

# 标题页

Ryan\*      Fan†

2020年5月7日

## 摘要

一般用于紧跟`\maketitle`命令之后介绍文档的摘要  
中文 L<sup>A</sup>T<sub>E</sub>X 排版。

## 1 用 L<sup>A</sup>T<sub>E</sub>X 排版文字

分段

换行`\`

冒号 “please press the ‘x’ key.”

连字符-用来组成复合词,

短破折号--用来连接数字表示范围,

长破折号---用来连接单词

省略号...和...

波浪号~

强调文字,但是无法换行,`ulem`宏包解决了这一问题,它提供的`uline`命令能够轻松生成自动换行的下划线。`em`命令用来将文字变为斜体以示强调。如果在本身已经用`emph`命令强调的文字内部嵌套使用`emph`命令,内部则使用

在合适的位置插入一个不会断行的空格 Fig. 1, Ryan Fan

断行

可以带可选参数 (`length`), 用于在换行处向下增加垂直间距

或者 `newline` 命令, 不用带参数

---

\*注脚

†注脚

断页，在在双栏排版中只起到另起一栏的作用

断词 I think this is: supercalifragilisticexpialidocious. And I think this is:  
supercalifragilisticexpialidocious.

# 目录

<b>1 用 L<sup>A</sup>T<sub>E</sub>X 排版文字</b>	<b>1</b>
<b>2 目录和页眉页脚</b>	<b>5</b>
2.1 子章节	5
2.1.1 子子章节	5
标题不带编号	5
<b>第一部分 分块</b>	<b>5</b>
<b>3 交叉引用</b>	<b>5</b>
<b>4 脚注和边注</b>	<b>5</b>
<b>5 特殊环境</b>	<b>5</b>
5.1 列表	5
5.2 对齐环境	6
5.3 引用环境	7
5.4 代码环境	7
5.5 表格	7
5.5.1 列表格	7
5.5.2 横线	8
5.5.3 合并单元格	9
5.5.4 嵌套表格	9
5.5.5 行距控制	9
5.6 图片	10
5.7 盒子	11
5.7.1 水平盒子	11
5.7.2 带框的水平盒子	11
5.7.3 垂直盒子	11
5.7.4 标尺盒子	12
5.8 浮动体	12
5.8.1 浮动体的标题	13
5.8.2 并排和子图表	13
<b>6 数学公式</b>	<b>14</b>
6.1 公式排版基础	14
6.1.1 行内和行间公式	14
6.2 数学模式	15
6.3 数学符号	15
6.3.1 一般符号	15
6.3.2 指数、上下标和导数	17

6.3.3	分式和根式	17
6.3.4	关系符	18
6.3.5	算符	19
6.3.6	巨算符	20
6.3.7	数学重音和上下括号	21
6.3.8	箭头	22
6.3.9	括号和定界符	23
6.4	多行公式	24
6.4.1	长公式折行	24
6.4.2	多行公式	24
6.4.3	公用编号的多行公式	25
6.5	数组和矩阵	25
6.6	公式中的间距	26
6.7	数学符号的字体控制	27
6.7.1	数学字母字体	27
6.7.2	数学符号的尺寸	27
6.7.3	加粗的数学符号	28
6.8	定理环境	28
6.8.1	$\LaTeX$ 原始的定理环境	28
6.8.2	amsthm 宏包	28
6.8.3	证明环境和证毕符号	29
<b>7</b>	<b>排版样式设定</b>	<b>30</b>
7.1	字体和字号	30
7.1.1	字体样式	30
7.1.2	字号	31
7.1.3	选用字体宏包	32
7.1.4	字体编码	33
7.1.5	使用 fontspec 宏包更改字体 (xelatex)	33
7.1.6	使用 xeCJK 宏包更改中文字体	33
<b>A</b>	<b>附录</b>	<b>34</b>

## 2 章节

### 2.1 子章节

#### 2.1.1 子子章节

#### 段落

#### 子段落

## 标题不带编号

## 第一部分 分块

## 3 交叉引用

A reference to this subsection looks like: “see section 3 on page 5.”

## 4 脚注和边注

“天地玄黄, 宇宙洪荒。日月盈昃, 辰宿列张。”<sup>1</sup>

有些情况下 (比如在表格环境、各种盒子内) 使用 `footnote` 并不能正确生成脚注。我们以分两步进行, 先使用 `footnotemark` 为脚注计数, 再在合适的位置用 `footnotetext` 生成脚注。<sup>2</sup>

边注较窄, 不要写过多文字, 最好设置较小的字号。

## 5 特殊环境

### 5.1 列表

有序列表

1. An item.
  - (a) A nested item.
  - \* A starred item.
2. Reference(1a).

无序列表

---

<sup>1</sup>出自《千字文》。

<sup>2</sup>出自《千字文》。

- An item.
  - A nested item.
  - + A ‘plus’ item. + A ‘plus’ item.
  - Another item. –Another item.
- Go back to upper level.

关键字环境

**Enumerate** Numbered list.

**Itemize** Non-numbered list.

重定义无序列表的符号

- † First item
  - ‡ Subitem
  - ‡ Subitem
- † Second item

重定义有序列表的符号

- A> First item
- B> Second item

## 5.2 对齐环境

`center`、`flushleft` 和 `flushright` 环境分别用于生成居中、左对齐和右对齐的文本环境。

Centered text using a `center` environment.

Left-aligned text using a `flushleft` environment.

Right-aligned text using a `flushright` environment.

还可以用以下命令直接改变文字的对齐方式: `Centered text paragraph.`

Left-aligned text paragraph.

Right-aligned text paragraph.

`center` 等环境会在上下文产生一个额外间距, 而 `\centering` 等命令不产生, 只是改变对齐方式。

比如在浮动体环境 `table` 或 `figure` 内实现居中对齐, 用 `\centering` 命令即可, 没必要再用 `center` 环境。

### 5.3 引用环境

**quote** 用于引用较短的文字, 首行不缩进

Francis Bacon says:

Knowledge is power.

**quotation** 用于引用若干段文字, 首行缩进

《木兰诗》:

万里赴戎机, 关山度若飞。朔气传金柝, 寒光照铁衣。将军百战死, 壮士十年归。

归来见天子, 天子坐明堂。策勋十二转, 赏赐百千强。

.....

### 5.4 代码环境

```
#include <iostream>
int main()
{
    std::cout << "Hello, world!"
                << std::endl;

    return 0;
}
```

```
~~~~~for_(int_i=0;_i<4;_i++)
~~~~~printf("Number_%d\n",i);
~~~~~
```

要排版简短的代码或关键字`\verb <delim><code></delim>`

`<delim>` 标明代码的分界位置, 前后必须一致, 除字母、空格或星号外, 可任意选择使得不与代码本身冲突, 习惯上使用 `|` 符号。

`\LaTeX`

`(a || b) (a_||_b)`

### 5.5 表格

#### 5.5.1 列表格

`tabular` 环境使用 `<column - spec>` 参数指定表格的列数以及每列的格式。

left	center	right	par box with fixed width
L	C	R	P

@ 格式可在单元格前后插入任意的文本, 但同时它也消除了单元格前后额外添加的间距。

```
1:1    one
11:3  eleven
```

格式参数重复

one	two	three	four	five	Hello!	Hello!
1	2	3	4	5	$\LaTeX$	$\LaTeX$
					hello!	$\LaTeX$

辅助格式  $>$  和  $<$ , 用于给列格式前后加上修饰命令

*italic\** normal

*column\** column

辅助格式甚至支持插入 `\centering` 等命令改变 p 列格式的对齐方式, 一般还要加额外的命令 `\arraybackslash` 以免出错。

`\centering` 等对齐命令会破坏表格环境里 `\` 换行命令的定义, `\arraybackslash` 用来恢复之。如果不加 `\arraybackslash` 命令, 也可以用 `\tabularnewline` 命令代替原来的 `\` 实现表格换行。

$\LaTeX$  本身提供了 `tabular*` 环境用来排版定宽表格, 但是不太方便使用, 比如要用到 @ 格式插入额外命令, 令单元格之间的间距为 `\fill`, 但即使这样仍然有瑕疵:

A	B	C	D
a	b	c	d

`tabularx` 宏包为我们提供了方便的解决方案。它引入了一个 X 列格式, 类似 p 列格式, 不过会根据表格宽度自动计算列宽, 多个 X 列格式平均分配列宽。X 列格式也可以用 `array` 里的辅助格式修饰对齐方式:

A	B	C	D
a	b	c	d

### 5.5.2 横线

`\cline{<i>j</i>}` 用来绘制跨越部分单元格的横线:

4	9	2
3	5	7
8	1	6

三线表由 `booktabs` 宏包支持, 它提供了 `\toprule`、`\midrule` 和 `\bottomrule` 命令用以排版三线表的三条线, 以及和 `\cline` 对应的 `\cmidrule`。除此之外, 最好不要用其它横线以及竖线:

Numbers			
	1	2	3
Alphbet	A	B	C
Roman	I	II	III

### 5.5.3 合并单元格

横向合并单元格较为容易,由 `\multicolumn{<n>}{<column-spec>}{<item>}` 命令实现:

其中  $\langle n \rangle$  为要合并的列数,  $\langle column-spec \rangle$  为合并单元格后的列格式,只允许出现一个 `l/c/r` 或 `p` 格式。如果合并前的单元格前后带表格线 `|`,合并后的列格式也要带 `|` 以使得表格的竖线一致。

形如 `\multicolumn{1}{<column-spec>}{<item>}` 的命令可以用来修改某一个单元格的列格式。

1	2	Center
3		Right
4	C	

纵向合并单元格需要用到 `multirow` 宏包提供的

`\multirow{<n>}{<width>}{<item>}` 命令:

$\langle width \rangle$  为合并后单元格的宽度,可以填 `*` 以使用自然宽度。

Item	Value	
	First	Second
A	1	2

### 5.5.4 嵌套表格

在单元格中嵌套一个小表格可以起到“拆分单元格”的效果。

注意要用 `\multicolumn` 命令配合 `@{}` 格式把单元格的额外边距去掉,使得嵌套的表格线能和外层的表格线正确相连:

a	b	c	
a	e	f	c
	e	f	
a	b	c	

如果不需要为“拆分的单元格”画线,并且只在垂直方向“拆分”的话, `makecell` 宏包提供的 `\makecell` 命令是一个简单的解决方案:

a	d1 d2
b	c

### 5.5.5 行距控制

$\text{\LaTeX}$  生成的表格看起来通常比较紧凑。修改参数 `\arraystretch` 可以得到行距更加宽松的表格:

Really loose
tabular rows.

另一种增加间距的办法是给换行命令 `\` 添加可选参数,在这一行下面加额外的间距,适合用于在行间不加横线的表格:

---

 Head lines

tabular lines

---

 tabular lines

## 5.6 图片

TeX 本身不支持插图功能, 需要由 `graphicx` 宏包辅助支持。

使用 `\includegraphics[options]{filename}` 命令加载图片了:

其中 *filename* 为图片文件名, 文件名有时需要使用相对路径或绝对路径。

`\graphicspath` 命令, 用于声明一个或多个图片文件存放的目录, 使用这些目录里的图片时可不用写路径:

`\includegraphics` 命令的可选参数 *options* 支持 *key*=*value* 形式赋值, 常用的参数如下:

参数	含义
<code>width=<i>width</i></code>	将图片缩放到宽度为 <i>width</i>
<code>height=<i>height</i></code>	将图片缩放到高度为 <i>height</i>
<code>scale=<i>scale</i></code>	将图片相对于原尺寸缩放 <i>scale</i> 倍
<code>angle=<i>angle</i></code>	令图片逆时针旋转 <i>angle</i> 度



## 5.7 盒子

### 5.7.1 水平盒子

`\mbox{...}`

`\makebox[⟨width⟩][⟨align⟩]{...}`

`\mbox` 生成一个基本的水平盒子, 内容只有一行, 不允许分段 (除非嵌套其它盒子) `\makebox` 更进一步, 可以加上可选参数用于控制盒子的宽度  $\langle width \rangle$ , 以及内容的对齐方式  $\langle align \rangle$ , 可选居中 `c` (默认值)、左对齐 `l`、右对齐 `r` 和分散对齐 `s`

[Test some words.]

| Test some words. |

[Test some words. |

| Test some words.]

[Test some words.]

### 5.7.2 带框的水平盒子

`\fbox` 和 `\framebox` 让我们可以为水平盒子添加边框。

`\fbox` Test some words.

`\framebox` Test some words.

可以通过 `\setlength` 命令调节边框的宽度 `\fboxrule` 和内边距 `\fboxsep`:

`\fboxrule` Test box.

`\fboxsep` Test box.

### 5.7.3 垂直盒子

排版一个文字可以换行的盒子:

`\parbox[⟨align⟩][⟨height⟩][⟨inner-align⟩][⟨width⟩]{...}`

`\begin{minipage}[⟨align⟩][⟨height⟩][⟨inner-align⟩][⟨width⟩]`

...

`\end{minipage}`

其中  $\langle align \rangle$  为盒子和周围文字的对齐情况 (类似 `tabular` 环境);  $\langle height \rangle$  和  $\langle inner-align \rangle$  设置盒子的高度和内容的对齐方式, 类似水平盒子 `\makebox` 的设置, 不过  $\langle inner-align \rangle$  接受的参数是顶部 `t`、底部 `b`、居中 `c` 和分散对齐 `s`。

天地玄黄

宇宙洪荒

三字经: 人之初 千字文:

性本善

性相近

习相远

如果在 `minipage` 里使用 `\footnote` 命令, 生成的脚注会出现在盒子底部, 编号是独立的, 并且使用小写字母编号。而在 `\parbox` 里无法正常使用 `\footnote`

命令,只能在盒子里使用`\footnotemark`,在盒子外使用`\footnotetext`。



### 5.7.4 标尺盒子

`\rule [⟨raise⟩]{⟨width⟩}{⟨height⟩}` 命令用来画一个实心的矩形盒子,也可适当调整以用来画线(标尺):

Black  box.

Upper  and lower  box.

A \_ line.

## 5.8 浮动体

$\text{\TeX}$  定义了两类浮动体环境 `figure` 和 `table`。习惯上 `figure` 里放图片,`table` 里放表格,但并没有严格限制,可以在任何一个浮动体里放置文字、公式、表格、图片等等任意内容。

```
\begin{table}[⟨placement⟩]
```

```
...
```

```
\end{table}
```

[⟨placement⟩] 参数提供了一些符号用来表示浮动体允许排版的位置,如 `hbp` 允许浮动体排版在当前位置、底部或者单独成页。`table` 和 `figure` 浮动体的默认设置为 `tbp`。

双栏排版环境下, $\text{\TeX}$  提供了 `table*` 和 `figure*` 环境用来排版跨栏的浮动体。它们的用法与 `table` 和 `figure` 一样,不同之处为双栏的 [⟨placement⟩] 参数只能用 `tp` 两个位置。

参数	含义
<code>h</code>	当前位置(代码所处的上下文)
<code>t</code>	顶部
<code>b</code>	底部
<code>p</code>	单独成页
<code>!</code>	在决定位置时忽视限制

`\clearpage` 命令会在另起一页之前,先将所有推迟处理的浮动体排版成页,此时 `htbp` 等位置限制被完全忽略。

`float` 宏包为浮动体提供了 `H` 位置参数,不与 `htbp` 及 `!` 混用。使用 `H` 位置参数时,会取消浮动机制,将浮动体视为一般的盒子插入当前位置。

<sup>3</sup>注脚来自 `fbox`

### 5.8.1 浮动体的标题

图表等浮动体提供了 `\caption{...}` 命令加标题, 并且自动给浮动体编号: 可以用带星号的命令 `\caption*` 生成不带编号的标题, 也可以使用带可选参数的形式 `\caption[...]{...}`, 使得在目录里使用短标题。 `\caption` 命令之后还可以紧跟 `\label` 命令标记交叉引用。

可通过修改 `\figurename` 和 `\tablename` 的内容来修改标题的前缀。标题样式的定制功能由 `caption` 宏包提供。

`table` 和 `figure` 两种浮动体分别有各自的生成目录的命令:

`\listoftables`

`\listoffigures`

### 5.8.2 并排和子图表



图 1: 图片标题

由于标题是横跨一行的, 用 `\caption` 命令为每个图片单独生成标题就需要借助前文提到的 `\parbox` 或者 `minipage` 环境, 将标题限制在盒子内。



图 2: 并排图 1



图 3: 并排图 2

给每个图片定义小标题时, 就要用到 `subfig` 宏包的功能



(a)



(b)

图 4: 使用 subfig 宏包的 \subfloat 命令排版子图。

## 6 数学公式

amsmath 宏包, 对多行公式的排版提供了有力的支持,amsfonts 宏包以及基于它的 amssymb 宏包提供了丰富的数学符号,amsthm 宏包扩展了 L<sup>A</sup>T<sub>E</sub>X 定理证明格式。

### 6.1 公式排版基础

#### 6.1.1 行内和行间公式

行内公式由一对 \$ 符号包裹:

The Pythagorean theorem is  $a^2 + b^2 = c^2$ .

行间公式在 L<sup>A</sup>T<sub>E</sub>X 里由 equation 环境包裹, equation 环境为公式自动生成一个编号, 这个编号可以用 \label 和 \ref 生成交叉引用。

amsmath 的 \eqref 命令甚至为引用自动加上圆括号; 还可以用 \tag 命令手动修改公式的编号, 或者用 \notag 命令取消为公式编号。

The Pythagorean theorem is:

$$a^2 + b^2 = c^2 \tag{1}$$

Equation (1) is called ‘Gougu theorem’ in Chinese.

It’s wrong to say

$$1 + 1 = 3 \tag{dumb}$$

or

$$1 + 1 = 4$$

直接使用不带编号的行间公式:

$$a^2 + b^2 = c^2$$

For short:

$$a^2 + b^2 = c^2$$

Or if you like the long one:

$$a^2 + b^2 = c^2$$

行内公式和行间公式的对比:

In text:  $\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$

In display:

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

## 6.2 数学模式

1. 数学模式中输入的空格被忽略。
2. 不允许有空行(分段)。
3. 所有的字母被当作数学公式中的变量处理, 想在数学公式中输入正体的文本, 可以用`\mathrm` 或者 `amsmath` 提供的 `\text` 命令。

$x^2 \geq 0$  for **all**  $x \in \mathbb{R}$

## 6.3 数学符号

### 6.3.1 一般符号

表 1: 文本/数学模式通用符号

{	\{	}	\}	\$	\\$	%	\%
†	\dag	§	\S	©	\copyright	...	\dots
‡	\ddag	¶	\P	£	\pounds	§	\S
*	\textasteriskcentered	·	\textperiodcentered	•	\textbullet		
®	\textregistered	™	\texttrademark				

表 2: 希腊字母

$\alpha$	<code>\alpha</code>	$\theta$	<code>\theta</code>	$o$	<code>o</code>	$\upsilon$	<code>\upsilon</code>
$\beta$	<code>\beta</code>	$\vartheta$	<code>\vartheta</code>	$\pi$	<code>\pi</code>	$\phi$	<code>\phi</code>
$\gamma$	<code>\gamma</code>	$\iota$	<code>\iota</code>	$\varpi$	<code>\varpi</code>	$\varphi$	<code>\varphi</code>
$\delta$	<code>\delta</code>	$\kappa$	<code>\kappa</code>	$\rho$	<code>\rho</code>	$\chi$	<code>\chi</code>
$\epsilon$	<code>\epsilon</code>	$\lambda$	<code>\lambda</code>	$\varrho$	<code>\varrho</code>	$\psi$	<code>\psi</code>
$\varepsilon$	<code>\varepsilon</code>	$\mu$	<code>\mu</code>	$\sigma$	<code>\sigma</code>	$\omega$	<code>\omega</code>
$\zeta$	<code>\zeta</code>	$\nu$	<code>\nu</code>	$\varsigma$	<code>\varsigma</code>		
$\eta$	<code>\eta</code>	$\xi$	<code>\xi</code>	$\tau$	<code>\tau</code>		
$\Gamma$	<code>\Gamma</code>	$\Lambda$	<code>\Lambda</code>	$\Sigma$	<code>\Sigma</code>	$\Psi$	<code>\Psi</code>
$\Delta$	<code>\Delta</code>	$\Xi$	<code>\Xi</code>	$\Upsilon$	<code>\Upsilon</code>	$\Omega$	<code>\Omega</code>
$\Theta$	<code>\Theta</code>	$\Pi$	<code>\Pi</code>	$\Phi$	<code>\Phi</code>		
以下命令依赖 <code>amsmath</code> 宏包							
$\Gamma$	<code>\varGamma</code>	$\Lambda$	<code>\varLambda</code>	$\Sigma$	<code>\varSigma</code>	$\Psi$	<code>\varPsi</code>
$\Delta$	<code>\varDelta</code>	$\Xi$	<code>\varXi</code>	$\Upsilon$	<code>\varUpsilon</code>	$\Omega$	<code>\varOmega</code>
$\Theta$	<code>\varTheta</code>	$\Pi$	<code>\varPi</code>	$\Phi$	<code>\varPhi</code>		
依赖 <code>amssymb</code> 宏包							
$F$	<code>\digamma</code>	$\varkappa$	<code>\varkappa</code>	$\beth$	<code>\beth</code>	$\gimel$	<code>\gimel</code>
$\daleth$	<code>\daleth</code>						

表 3: 其它符号

$\dots$	<code>\dots</code>	$\cdots$	<code>\cdots</code>	$\vdots$	<code>\vdots</code>	$\ddots$	<code>\ddots</code>
$\hbar$	<code>\hbar</code>	$\imath$	<code>\imath</code>	$\jmath$	<code>\jmath</code>	$\ell$	<code>\ell</code>
$\Re$	<code>\Re</code>	$\Im$	<code>\Im</code>	$\aleph$	<code>\aleph</code>	$\wp$	<code>\wp</code>
$\forall$	<code>\forall</code>	$\exists$	<code>\exists</code>	$\partial$	<code>\partial</code>	$\nabla$	<code>\nabla</code>
$\prime$	<code>\prime</code>	$\emptyset$	<code>\emptyset</code>	$\infty$	<code>\infty</code>	$\nabla$	<code>\nabla</code>
$\triangle$	<code>\triangle</code>	$\perp$	<code>\perp</code>	$\top$	<code>\top</code>	$\angle$	<code>\angle</code>
$\sqrt{\quad}$	<code>\surd</code>	$\diamond$	<code>\diamondsuit</code>	$\heartsuit$	<code>\heartsuit</code>	$\clubsuit$	<code>\clubsuit</code>
$\spadesuit$	<code>\spadesuit</code>	$\neg$	<code>\neg</code> or <code>\lnot</code>	$\flat$	<code>\flat</code>	$\natural$	<code>\natural</code>
$\sharp$	<code>\sharp</code>						
以下命令依赖 <code>latexsym</code> 宏包							
$\mho$	<code>\mho</code>	$\Box$	<code>\Box</code>	$\Diamond$	<code>\Diamond</code>		

$$a_1, a_2, \dots, a_n$$

$$a_1 + a_2 + \dots + a_n$$

### 6.3.2 指数、上下标和导数

$\wedge$  和  $\_$  标明上下标。注意上下标的内容(子公式)一般需要用花括号包裹, 否则上下标只对后面的一个符号起作用:

$$p_{ij}^3 \quad m_{\text{Kunth}} \quad \sum_{k=1}^3 k$$

$$a^x + y \neq a^{x+y} \quad e^{x^2} \neq e^{x^2}$$

导数符号' 是一类特殊的上标, 可以适当连用表示多阶导数, 也可以在其后连用上标:

$$f(x) = x^2 \quad f'(x) = 2x \quad f''(x) = 4$$

### 6.3.3 分式和根式

分式使用  $\frac{\text{分子}}{\text{分母}}$  来书写。分式的大小在行间公式中是正常大小, 而在行内被极度压缩。amsmath 提供了方便的命令  $\dfrac$  和  $\tfrac$ , 令用户能够在行内使用正常大小的分式, 或是反过来。

In display style:

$$3/8 \quad \frac{3}{8} \quad \frac{3}{8}$$

In text style:  $1\frac{1}{2}$  hours  $1\frac{1}{2}$  hours

一般的根式使用  $\sqrt{\dots}$ ; 表示 n 次方根时写成  $\sqrt[n]{\dots}$ 。

$$\sqrt{x} \Leftrightarrow x^{1/2} \quad \sqrt[3]{2} \quad \sqrt{x^2 + y}$$

特殊的分式形式, 如二项式结构, 由 amsmath 宏包的  $\binom$  命令生成:

Pascal's rule is

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

## 6.3.4 关系符

表 4: 二元关系符

<	<	>	>	=	=
≡	\equiv	≤	\leq or \le	≥	\geq or \ge
≪	\ll	≫	\gg	≐	\doteq
<	\prec	>	\succ	~	\sim
≲	\preceq	≳	\succeq	≈	\simeq
⊂	\subset	⊃	\supset	≈	\approx
⊆	\subseteq	⊇	\supseteq	≅	\cong
⊆	\sqsubseteq	⊇	\sqsupseteq	⋈	\bowtie
∈	\in	∋	\ni , \owns	∝	\propto
⊢	\vdash	⊣	\dashv	⊨	\models
	\mid		\parallel	⊥	\perp
⌣	\smile	⌣	\frown	∝	\asymp
:	:	∉	\notin	≠	\neq or \ne

以下命令依赖 latexsym 宏包

⊆	\sqsubset	⊇	\sqsupset	⋈	\Join
---	-----------	---	-----------	---	-------

以下命令依赖 amssymb 宏包

⋖	\lessdot	⋗	\gtrdot	≐	\doteqdot
⋘	\leqslant	⋙	\geqslant	≐	\risingdotseq
⋚	\eqslantless	⋛	\eqslantgtr	≐	\fallingdotseq
≐	\leqq	≐	\geqq	≐	\eqcirc
≐	\circeq	≫	\ggg	≐	\lll or \llless
⋗	\lesssim	⋗	\gtrsim	≐	\triangle
⋗	\lessapprox	⋗	\gtrapprox	≐	\bumpeq
⋗	\lessgtr	⋗	\gtrless	≐	\Bumpeq
⋗	\lesseqgtr	⋗	\gtreqless	~	\thicksim
⋗	\lesseqqgtr	⋗	\gtreqqless	≈	\thickapprox
⋗	\preccurlyeq	⋗	\succcurlyeq	≈	\approxeq
⋗	\curlyeqprec	⋗	\curlyeqsucc	~	\backsim
⋗	\precsim	⋗	\succsim	≈	\backsimeq
⋗	\precapprox	⋗	\succapprox	≐	\vDash
⋗	\subseteqq	⋗	\supseteqq	≐	\Vdash
	\shortparallel	⋗	\Supset	≐	\Vvdash
◀	\blacktriangleleft	⊆	\sqsupset	∋	\backepsilon
▷	\vartriangleright	∴	\because	∝	\varpropto
▶	\blacktriangleright	⊆	\Subset	∅	\between
▷	\trianglerighteq	⌣	\smallfrown	⌣	\pitchfork
△	\vartriangleleft		\shortmid	⌣	\smallsmile
△	\trianglelefteq	∴	\therefore	⊆	\sqsubset

TeX 还提供了自定义二元关系符的命令 `\stackrel`, 用于将一个符号叠加在原有的二元关系符之上:

$$f_n(x) \stackrel{*}{\approx} 1$$

### 6.3.5 算符

表 5: 二元运算符

+	+	-	-	±	<code>\pm</code>
≠	<code>\mp</code>	◁	<code>\triangleleft</code>	·	<code>\cdot</code>
÷	<code>\div</code>	▷	<code>\triangleright</code>	×	<code>\times</code>
\	<code>\setminus</code>	★	<code>\star</code>	∪	<code>\cup</code>
∩	<code>\cap</code>	*	<code>\ast</code>	⊔	<code>\sqcup</code>
⊐	<code>\sqcap</code>	○	<code>\circ</code>	∨	<code>\vee</code>
∨	<code>\lor</code>	∧	<code>\wedge</code>	∧	<code>\land</code>
•	<code>\bullet</code>	⊕	<code>\oplus</code>	⊖	<code>\ominus</code>
◇	<code>\diamond</code>	⊙	<code>\odot</code>	⊗	<code>\otimes</code>
⊕	<code>\uplus</code>	⊗	<code>\otimes</code>	○	<code>\bigcirc</code>
∏	<code>\amalg</code>	△	<code>\bigtriangleup</code>	▽	<code>\bigtriangledown</code>
†	<code>\dagger</code>	‡	<code>\ddagger</code>	∷	<code>\wr</code>

以下命令依赖 `latexsym` 宏包

◁	<code>\lhd</code>	▷	<code>\rhd</code>
⊑	<code>\unlhd</code>	⊒	<code>\unrhd</code>

以下命令依赖 `amssymb` 宏包

‡	<code>\dotplus</code>	·	<code>\centerdot</code>		
×	<code>\ltimes</code>	×	<code>\rtimes</code>	※	<code>\divideontimes</code>
∪	<code>\doublecup</code>	∩	<code>\doublecap</code>	∖	<code>\smallsetminus</code>
∨	<code>\veebar</code>	∧	<code>\barwedge</code>	∧	<code>\doublebarwedge</code>
⊕	<code>\boxplus</code>	⊖	<code>\boxminus</code>	⊖	<code>\circleddash</code>
⊗	<code>\boxtimes</code>	⊙	<code>\boxdot</code>	⊙	<code>\circledcirc</code>
⊔	<code>\intercal</code>	⊗	<code>\circledast</code>	∠	<code>\rightthreetimes</code>
∨	<code>\curlyvee</code>	∧	<code>\curlywedge</code>	∧	<code>\listoffigures</code>

表 6: L<sup>A</sup>T<sub>E</sub>X 作为算符的函数名称

不带上下限的算符							
sin	\sin	arcsin	\arcsin	sinh	\sinh	exp	\exp
dim	\dim	cos	\cos	arccos	\arccos	cosh	\cosh
log	\log	ker	\ker	tan	\tan	arctan	\arctan
tanh	\tanh	lg	\lg	hom	\hom	cot	\cot
arg	\arg	coth	\coth	ln	\ln	deg	\deg
sec	\sec	csc	\csc				
带上下限的算符							
lim	\lim	lim sup	\limsup	lim inf	\liminf	sup	\sup
inf	\inf	min	\min	max	\max	det	\det
Pr	\Pr	gcd	\gcd				

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

对于求模表达式,L<sup>A</sup>T<sub>E</sub>X 提供了 \pmod 和 \bmod 命令:

$$a \bmod b \quad x \equiv a \pmod{b}$$

amsmath 允许用户在导言区用 \DeclareMathOperator 在定义自己的算符,其中带星号的命令定义带上下限的算符:

$$\operatorname{argh} 3 = \operatorname{Nnt}_{x=1} 4x$$

### 6.3.6 巨算符

In text:  $\sum_{i=1}^n \int_0^{\frac{\pi}{2}} \oint_0^{\frac{\pi}{2}} \prod_{\epsilon}$

In display:

$$\sum_{i=1}^n \int_0^{\frac{\pi}{2}} \oint_0^{\frac{\pi}{2}} \prod_{\epsilon}$$

巨算符的上下标位置可由 \limits 和 \nolimits 控制

In text:  $\sum_{i=1}^n \int_0^{\frac{\pi}{2}} \oint_0^{\frac{\pi}{2}} \prod_{\epsilon}$

In display:

$$\sum_{i=1}^n \int_0^{\frac{\pi}{2}} \oint_0^{\frac{\pi}{2}} \prod_{\epsilon}$$

amsmath 宏包还提供了 \substack, 能够在下限位置书写多行表达式:

$$\sum_{\substack{0 \leq i \leq n \\ j \in \mathbb{R}}} P(i, j) = Q(n)$$

subarray 环境更进一步, 令多行表达式可选择居中 (c) 或左对齐 (l):

$$\sum_{\substack{0 \leq i \leq n \\ j \in \mathbb{R}}} P(i, j) = Q(n)$$

### 6.3.7 数学重音和上下括号

表 7: 数学重音符号

$\hat{a}$	<code>\hat{a}</code>	$\check{a}$	<code>\check{a}</code>	$\tilde{a}$	<code>\tilde{a}</code>
$\acute{a}$	<code>\acute{a}</code>	$\grave{a}$	<code>\grave{a}</code>	$\breve{a}$	<code>\breve{a}</code>
$\bar{a}$	<code>\bar{a}</code>	$\vec{a}$	<code>\vec{a}</code>	$\mathring{a}$	<code>\mathring{a}</code>
$\dot{a}$	<code>\dot{a}</code>	$\ddot{a}$	<code>\ddot{a}</code>		
$\widehat{AAA}$	<code>\widehat{AAA}</code>	$\widetilde{AAA}$	<code>\widetilde{AAA}</code>		
以下命令依赖 amsmath 宏包					
$\ddot{a}$	<code>\ddot{a}</code>	$\overset{\cdot\cdot}{a}$	<code>\overset{\cdot\cdot}{a}</code>		
以下命令依赖 yhmath 宏包					
$\wideparen{AAA}$	<code>\wideparen{AAA}</code>				

使用时要注意重音符号的作用区域, 一般应当对某个符号而不是“符号加下标”使用重音:

$$\begin{aligned} \bar{x}_0 & \quad \bar{x}_0 \\ \vec{x}_0 & \quad \vec{x}_0 \\ \hat{e}_x & \quad \hat{e}_x \end{aligned}$$

$\LaTeX$  也能为多个字符加重音, 包括直接画线的 `\overline` 和 `\underline` 命令 (可叠加使用)、宽重音符号 `\widehat`、表示向量的箭头 `\overrightarrow` 等。

表 8: 作为重音的箭头符号

$\overrightarrow{AB}$	<code>\overrightarrow{AB}</code>	$\overleftarrow{AB}$	<code>\overleftarrow{AB}</code>
以下命令依赖 amsmath 宏包			
$\underlineleftarrow{AB}$	<code>\underlineleftarrow{AB}</code>	$\underrightarrow{AB}$	<code>\underrightarrow{AB}</code>
$\overleftarrow{\overrightarrow{AB}}$	<code>\overleftarrow{\overrightarrow{AB}}</code>	$\underleftarrow{\overrightarrow{AB}}$	<code>\underleftarrow{\overrightarrow{AB}}</code>

$$\begin{aligned} 0.\bar{3} & = \frac{1}{\underline{\underline{3}}} \\ x\hat{Y} & \quad \widehat{XY} \end{aligned}$$

$$\overrightarrow{AB} \quad \overleftarrow{AB}$$

`\overbrace` 和 `\underbrace` 命令用来生成上/下括号, 各自可带一个上/下标公式:

$$\underbrace{a+b+c \cdot d+e+f}_{\text{meaning of life}} = 42$$

### 6.3.8 箭头

表 9: 箭头

$\leftarrow$	<code>\leftarrow</code> or <code>\gets</code>	$\longleftarrow$	<code>\longleftarrow</code>
$\rightarrow$	<code>\rightarrow</code> or <code>\to</code>	$\longrightarrow$	<code>\longrightarrow</code>
$\leftrightarrow$	<code>\leftrightarrow</code>	$\longleftrightarrow$	<code>\longleftrightarrow</code>
$\Leftarrow$	<code>\Leftarrow</code>	$\Lleftarrow$	<code>\Lleftarrow</code>
$\Rightarrow$	<code>\Rightarrow</code>	$\Rrightarrow$	<code>\Rrightarrow</code>
$\Leftrightarrow$	<code>\Leftrightarrow</code>	$\Llongleftrightarrow$	<code>\Llongleftrightarrow</code>
$\mapsto$	<code>\mapsto</code>	$\longmapsto$	<code>\longmapsto</code>
$\hookrightarrow$	<code>\hookrightarrow</code>	$\hookrightarrow$	<code>\hookrightarrow</code>
$\lhookrightarrow$	<code>\lhookrightarrow</code>	$\rhookrightarrow$	<code>\rhookrightarrow</code>
$\lharpoonup$	<code>\lharpoonup</code>	$\rharpoonup$	<code>\rharpoonup</code>
$\lharpoondown$	<code>\lharpoondown</code>	$\rharpoondown$	<code>\rharpoondown</code>
$\rightleftharpoons$	<code>\rightleftharpoons</code>	$\iff$	<code>\iff</code>
$\uparrow$	<code>\uparrow</code>	$\downarrow$	<code>\downarrow</code>
$\Uparrow$	<code>\Uparrow</code>	$\uparrow$	<code>\uparrow</code>
$\Downarrow$	<code>\Downarrow</code>	$\downarrow$	<code>\downarrow</code>
$\nearrow$	<code>\nearrow</code>	$\searrow$	<code>\searrow</code>
$\swarrow$	<code>\swarrow</code>	$\nwarrow$	<code>\nwarrow</code>

以下命令依赖 `latexsym` 宏包

$\leadsto$  `\leadsto`

`amsmath` 的 `\xleftarrow` 和 `\xrightarrow` 命令提供了长度可以伸展的箭头, 并且可以为箭头增加上下标:

$$a \xleftarrow{x+y+z} b$$

$$c \xrightarrow[x<y]{a*b*c} d$$

## 6.3.9 括号和定界符

表 10: 定界符

( ( ) )	↑	\uparrow
[ \lbrack or [ ] \rbrack or ]	↓	\downarrow
{ \lbrace or \{ } \rbrace or \}	↕	\updownarrow
< \langle > \rangle	↕	\Uparrow
\or	⇓	\Downarrow
/ / \ \backslash	⇕	\Updownarrow
⌊ \lfloor ⌋ \rfloor	⌈	\rceil
⌈ \lceil		

表 11: 用于行间公式的大定界符

{	\lgroup		\arrowvert
}	\rgroup		\Arrowvert
{	\lmoustache	⌋	\bracevert
}	\rmoustache		

$$a, b, c \neq \{a, b, c\}$$

使用 `\left` 和 `\right` 命令可令括号 (定界符) 的大小可变, 在行间公式中常用。`LaTeX` 会自动根据括号内的公式大小决定定界符大小。`\left` 和 `\right` 必须成对使用。需要使用单个定界符时, 另一个定界符写成 `\left.` 或 `\right.`

$$1 + \left( \frac{1}{1-x^2} \right)^3 \quad \left. \frac{\partial f}{\partial t} \right|_{t=0}$$

还可以用 `\big`、`\bigg` 等命令生成固定大小的定界符。更常用的形式是类似 `\left` 的 `\bigl`、`\biggl` 等, 以及类似 `\right` 的 `\bigr`、`\biggr` 等 (`\bigl` 和 `\bigr` 不必成对出现)。

$$\left( (x+1)(x-1) \right)^2$$

((( ( ))) ||||| ⇓⇓⇓⇓

## 6.4 多行公式

### 6.4.1 长公式折行

amsmath 宏包的 `multline` 环境提供了书写折行长公式的方便环境。它允许用 `\\` 折行, 将公式编号放在最后一行。多行公式的首行左对齐, 末行右对齐, 其余行居中。

$$\begin{aligned}
 a + b + c + d + e + f + g + h + i \\
 &= j + k + l + m + n \\
 &= o + p + q + r + s \\
 &= t + u + v + w + x \quad (2)
 \end{aligned}$$

类似 `equation*`, `multline*` 环境排版不带编号的折行长公式。

### 6.4.2 多行公式

罗列一系列公式, 并令其按照等号对齐。

目前最常用的是 `align` 环境, 它将公式用 `&` 隔为两部分并对齐。分隔符通常放在等号左边:

$$a = b + c \quad (3)$$

$$= d + c \quad (4)$$

`align` 环境会给每行公式都编号。我们仍然可以用 `\notag` 去掉某行的编号。为了对齐加号, 我们将分隔符放在等号右边, 这时需要给等号后添加一对括号 `{}` 以产生正常的间距:

$$a = b + c \quad (5)$$

$$= d + e + f + g + h + i + j + k + l \\ + m + n + o \quad (6)$$

$$= p + q + r + s \quad (7)$$

`align` 还能够对齐多组公式, 除等号前的 `&` 之外, 公式之间也用 `&` 分隔:

$$a = 1 \quad b = 2 \quad c = 3 \quad (8)$$

$$d = -1 \quad e = -2 \quad f = -5 \quad (9)$$

如果我们不需要按等号对齐, 只需罗列数个公式, `gather` 将是一个很好用的环境:

$$a = b + c \quad (10)$$

$$d = e + f + g \quad (11)$$

$$h + i = j + k \quad (12)$$

$$l + m = n \quad (13)$$

`align` 和 `gather` 有对应的不带编号的版本 `align*` 和 `gather*`

### 6.4.3 公用编号的多行公式

多个公式组在一起公用一个编号, 编号位于公式的居中位置。

amsmath 宏包提供了诸如 aligned、gathered 等环境, 与 equation 环境套用。

以 -ed 结尾的环境用法与前一节不以 -ed 结尾的环境用法一一对应。

$$\begin{aligned} a &= b + c \\ d &= e + f + g \\ h + i &= j + k \\ l + m &= n \end{aligned} \tag{14}$$

split 环境和 aligned 环境用法类似, 也用于和 equation 环境套用, 区别是 split 只能将每行的一个公式分两栏, aligned 允许每行多个公式多栏。

## 6.5 数组和矩阵

二维数组, LaTeX 提供了 array 环境, 用法与 tabular 环境极为类似, 也需要定义列格式, 并用 \换行。数组可作为一个公式块, 在外套用 \left、\right 等定界符:

$$\mathbf{X} = \left( \begin{array}{cccc} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nn} \end{array} \right)$$

amsmath 提供的 cases 环境

$$|x| = \begin{cases} -x & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ x & \text{if } x > 0. \end{cases}$$

amsmath 宏包提供了多种排版矩阵的环境, 包括不带定界符的 matrix, 以及带各种定界符的矩阵 pmatrix (())、bmatrix ([])、Bmatrix ({}). 使用这些环境时, 无需给定列格式:

$$\begin{array}{cc} 1 & 2 \\ 3 & 4 \end{array} \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nn} \end{bmatrix}$$

在矩阵中的元素里排版分式时, 一来要用到 \dfrac 等命令, 二来行与行之间有可能紧贴着, 这时要调节间距:

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}$$

## 6.6 公式中的间距

生成间距的命令 `\quad` 和 `\qquad` 和 `\,` 在文本和数学环境中可用

`\,`、`\:`、`\;` 以及负间距 `\!` 命令只用于数学环境。文本中的 `\,` 也能使用在数学公式中。

无额外间距	<code>aa</code>	<code>\,</code>	<code>aa</code>
<code>\quad</code>	<code>a a</code>	<code>\:</code>	<code>aa</code>
<code>\qquad</code>	<code>a a</code>	<code>\;</code>	<code>aa</code>
<code>\,</code>	<code>aa</code>	<code>\!</code>	<code>aa</code>

一个常见的用途是修正积分的被积函数  $f(x)$  和微元  $dx$  之间的距离。

$$\int_a^b f(x)dx \quad \int_a^b f(x) dx$$

另一个用途是生成多重积分号。如果我们直接连写两个 `\int`，之间的间距将会过宽，此时可以使用负间距 `\!` 修正之。

$$\int \int f(x)g(y) dx dy$$

$$\iint f(x)g(y) dx dy$$

`amsmath` 提供了更方便的多重积分号，如二重积分 `\iint`、三重积分 `\iiint` 等。

$$\iint f(x)g(y) dx dy$$

$$\iint \iiint \int \cdots \int$$

## 6.7 数学符号的字体控制

### 6.7.1 数学字母字体

表 12: 数学字母字体

示例	命令	依赖的宏包
<i>ABCDEabcde</i> 1234	<code>\mathnormal{...}</code>	
ABC <i>DE</i> abcde1234	<code>\mathrm{...}</code>	
<i>ABC</i> <b>DE</b> abcde1234	<code>\mathit{...}</code>	
<b>ABCDEabcde</b> 1234	<code>\mathbf{...}</code>	
ABC <i>DE</i> abcde1234	<code>\mathsf{...}</code>	
ABC <i>DE</i> abcde1234	<code>\mathhtt{...}</code>	
<i>ABC</i> $\mathcal{DE}$	<code>\mathcal{...}</code>	只大写字母
<i>ABC</i> $\mathcal{DE}$	<code>\mathcal{...}</code>	eucal, 只大写字母
<i>ABC</i> $\mathcal{DE}$	<code>\mathscr{...}</code>	mathrsfs, 只大写字母
$\mathfrak{ABCDE}$ abcdegyvw	<code>\mathfrak{...}</code>	amssymb 或 eufrak
<b>ABCDE</b>	<code>\mathbb{...}</code>	amssymb, 只大写字母

$$\mathcal{R} \quad \mathfrak{R} \quad \mathbb{R}$$

$$\mathcal{L} = -\frac{1}{4}F_{\mu\nu}F^{\mu\nu}$$

$\mathfrak{su}(2)$  and  $\mathfrak{so}(3)$  Lie algebra

### 6.7.2 数学符号的尺寸

表 13: 数学符号尺寸

命令	尺寸	示例
<code>\displaystyle</code>	行间公式尺寸	$\sum a$
<code>\textstyle</code>	行内公式尺寸	$\sum a$
<code>\scriptstyle</code>	上下标尺寸	$a$
<code>\scriptscriptstyle</code>	次级上下标尺寸	$a$

$$P = \frac{\sum_{i=1}^n (x_i - x)(y_i - y)}{\left[ \sum_{i=1}^n (x_i - x)^2 \sum_{i=1}^n (y_i - y)^2 \right]^{1/2}}$$

### 6.7.3 加粗的数学符号

L<sup>A</sup>T<sub>E</sub>X 提供了一个命令 `\boldmath` 令用户可以将整套数学字体切换为粗体版本。但这个命令只能在公式外使用:

$\mu, M$      $\mu, \mathbf{M}$      $\boldsymbol{\mu}, \mathbf{M}$

`amsmath` 提供了一个 `\boldsymbol` 命令 (由调用的 `amsbsy` 宏包提供), 用于打破 `\boldmath` 的限制, 在公式内部将一部分符号切换为粗体。

$\mu, M$      $\boldsymbol{\mu}, \mathbf{M}$

## 6.8 定理环境

### 6.8.1 L<sup>A</sup>T<sub>E</sub>X 原始的定理环境

`\newtheorem{<theoremenvironment>}{<title>}[<section - level>]`

`\newtheorem{<theoremenvironment>}{<counter>}{<title>}`

`<theoremenvironment>` 为定理环境的名称。`<title>` 是定理环境的标题 (“定理”, “公理” 等)。

定理的序号由两个可选参数之一决定, 它们不能同时使用:

- `<section - level>` 为章节级别, 如 `chapter`、`section` 等, 定理序号成为章节的下一级序号;
- `<counter>` 为用 `\newcounter` 自定义的计数器名称, 定理序号由这个计数器管理。

**My Theorem 6.8.1.** *The light speed in vacuum is 299,792,458 m/s.*

**My Theorem 6.8.2** (Energy-momentum relation). *The relationship of energy, momentum and mass is*

$$E^2 = m_0^2 c^4 + p^2 c^2$$

where  $c$  is the light speed described in theorem 6.8.1.

### 6.8.2 amsthm 宏包

`amsthm` 提供了 `\theoremstyle` 命令支持定理格式的切换, 在用 `\newtheorem` 命令定义定理环境之前使用。

`amsthm` 预定义了三种格式用于 `\theoremstyle`:

- `plain` 和 L<sup>A</sup>T<sub>E</sub>X 原始的格式一致
- `definition` 使用粗体标签、正体内容
- `remark` 使用斜体标签、正体内容

另外 `amsthm` 还支持用带星号的 `\newtheorem*` 定义不带序号的定理环境

**Law 1.** Don't hide in the witness box.

**Jury 2** (The Twelve). *It could be you! So beware and see law 1.*

*Margaret.* No, No, No

*Margaret.* Denis!

以上例子定义的 jury 环境与 law 环境共用编号,mar 环境不编号  
amsthm 还支持使用 \newtheoremstyle 命令自定义定理格式,更为方便使用的是 ntheorem 宏包。

### 6.8.3 证明环境和证毕符号

amsthm 还提供了一个 proof 环境用于排版定理的证明过程。proof 环境末尾自动加上一个 □ 证毕符号:

*Proof.* For simplicity, we use

$$E = mc^2$$

That's it. □

如果行末是一个不带编号的公式,符号会另起一行,这时可使用 \qedhere 命令将号放在公式末尾:

*Proof.* For simplicity, we use

$$E = mc^2 \quad \square$$

\qedhere 对于 align\* 等命令也有效:

*Proof.* Assuming  $\gamma = 1/\sqrt{a - v^2/c^2}$ , then

$$\begin{aligned} E &= \gamma m_0 c^2 \\ p &= \gamma m_0 v \end{aligned} \quad \square$$

在使用带编号的公式时,建议最好不要在公式末尾使用 \qedhere 命令。对带编号的公式使用 \qedhere 命令会使符号放在一个难看的位置,紧贴着公式:

*Proof.* For simplicity, we use

$$E = mc^2 \quad (15) \quad \square$$

*Proof.* For simplicity, we use

$$E = mc^2 \quad (16) \quad \square$$

在 align 等环境中使用 \qedhere 命令会使 □ 盖掉公式的编号:

*Proof.* Assuming  $\gamma = 1/\sqrt{a - v^2/c^2}$ , then

$$\begin{aligned} E &= \gamma m_0 c^2 \\ p &= \gamma m_0 v \end{aligned} \quad (17) \quad \square$$

使用 equation 嵌套 aligned 等环境时,\qedhere 命令会将 □ 直接放在公式后:

*Proof.* Assuming  $\gamma = 1/\sqrt{a - v^2/c^2}$ , then

$$\begin{aligned} E &= \gamma m_0 c^2 \\ p &= \gamma m_0 v \quad \square \end{aligned} \tag{18}$$

证毕符号  $\square$  本身被定义在命令 `\qedsymbol` 中, 如果有使用实心符号作为证毕符号的需求, 需要自行用 `\renewcommand` 命令修改

*Proof.* For simplicity, we use

$$E = mc^2 \quad \blacksquare$$

## 7 排版样式设定

### 7.1 字体和字号

The small and **bold** Romans ruled all of great big *Italy*.

#### 7.1.1 字体样式

表 14: 字体命令

<code>\rmfamily</code>	<code>\textrm{...}</code>	<code>\rm{...}</code>	roman	衬线字体 (罗马体)
<code>\sffamily</code>	<code>\textsf{...}</code>	<code>\sf{...}</code>	sans serif	无衬线字体
<code>\ttfamily</code>	<code>\texttt{...}</code>	<code>\tt{...}</code>	typewriter	等宽字体
<code>\mdseries</code>	<code>\textmd{...}</code>		medium	正常粗细 (中等)
<code>\bfseries</code>	<code>\textbf{...}</code>	<code>\bf{...}</code>	<b>bold face</b>	粗体
<code>\upshape</code>	<code>\textup{...}</code>		upright	直立体
<code>\itshape</code>	<code>\textit{...}</code>	<code>\it{...}</code>	<i>italic</i>	意大利斜体
<code>\slshape</code>	<code>\textsl{...}</code>	<code>\sl{...}</code>	<i>slanted</i>	倾斜体
<code>\scshape</code>	<code>\textsc{...}</code>	<code>\sc{...}</code>	SMALL CAPS	小字母大写
<code>\em</code>	<code>\emph{...}</code>		<u>emphasized</u>	强调, 默认斜体
<code>\normalfont</code>	<code>\textnormal{...}</code>		normal font	默认字体

诸如 `\bfseries` 形式的命令将会影响之后所有的字符, 如果想要让它在局部生效, 需要用花括号分组, 也就是写成 `{\bfseries <some text>}` 这样的形式; 对应的 `\textbf` 形式带一个参数, 只改变参数内部的字体, 更为常用。在公式中, 直接使用 `\textbf` 等命令不会起效, 甚至报错。

## 7.1.2 字号

表 15: 字号

字号		10pt 选项 (默认)	11pt 选项	12pt 选项
<code>\tiny</code>	tiny font	5pt	6pt	6pt
<code>\scriptsize</code>	very small font	7pt	8pt	8pt
<code>\footnotesize</code>	quite small font	8pt	9pt	10pt
<code>\small</code>	small font	9pt	10pt	10.95pt
<code>\normalsize</code>	normalfont	10pt	10.95pt	12pt
<code>\large</code>	large font	12pt	12pt	14.4pt
<code>\Large</code>	larger font	14.4pt	14.4pt	17.28pt
<code>\LARGE</code>	very large font	17.28pt	17.28pt	20.74pt
<code>\huge</code>	huge	20.74pt	20.74pt	24.88pt
<code>\huge</code>	largest	24.88pt	24.88pt	24.88pt

字号命令实际大小依赖于所使用的文档类及其选项。

使用字号命令的时候,通常也需要用花括号进行分组,如同 `\rmfamily` 那样。

He likes **large and** small **letters**.

还提供了一个基础的命令 `\fontsize` 用于设定任意大小的字号:

```
\fontsize{<size>}{<line - skip>}
```

`<size>` 为字号,`<line - skip>` 为基础行距。

表 15 中的命令也都各自设定了与字号对应的基础行距,大小为字号的 1.2 倍。

如果不是在导言区,`\fontsize` 设定需要 `\selectfont` 命令才能立即生效,而表 15 的字号设定都是立即生效的。

## 7.1.3 选用字体宏包

表 16: 常见的 L<sup>A</sup>T<sub>E</sub>X 字体宏包

文本 / 数学字体搭配的宏包	
lmodern	Latin Modern 字体, 对 Computer Modern 字体的扩展
cmbright	仿 Computer Modern 风格的无衬线字体
euler	Euler 风格数学字体, 也出自于高德纳之手
ccfonts	Concrete 风格字体
txfonts	Times 风格的字体宏包
pxfonts	Palatino 风格的字体宏包
stix	Times 风格的字体宏包
newtxtext,newtxmath	txfonts 的改进版本, 分别设置文本和数学字体
newpxtext,newpxmath	pxfonts 的改进版本, 分别设置文本和数学字体
mathptmx	psnfss 字体宏集之一, Times 风格, 较为陈旧, 不推荐使用
mathpazo	psnfss 字体宏集之一, Palatino 风格, 较为陈旧, 不推荐使用
fourier	fourier 风格数学字体, 配合 Utopia 正文字体
fouriernc	fourier 风格数学字体, 配合 New Century Schoolbook 正文字体
arev	Arev 无衬线字体宏包, Vera Sans 风格
mathdesign	配合 Charter / Garamond / Utopia 正文字体的数学字体宏
文本字体宏包	
以下字体包括传统的 L <sup>A</sup> T <sub>E</sub> X 字体格式以及 TrueType / OpenType 格式。	
cm-unicode	Computer Modern 风格的 Unicode 字体, 支持多种西方语言
dejavu	DejaVu 开源字体
droid	Droid 开源字体
inconsolata	Inconsolata 开源等宽字体
libertine	Linux Libertine / Linux Biolium 开源字体
roboto	Roboto 开源无衬线字体
sourcesanspro	Source Sans Pro 开源无衬线字体
sourcecodepro	Source Code Pro 开源等宽字体
符号宏包	
mathabx	数学符号宏包之一
MnSymbol	数学符号宏包之一, 配合 Minion Pro 文本字体
fdsymbol	数学符号宏包之一
pifont	Zapf Dingbats 符号宏包

### 7.1.4 字体编码

字体编码规定了一个字体里包含的符号, 并将若干符号用  $\LaTeX$  命令定义。常见的正文字体编码有 OT1 和 T1 等。 $\LaTeX$  默认使用兼容 plain  $\TeX$  的 OT1 编码

切换字体编码要用到 fontenc 宏包:

```
\usepackage[T1]{fontenc}
```

fontenc 宏包是用来配合传统的  $\LaTeX$  字体的, 如表 16 中的一些传统字体宏包。如果使用 xelatex 编译方式, 并使用 fontspec 宏包调用 ttf 或 otf 格式字体, 就不要再使用 fontenc 宏包。

### 7.1.5 使用 fontspec 宏包更改字体 (xelatex)

xelatex 命令下支持用户调用字体的宏包是 fontspec。宏包提供了几个设置全局字体的命令, 设置 \rmfamily 等对应命令的默认字体:<sup>4</sup>

```
\setmainfont[⟨fontfeatures⟩]{⟨fontname⟩}
```

```
\setsansfont[⟨fontfeatures⟩]{⟨fontname⟩}
```

```
\setmonofont[⟨fontfeatures⟩]{⟨fontname⟩}
```

⟨fontname⟩ 使用字体的文件名 (带扩展名) 或者字体的英文名称,

⟨fontfeatures⟩ 用来手动配置对应的粗体或斜体。

### 7.1.6 使用 xeCJK 宏包更改中文字体

```
\setCJKmainfont[⟨fontfeatures⟩]{⟨fontname⟩}
```

```
\setCJKsansfont[⟨fontfeatures⟩]{⟨fontname⟩}
```

```
\setCJKmonofont[⟨fontfeatures⟩]{⟨fontname⟩}
```

由于中文字体少有对应的粗体或斜体, ⟨fontfeatures⟩ 里多用其他字体来配置, 比如将宋体对应的 BoldFont 配置为黑体, 而 ItalicFont 配置为楷体。

---

<sup>4</sup>新版本 fontspec 的命令支持 (并且推荐) 把必选参数 ⟨fontname⟩ 放在可选参数 ⟨fontfeatures⟩ 的前面。

## A 附录