

week 4: latex

Mayrain

vscode*

2023 年 11 月 27 日

1 What is latex

latex 的前身是 tex，是一种排版软件，用于生成高质量的文档，比如科技论文，书籍等。latex 是 tex 的一种宏包，是一种对 tex 的封装，使得 tex 更加容易使用。

latex 的优点是可以生成高质量的文档，而且可以使用代码的方式来排版文档，可以很方便的生成数学公式，表格，图片等。

latex 的缺点是学习曲线比较陡峭，而且不适合用来写小文档，比如笔记等。我用 tex 做笔记的原因主要是为了熟练掌握其技巧。

以上都是 copilot 写的，我只是复制粘贴了一下（笑）。

一般来说发行版就是打包好的套装，包含了 latex 引擎，宏包，字体等。

也可以直接安装引擎，比如 xetex，pdftex。他们起到编译器的作用。编写的方法是随意的，任何一个文本编辑器都可以完成这个事情。

2 latex command

latex 的命令以 \ 开头。而且对大小写敏感

这里的 \ 最好是用 verb 命令，而不用 text 或者 \textbackslash。中者无法实现（命令在大括号中也有效），后者太麻烦了。

latex 可以用 {} 限定作用范围，而且也可以用 [] 表达可选参数。在 {} 中的表达是必选的。

环境就是一种命令！

3 latex space

有关 latex 中的空格/段落，需要注意：

- 1 个或多个空格，latex 会当做一个空格处理
- 段首空格不处理。我们必须要用 \hspace{2em} 这样的命令才能使其空格。

*mayrain@gmail.com

- 换行符视为一个空格，也就是说只换行，`latex` 不会换行，必须要再换一行以空出一行。（连续两个换行符，`latex` 会认为是一个段落的结束，会自动空出一行。）
- `\par`或者是空行，`latex` 会认为是一个段落的结束，会自动空出一行。
- `\\ \newline`则是所谓断行，相当于段落内换行，不产生新段落。
- `\newpage \clearpage`则是手动断页，但是前者在双栏状态下左页换右页，还在同一页；后者则是直接该页都不要了。

4 latex strange symbol

4.1 quotation symbol

`latex` 的重要特点是“非二义性”。在 `word` 中，左右引号是自动识别的，但是对于 `latex` 中这样的识别仍然不够准确。如果直接键入单/双引号，`latex` 只会将其识别为右引号，而没有左引号。

左引号的出现应当使用反引号，也就是```，输入上：“We bands of brothers”和`'We bands of brothers'`才是正确的。

不过，双引号也可以使用两个单引号来作为右引号，例如“`We bands of brothers`”。

如果搞错了，情况就是这样：

`"We bands of brothers"`

`'We bands of brothers'`

危险的地方在于这甚至不会报错。

4.2 prevent connecting（防止连字）

在两个字母中间使用 `{}` 可以避免连字，例如：

`difficult`

`difficult`

下面的单词在两个 `f`，`f` 和 `i` 之间加入了 `{}` 用于规避连字，看起来美观多了。

4.3 mandatory distance

长度单位有 `pt`, `in`, `cm`, `mm`, `em`（当前 M 宽度）, `ex`（当前 x 高度）

几种修改：

1. 行距：`\linespread{factor}`。默认行距为 1.2 倍字体大小。

2. 水平间距：`\hspace{length}`。

`\hspace*{length}`可以规避当只有一边有内容时，距离会被自动吞掉的情况。

`\quad`代表 `1em`，两个 `q` 则是 `2em`

`\hspace{\fill}lol`这样的操作则会将 `lol` 放在本行最后，中间全部用空格填充。

3. 竖直间距：`\vspace`

也可以使用类似于`\\[1em]`这样的命令来换行时插入垂直间距。

4. 分栏：

全局分栏：`\twocolumn \onecolumn`很明显了

局部分栏：需要引入宏包 `multicol`，然后使用 `begin` 包含 `multicols` 环境，注释里可以看到。

5 docu structure

5.1 Chap. and index

+ `\chapter`(仅用于 book 和 report)

+ `\section`

+ `\subsection`

+ `\subsubsection`

- `\paragraph`

- `\subparagraph`

iii. 以上内容可以加入参数表示短标题，显示在目录和页脚中，代替原有内容。

iv. 还可以加入 * 表示不编号。

v. 利用`\tableofcontents`可以生成目录。，不过需要编译两次。

vi. 生成目录后，还可以通过`\addcontentsline{toc}{section}{name}`手动添加目录项。

vii. `\appendix`命令可以生成附录。

以上内容，前者是无序列表（`itemize`），后者是有序列表（`enumerate`）。最多嵌套 4 层。

无序列表的符号可以自定义，在 `item` 上增加参数即可。`enumerate` 也可以指定参数，不过一般都是 `i.` , `a.` , `a)` , `(1)` 云云

5.2 center, left and right

以下是命令：

- `\centering`
- `\raggedright`¹
- `\raggedleft`

以下是环境：

- `\begin{center}`
- `\begin{flushleft}`
- `\begin{flushright}`

对于环境来说，会在上下额外产生间距。命令则完全不会。

命令会对后方所有部分都产生影响，而不是只影响一个 `para.`

¹注意这里 `raggedright` 对应的是左对齐，而不是右对齐！非常反直觉。

5.3 code environment

代码块可直接上 verbatim 环境。

```
#include <stdio.h>

int main()
{
    printf("Hello World!\n");
    return 0;
}
```

行内公式则可用\verb

```
printf("Hello          World!\n");
```

（我受不了了，原来用了这么久的 verb 命令是用在代码上。。。）

利用\verb*则可以令其显示（是真正的“显示”！）空格。

```
printf("Hello          World!\n");
```

listing 宏包则可实现代码高亮，但是既然 latex 自己已经这么好用了那我们就不管了。

5.4 tabular

相关内容我放在另一个文档。

5.5 float block(浮动体)

浮动体使得图片和表格脱离文本，独立寻找合适的位置排放。

- 利用 figure 环境包裹图片
- 利用 table 环境包裹表格

一般来说，我们使用浮动体，是为了“让他们自由排版”。一些图片硬和文字一起排版将会使得版面非常难看，因此可以利用浮动体让他们自由排版。等到需要使用的时候我们就加 label 和 ref 命令来引用。

（注意，\pageref{ref}可以引用页码。）所以浮动体 +label+ref 是一套完整的流程!!!!

另外也可以加 caption 命令来添加标题。

（这个过程也叫做交叉引用。）

一个非常重要的提醒：

\ref一定要紧跟在\caption的后面，否则图片编号是图 1，但是 ref 引用时将会变成章节号（也就是不能成功引用的意思）

```
\begin{figure}[placement]
...
\end{figure}
```

placement 则是用 htbp 表示，分别代表 here,top,bottom,page。

使用! 忽略限制（限制：每页浮动体数量，占用页面比例，浮动体间距）

另外，对于浮动体来说，还可以使用`\listoffigures`和`\listoftables`来生成浮动体的目录。

Tip: `\label`可以记录的位置:

- 章节标题后面紧接着使用
- 行间公式中任意位置
- 有序列表的 item 后面（必须有序）
- 浮动体 caption 后面
- 定理环境内部

注意不记录编号的命令是不能使用的，包括 `section*` 这种手动去掉的。

5.6 math expression

行内公式:

一对 `$` 包裹即可。

也可以用`\(和\)`包裹

行间公式:

- 一对 `$$` 包裹
- `\[和\]` 包裹
- `equation` 环境。带编号。使用 `equation*` 可以不带编号，用`\notag`也可以。
- `align` 环境也可以。

Tip: 如果说上下公式有联系的话，注意中间不要留空！

公式中空格一定会被忽略！一些手动加空格的方法:

1. ab 使用,
2. $a\,b$ 使用:
3. $a\,b$ 使用;
4. $a\,b$ 使用空格
5. $a\quad b$ 使用 `quad`
6. $a\qquad b$ 使用 `qquad`
7. $a\!b$ 使用!

以上部分空格分别变大。但是最后一个负空格，也就是反而缩紧了。

另外，数学模式中，一般输入英文会显示斜体（被视为变量）。对于 `\sin` 等函数，命令输入可以让他们保持正体。

如果有需要输入的函数（但是函数库里没有），需要使用`\mathrm{text}`命令，也就是数学模式的正体。。。

数学模式中有两种显示方式：`\displaystyle`和`\textstyle`。一般来说前者适用于行间公式，后者适用于行内公式。前者的占用空间较大，后者的占用空间较小，另外在上下标的位置上面可能会有些区别。

如果想要在行内强行使用`\displaystyle`，则将行内公式写作`$\displaystyle \sum_{i=1}^{100}`即可。不过可能造成

行距混乱。

最后，不要滥用 `text`！而且数学模式中不能用 `textbf` 之类的方式搞字体！

一些常用的公式如下图 1 所示。注意诸如 `mod` 中涉及到的三横等号不要自己瞎写，一定要用公式生成才最规范！

图中的 `\left(` 和 `\right)` 代表自动匹配括号大小。当然也可以用 `\left(` 和 `\right.` 这样的方式放弃一边的括号，但是

常用数学符号

- 希腊字母: `\alpha` α `\beta` β ... `\Gamma` Γ `\Delta` Δ ...
- 无穷大 `\infty` ∞ ; 根式 `\sqrt{...}` $\sqrt{...}$ `\sqrt[n]{...}` $\sqrt[n]{...}$
- 一些省略号 `\dots` \dots `\cdots` \cdots `\vdots` \vdots `\ddots` \ddots
- 分式 `\frac{分子}{分母}` $\frac{分子}{分母}$ `\dfrac{分子}{分母}` $\dfrac{分子}{分母}$ `\tfrac{分子}{分母}` $\tfrac{分子}{分母}$
- `a\bmod b` $a \bmod b$; `x\equiv a\pmod{b}` $x \equiv a \pmod{b}$
- `\bar{}` $\bar{}$ `\vec{}` $\vec{}$ `\hat{}` $\hat{}$ `\overline{}` $\overline{}$ `\underline{}` $\underline{}$ `\widehat{}` $\widehat{}$ `\overrightarrow{}` $\overrightarrow{}$
- `\left(` $\left($ `\right)` $\right)$ 等自动匹配大小 `\left.` $\left.$ `\right.` $\right.$ 取消一侧
- `\bigl(` $\bigl($ `\Bigr)` $\Bigr)$ `\biggl(` $\biggl($ `\Biggr)` $\Biggr)$ 等手动大小
- ...
- 更多参考 `\lshort` 或符号大全
- LaTeX 数学公式大全 - `lowa_BattleShip` 也是一个不错的整合

图 1: frequently used math expression

绝对不能不匹配！（不能不写某一侧！）

对齐模式: `align/aligned` 对齐环境。可使用 `\notag` 或者 `\align*` 来取消编号。

`aligned` 本身不自带数学环境！因此需要嵌套进入 `equation` 之类的环境才能使用！

矩阵模式: 参照另一份学习文档。

5.7 standard of math expression

1. 特殊函数用特殊命令，或者是利用 `\mathrm` 命令写为正体。能有专门的命令甚至都不能用 `\mathrm`！变量以外，全部正体！（尤其是各类算子）
2. 正确使用 `\displaystyle` 以及 `\dfrac`（性质一致）
3. 注意区分 `\colon` 以及手打冒号。前者被用于映射（一个标点），后者常被用于集合，例如 $\{x : x = 1\}$
4. `left` 和 `right` 的括号要经常用，只要内容不为一个字符高时，就必须要用自适应括号！
5. 行间公式之间，如果有关联，不可直接加空行。