

LATEX



SOOCHOW UNIVERSITY

排版手記

吳昊

目 录

第一章 L^AT_EX 基础	1	4.2 array 宏包	9
1.1 命令	1	4.3 浮动体	10
1.1.1 命令形式	1	4.4 表格	10
1.2 文类	1	4.4.1 跨行和跨列表格	11
1.3 字体	1	4.4.2 彩色表格	12
1.3.1 全局字体设置	2	4.4.3 斜线表头	14
1.3.2 局部字体设置	2	4.4.4 表格标题	14
1.3.3 在数学环境中使用中文	2	第五章 正文工具	15
1.3.4 汉字“斜体”	2	5.1 目录	15
1.4 长度	2	5.2 脚注	15
1.5 空格、换行、分段、分页与对齐	3	5.3 边注	15
第二章 绘图	3	5.4 参考文献	16
2.1 函数图	3	5.5 链接	17
2.1.1 二维函数图	3	5.6 引用功能	17
2.1.2 三维函数图	4	5.7 列表	17
2.1.3 隐函数图像	5	5.7.1 常规列表	18
2.2 线图	5	5.7.2 排序列表	18
2.3 条形图	5	5.7.3 解说列表	19
2.4 从外部数据作图	6	5.7.4 列表宏包	19
第三章 版面和格式	7	5.8 附录	20
3.1 页眉页脚	7	5.9 代码环境	20
3.2 颜色	7	第六章 数学排版	23
3.3 文本格式	8	6.1 数学模式	23
3.3.1 下画线	8	6.2 数学宏包	23
3.3.2 符号	8	6.3 数学符号	23
3.4 标题	9	6.4 公式环境	23
第四章 图和表	9	6.5 矩阵环境	23
4.1 插图	9	6.6 定理环境	23
		第七章 Tikz	23
		7.1 带箭头的注释效果	24
		第八章 宏包	24
		8.1 mhchem	24
		8.2 Chemfig	28
		8.3 C _T _E Xzhnumber	28

第一章 L^AT_EX 基础

1.1 命令

1.1.1 命令形式

命令的一般形式如下。

```
1 \命令名[可选参数1,可选参数2,...]{必要参数1,必要参数2,...}
```

同时, 根据命令的作用范围的不同, 可大致把命令分为如下四种形式。

1.2 文类

C_T_EX 提供四个中文文类, `ctexart`、`ctexrep`、`ctexbook` 和 `ctexbeamer`

1.3 字体

等宽字体 Typewriter Family 英文的 a 和 i 在非等宽字体里面肯定宽度不一样, 这样在大段文本里就不好辨认, 等宽字体的所以字母宽度一样, 笔画的起止还有装饰衬线 (所以等宽字体多数属于衬线字体), 易读性高

等线字体 无字头字脚, 笔画圆润, 粗细均匀, 例如 Windows 自带的 Arial、黑体和幼圆

衬线字体 serif 在字的笔画开始、结束的地方有额外的装饰, 而且笔画的粗细会有所不同, 宋体就是一种最标准的 serif 字体

无衬线字体 sans serif 在字的笔画开始、结束的地方没有这些额外的装饰, 而且笔画的粗细差不多

等宽字体一般用来书写代码, 特别是使用缩进控制语法的 *python* 语言, 更需要等宽字体来书写代码了。

科学书写中文文档的第一步应该是调用 C_T_EX 宏包, 其提供四种命令来调用在中文文档中常用的四种字体。

```
1 {\songti 爆竹声中一岁除, 春风送暖入屠苏。}  
2 {\fangsong 家家乞巧望秋月, 穿尽红丝几万条。}  
3 {\heiti 黄沙百战穿金甲, 不破楼兰终不还。}  
4 {\kaishu 君不见走马川行雪海边, 平沙莽莽黄入天。}
```

效果如下:

爆竹声中一岁除, 春风送暖入屠苏。
家家乞巧望秋月, 穿尽红丝几万条。
黄沙百战穿金甲, 不破楼兰终不还。
君不见走马川行雪海边, 平沙莽莽黄入天。

分全局和局部字体设置。

1.3.1 全局字体设置

中文的文档都要调用 *ctex* 宏包，该宏包提供一个简单的参数可以设置全部正文的字体。

```
1 \setmainfont{Times New Roman} %设置主字体，仅对西文起作用
2 \setCJKmainfont{SimSun} %设置主字体，仅对中文起作用
```

有时候需要改变 L^AT_EX 默认的等宽字体，如本文档的等宽字体设置。更改等宽字体之后，将会影响 `\texttt{text}`, `\ttfamily`, `\tt` 这些命令所作用的字体，还会影响默认使用等宽字体（如脚注，抄录环境）的环境。

```
1 \setmonofont{Source Code Pro} %英文等宽
2 \setCJKmonofont{simfang.ttf} %中文等宽，仿宋
```

1.3.2 局部字体设置

```
1 \newfontfamily\daima{Consolas} %使用\daima直接调用
```

1.3.3 在数学环境中使用中文

默认情况下，数学环境中是不允许输入汉字的。当我们需要输入汉字作为变量的标识时，可以使用 `\text{要输入的汉字字符}` 来完成这项工作。

$$f_{\text{高温}}$$

$$f_{\text{高温}}$$

1.3.4 汉字“斜体”

汉字没有加斜体。平常我们看到的加斜汉字，通常是几何变换得到的结果，非常的粗糙，并不严格满足排版要求；而真正的字形是需要精细的设计的。同时，汉字字体里面也很少有加粗体的设计。但是，有时候却又有所谓的“斜体”需求。L^AT_EX 也是可以实现这种伪斜体的。虽然可以实现，但排版规范并不推荐我们使用斜体来强调某个元素。如果想要强调某个元素，可以使用黑体。

汉字伪斜体

```
1 {\CJKfontspec[FakeSlant=0.4]{SimSun}\zihao{1} 汉字伪斜体}
```

西文一般设有加斜，但是这与“斜体”并不是同一回事。加斜是指某种字族的 *Slant* 字系；而斜体，是指 *Italy* 字族。所以 `\textit{}`, `\itshape`, `\it` 是斜体 (*Italic Shape*)，而 `\textsl{}`, `\slshape`, `\sl` 是字体的倾斜形状。

Italy Shape *Slanted Shape*

1.4 长度

我一般只用如下几个：

em 当前字号下，大写字母 M 的宽度，用于水平间距调整

ex 当前字号下，小写字母 x 的高度，用于垂直间距调整

mm 毫米，都懂

1.5 空格、换行、分段、分页与对齐

L^AT_EX 将多个空格视为一个，多个换行也会被视为一个。一般习惯使用~产生一个空格，使用\mbox{}产生一个空白段落（空白行），使用\par产生一个带缩进的新段，使用\\来强制换行，但下一段的缩进会消失。

段落之间的距离一般这样控制：

```
1 \setlength{\parskip}{0pt plus 1pt}%默认值
```

宏包 *lettrine* 能产生首字下沉效果：

\lettrine{吾}{}生也有涯，而知也无涯。
以有涯随无涯，殆已！

吾生也有涯，而知也无涯。以有涯随
无涯，殆已！

用\newpage命令开始新的一页。用\clearpage命令清空浮动体队列，并开始新的一页。用\cleardoublepage命令清空浮动体队列，并在偶数页上开始新的一页。注意：以上命令都是基于\vfill的。如果要连续新开两页，请在中间加上一个空的箱，如\newpage\mbox{}\newpage。

L^AT_EX 默认使用两端对齐来排版，我们可以用flushleft,flushright,center这三个环境来构造居左，居右，居中三种版式。特殊情况可以使用\centering,\raggedleft,\raggedright来实现居中，居右，居左。

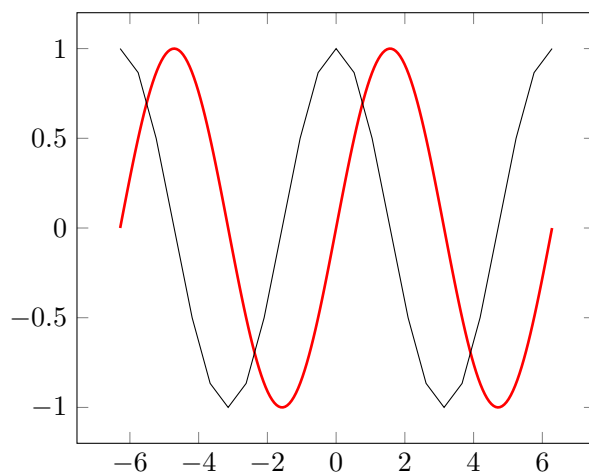
第二章 绘图

2.1 函数图

2.1.1 二维函数图

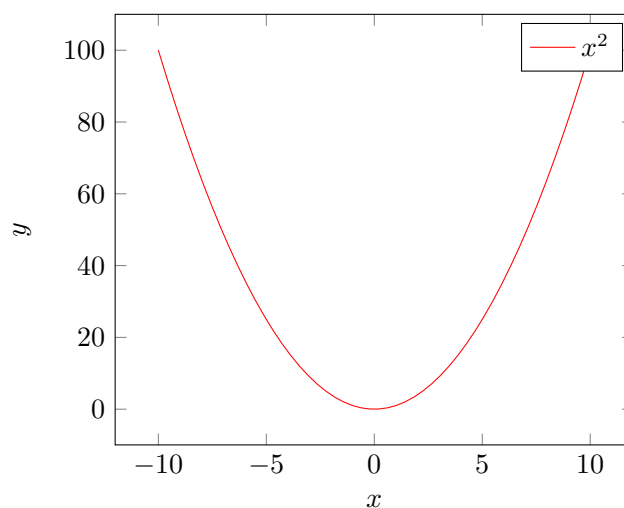
绘制函数 $f(x) = \sin(x)$ 和 $g(x) = \cos(x)$ 在定义域 $[-2\pi, 2\pi]$ 上的图像。对于函数 $f(x) = \sin(x)$ ，我们使用 red 和/tikz/line width={1pt}来使其更显眼。L^AT_EX 默认取点少，我们使用 samples=x来控制取点数，使函数图像更为平滑。

```
1 \addplot[domain=-2*pi:2*pi,samples=200,red,/tikz/line width={1pt}]{sin(deg(x))};  
2 \addplot[domain=-2*pi:2*pi]{cos(deg(x))};
```



绘图命令里面是可以直接输入函数表达式的，使用如下命令可以直接绘制函数 x^2 在定义域 $[-10:10]$ 上的图像。

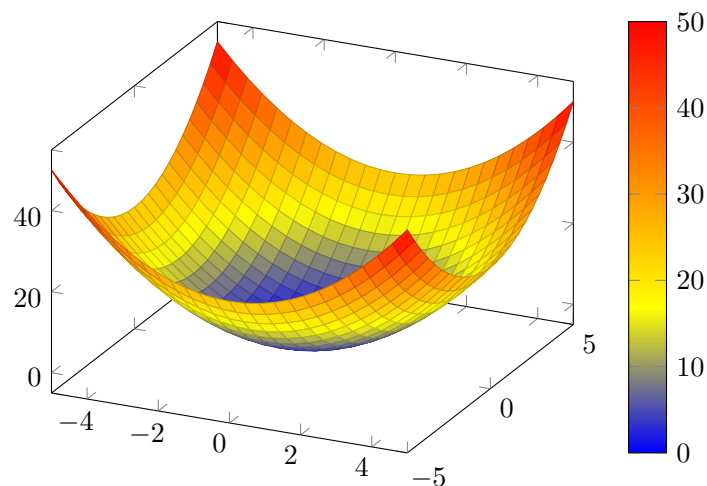
```
1 \addplot[domain=-10:10,red,samples=50]{x^2}
```



2.1.2 三维函数图

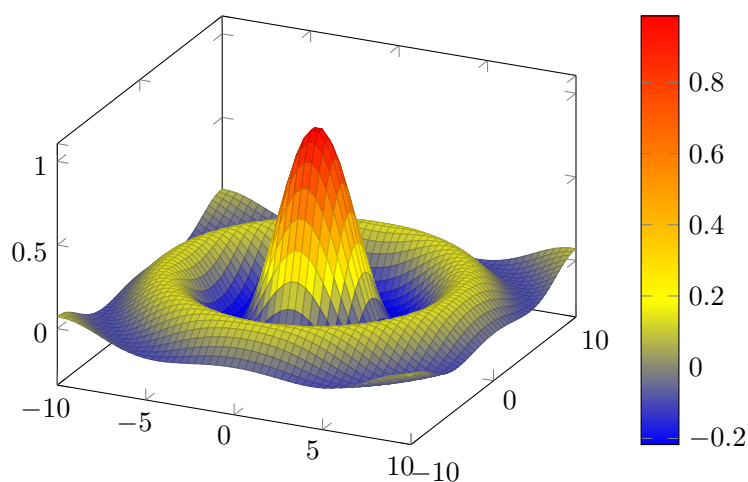
绘制函数 $f(x, y) = x^2 + y^2$ 的三维函数图像。其中参数 *colorbar* 可以为函数图添加一个彩色条。

```
1 \begin{tikzpicture}
2 \begin{axis}[colorbar]
3 \addplot3[surf]{x^2+y^2};
4 \end{axis}
5 \end{tikzpicture}
```



绘制函数 $g(x, y) = \frac{\sin(\sqrt{x^2 + y^2})}{\sqrt{x^2 + y^2}}$ 的三维函数图像。

```
1 \addplot3[surf,samples=50,domain=-10:10]{sin(deg(sqrt(x^2+y^2)))/sqrt(x^2+y^2)};
```



2.1.3 隐函数图像

2.2 线图

对于折线图而言，使用 `sharp plot` 可以得到折线图，而使用 `smooth` 可以得到平滑曲线图。这些命令在 `addplot` 的设置里面添加。一些其他基础设置如下所示。

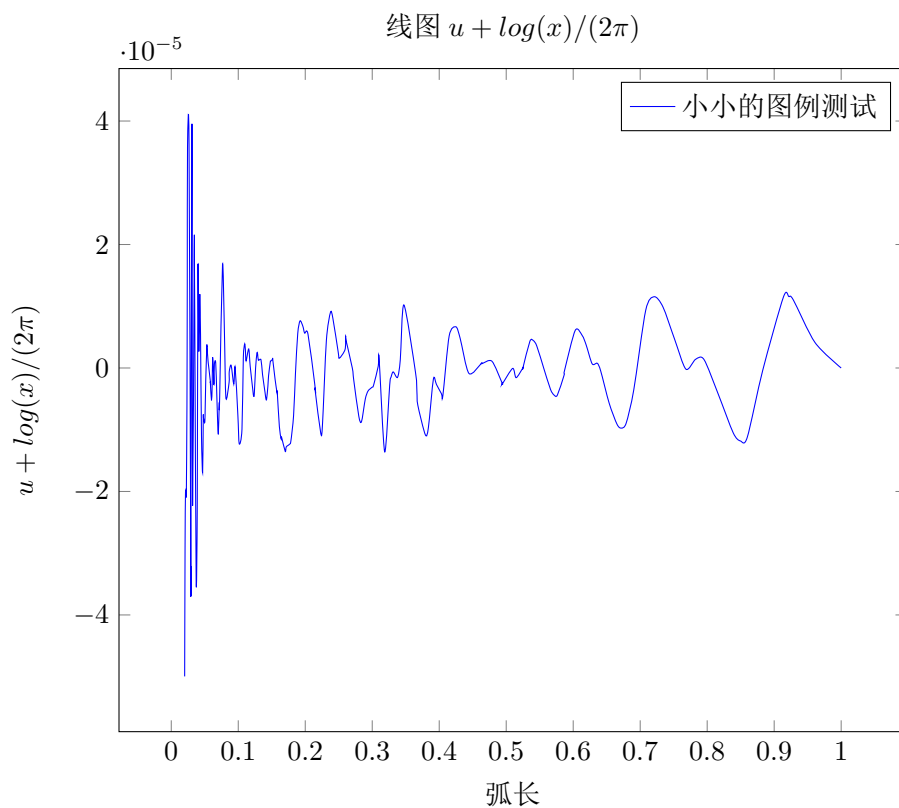
```
1 \item grid=major      %显示网格线
2 \item mark=x          %设置数据点标记类型
```

2.3 条形图

2.4 从外部数据作图

从外部文件加载数据，例如从 COMSOL 导出数据为 txt 文档，在 \LaTeX 中可以直接读取其中的数据作图，只要指定正确的数据路径即可。

```
1 \addplot[blue,smooth,domain=0:1,/tikz/line width={0.4pt}] table[x=x,y=y] {
  body/data.txt};
```



第三章 版面和格式

3.1 页眉页脚

一般来说，设置页眉页脚需要调用使用比较广泛的 *fancyhdr* 宏包。我习惯使用如下代码先清空默认定义，然后自己重新定义。页眉页脚线的粗细也可以重新定义。

```
1 \usepackage{fancyhdr}
2 \pagestyle{fancy}
3 \fancyhf{} %清空当前设置
4 %单页文档
5 \lhead{} %l, r, c, 左中右
6 \cfoot{}
7 %双页文档
8 \fancyhead[R0,LE]{} %E, O, 左、右页
9 \fancyfoot[LE,R0]{\thepage}
10 \renewcommand{\headrulewidth}{0.4 pt}
11 \renewcommand{\footrulewidth}{0.4 pt}
```

我们可以将章节标题和序号插入到页眉或者页脚中去，其格式与正文中章节标题的定义一样。如果需要更改，要重新定义。例如，可以使用如下代码重新定义页眉内的章标题样式，用在在本书中，这将会使页眉的“第 X 章版式”更改为“X 版式”。

具体更改页眉页脚区域章节显示样式的代码如下。

```
1 \renewcommand{\chaptermark}[1]{\markleft{\thesection.\#1}}
2 %两种一样，\markleft影响\leftmark，而\makeboth影响两着，需要选一
3 \renewcommand{\chaptermark}[1]{\markboth{\thechapter.\#1}{节样式空置表示修改
  章样式}}
4 \renewcommand{\chaptermark}[1]{\markboth{章样式}{节样式}}
```

在 book 文件类别下，第二章 绘图自动存录各章之章名，2.4 从外部数据作图记录节标题。所以，想要在页眉上显示章节标题是很容易实现的。

```
1 \lhead{\leftmark} %左页眉显示章
2 \rhead{\rightmark} %右页眉显示节
```

3.2 颜色

一般来说，我们调用下 *xcolor* 这个宏包。如果对内置的颜色了解，或者现有 RGB 颜色值，一般使用如下代码直接调用颜色。

Color Text 中文测试

```
1 \color[RGB]{204, 128, 92}{Color Text中文测试}
```

但是每次调用颜色都写颜色代码似乎不方便，我们可以先定义，基本定义形式如下。

```

1 \usepackage{xcolor} %颜色宏包
2 \definecolor{backcolor}{RGB}{242,242,242} %背景色
3 \definecolor{comment}{RGB}{0,128,0} %注释
4 \definecolor{keyword}{RGB}{0,0,255} %关键词
5 \definecolor{name名字随意}{model色值类型}{color-spec色值范围}

```

然后，我们就可以直接调用我们定义的颜色名称来设定颜色了。

function, return, if, true, false

```
1 \color{keyword}{\slshape function, return, if, true, false}
```

3.3 文本格式

3.3.1 下画线

L^AT_EX 自身提供的下划线之类的命令并不好用，我们使用 *ulem* 或者 *CJKfntef* 宏包来代替。对于中文排版，在 *ctex* 宏包的调用命令中添加 *fntef* 选项即可调用 *CJKfntef* 宏包来画线，*CJKfntef* 宏包的优点是其画线命令中的中文能自动中断正确换行。本书以 *CJKfntef* 宏包为例。

```

\CJKunderdot{important 非常重要}\\
\CJKunderline{notice 注意}\\
\CJKunderdblline{urgent 必须}\\
\CJKunderwave{prompt 提示}\\
\CJKsout{wrong 错误}\\
\CJKxout{removed 删除}

```

important 非常重要
notice 注意
urgent 必须
prompt 提示
~~wrong 错误~~
~~removed 删除~~

对于前四种下划线，该宏包还提供了调节下划线与文本之间距离的参数。默认值都是 0.2em，需要使用重新定义命令来使参数生效。

```

1 \CJKunderdotbasesep
2 \CJKunderlinebasesep
3 \CJKunderdbllinebasesep
4 \CJKunderwavebasesep
5 \renewcommand{\CJKunderlinebasesep}{0.5em}

```

同时，还提供了 6 条颜色命令。大致命令形式和调用方法如下所示，其中 *color* 命令有多种调用模式，详见本书颜色部分。

```

1 \CJKunderlinecolor
2 \renewcommand{\CJKunderline}{\color{blue}}

```

3.3.2 符号

英文的短横可以产生三种符号：

短横

- 连字符：输入一个短横，-，效果如 daughter-in-law
- 数学起止符：输入两个短横，--，效果如 page1-2
- 英文破折号：输入三个短横：—，效果如 Listen—I'm serious

中文破折号，省略号一般直接用中文输入法输入，但英文的省略号值得注意，需要使用`\ldots`来输入。

这两个符号需要借助数学模式`$...$`来输入：

角度符号, 摄氏度符号

<code>\$30\,^{\circ}\$</code> <code>\$37\,^{\circ}\mathrm{C}\$</code>	<code>30^{\circ} 37^{\circ}\mathrm{C}</code>
--	--

3.4 标题

CT_EX 宏包提供标题修改功能，所以中文文档很容易实现标题的修改。如果是直接使用 CT_EX 提供的文类，标题是可以直接修改的；如果仅仅是调用了 CT_EX 宏包，需要给宏包加上调用参数才能修改标题。

```
1 \usepackage[
2 heading=true,%启用修改章节标题的接口
3 ]{ctex}
```

第四章 图和表

4.1 插图

4.2 array 宏包

数组宏包 *array* 改进和扩展了 L^AT_EX 的 *tabular*、*tabular**、*array* 环境的功能，增强了列格式的功能和一些其他表格参数的调整功能。

表 4.1 array 宏包基本参数

选项	说明
<code>l</code>	左对齐
<code>c</code>	居中
<code>r</code>	右对齐
<code>p{列宽}</code>	顶对齐
<code>m{列宽}</code>	居中对齐
<code>b{列宽}</code>	底对齐
<code>@{声明}</code>	该列每行都插入声明中的文本
<code>>{声明}</code>	命令或需要插入列元素前的文本
<code><{声明}</code>	命令或需要插入列元素后的文本
<code> </code>	在列边或列间插入垂直线
<code>!{声明}</code>	在列间插入声明要求的样式

4.3 浮动体

浮动体将图或表与其标题定义为整体，然后动态排版，以解决图、表卡在换页处造成的过长的垂直空白的问题。但有时它也会打乱你的排版意图，因此使用与否需要根据情况决定。图片的浮动体是 *figure* 环境，而表格的浮动体是 *table* 环境。

对表格来说，输出表格内容的是 *tabular* 环境，*table* 只是一个会浮动体（到处乱跑盒子）而已。没有 *tabular* 环境，*table* 环境一样会乱跑；没有 *table* 环境，*tabular* 环境一样会输出表格内容。

下面是一个典型的浮动体例子。参数 *!htb* 含义是：！表示忽略内部参数（比如内部参数对一页中浮动体数量的限制）；h、t、b 分别表示插入此处、插入页面顶部、插入页面底部，故 *htb* 表示优先插入此处，再尝试插入到某页顶，最后尝试插入到页底。此外还有参数 *p*，表示允许为浮动体单独开一页。L^AT_EX 的默认参数是 *tbp*。请不要单独使用 *htbp* 中的某个参数，以免造成不稳定。另外需要注意的是 *label* 命令写在 *caption* 命令下方，否则交叉引用会出现问题。

```
1 \begin{table }[!htb]           %浮动体环境
2     \begin{center}             %居中环境
3         \caption{标题}
4         \label{用于交叉引用的标签}
5         \begin{tabular }{}     %表格环境
6             \end{tabular}
7     \end{center}
8 \end{table}
```

4.4 表格

L^AT_EX 原生的表格功能非常有限，甚至不支持单元格跨行和表格跨页，我们必须通过宏包来解决。如有需求，可在 *tabular* 环境外定义全部表格线的粗细，例如，`\setlength{\arrayrulewidth}{2pt}` 或者直接写 `\arrayrulewidth=2pt`。

```
\centering
\arrayrulewidth=1pt%表格线宽度
\begin{tabular}
{ | C{6mm} | C{6mm} | C{6mm} | }
\hline
\multicolumn{3}{|c|}{整体表格线宽}\\
\hline
7&5&3\\
\hline
6&1&8\\
\hline
\end{tabular}
```

整体表格线宽		
7	5	3
6	1	8

如果需要单独定义某一条表格线的粗细，必须要做额外的设置。

如果我们要更改垂直表格线的粗细，可以利用 *array* 宏包提供的新列格式选项定义命令。其中的新选项名只能用一个字母来表示。使用该命令更改中间两条垂直线粗细为 2pt。

```
1 \newcolumntype{新选项名称}[参数数量]{列格式}
```

```
2 \newcolumntype{I}{!{\vrule width 4pt}}
```

```
\centering
\newcolumntype{I}{!{\vrule width 2pt}}
\begin{tabular}
{|C{6mm}|C{6mm}|C{6mm}|}
\hline
\multicolumn{3}{IcI}{垂直线粗细}\\
\hline
7&5&3\\
\hline
6&1&8\\
\hline
\end{tabular}
```

垂直线粗细		
7	5	3
6	1	8

水平表格线的粗细较难修改，需要使用 *booktabs* 宏包，该宏包可以任意修改水平线粗细，还可以在其上、下方附加一段垂直空白。

```
\centering
\begin{tabular}
{|C{8mm}|C{8mm}|C{8mm}|}
\hline
\multicolumn{3}{|c|}{水平线宽}\\
\specialrule{2pt}{0pt}{0pt}
7&5&3\\
\hline
6&1&8\\
\hline
\end{tabular}
```

水平线宽		
7	5	3
6	1	8

array 包重新实现了 *tabular* 环境，加了不少新选项进去。比如我们可以定义 *F* 为一个居中且在数学环境中的列类型。然后在 *tabular* 中调用 *F* 即可在表格环境中排出数学样式。

```
\centering
\newcolumntype{F}{>{\$}c<{\$}}
\begin{tabular}{FFF}
\alpha & \beta & \gamma \\
\delta & \epsilon & \upsilon \\
\sigma & \tau & \phi
\end{tabular}
```

α	β	γ
δ	ϵ	υ
σ	τ	ϕ

4.4.1 跨行和跨列表格

既跨行又跨列时，必须把 *multirow* 命令放在 *multicolumn* 内部，始终记住跨列享受最高的优先级。

```

\centering
\begin{center}
\begin{tabular}{|c|c|c|}
\hline
\multirow{2}{2cm}{A Text!}
& ABC & DEF \\
& abc & def \\
\hline
\multicolumn{2}{|c|}
{ \multirow{2}{*}{Nothing}} & XYZ \\
\multicolumn{2}{|c|}
& xyz \\
\hline
\end{tabular}
\end{center}

```

A Text!	ABC	DEF
	abc	def
Nothing		XYZ
		xyz

```

\centering
\begin{tabular}{|ccc|}
\hline
2&9&4\\
7&\multicolumn{2}{c|}
{ \multirow{2}{*}{?}} \\
6&&\\
\hline
\end{tabular}

```

2	9	4
7	?	
6		

4.4.2 彩色表格

彩色表格。该宏包主要使用的命令是 *columncolor* 和 *rowcolor*，一个用来给列进行着色，一个是给行进行着色，下面这个例子已经全部涉及到了。

```

\centering
\begin{tabular}{ccc}
\rowcolor[gray]{.9}
2&9&4\\
\rowcolor[gray]{.8}
7&5&3\\
\rowcolor[gray]{.7}
6&1&8\\
\end{tabular}

```

2	9	4
7	5	3
6	1	8

```

\centering
\begin{tabular}%
{>{\columncolor[gray]{.9}}c%
>{\columncolor[gray]{.8}}c%
>{\columncolor[gray]{.7}}c}
2&9&4\\
7&5&3\\
6&1&8\\
\end{tabular}

```

2	9	4
7	5	3
6	1	8

```

\centering
\begin{tabular}{ccc}
\cellcolor[rgb]{.9,.9,.9}2&
\cellcolor[rgb]{.8,.9,.9}9&
\cellcolor[rgb]{.7,.9,.9}4\\
\cellcolor[rgb]{.9,.8,.9}7&
\cellcolor[rgb]{.8,.8,.9}5&
\cellcolor[rgb]{.7,.8,.9}3\\
\cellcolor[rgb]{.9,.7,.9}6&
\cellcolor[rgb]{.8,.7,.9}1&
\cellcolor[rgb]{.7,.7,.9}8\\
\end{tabular}

```

2	9	4
7	5	3
6	1	8

一个复杂的彩色表格例子，代码留着以后仔细看，彩色表格应该用的不多。

```

1 %使用array宏包的特性来定义几个表格属性，只适用于本环境
2 \newcommand*{\arraycolor}[1]{\protect\leavevmode\color{#1}}
3 \newcolumnntype{A}{>{\columncolor{blue!50!white}}c}
4 \newcolumnntype{B}{>{\columncolor{LightGoldenrod}}c}
5 \newcolumnntype{C}{>{\columncolor{FireBrick!50}}c}
6 \newcolumnntype{D}{>{\columncolor{Gray!42}}c}
7 \begin{center}
8   \sffamily
9   \arrayrulecolor{white}
10  \arrayrulewidth=1pt
11  \renewcommand{\arraystretch}{1.5}
12  \rowcolors[\hline]{3}{.!50!White}{}
13  \begin{tabular}{A|B|C}
14    \multicolumn{3}{D}{\bfseries Example table}\\
15    \rowcolor{.!50!Black}
16    \arraycolor{White}\bfseries First column &
17    \arraycolor{White}\bfseries Second column &
18    \arraycolor{White}\bfseries Third column\\
19    1 & A & E\\
20    2 & B & F\\
21    3 & C & G\\
22    4 & D & H\\
23  \end{tabular}
24 \end{center}

```

4.4.3 斜线表头

虽然斜线表头是不符合国标的,但在非正式场合用得还挺多的。制作斜线表头需要 *diagbox* 宏包,刘海洋写的,中文说明。

```
\centering
\begin{tabular}{|l|ccc|}
\hline
\diagbox{Time}{Room}{Day}
&Mon&Tue&Wed\\
\hline
Morning&used&used&\\
Afternoon& &used&used\\
\hline
\end{tabular}
```

Room \ Day	Mon	Tue	Wed
Time			
Morning	used	used	
Afternoon		used	used

4.4.4 表格标题

表格标题命令默认只能在浮动体内使用,在导言中添加如下命令,便可以在浮动体外使用 `\figcaption` 和 `\tabcaption` 命令来为图标添加标题。为了防止标题和图表不在一页,我们也可以用 `minipage` 环境把它们包起来。

```
1 \makeatletter
2 \newcommand\figcaption{\def\@capttype{figure}\caption}
3 \newcommand\tabcaption{\def\@capttype{table}\caption}
4 \makeatother
1 \begin{tabular}{|C{1cm}|C{1cm}|C{1cm}|}
2   \hline
3   \multirow{2}{*{时间}} & \multicolumn{2}{c|}{星期}\\
4   \cline{2-3} & 一 & 二 \\
5   \hline
6   8:30 & 化学 & 物理 \\
7   9:30 & 韩语 & 数学 \\
8   \hline
9 \end{tabular}
```

表 4.2 一张课表

时间	星期	
	一	二
8:30	化学	物理
9:30	韩语	数学

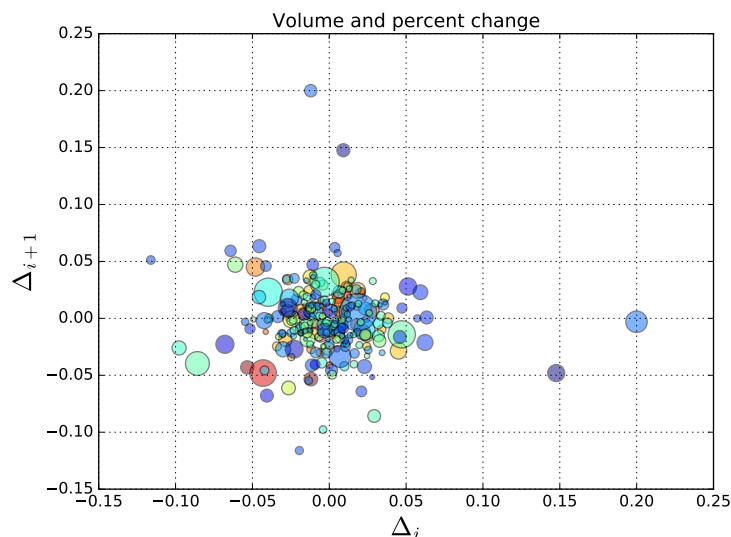


图 4.1 一副图像

第五章 正文工具

5.1 目录

5.2 脚注

脚注是对正文中词语的补充说明。系统提供的脚注命令如下，序号用于自行设定脚注序号，通常不需要给出。

```
1 \footnote[number]{text}
```

例如，为本文作者¹添加脚注。

如果要在脚注中输入带反斜杠的字符串，可使用等宽字体命令加字符串命令输入²。代码如下。如果需要更多的设置，可以调用脚注宏包 *footmisc*，对脚注命令 `\footnote` 进行扩展功能。

```
1 \footnote{\texttt{\string\footnote}}
```

5.3 边注

L^AT_EX 本身提供边注命令：

```
1 \marginpar[左边注]{右边注}
```

边注测试。

这是边注啊

¹邹思宇，男，L^AT_EX 爱好者

²脚注命令 `\footnote`

调用 `marginnote` 宏包，新定义一个边注。使用 `\bz` 调用，将会在与段落平齐的地方生成一个边注。例如：

```
1 % 边注和索引，来自重庆大学 LaTeX 团队
2 \renewcommand*{\marginfont}{\color{Note}\sffamily\heiti}
3 \DeclareDocumentCommand{\bz}{s o m}{%
4   \IfBooleanTF {#1}
5     {%true
6       \IfNoValueTF{#2}{\marginnote[#3]{#3}}{\marginnote[#2]{#3}}
7     }{%false
8       \IfNoValueTF{#2}{\marginnote[#3]{#3}}{\marginnote[#2]{#3}}
9     }
10 }%
11 }
```

从这一行开始是用于重新定义边注的代码

5.4 参考文献

中文著作肯定要符合《GB7714-2015 信息与文献参考文献著录规则》的要求，我习惯使用 `biblatex` 来生成参考文献。在导言区或者自定义的类文件中添加如下 1–5 行的代码，调用 `biblatex` 宏包并指定 `bib` 数据库路径³和名称。在正文中使用⁴第 7 行代码打印参考文献。

本书主要参考了刘海洋^[1]和胡伟^[2]编写的教程。使用的参考文献样式是胡振震编写的，源码托管在 [Github-hushidong/biblatex-gb7714-2015](https://github.com/hushidong/biblatex-gb7714-2015) 上。

```
1 \usepackage[
2   backend=biber,%处理方式
3   style=gb7714-2015%样式
4 ]{biblatex}
5 \addbibresource{Zousiyu.bib}
6
7 \printbibliography%打印参考文献
```

`bib` 参考文献数据格式如下所示，为分字段显示。各字段可以顾名思义，第一行的“刘海洋”是参考文献标识，你在文中引用参考文献时需要使用此标识。

```
1 @book{刘海洋,
2   title={LATEX入门},
3   author={刘海洋},
4   publisher={电子工业出版社},
5   year={2013},
6 }
```

参考文献使用范例，单独列出^{[1][2]}，一起列出^[1,2]

范例中使用参考文献标识引用参考文献，具体实现如下。

```
1 单独列出 \cite{刘海洋}\cite{胡伟}
2 一起列出 \cite{刘海洋,胡伟}
```

³文中采用的是相对路径，即数据库为我编译的 `tex` 文件的同一目录下的 `Zousiyu.bib` 文件

⁴一般写在 `\end{document}` 之前

5.5 链接

这部分内容主要用 *hyperref* 宏包来实现。

5.6 引用功能

在论文写作中，章节、插图、表格、公式和文本经常要前后调整或增添删减，这些引用的位置难以一次确定，所以不能进行直接编号。L^AT_EX 提供很智能的方法来解决这个问题，你不用担心引用的编号问题，只管引用就好了，L^AT_EX 系统会帮你编号。

在你的导言区添加如下代码，重新定义自动引用的名字。

```

1 \AtBeginDocument{%
2   \def\figureautorefname{图}
3   \def\tableautorefname{表}
4   \def\partautorefname{卷}
5   \def\appendixautorefname{附录}
6   \def\equationautorefname{式}
7   \def\Itemautorefname{列表}
8   \def\chapterautorefname{章}
9   \def\sectionautorefname{节}
10  \def\subsectionautorefname{小节}
11  \def\subsubsectionautorefname{条目}
12  \def\paragraphautorefname{自然段}
13  \def\Hfootnoteautorefname{脚注}
14  \def\AMSautorefname{式}
15  \def\theoremautorefname{定理}
16  \def\pageautorefname{页}
17 }
```

我们可以使用命令引用一个表格、公式、图片等。如使用如下命令分别引用一张表和一个带编号的公式。引用结果：如节 5.6 中式 5.1，表 5.1，图 5.1 所示。

占位

占位

占位

占位

```

1 \ref{tools-equation}
2 \ref{tools-tabular}
```

$$\int \operatorname{arccsc} x \, dx = x \operatorname{arccsc} x + \ln(x + \sqrt{x^2 - 1}) + C \quad (5.1)$$

表 5.1 T_EX 家族标识符

T _E X 家族标识符	
L ^A T _E X	L ^A T _E X 2 _ε
T _E X	X _Ǝ L ^A T _E X

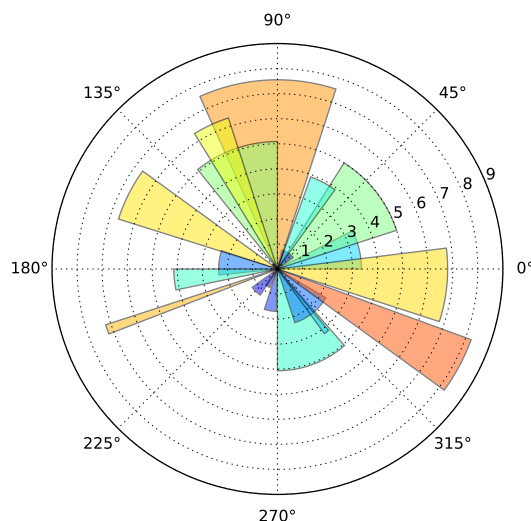


图 5.1 Demo of bar plot on a polar axis

5.7 列表

5.7.1 常规列表

常规列表环境。

```
\begin{itemize}
\item[记号] 条目1
\item[-] description1
\item[*] description2
\end{itemize}
```

```
记号 条目 1
- description1
* description2
```

常规列表的条目之间间距较大，可以使用长度赋值命令将条目环境额外的垂直空白设置为 0pt，达到与正文间距一致。

```
1 \itemsep=0pt
2 \parskip=0pt
```

5.7.2 排序列表

排序列表的基本形式。

```
\begin{enumerate}
\item 条目1
\item 条目2
\item 条目3
\end{enumerate}
```

```
1. 条目 1
2. 条目 2
3. 条目 3
```

排序列表可以嵌套，各层条目序号不一，我们可对其序号、标号和前缀进行重新定义，以生成所需要的条目样式。但是重新定义命令使用起来麻烦，排序列表默认命令也复杂，不便记忆，更不利于重定义。

为了方便，我们直接使用 *paralist* 宏包，我们只需要将标号样式填入方括号中，即可对标号进行重定义。此法在后面说明。

5.7.3 解说列表

示例如下，该类型列表用于对专业术语进行解释。具体设置不做详细说明，因为使用不便，后面有更好的宏包可以对以上所提到的三类列表进行更简便地进行设置。

```
\begin{description}
\item[APLL]
Automatic Phase-Locked Loop
自动锁相环
\item[GPS] Global
Positioning System 全球定位系统
\item[SPACETRACK]
Space Tracking 空间跟踪
\end{description}
```

APLL Automatic Phase-Locked Loop 自动锁相环

GPS Global Positioning System 全球定位系统

SPACETRACK Space Tracking 空间跟踪

5.7.4 列表宏包

最常用的列表宏包是 *paralist*，使用该宏包生成列表，条目与条目之间默认没有额外间距，不需要额外调整。此外，该宏包还提供行内列表，类似于行内公式，非常方便。

5.7.4.1 三种常规列表

paralist 提供 *compactitem*, *asparaitem*, *inparaitem* 三种常规列表环境。前两种无额外行距，*compactitem* 的列表换行也缩进，*asparaitem* 换行后无缩进。其中 *compactitem* 环境可用命令 `\pltopsep=12pt` 来附加上下额外行距。

在此，我们只看行内列表 *inparaitem* 一个例子。

```
\begin{inparaitem}[\S]
\item 一种以脉冲力相互作用的过程。
\item 两个质点交换它们动量和能量的持续过程。
\end{inparaitem}
我们倾向于第二种说法。
```

对于碰撞的物理定义已有许多种，例如：
 § 一种以脉冲力相互作用的过程。
 § 两个质点交换它们动量和能量的持续过程。
 我们倾向于第二种说法。

5.7.4.2 三种排序列表

该宏包同样提供三种排序列表环境，*compactenum*, *asparaenum*, *inparaenum*。*compactenum* 无首行缩进但有换行缩进，*asparaenum* 有首行缩进但换行不缩进。

对于碰撞的物理定义已有许多种，例如：

```
\begin{compactenum}[(1)]
\item 一种以脉冲力相互作用的过程。
\item 两个质点交换它们动量和能量的持续过程。
\end{compactenum}
```

我们倾向于第二种说法。

对于碰撞的物理定义已有许多种，例如：

- (1) 一种以脉冲力相互作用的过程。
- (2) 两个质点交换它们动量和能量的持续过程。

我们倾向于第二种说法。

对于碰撞的物理定义已有许多种，例如：

```
\begin{asparaenum}[(1)]
\item 一种以脉冲力相互作用的过程。
\item 两个质点交换它们动量和能量的持续过程。
\end{asparaenum}
```

我们倾向于第二种说法。

对于碰撞的物理定义已有许多种，例如：

- (1) 一种以脉冲力相互作用的过程。
- (2) 两个质点交换它们动量和能量的持续过程。

我们倾向于第二种说法。

宏包提供了一个参数位置，可填入序号和命令来控制排序列表样式。数字必须是 A, a, I, i, l 这几种，否则无法自动排序。如有字符串需要正常输出，可用花括号括起来，如 $\{ \text{例} \}$ 1。

对于碰撞的物理定义已有许多种，例如：

```
\begin{compactenum}[\text{定义} \ \itshape (1)]
\item 一种以脉冲力相互作用的过程。
\item 两个质点交换它们动量和能量的持续过程。
\end{compactenum}
```

我们倾向于第二种说法。

对于碰撞的物理定义已有许多种，例如：

- 定义 (1) 一种以脉冲力相互作用的过程。
- 定义 (2) 两个质点交换它们动量和能量的持续过程。

我们倾向于第二种说法。

5.7.4.3 三种解说列表环境

提供 *compactdesc*, *asparadesc*, *inparadesc* 三种解说列表环境，差异见 L^AT_EX 2_ε 完全学习手册列表宏包 *paralist* 一节。

5.7.4.4 其他特点

(1) 调用 *paralist* 宏包后，系统自带的列表环境也可以使用可选参数来修改条目标志和样式。

(2) 支持交叉引用。

(3) 常规列表和排序列表可以相互嵌套。

调用 $\backslash\text{emph}\{paralist\}$ 宏包后，系统自带的列表环境的标记更改示例。

```
\begin{enumerate}[(1)]
\itemsep=0pt
\parskip=0pt
\item 条目1
\item 条目2
\item 条目3
\end{enumerate}
```

调用 *paralist* 宏包后，系统自带的列表环境的标记更改示例。

- (1) 条目 1
- (2) 条目 2
- (3) 条目 3

5.8 附录

5.9 代码环境

首先载入 *listings* 宏包，定义基础代码环境，我取名为 *CodeBase*，这个基础代码环境定义的样式能被后续的代码环境调用，免去重复设置。也正是因为基础代码环境的通用性，所以这里只适合定义在所有代码环境中都适用的样式，如字体、各种边距、换行和标识等。

```

1 \lstdefinestyle{CodeBase}
2 {
3   basicstyle=\small\ttfamily,
4   frame=l,
5   aboveskip=0pt,%上边距
6   belowskip=0pt,%下边距
7   lineskip=0pt,
8   tabsize=4,%设置tab空格数
9   showtabs=false,%Tab
10  showspace=false,%空格标识
11  showstringspaces=false,
12  numbers=left,
13  numbersep=5pt,%行号与代码距离
14  numberstyle=\small\ttfamily,
15  rulecolor=\color{cyan},
16  boxpos=c,
17  xleftmargin=1em,%左边距
18  xrightmargin=0pt,
19  breaklines=true,%自动换行
20  breakindent=0pt,%换行后缩进为0
21  extendedchars=false,%解决代码跨页时，章节标题，页眉等汉字不显示的问题
22  framesep=3pt,
23  rulesep=2pt,
24  framerule=1pt,
25  %代码颜色设置
26  backgroundcolor=\color{gray!5},
27  stringstyle=\color{green!40!black!100},
28  keywordstyle=\bfseries\color{RGB}{0,0,255},
29  commentstyle=\slshape\color{black!60},
30 }
```

接下来，我们就可以用这个基本样式来定义一个专用于 L^AT_EX 代码书写的样式和相应的环境。

```

1 %LaTeX代码环境用
2 \lstdefinestyle{LaTeX}
3 {
4   style=CodeBase,
5   language=[LaTeX]TeX,
6   classoffset=0,
7   morekeywords={addplot, begin, end},
8 }
```

```

9
10 %定义 latex 代码专用环境
11 \lstnewenvironment{latex}[1]{\lstset{style=LaTeX}}{}

```

最后，直接在正文中使用新定义的环境 *latex* 框住所需要展示的代码即可。

上面定义了一个 L^AT_EX 专用的代码环境，实际使用肯定不只 L^AT_EX 代码需要展示，还有诸如 *Python*, *MATLAB* 等大量其他代码需要展示。这里我们在定义一个用于展示 *MATLAB* 代码的环境，同样也是从基础样式 *CodeBase* 进行衍生，只需要几条简单的命令即可。

```

1 %matlab 代码展示
2 \lstdefinestyle{Matlab}{
3     style=CodeBase,
4     language=Matlab
5 }
6
7 %定义 Matlab 代码专用环境
8 \lstnewenvironment{Matlab}[1]{\lstset{style=Matlab}}{}

```

MATLAB 代码高亮测试。

```

1 t=0:pi/10:2*pi;
2 [X,Y,Z]=cylinder(4*cos(t));
3 subplot(1,2,1);mesh(X);title('X');
4 subplot(1,2,2);mesh(Y);title('Y');

```

从 *CodeBase* 定义的新样式 X，其设置可以覆盖 *CodeBase* 中的设置，如下面这段 *Python* 代码高亮测试中，我们在代码中定义了一句 `keywordstyle=\slshape\color[RGB]{0,0,255}`，让 *Python* 代码中的关键词变为斜体，其他代码环境不受影响。

```

1 \lstdefinestyle{python}{
2     style=CodeBase,
3     keywordstyle=\slshape\color[RGB]{0,0,255},%就是这句
4     language=Python,
5     morekeywords={def},
6 }
7 \lstnewenvironment{python}[1]{\lstset{style=python}}{}

```

Python 代码展示。

```

1 def ffmpeg_concat_av(files, output, ext):
2     print('Merging video parts...', end='', flush=True)
3     params = [FFMPEG] + LOGLEVEL
4     for file in files:
5         if os.path.isfile(file): params.extend(['-i', file])
6     params.extend(['-c:v', 'copy'])
7     if ext == 'mp4':
8         params.extend(['-c:a', 'aac'])
9     elif ext == 'webm':
10        params.extend(['-c:a', 'vorbis'])
11    params.extend(['-strict', 'experimental'])
12    params.append(output)
13    return subprocess.call(params)

```

listings 宏识别的代码关键词肯定是有限的，但好在它提供一个参数可以扩充关键词。比如我们为 *c++* 语言添加更多的关键词，只需要在设置里面写下如下代码。关键词想要多少都

行，依据实际情况补充。

```
1 \lstset{
2   morekeywords={alignas,continue,friend,register,true,alignof,decltype,
   goto,reinterpret_cast,try,asm,default,if,return,typedef,auto,delete,inline
   ,short,typeid,bool,do,int,signed,typename,break,double,long,sizeof,union,
   case,dynamic_cast,mutable,static,unsigned,catch,else,namespace,static_
   assert,using,char,enum,new,static_cast,virtual,char16_t,char32_t,explicit,
   noexcept,struct,void,export,nullptr,switch,volatile,class,extern,operator
   ,template,wchar_t,const,false,private,this,while,constexpr,float,
   protected,thread_local,const_cast,for,public,throw,std}
3 },
```

第六章 数学排版

终于到了 \LaTeX 最擅长的部分，数学排版。

6.1 数学模式

分行内公式和行间公式。

行内公式，即： $\sum_{i=1}^n a_i$ ，得到： $\sum_{i=1}^n a_i$ 。

行间公式，即：
$$\sum_{i=1}^n a_i$$
，得到：

$$\sum_{i=1}^n a_i$$

6.2 数学宏包

6.3 数学符号

6.4 公式环境

6.5 矩阵环境

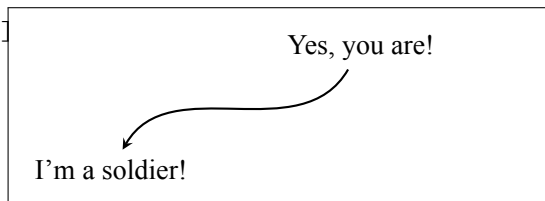
6.6 定理环境

第七章 Tikz

Tikz 包非常强大且异常复杂，单独列一章出来进行学习。

7.1 带箭头的注释效果

```
\usetikzlibrary{positioning}
\tikzset{>=stealth}
\begin{tikzpicture}[node distance=1.5cm]
\node (test){I'm a soldier!};
\node (testDesc)[above right=of test]
{Yes, you are!};
\draw [->,thick] (testDesc)
to[in=60,out=-120] (test);
\end{tikzpicture}
```



在这里,我们引入了 `tikz` 宏包,以及它的 `positioning` 库,用来绘制和定位 `nodes`。在 `tikzpicture` 中,我们建立了两个 `node`: `test` 和 `testDesc`, 后者的位于前者的右上方。最后,我们用绘制了从 `testDesc` 到 `test` 的曲线箭头, 其样式由之前的 `\tikzset{>=stealth}` 指定。

第八章 宏包

\LaTeX 的强大之处在于有各种各样的宏包进行扩展, 这些宏包能帮助 \LaTeX 完成多样的排版任务。

例如, 在文本中排版化学式, 有两个宏包比较流行, *mhchem* 和 *chemfig*。前者功能简单, 用法也简单, 书写无机化学式首选; 后者功能众多, 用法也较为繁复, 主要用来书写复杂的有机化学式。

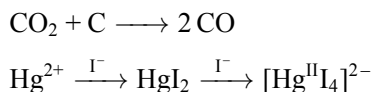
8.1 mhchem

使用下面的语句调用 *mhchem* 宏包, 添加 `version=4` 参数是为了使用宏包的一些新特性, 编写的文档如果要照顾老旧的 \LaTeX 版本用户, 可以酌情降低版本。

```
1 \usepackage[version=4]{mhchem}
```

化学方程式

```
\ce{CO2 + C -> 2 CO}\par
\ce{Hg^{2+} ->[I-] HgI2}
->[I-] [Hg^{II}I4]^{2-}
```



分子式可用在文本模式、数学模式和标题里面。

化学分子式

```
\ce{H2O}\par
\ce{Sb2O3}
```



离子

```
\ce{H+}\par
\ce{CrO4^2-}\par
\ce{[AgCl2]-}\par
\ce{Y^99+}\par
\ce{Y^{99+}}
```

H^+
 CrO_4^{2-}
 $[\text{AgCl}_2]^-$
 Y^{99+}
 Y^{99+}

氧化价态

```
\ce{Fe^{II}Fe^{III}2O4}
```

$\text{Fe}^{\text{II}}\text{Fe}^{\text{III}}_2\text{O}_4$

化学计量数

```
\ce{2H2O}\par
\ce{2 H2O}\par
\ce{0.5H2O}\par
\ce{1/2H2O}\par
\ce{(1/2)H2O}\par
\ce{$n$H2O}
```

$2\text{H}_2\text{O}$
 $2\text{H}_2\text{O}$
 $0.5\text{H}_2\text{O}$
 $\frac{1}{2}\text{H}_2\text{O}$
 $(1/2)\text{H}_2\text{O}$
 $n\text{H}_2\text{O}$

同位素

```
\ce{^{227}_{90}Th+}\par
\ce{^{227}_{90}Th+}\par
\ce{^{0}_{-1}n^{+}}\par
\ce{^{0}_{-1}n-}
```

$^{227}_{90}\text{Th}^+$
 $^{227}_{90}\text{Th}^+$
 $^{0}_{-1}\text{n}^-$
 $^{0}_{-1}\text{n}^-$

$()$, $[]$ 可以正常表示, 但需要使用 `\{ \}` 来表示花括号。大型的花括号只能在数学环境里面使用。

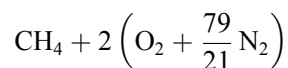
括弧, 方括号, 花括号

```
\ce{(NH4)2S}\par
\ce{[\{(X2)3\}2]^3+}
```

$(\text{NH}_4)_2\text{S}$
 $[\{(X_2)_3\}_2]^{3+}$

```
1| \[\ce{CH4 + 2 $\left( \ce{O2 + 79/21 N2} \right)$}\]
```

大花括号



聚合态

```
\ce{H2(aq)}\par
\ce{CO3^2-_{(aq)}}\par
\ce{NaOH(aq,$\infty$)}
```

$\text{H}_2(\text{aq})$
 $\text{CO}_3^{2-}(\text{aq})$
 $\text{NaOH}(\text{aq}, \infty)$

未成对电子,
自由基

```
\ce{OCO^{.-}}\par
\ce{NO^{(2.)-}}
```

OCO^{•-}
NO^{(2•)-}

排版约定, 变量使用斜体排版, 而其他元素 (如化学式) 则使用直立字体排版。

变量

```
$\ce{NO_x}$\par
$\ce{Fe^{n+}}$
```

NO_x
Feⁿ⁺

```
1 | $\ce{x Na(NH4)HPO4 ->[\Delta] (NaPO3)_x + x NH3 ^ + x H2O}$
```



希腊字符

```
\ce{\mu-Cl}\par
\ce{[Pt(\eta^2-C2H4)Cl3]-}
```

μ-Cl
[Pt(η²-C₂H₄)Cl₃]⁻

加成化合物

```
\ce{KCr(SO4)2*12H2O}\par
\ce{KCr(SO4)2.12H2O}\par
\ce{KCr(SO4)2 * 12 H2O}
```

KCr(SO₄)₂ · 12 H₂O
KCr(SO₄)₂ · 12 H₂O
KCr(SO₄)₂ · 12 H₂O

化学键

```
\ce{C6H5-CHO}\par
\ce{A-B=C#D}\par
\sffamily\bfseries\ce{A-B=C#D}
```

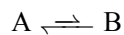
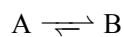
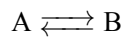
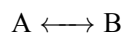
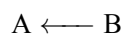
C₆H₅-CHO
A-B=C≡D
A-B=C≡D

```
\ce{A\bond{-}B\bond{=}C\bond{#}D}\par
\ce{A\bond{1}B\bond{2}C\bond{3}D}\par
\ce{A\bond{~}B\bond{~~}C}\par
\ce{A\bond{---}B\bond{~=}C\bond{---}D}\par
\ce{A\bond{...}B\bond{...}C}\par
\ce{A\bond{->}B\bond{<-}C}
```

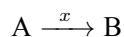
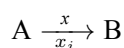
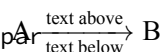
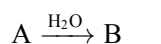
A-B=C≡D
A-B=C≡D
A...B=C
A≡B≡C≡D
A...B...C
A→B←C

反应箭头

```
\ce{A -> B}\par
\ce{A <- B}\par
\ce{A <-> B}\par
\ce{A <--> B}\par
\ce{A <=>> B}\par
\ce{A <<=> B}
```



```
\ce{A ->[H2O] B}\par
\ce{A ->[{\text above}][{\text below}] B}\par
\ce{A ->[$x$][$x_i$] B}\par
\ce{A ->[$x$] B}
```

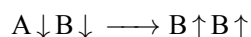
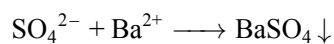


带参数的反应箭头

```
\ce{A + B}\par
\ce{A - B}\par
\ce{A = B}\par
\ce{A \pm B}
```

化学方程式
计算符

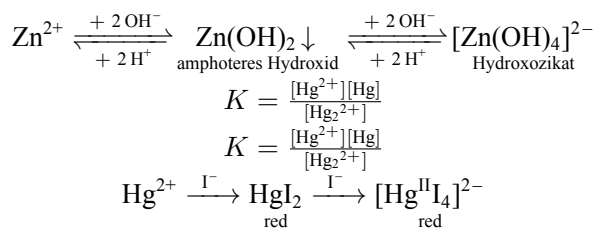
```
\ce{SO4^2- + Ba^2+ -> BaSO4 v}\par
\ce{A v B (v) -> B ^ B (^)}
```



沉淀和气体

```
1 \ce{Zn^2+}
2   <=>[+ 2OH-][+ 2H+]
3   $\underset{\text{amphoterer Hydroxid}}{\ce{Zn(OH)2 v}}$
4   <=>[+ 2OH-][+ 2H+]
5   $\underset{\text{Hydroxozikat}}{\ce{[Zn(OH)4]^2-}}$
6 }
7
8 $K = \frac{\ce{Hg^2+}}{\ce{Hg}} \frac{\ce{Hg^2+}}{\ce{Hg^2+}}$
9
10 $K = \ce{\frac{[Hg^2+][Hg]}{[Hg^2+]}}$
11
12 \ce{Hg^2+ ->[I-]}
13   $\underset{\text{red}}{\ce{HgI2}}$
14   ->[I-]
15   $\underset{\text{red}}{\ce{[Hg^II]I4^2-}}$
16 }
```

极好的示例



8.2 Chemfig

8.3 C_TE_Xzhnumber

以中文格式输出数字。这里的数字可以是整数、小数和分数。

```
\zhnumber{2012020120} \\
\zhnumber{2 012 020 120} \\
\zhnumber{2,012,020,120} \\
\zhnumber{2012.020120} \\
\zhnumber{2012.} \\
\zhnumber{.2012} \\
\zhnumber{20120/20120} \\
\zhnumber{/2012} \\
\zhnumber{2012/} \\
\zhnumber{201;2020/120}
```

二十亿零一千二百零二万零一百二十
二十亿零一千二百零二万零一百二十
二十亿零一千二百零二万零一百二十
二千零一十二点零二零一二零
二千零一十二点零
零点二零一二
二万零一百二十分之二万零一百二十
二千零一十二分之零
零分之二千零一十二
二百零一又一百二十分之二千零二十

将阿拉伯数字转换为中文字符串。

```
\zhdigits{2012020120} \\
\zhdigits*{2012020120}
```

二〇一二〇二〇一二〇
二零一二零二零一二零

带 * 才能映
射 0 为零

将 L^AT_EX 计数器数值转换为中文。

```
1 \zhnum{section}
```

输出当天的星期。

```
\zhweekday{2012/5/20}
```

星期日

以中文格式输出日期。

```
\zhdate{2012/5/21} \\
\zhdate*{1995/01/26}
```

二〇一二年五月二十一日
一九九五年一月二十六日星期四

带 * 的可以
输出星期

以中文输出当天日期。

```
\zhtoday
```

二〇一六年十月二十九日

以中文输出时间。

```
\zhtime{23:56}
```

二十三时五十六分

输出当前时间。

```
\zhcurrtime
```

十三时二十分

输出天干计数，数字范围是 1–10。

```
\zhtiangan{1} \zhtiangan{2}
\zhtiangan{3} \zhtiangan{4}
\zhtiangan{5} \zhtiangan{10}
```

甲 乙 丙 丁 戊 癸

输出地支计数，数字范围 1–12。

```
\zhdizhi{1} \zhdizhi{2} \zhdizhi{3}
\zhdizhi{4} \zhdizhi{5} \zhdizhi{12}
```

子 丑 寅 卯 辰 亥

输出干支计数，数字范围 1–60。

```
\zhganzhi{1} \zhganzhi{2}
\zhganzhi{3} \zhganzhi{4}
\zhganzhi{5} \zhganzhi{60}
```

甲子 乙丑 丙寅
丁卯 戊辰 癸亥

输出公元纪年对应的干支纪年，公元前用负数。

```
\zhganzhinian{1898}
\zhganzhinian{-246} \
\zhganzhinian{-2697}
\zhganzhinian{\year}
```

戊戌 乙卯
甲子 丙申

zhnumsetup 的样式控制选项。

Simplified 以简体中文输出数字（对 Big5 编码无效）

Traditional 以繁体中文输出数字（对 Big5 编码无效）

Normal 以小写形式输出中文数字

Financial 以大写形式输出中文数字

Ancient 以廿输出 20，以卅输出 30，以卌输出 40，以𠫪输出 200

```
\zhnumsetup{
style={Traditional,Normal}}
\zhnumber{62012.3} \
\zhnumsetup{style=Ancient}
\zhnumber{21}
```

六萬二千零一十二點三
廿一

```
\zhnumsetup{
style={Traditional,Financial}}
\zhnumber{62012.3} \
\zhnumsetup{style=Ancient}
\zhnumber{21}
```

陸萬貳仟零壹拾貳點叁
廿一

Bibliography

- [1] 刘海洋. LATEX 入门[M]. [S.l.]: 电子工业出版社, 2013.
- [2] 胡伟. LATEX2e 完全学习手册 (第 2 版)[M]. [S.l.]: 清华大学出版社, 2013.