

邹思宇的 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ 学习笔记

Power by $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ Texstudio

作者： 邹思宇

地点： 苏州大学独墅湖校区

版本： 二〇一六年九月二十七日，Nightly Build

Copyright © 2016 Zou.siyu.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

目 录

第一章 \LaTeX 基础	1
1.1 命令	1
1.1.1 命令形式	1
第二章 绘图	1
2.1 函数图	1
2.1.1 二维函数图	1
2.1.2 三维函数图	2
2.1.3 隐函数图像	3
2.2 线图	3
2.3 条形图	3
2.4 从外部数据作图	3
第三章 版面和格式	5
3.1 页眉页脚	5
3.2 代码展示	5
3.3 字体设置	6
3.3.1 全局字体设置	6
3.3.2 局部字体设置	6
3.3.3 在数学环境中使用中文	6
3.3.4 汉字“斜体”	6
3.4 颜色	6
3.5 文本格式	7
3.5.1 下画线	7
第四章 图和表	8
4.1 插图	8
4.2 <code>array</code> 宏包	8
4.3 浮动体	8
4.4 表格	9
4.4.1 跨行和跨列表格	10
4.4.2 彩色表格	11
4.4.3 斜线表头	12
4.4.4 表格标题	13
第五章 正文工具	14
5.1 目录	14
5.2 脚注	14

5.3 尾注	14
5.4 边注	15
5.5 参考文献	15
5.6 链接	15
5.7 引用功能	15
5.8 列表	15
5.8.1 常规列表	15
5.8.2 排序列表	16
5.8.3 解说列表	16
5.8.4 列表宏包 <code>paralist</code>	16
5.8.4.1 三种常规列表	17
5.8.4.2 三种排序列表	17
5.8.4.3 三种解说列表环境	18
5.8.4.4 其他特点	18
5.9 附录	18
第六章 数学排版	18
6.1 数学模式	18
6.2 数学宏包	19
6.3 数学符号	19
6.4 公式环境	19
6.5 矩阵环境	19
6.6 定理环境	19
第七章 <code>Tikz</code>	19
7.1 带箭头的注释效果	19

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。

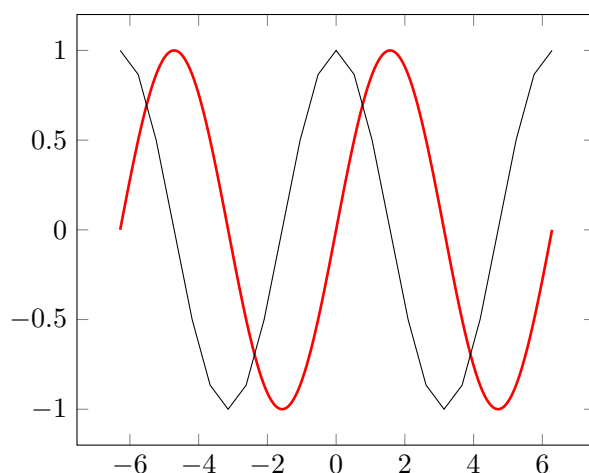
第二章 绘图

2.1 函数图

2.1.1 二维函数图

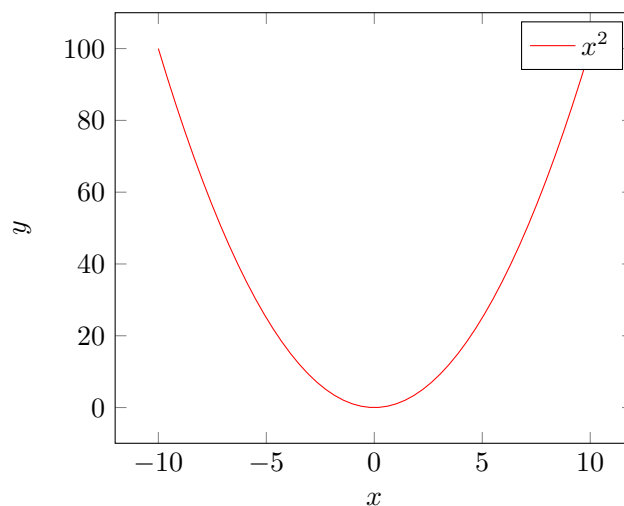
绘制函数 $f(x) = \sin(x)$ 和 $g(x) = \cos(x)$ 在定义域 $[-2\pi, 2\pi]$ 上的图像。对于函数 $f(x) = \sin(x)$, 我们使用 `red` 和 `/tikz/line width={1pt}` 来使其更显眼。L^AT_EX 默认取点少, 我们使用 `samples=x` 来控制取点数, 使函数图像更为平滑。

```
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。

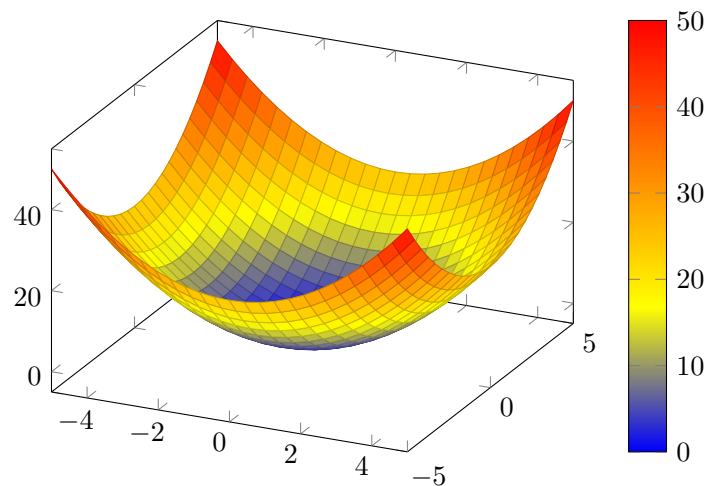
```
1 \addplot[domain=-10:10,red,samples=50]{x^2}
```



2.1.2 三维函数图

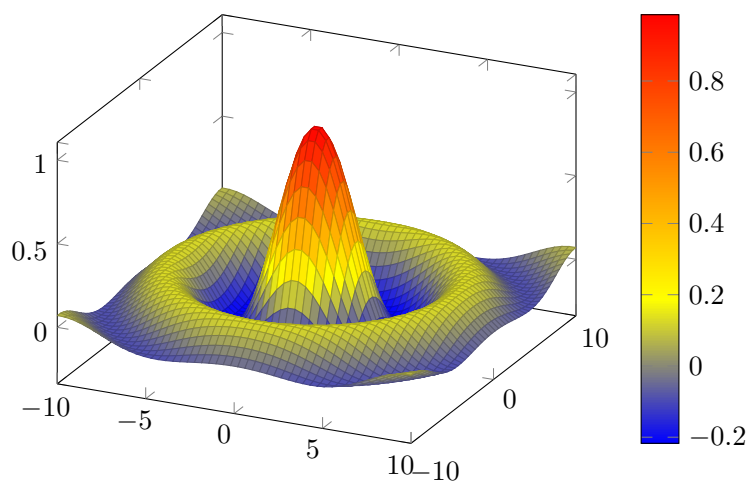
绘制函数 $f(x, y) = x^2 + y^2$ 的三维函数图像。其中参数 *colorbar* 可以为函数图添加一个彩色条。

```
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。

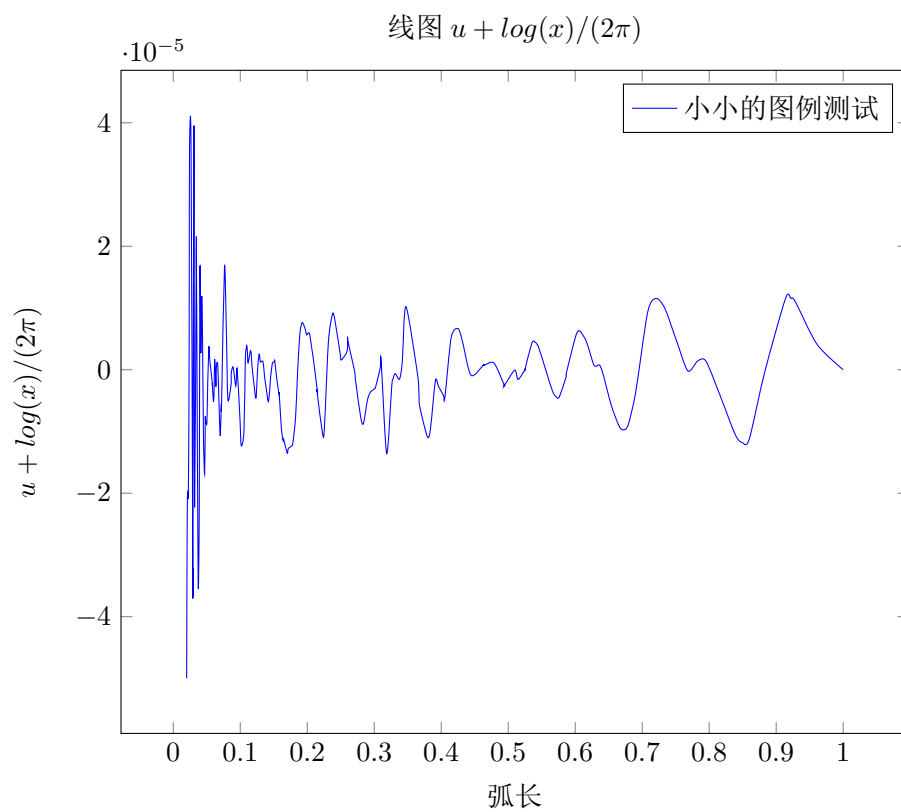
```
1 \item grid=major      %显示网格线
2 \item mark=x          %设置数据点标记类型
```

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 L^AT_EX 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[RO,LE]{} %E, O, 左、右页
9 \fancyfoot[LE,RO]{\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 代码展示

我们需要使用一个宏包——*lstlisting* 来达到这个目的。记住，不要再 *lstlisting* 环境的第一行代码前面使用 Tab 或者空格，因为会无条件还原，导致代码不能贴合框线，丑陋。

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');
```


3.3 字体设置

分全局和局部字体设置。

3.3.1 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} %仅对西文起作用
2 \setCJKmainfont{SimSun} %仅对中文起作用
```

3.3.2 局部字体设置

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

3.3.3 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。

```
$f_{\text{高温}}$
```

 $f_{\text{高温}}$

3.3.4 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。

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 *Italy* 字系；而斜体，是指 *Slant* 字族。所以 `\textit{}` 是加斜字体，而 `\textsl{}` 或者 `\slshape` 才是真正的斜体。

Italy Slant *Italy Slant*

3.4 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}
```

3.5 文本格式

3.5.1 下画线

LaTeX 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}}
```

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（到处乱跑的盒子）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，一面造成意想不到的问题。

```

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|}{表格线宽度修改}\\
\hline
7&5&3\\
\hline
6&1&8\\
\hline
\end{tabular}

```

表格线宽度修改		
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}|IC{6mm}|IC{6mm}|}
\hline
\multicolumn{3}{|c|}{垂直线粗细更改}\\
\hline
7&5&3\\
\hline
6&1&8\\
\hline
\end{tabular}

```

垂直线粗细更改		
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|}{全局表格线粗细更改}\\
\specialrule{2pt}{0pt}{0pt}
7&5&3\\
\hline
6&1&8\\
\hline
\end{tabular}
```

全局表格线粗细更改		
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 \\
\cline{2-3} & 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 \diagdown 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


```

\begin{center}
\tabcaption{一张课表}
\begin{tabular}{|C{1cm}|C{1cm}|C{1cm}|}
→ \hline
→ \multirow{2}{2}*{时间} & \multicolumn{2}{c|}{星期}\\
→ \cline{2-3} & 一 & 二 \\
→ \hline
→ 8:30 & 化学 & 物理\\
→ 9:30 & 韩语 & 数学\\
→ \hline
\end{tabular}
\end{center}

```

图 4.1 一张图片

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 尾注

LaTeX 没有尾注功能，但是可以调用 *endnotes* 宏包来生成尾注。用下面这段话作为尾注例子。

南北朝时著名数学家祖冲之¹发明了一种计算方法，从而得到了更为准确的圆周率²值。

本章注释

¹公元 429-500

²圆周长与直径之比

5.4 边注

LaTeX 本身提供边注命令：

¹邹思宇，男，LaTeX 爱好者

²脚注命令 `\footnote`

```
1 \marginpar[左边注]{右边注}
```

边注测试。

这是边注啊

5.5 参考文献

5.6 链接

这部分内容主要用 *hyperref* 宏包来实现。

5.7 引用功能

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Equation 5.1和??所示。

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

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

(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 _Y L ^A T _E X

5.8 列表

5.8.1 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
```

5.8.2 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。

5.8.3 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

5.8.4 列表宏包 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。

5.8.4.1 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。

对于碰撞的物理定义已有许多种，例如：

```
\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。

5.8.4.2 三种排序列表

该宏包同样提高三种排序列表环境，*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 `{例 1}`。

对于碰撞的物理定义已有许多种，例如：

```
\begin{compactenum}[{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。

5.8.4.3 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。

5.8.4.4 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`\latexline{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

5.9 附录

第六章 数学排版

终于到了 L^AT_EX 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

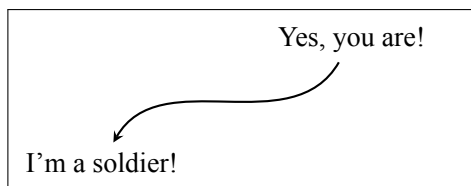
6.6 定理环境

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。