况。其中有的是需要在文本环境下来给出的, 有些是应该在数学环境下给出的, 还有一些既可以在文本下给出, 也可以在数学下给出, 但是排版的效果略有不同。

- 注音符号: 这里只以 o 为例, 这种修饰方法还可以用于其它字符。

ó \'{o}	ò \'{o}	ô \^{o}	ö \"{o}	õ \~{o}
ō \={o}	ô \.{o}	ö \u{o}	ö \v{o}	ö \H{o}
ôo \t{oo}	q \c{o}	ó \'{o}	q \d{o}	q \b{o}

- 特殊符号: 一般可以用于文本和数学模式。

† \dag	‡ \ddag	§ \S	¶ \P	© \copyright
£ \pounds	œ \oe	Œ \OE	æ \ae	Æ \AE
å \aa	Å \AA	ø \o	Ø \O	l \l
L \L	ß \ss	ı \i	J \j	

- 数学模式下的修饰：可以用于数学模式，这种修饰对任何字符都适用，在这里只给出对 a 的显示。

$\hat{a}$ <code>\hat{a}</code>	$\check{a}$ <code>\check{a}</code>	$\breve{a}$ <code>\breve{a}</code>	$\acute{a}$ <code>\acute{a}</code>
$\grave{a}$ <code>\grave{a}</code>	$\tilde{a}$ <code>\tilde{a}</code>	$\bar{a}$ <code>\bar{a}</code>	$\vec{a}$ <code>\vec{a}</code>
$\dot{a}$ <code>\dot{a}</code>	$\ddot{a}$ <code>\ddot{a}</code>	$\widetilde{a}$ <code>\widetilde{a}</code>	$\widehat{a}$ <code>\widehat{a}</code>

- 2). **小写希腊字符**：L<sup>A</sup>T<sub>E</sub>X 允许在数学模态下使用大写和小写希腊字符，但在文本中不可以直接使用它们，如果想使用应该用单个 \$ 号括起来。

$\alpha$ <code>\alpha</code>	$\beta$ <code>\beta</code>	$\gamma$ <code>\gamma</code>	$\delta$ <code>\delta</code>
$\epsilon$ <code>\epsilon</code>	$\varepsilon$ <code>\varepsilon</code>	$\zeta$ <code>\zeta</code>	$\eta$ <code>\eta</code>
$\theta$ <code>\theta</code>	$\vartheta$ <code>\vartheta</code>	$\iota$ <code>\iota</code>	$\kappa$ <code>\kappa</code>
$\lambda$ <code>\lambda</code>	$\mu$ <code>\mu</code>	$\nu$ <code>\nu</code>	$\xi$ <code>\xi</code>
$o$ <code>o</code>	$\pi$ <code>\pi</code>	$\varpi$ <code>\varpi</code>	$\rho$ <code>\rho</code>
$\iota$ <code>\iota</code>	$\kappa$ <code>\kappa</code>	$\varrho$ <code>\varrho</code>	$\sigma$ <code>\sigma</code>
$\varsigma$ <code>\varsigma</code>	$\tau$ <code>\tau</code>	$\upsilon$ <code>\upsilon</code>	$\phi$ <code>\phi</code>
$\varphi$ <code>\varphi</code>	$\chi$ <code>\chi</code>	$\psi$ <code>\psi</code>	$\omega$ <code>\omega</code>

- 3). **大写希腊字符**：一些大写的希腊字符同英文中的是一致的，所以在 L<sup>A</sup>T<sub>E</sub>X 中并没有必要配备相应的命令，下面只列出一些不同的符号。

$\Gamma$ <code>\Gamma</code>	$\Delta$ <code>\Delta</code>	$\Theta$ <code>\Theta</code>	$\Lambda$ <code>\Lambda</code>
$\Xi$ <code>\Xi</code>	$\Pi$ <code>\Pi</code>	$\Sigma$ <code>\Sigma</code>	$\Upsilon$ <code>\Upsilon</code>
$\Phi$ <code>\Phi</code>	$\Psi$ <code>\Psi</code>	$\Omega$ <code>\Omega</code>	

- 4). **常用数学符号**

$\aleph$ <code>\aleph</code>	$\hbar$ <code>\hbar</code>	$\prime$ <code>\prime</code>	$\emptyset$ <code>\emptyset</code>
$\forall$ <code>\forall</code>	$\exists$ <code>\exists</code>	$\imath$ <code>\imath</code>	$\jmath$ <code>\jmath</code>
$\ell$ <code>\ell</code>	$\wp$ <code>\wp</code>	$\Re$ <code>\Re</code>	$\Im$ <code>\Im</code>
$\partial$ <code>\partial</code>	$\infty$ <code>\infty</code>	$\nabla$ <code>\nabla</code>	$\surd$ <code>\surd</code>
$\top$ <code>\top</code>	$\bot$ <code>\bot</code>	$\parallel$ <code>\parallel</code>	$\angle$ <code>\angle</code>
$\triangle$ <code>\triangle</code>	$\backslash$ <code>\backslash</code>	$\neg$ <code>\neg</code>	$\flat$ <code>\flat</code>
$\natural$ <code>\natural</code>	$\sharp$ <code>\sharp</code>	$\clubsuit$ <code>\clubsuit</code>	$\diamondsuit$ <code>\diamondsuit</code>
$\heartsuit$ <code>\heartsuit</code>	$\spadesuit$ <code>\spadesuit</code>		

- 5). **较大的数学符号**：这样一些符号往往可以带有上下标以起到特殊的作用，如求和号上下标的作用。这些符号上下标的位置在方程下和在文本行中是不同的，除非使用者另加修饰。

$\sum$ <code>\sum</code>	$\prod$ <code>\prod</code>	$\coprod$ <code>\coprod</code>	$\int$ <code>\int</code>
$\oint$ <code>\oint</code>	$\bigcap$ <code>\bigcap</code>	$\bigvee$ <code>\bigvee</code>	$\bigwedge$ <code>\bigwedge</code>
$\bigsqcup$ <code>\bigsqcup</code>	$\bigcup$ <code>\bigcup</code>	$\bigodot$ <code>\bigodot</code>	$\bigotimes$ <code>\bigotimes</code>
$\bigoplus$ <code>\bigoplus</code>	$\biguplus$ <code>\biguplus</code>	$\lim$ <code>\lim</code>	$\sin$ <code>\sin</code>

- 6). **二元运算符号**

$\pm$	<code>\pm</code>	$\mp$	<code>\mp</code>	$\setminus$	<code>\setminus</code>	$\cdot$	<code>\cdot</code>
$\times$	<code>\times</code>	$\div$	<code>\div</code>	$*$	<code>\ast</code>	$\star$	<code>\star</code>
$\diamond$	<code>\diamond</code>	$\circ$	<code>\circ</code>	$\bullet$	<code>\bullet</code>	$\wr$	<code>\wr</code>
$\cup$	<code>\cup</code>	$\cap$	<code>\cap</code>	$\vee$	<code>\vee</code>	$\wedge$	<code>\wedge</code>
$\sqcup$	<code>\sqcup</code>	$\sqcap$	<code>\sqcap</code>	$\odot$	<code>\odot</code>	$\otimes$	<code>\otimes</code>
$\oplus$	<code>\oplus</code>	$\uplus$	<code>\uplus</code>	$\dagger$	<code>\dagger</code>	$\ddagger$	<code>\ddagger</code>
$\triangleleft$	<code>\triangleleft</code>	$\triangleright$	<code>\triangleright</code>	$\triangle$	<code>\bigtriangleup</code>	$\bigcirc$	<code>\bigcirc</code>
$\ominus$	<code>\ominus</code>	$\oslash$	<code>\oslash</code>	$\nabla$	<code>\bigtriangledown</code>	$\amalg$	<code>\amalg</code>
$\dots$	<code>\ldots</code>	$\cdots$	<code>\cdots</code>	$\vdots$	<code>\vdots</code>	$\ddots$	<code>\ddots</code>

### 7). 关系数学符号

$\leq$	<code>\leq</code>	$\geq$	<code>\geq</code>	$\equiv$	<code>\equiv</code>	$\prec$	<code>\prec</code>
$\succ$	<code>\succ</code>	$\sim$	<code>\sim</code>	$\preceq$	<code>\preceq</code>	$\succeq$	<code>\succeq</code>
$\simeq$	<code>\simeq</code>	$\ll$	<code>\ll</code>	$\gg$	<code>\gg</code>	$\asymp$	<code>\asymp</code>
$\subset$	<code>\subset</code>	$\supset$	<code>\supset</code>	$\approx$	<code>\approx</code>	$\subseteq$	<code>\subseteq</code>
$\supseteq$	<code>\supseteq</code>	$\sqsubseteq$	<code>\sqsubseteq</code>	$\sqsupseteq$	<code>\sqsupseteq</code>	$\sqsubseteq$	<code>\sqsubseteq</code>
$\bowtie$	<code>\bowtie</code>	$\in$	<code>\in</code>	$\ni$	<code>\ni</code>	$\propto$	<code>\propto</code>
$\vdash$	<code>\vdash</code>	$\dashv$	<code>\dashv</code>	$\models$	<code>\models</code>	$\smile$	<code>\smile</code>
$\mid$	<code>\mid</code>	$\doteq$	<code>\doteq</code>	$\frown$	<code>\frown</code>	$\parallel$	<code>\parallel</code>
$\perp$	<code>\perp</code>						

### 8). 否定关系运算符号: 这类符号在前面给出的关系符前加一个 `\not` 命令即可。

$\not<$	<code>\not&lt;</code>	$\not>$	<code>\not&gt;</code>	$\not=$	<code>\not=</code>	$\not\leq$	<code>\not\leq</code>
$\not\geq$	<code>\not\geq</code>	$\not\equiv$	<code>\not\equiv</code>	$\not\prec$	<code>\not\prec</code>	$\not\succ$	<code>\not\succ</code>
$\not\sim$	<code>\not\sim</code>	$\not\preceq$	<code>\not\preceq</code>	$\not\succeq$	<code>\not\succeq</code>	$\not\simeq$	<code>\not\simeq</code>
$\not\asymp$	<code>\not\asymp</code>	$\not\subset$	<code>\not\subset</code>	$\not\supset$	<code>\not\supset</code>	$\not\approx$	<code>\not\approx</code>
$\not\subseteq$	<code>\not\subseteq</code>	$\not\supseteq$	<code>\not\supseteq</code>	$\not\sqsubseteq$	<code>\not\sqsubseteq</code>	$\not\cong$	<code>\not\cong</code>
$\not\sqsupseteq$	<code>\not\sqsupseteq</code>	$\not\in$	<code>\not\in</code>	$\not\ni$	<code>\not\ni</code>		

### 9). 箭头符号

$\leftarrow$	<code>\leftarrow</code>	$\longleftarrow$	<code>\longleftarrow</code>	$\uparrow$	<code>\uparrow</code>
$\Leftarrow$	<code>\Leftarrow</code>	$\Longleftarrow$	<code>\Longleftarrow</code>	$\Uparrow$	<code>\Uparrow</code>
$\rightarrow$	<code>\rightarrow</code>	$\longrightarrow$	<code>\longrightarrow</code>	$\downarrow$	<code>\downarrow</code>
$\Rightarrow$	<code>\Rightarrow</code>	$\Longrightarrow$	<code>\Longrightarrow</code>	$\Downarrow$	<code>\Downarrow</code>
$\leftrightarrow$	<code>\leftrightarrow</code>	$\longleftrightarrow$	<code>\longleftrightarrow</code>	$\updownarrow$	<code>\updownarrow</code>
$\mapsto$	<code>\mapsto</code>	$\longmapsto$	<code>\longmapsto</code>	$\nearrow$	<code>\nearrow</code>
$\hookleftarrow$	<code>\hookleftarrow</code>	$\hookrightarrow$	<code>\hookrightarrow</code>	$\searrow$	<code>\searrow</code>
$\leftharpoonup$	<code>\leftharpoonup</code>	$\rightharpoonup$	<code>\rightharpoonup</code>	$\swarrow$	<code>\swarrow</code>
$\leftharpoondown$	<code>\leftharpoondown</code>	$\rightharpoondown$	<code>\rightharpoondown</code>	$\nwarrow$	<code>\nwarrow</code>
$\rightleftharpoons$	<code>\rightleftharpoons</code>				

## 10). 边界符号

[ \lbrack	[ \lfloor	{ \lbrace	[ \lceil
< \langle	] \rbrack	] \rfloor	] \rceil
} \rbrace	> \rangle		

## 附录 B 如何从计算机网络上获得信息

L<sup>A</sup>T<sub>E</sub>X的开发者已经将其开发的各个版本和工具包 (package) 放到国际互联网的一些公共网址下, 用户可以通过匿名的 (anonymous) ftp 方式去获得这些软件。

常用的网址为

- 德国网址 ftp.dante.de:
- 英国网址 ftp.tex.ac.uk:
- 美国网址 ftp.shsu.edu:

匿名进入该网址后还应再进入 tex-archive/macros/latex 目录, 在该目录下可以进一步查找有关的文件。例如很多工具包可以在其下面的 contrib/others, contrib/packages 或 contrib/supported 等目录下查出。

中文版 L<sup>A</sup>T<sub>E</sub>X的开发者们也将最新版本的软件放到网上, 其地址为

ftp.cc.ac.cn:/pub/cct

ftp.cnc.ac.cn:/CASnet/cct

由于中科院的计算机网络和我们常用的教育网之间信息传递速度不是很理想, 所以软件下载 (download) 是相当慢的, 经开发者张林波先生许可, 特将 5.11 版的中文版 L<sup>A</sup>T<sub>E</sub>X装载到东北大学校园网下, 以供用户下载并使用。

网络地址: ftp2.synet.edu.cn:/pub/pc/format/cct511

除了 CCT 5.11 版以外, 这里提到的其它软件如各种工具包、GhostView 及 DVIPS 所需要的工具等也放到同一目录下, 以备用户下载。

在国际上 T<sub>E</sub>X用户组织了一个用户组 (TUG - T<sub>E</sub>X users' group), 并定期在网络上刊出其杂志 TUGboat, 交流并探讨 T<sub>E</sub>X的应用与技巧, 这些信息用户也可以通过网络免费获得。用户还可以键入 tin -r 命令并查找 .tex 来查询最新信息, 阅读一些高级用户关于 L<sup>A</sup>T<sub>E</sub>X的问答。

## 附录 C 标准排版注意事项

根据国际排版上的惯例, 下面的排版结果都存在问题:

$$\int_0^{\infty} f(x)dx, \quad A = e^x$$

原因是 d 和 e 均不是变量名, 所以应该采用正体, 而不应该采用斜体。上面公式的正确排版格式为

$$\int_0^{\infty} f(x)dx, \quad A = e^x$$

这个部分将根据清华大学出版社的《著译者手册》<sup>[6]</sup> 介绍排版的标准, 并以举例的方式列举出一些常用错误。

• **文字与数字:** 下面情况应采用阿拉伯数字表示

- 1). 表示世纪、年代、年月日等: 如 20 世纪 90 年代、1994 年 2 月 14 日下午 3 时 40 分等, 1994 年不得写成 94 年, 1980-1994 年不能写成 1980-94 年。
- 2). 记数与计量等: 如 -312,  $1/3$ , 98%, 5.5 倍, 1 岁 10 个月。

下列情况应使用汉字表示数字

- 1). 固定词组等, 如二氧化碳, 第一书记, “八五”规划, 相差十万八千里。
- 2). 邻近两个数字并列使用, 表示概数 (不能用顿号分隔) 如三五天, 十之八九等。
- 3). 不是出现在一组表示科学计量和具有统计意义数字中的一位数, 如一个人,

• **正体与斜体:**

- 1). 数学公式中的变量名或以字母代表的函数名应采用斜体表示, 如  $x, y, f(x)$  等。
- 2). 单位制应采用正体, 如 Nm (牛顿米)、kg (公斤)、kPa (千帕) 等。
- 3). 指数、对数及三角函数名等应采用正体, 如  $\sin(\alpha t)$ ,  $\ln 2$ ,  $\det(A)$  (即  $A$  矩阵的行列式) 等<sup>10</sup>。
- 4). 算子符号应采用正体, 如  $dx$  而不是  $dx$ ,  $\lim_{x \rightarrow \infty} p(x)$  而不是  $\lim_{x \rightarrow \infty} p(x)$ 。
- 5). 常数符号应使用正体, 如  $e, j$  和  $\pi$  等。

• **数学公式处理:**

- 1). 较简单的、叙述性的公式可串写于正文行中, 结论性或较长的公式应单独占行居中书写。公式末尾可以加标点符号, 也可以不加, 但应该全书保持一致。
- 2). 较长的公式需要转行时, 应首先考虑等号 (或大于、小于号) 处, 其次考虑在运算符号处, 在乘号处转行时, 乘号应写为  $\times$ 。
- 3). 文本行中数学公式应尽量写成横式  $1/a$  或负幂次  $a^{-1}$ , 少用叠式  $\frac{1}{a}$ , 以节省版面并美观。必要时分子分母应加括号以免发生歧义, 如  $x/y/z$  可理解成  $(x/y)/z$ , 也可理解成  $x/(y/z)$ , 故书写时应注意。
- 4). 较复杂的指数函数应尽量用  $\exp$  写成横排式, 如  $e^{\frac{c+d}{a+b}}$  应写成  $\exp[(c+d)/(a+b)]$ 。
- 5). 乘式中字母与数符之间、字母数符与前面的数字之间以及两个括号之间不能用乘号, 如  $2\alpha \cos \alpha$ 。
- 6). 数字与数字之间、字母数符与后面的数字之间应该用乘号分隔, 如  $3b \sin \alpha \times 10^{-2}$ 。

• **参考文献格式:** 在写参考文献时, 若著者不超过 3 人时应全部著录, 其名字之间用逗号分隔, 若著者为 4 个或以上时, 只著录前 3 人, 其后加“等” (英文加 *et al*)。西文著者姓在前, 名字采用缩写字母 (不加点), 姓与名间用空格分隔, 不应使用逗号。这里将分下列 3 项给出标准的中国图书参考文献格式 (注意其中标点符号运用)

- 1). **图书格式:** 著者. 书名. 版本 (第 1 版不注). 出版地: 出版社, 出版年

例如: [1] 薛定宇. 控制系统计算机辅助设计—MATLAB 语言及应用. 北京: 清华大学出版社, 1996

- 2). **期刊文献格式:** 著者. 题目 (可省略). 期刊名. 出版年, 卷号 (期号), 页码

例如: [2] Xue D, Atherton D P. A suboptimal reduction algorithm for linear systems with a time delay. *International Journal of Control*, 1994, 60(2): 181-196

- 3). **论文集中文章格式:** 著者. 题目. 见 (In): 论文集编者. 论文集名. 版本 (第 1 版不注). 出版地: 出版者, 出版年, 页码



例如: [3] Xue D, Atherton D P. A menu-driven model reduction program and its applications. In: Barker H A, ed. Computer-aided design in control systems. Oxford: Pergamon Press, 1992. 232-238

- **各种环境的字号大小:** 规范的中文排版中要求正文使用 5 号字, 而图、表的标题和其中的内容应选择 6 号字。
- **清华大学出版社版面格式定义:** 这里以典型的清华大学出版社要求的页面设置为例, 给出下列 L<sup>A</sup>T<sub>E</sub>X 相应的命令及文件框架, 以便参考。当然用户也可以仿照此文件建立他自己学位论文的 L<sup>A</sup>T<sub>E</sub>X 文件框架。

```
\documentclass[11pt]{cbook}
\usepackage{emlines2}
\usepackage{epsf} % and other packages
\topmargin 00mm
\oddsidemargin 0mm
\evensidemargin 0mm
\textwidth 155mm
\textheight 210mm
\newcommand{\double}{\baselineskip 1.3 \baselineskip}
\parindent 2\ccwd
\parskip 0mm
\hyphenpenalty=1500
\flushbottom
\newcommand{\ispace}{\protect \hspace{2\ccwd}}
\newcommand{\citeu}[1]{${\mbox{\protect \scriptsize \cite{#1}}}$}
\begin{document}\zihao{5}\double
-----
    Main Body of Text
-----
\end{document}
```

## 参 考 文 献

- [1] Adobe Systems Incorporated. PostScript language reference manual (2nd edition). Reading MA: Addison-Wesley Publishing Company, 1990
- [2] Goossens M, Mittelbach F, Samarin A. The L<sup>A</sup>T<sub>E</sub>X companion. Reading MA: Addison-Wesley Publishing Company, 1994
- [3] Knuth D E, The T<sub>E</sub>Xbook: Volume A of computers and typesetting. Reading MA: Addison-Wesley Publishing Company, 1986
- [4] Lamport L. L<sup>A</sup>T<sub>E</sub>X: a document preparation system — user's guide and reference manual, Reading MA: Addison-Wesley Publishing Company, (1st edition 1985, 2nd edition 1994)
- [5] Lamport L, Mattes E. Using L<sup>A</sup>T<sub>E</sub>X with emT<sub>E</sub>X. Internal Report , 1989
- [6] 清华大学出版社. 译著者手册, 1994 年
- [7] 张林波等. CCT 中外文科技排版系统. 北京: 海洋出版社, 1993