

# 杂谈勾股定理

张三\*

厦门大学

2020 年 11 月 13 日

---

\*厦门大学学生。

## 摘要

这是一篇关于勾股定理的小短文。

本文介绍  $\text{\LaTeX}$  的使用。

摘要的标题由 `\abstractname` 定义，英文默认是 “Abstract”，中文默认是 “摘要”，可通过重定义 `\abstractname` 来设置，如：

```
\CTEXoptions=[abstractname={摘\quad 要}]
```

# 目录

1	勾股定理在古代	1
2	勾股定理的近代形式	1
3	组织文本	2
3.1	标点符号	2
3.1.1	引号	2
3.1.2	减号	2
3.1.3	省略号	2
3.1.4	不能直接录入的标点符号	2
3.1.5	特殊字符	2
3.1.6	标点格式	3
3.1.7	带子	3
3.1.8	幻影	3
3.1.9	分段	3
3.1.10	西文句末	4
3.1.11	中英文间隔	4
3.2	字体	4
3.2.1	预定义命令的字体信息	4
3.2.2	字体坐标信息	4
3.2.3	字体处理	5
3.2.4	中文字体	5
3.2.5	更换字体	6
3.2.6	强调文字	6
3.2.7	字号	7
3.3	段落与环境	9
3.3.1	水平间距	9
3.3.2	盒子	11
3.3.3	正文段落	12
3.3.4	文本环境	13
3.3.5	列表环境	14
3.3.6	定理类环境	16
3.3.7	抄录环境	18
3.3.8	代码抄录	18
3.3.9	tabbing 环境	19

3.3.10	脚注与边注	19
3.3.11	垂直间距与垂直盒子	21
3.4	文档的结构层次	23
3.4.1	划分章节	23
3.4.2	多文件编译	24
3.4.3	定制章节格式	24
3.5	文档类与整体格式设置	25
3.5.1	页面格式	25
3.5.2	分栏控制	27
3.5.3	定义命令	27
3.5.4	定义环境	28
<b>4</b>	<b>自动化工具</b>	<b>29</b>
4.1	目录	29
4.1.1	目录和图表目录	29
4.1.2	控制目录内容	30
4.2	交叉引用	30
4.2.1	标签与引用	30
4.2.2	更多交叉引用	31
4.2.3	电子文档与超链接	31
4.3	BibTeX 与文献数据库	34
4.3.1	BibTeX 基础	34
4.3.2	用 natbib 定制文献格式	35
<b>5</b>	<b>玩转数学公式</b>	<b>36</b>
5.1	数学模式概说	36
5.2	数学结构	37
5.2.1	上下标	37
5.2.2	上下划线和花括号	39
5.2.3	分式	40
5.2.4	根式	42
5.3	符号与类型	43
5.3.1	字母表与普通符号	43
5.3.2	L <sup>A</sup> T <sub>E</sub> X 数学符号对应表	45
5.3.3	数学算子	47

<b>6 绘制图表</b>	<b>47</b>
6.1 L <sup>A</sup> T <sub>E</sub> X 中的表格 . . . . .	47
6.2 插图与变换 . . . . .	47
6.3 浮动体与标题控制 . . . . .	47
6.4 使用彩色 . . . . .	47
<b>参考文献</b>	<b>47</b>

## 1 勾股定理在古代

西方称勾股定理为毕达哥拉斯定理，将勾股定理的发现归功于公元前 6 世纪的毕达哥拉斯学派。该学派得到了一个法则，可以求出可排成直角三角形三边的三元数组。毕达哥拉斯学派没有书面著作，该定理的严格表述和证明则见于欧几里德<sup>1</sup>《几何原本》的命题 47：“直角三角形斜边上的正方形等于两直角边上的两个正方形之和。”证明是用面积做的。

我国《周髀算经》载商高（约公元前 2 世纪）1答周公问

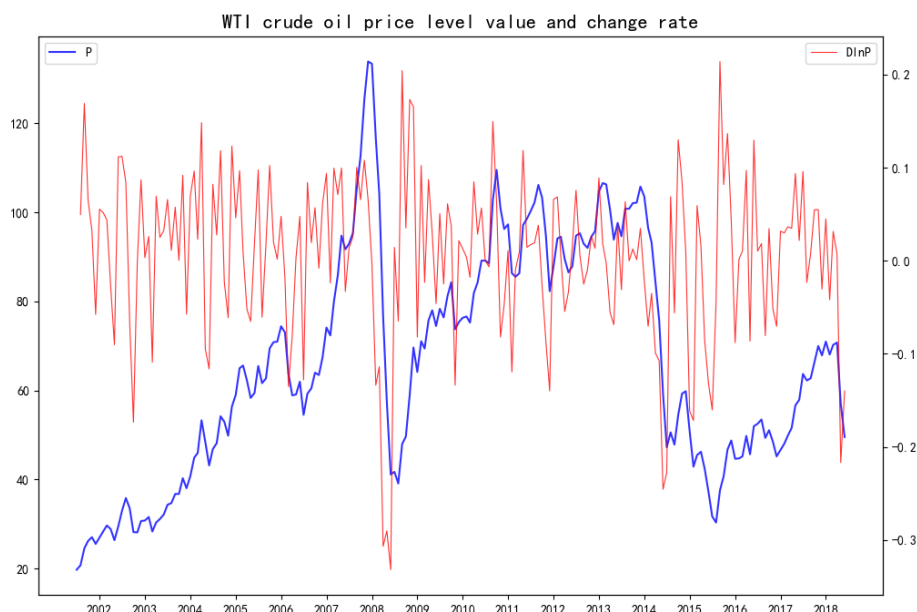


图 1 WTI crude oil price level value and change rate

下表列出了一些较小的勾股数Kumar et al. (2020):

直角边 $a$	直角边 $b$	斜边 $c$
3	4	5
5	12	13

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

$$a^2 + b^2 = c^2 \quad (1.1)$$

## 2 勾股定理的近代形式

引用的内容

<sup>1</sup>欧几里得，约公元前 330–275 年。

**定理 1 (勾股定理)** 直角三角形斜边的平方等于两腰的平方和。

可以用符号语言表述为(1.1):

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

## 3 组织文本

### 3.1 标点符号

#### 3.1.1 引号

“‘a’ and ‘b’”

#### 3.1.2 减号

An inter-word dash or, hyphen, as in X-ray.

An medium dash for number range, like 1–2, 2 ~ 3.

A punctuation dash—like this.

#### 3.1.3 省略号

She ... she got it

I've no idea ....

#### 3.1.4 不能直接录入的标点符号

# \$ % & { } \_ \~ ^ ~

#### 3.1.5 特殊字符

表 1: 特殊字符

重音命令			
ò \‘o	ó \’o	ô \^o	’o \”o
õ \~o	ō \=o	ó \.o	ö \u{o}
ǒ \v{o}	ǒ \H{o}	ôo \t{oo}	ô \r{o}
ô \b{o}	ô \d{o}		
特殊字母			
Å \AA	å \aa	Æ \AE	æ \ae

接下一页表格

(续表)

Œ \OE	œ \oe	ı \i	Ÿ \j
IJ \IJ	ij \ij	Ł \L	ł \l
Ø \O	ø \o	ß \ss	
特殊字符			
§ \S	† \dag	‡ \ddag	¶ \P
© \copyright	® \textregistered	™ \texttrademark	£ \pounds
• \textbullet	× \times	~ \sim	① \textcircled{1}
① \ding{\numexpr172}			

3.1.6 标点格式

全角式，所有标点全角，有挤压。例如，“标点挤压”。又如《标点符号用法》。

半角式，所有标点半角，有挤压。例如，“标点挤压”。又如《标点符号用法》。

开明式，部分的标点半角，有挤压。例如，“标点挤压”。又如《标点符号用法》。

行末半角式，仅行末挤压。例如，“标点挤压”。又如《标点符号用法》。

无格式，只有禁则，无挤压。例如，“标点挤压”。又如《标点符号用法》。

3.1.7 带子

Question 1

Donald E. Knuth

Mr. Knuth

function  $f(x)$

1, 2, and 3

3.1.8 幻影

幻影      快速隐形

幻影参数快速隐形

3.1.9 分段

这是一行文字

[] 另一行

这是一行文字

另一行

这                    是                    一                    行                    文                    字

另一行



### 3.1.10 西文句末

A sentence, And anotehr.

U.S.A. means United States Army?

Tinker et al. mad the double play.

Roman number XII. Yes.

### 3.1.11 中英文间隔

中文和 English 的混排效果并不依赖于 space 的有无。

个别时候需要忽略汉字和其他内容之间由 ‘xeCJK’ 产生的空格，这是可以把汉字放进一个盒子里：条目-a 不同于条目-b。

## 3.2 字体

### 3.2.1 预定义命令的字体信息

*Italic font test*

**Bond font test**

表 2: 预定义命令的字体信息

预定义命令的字体族			
字体族	带参数命令	声明命令	效果
罗马字体族（默认）	<code>\textrm&lt;text&gt;</code>	<code>\rmfamily</code>	Roman font family
无衬线字体族	<code>\textsf&lt;text&gt;</code>	<code>\sffamily</code>	Sans serif font family
打字机字体族	<code>\texttt&lt;text&gt;</code>	<code>\ttfamily</code>	Typewriter font family
预定义命令的字体形状			
字体形状	带参数命令	声明命令	效果（Latin Modern Roman 字体族）
直立（默认）	<code>\textup&lt;text&gt;</code>	<code>\upshape</code>	Upright shape
意大利	<code>\textit&lt;text&gt;</code>	<code>\itshape</code>	<i>Italic shape</i>
倾斜	<code>\textsl&lt;text&gt;</code>	<code>\slshape</code>	<i>Slanted shape</i>
小型大写	<code>\textsc&lt;text&gt;</code>	<code>\scshape</code>	SMALL CAPITALS SHAPE
预定义命令的字体系列			
字体系列	带参数命令	声明命令	效果
中等（默认）	<code>\textmd&lt;text&gt;</code>	<code>\mdseries</code>	Medium series
加宽加粗	<code>\textbf&lt;text&gt;</code>	<code>\bfseries</code>	<b>Bold extended series</b>

### 3.2.2 字体坐标信息

字体的族、形状和系列三种性质犹如确定字体的三维坐标，同一维度内的性质不能重叠，但不同类的性质则可以叠加，三种性质的组合效果见表 3

表 3: 字体坐标

字体族	\rmfamily		\sffamily		\ttfamily	
系列 形状	\mdseries	\bfseries	\mdseries	\bfseries	\mdseries	\bfseries
\upshape	Text	<b>Text</b>	Text	<b>Text</b>	Text	<b>Text</b>
\itshape	<i>Text</i>	<b><i>Text</i></b>	<i>Text</i> '	<b><i>Text</i></b> '	<i>Text</i>	<b><i>Text</i></b> '
\slshape	<i>Text</i>	<b><i>Text</i></b>	<i>Text</i>	<b><i>Text</i></b>	<i>Text</i>	<b><i>Text</i></b>
\scshape	TEXT	None	None	None	TEXT	None

**Note:** 表中以 Latin Modern 字体为例，这是 L<sup>A</sup>T<sub>E</sub>X 的字体默认值。表中实际共有 17 种不同的字体，如果使用其他的字体，表中的缺省值可能会有所不同。

### 3.2.3 字体处理

- 恢复默认字体

**This is a paragraph of bold and *italic font*, sometimes returning to normal font is necessary.**

- 斜体倾斜校正

MM MM MM

很少情况下，`\textit` 命令自动加入的倾斜校正时不必要的，此时可以使用 `\nocorr` 命令禁止校正

MM

### 3.2.4 中文字体

中文字体通常没有西文字体那样复杂的成套的变体，各个字体之间一般都是独立的。对于中文字体，一般只使用不同的字体族进行区分。xeCJK 和 CJK 宏包机制下，中文字体的选择命令和西文字体是份力的，选择中文字体用 `\CJKfamily` 命令，如：

这是黑体

这是楷书

在 `ctex` 宏包及文档类下有一些预定义，在默认情况下（`winfonts` 选项）针对 Windows 常用字体配置了的四种字体族：`zhsong`（宋体）、`zhhei`（黑体）、`zhkai`（楷书）、`zhfs`（仿宋）。为了方便使用，`ctex` 提供了简化的命令：

宋体 黑体 仿宋 楷书

`ctex` 宏包及文档类（如 ‘`ctexart`’）另外定义了一些组合字体，可以让中文也具备使用粗体（`\bfseries`）和意大利体（`\itshape`）的功能。默认中文字体族为 `rm`，其正常字体是宋体，粗体是黑体，意大利体是楷体。

### 3.2.5 更换字体

通过 fontspec 宏包的机制来调用字体，最基本的是设置正文罗马字体族、无衬线字体族和打字机字体族的命令：

```
\setmainfont[<可选选项>]{<字体名>}
```

```
\setsansfont[<可选选项>]{<字体名>}
```

```
\setmonofont[<可选选项>]{<字体名>}
```

此时 \rmfamily, \sffamily, \ttfamily 就分别对应设置的三种字体，而且 fontspec 会自动找到并匹配对应的粗体、斜体等编题，尽量使 \bfseries, \itshape 等命令也有效。

除此之外，还可以定义新的字体族命令：

```
\newfontfamily<命令>[<可选选项>]{<字体名>}
```

xeCJK 宏包（ctex 宏包或文档类会自动调用）提供了与 fontspec 对应的中文字体设置命令：

```
\setCJKmainfont[<可选选项>]{<字体名>}
```

```
\setCJKsansfont[<可选选项>]{<字体名>}
```

```
\setCJKmonofont[<可选选项>]{<字体名>}
```

```
\setCJKfamilyfont[<可选选项>]{<字体名>}
```

### 3.2.6 强调文字

#### (1) \emph and \em

命令形式：

You *should* use fonts carefully.

You should *use fonts carefully*.

声明形式：

This is *emphasized* text

#### (2) Redefine \Emph

有时仍然使用大写、小型大写或粗体进行更醒目的强调，此时可以定义一个新的命令：

This is **emphasize** text.

#### (3) \underline

Emphasized text and another.

`\underline` 的一个很大确定是下划线部分不能换行，并且下划线与文字的距离参差不齐。

#### (4) `\uline`

`\ulem` 宏包的 `\uline` 命令解决了上述 `\underline` 的命令，并把默认的 `\emph` 命令也改为使用下划线方式：

Emphasized text and another.

A very very very very very very very very very very very very very very very very very very long sentence.

`\ulem` 宏包还提供了其他一些修饰文字的命令：

urgent      boat      ~~wrong~~      ~~removed~~      dashing      dotty

`CJKfntef` 宏包对汉字也提供了类似的功能，同时也进行了一些扩充：

汉字，下点线                  汉字，单下划线                  汉字，双下划线  
 汉字，下划波浪线              汉字，删除线                  汉字，斜删除线

此外，`CJKfntef` 宏包，还提供了 `CJKfilltwosides` 环境，让汉字分散对齐：

汉 字 ， 分 散 对 齐

在 `ctex` 宏包及文档中，可以使用 `\CTEXunderline` 等以 `\CTEX` 开头的命令代替以 `\CJK` 命令开头的命令

### 3.2.7 字号

#### 1. 西文字体大小调整

基本的  $\text{\LaTeX}$  提供了 10 个简单的声明式命令调整文字的大小见表4。

表 4: 预定义调整西文字号

命令	效果	命令	效果
<code>\tiny</code>	tiny	<code>\large</code>	large
<code>\scriptsize</code>	script size	<code>\Large</code>	larger
<code>\footnotesize</code>	footnote size	<code>\LARGE</code>	even larger
<code>\small</code>	small	<code>\huge</code>	huge
<code>\normalsize</code>	normal size	<code>\Huge</code>	largest

#### 2. 中文字号大小

表 5: 预定义中文字号

命令	大小 (bp)	效果
<code>\zihao{0}</code>	42	初号
<code>\zihao{-0}</code>	36	小初号
<code>\zihao{1}</code>	26	一号
<code>\zihao{-1}</code>	24	小一号
<code>\zihao{2}</code>	22	二号
<code>\zihao{-2}</code>	18	小二号
<code>\zihao{3}</code>	16	三号
<code>\zihao{-3}</code>	15	小三号
<code>\zihao{4}</code>	14	四号
<code>\zihao{-4}</code>	12	小四号
<code>\zihao{5}</code>	10.5	五号
<code>\zihao{-5}</code>	9	小五号
<code>\zihao{6}</code>	7.5	六号
<code>\zihao{-6}</code>	6.5	小六号
<code>\zihao{7}</code>	5.5	七号
<code>\zihao{8}</code>	5	八号

### 3. 不容文档类选项下的字号命令

字号命令表示的具体尺寸随所使用的文档类和大小选项不同而不同 (见表 6)。在标准 L<sup>A</sup>T<sub>E</sub>X 文档类 `article`, `report` 和 `book` 中, 可以设置文档类选项 `10pt`, `11pt` 和 `12pt`, 全局地设置文档内的字号, 默认为 `10pt`, 即 `\normalsize` 的大小为 `10pt`。

表 6: 不同文档类选项下的字号命令

命令	10pt(En)	12pt	12pt	cs5size(ctex)	cs4size
<code>\tiny</code>	5	6	6	七	小六
<code>\scriptsize</code>	7	8	8	小六	六
<code>\footnotesize</code>	8	9	10	六	小五
<code>\small</code>	9	10	10.95	小五	五
<code>\normalsize</code>	10	10.95	12	五	小四

命令	10pt(En)	12pt	12pt	cs5size(ctex)	cs4size
<code>\large</code>	12	12	14.4	小四	小三
<code>\Large</code>	14.4	14.4	17.28	小三	小二
<code>\LARGE</code>	17.28	17.28	20.74	小二	二
<code>\huge</code>	20.74	20.74	24.88	二	小一
<code>\Huge</code>	24.88	24.88	24.88	一	一

Note: normal size 为 ctex 文档类默认字号

### 3.3 段落与环境

#### 3.3.1 水平间距

在  $\text{\LaTeX}$  中，可以使用的长度单位见表 7:

表 7: 可使用的长度单位

Symbol	Description
pt	point 点（欧美传统排版的长度单位，有时叫做“磅”）
pc	pica（1 pc = 12pt，相当于四号字体）
in	inch 英寸（1 in = 72.27 pt）
bp	big point 大点（在 PostScript 等其他电子排版领域的 point 都值大点，1 in = 72 bp）
cm	centimeter 厘米（2.54 cm = 1 in）
mm	millimeter 毫米（10 mm = 1 cm）
dd	didot point（欧洲大陆使用，1157 dd = 2338 pt）
cc	cicero（欧洲大陆使用，pica 的对应物，1 cc = 12 dd）
sp	scaled point（ $\text{\TeX}$ 中最小的长度单位，所有长度都是他的倍数，65536 sp = 1 pt）
em	全身（字号对应的长度，等于一个 <code>\quad</code> 的长度，也称为“全方”。本义是大写字母 M 的宽度）
ex	x-height（与字号相关，由字体定义。本意是小写字母 x 的高度）

正文中不同的水平间距见图 2:

在正文中可以使用下面的命令表示不可换行的水平间距：

<code>\thinspace</code> 或 <code>\,</code>	0.1667 em	$\rightarrow \leftarrow$
<code>\negthinspace</code>	-0.1667 em	$\leftarrow \rightarrow$
<code>\enspace</code>	0.5 em	$\rightarrow \leftarrow$
<code>\nobreakspace</code> 或 <code>~</code>	空格	$\rightarrow \leftarrow$

可以使用下面的命令表示可以换行的水平间距：

<code>\quad</code>	1 em	$\rightarrow \leftarrow$
<code>\qquad</code>	2 em	$\rightarrow \leftarrow$
<code>\enskip</code>	0.5 em	$\rightarrow \leftarrow$
<code>\_</code>	空格	$\rightarrow \leftarrow$

图 2 不同水平间距

使用水平间距的命令要注意适用，如 `\,` 是不可断行的，因而就不适用于分隔很长的内容，单用来代替逗号给长数字分段就很合适：1 234 567 890。

负距离 `\negthinspace` 则可以用来细调符号距离或拼接两个符号，如把两个“剑号”拼起来： $\ddagger$ versus  $\ddagger$ 。

- `hspace`

当上面的命令没有合适的距离是，还可以使用 `\hspace{distance}` 命令来产生指定的水平间距，该命令产生的距离是**可断行的**，如：`Space` 1 cm

`\hspace` 的作用是分隔左右的内容，在某些只有一边内容的地方（如强制断行的行首）， $\text{\LaTeX}$  会忽略产生的距离，此时可以用带星号的命令 `\hspace*distance` 阻止命令被忽略，如：

```
Space
1 cm
Space
1 cm
```

`\hspace` 可以产生随内容可伸缩的长度，即橡皮长度，这样才能保证在分行行末的对齐，语法为：

`<normal length>plus< 可伸长长度 >minus< 可缩短长度 >`

```
longggggggggggg    longggggggggggg    longggggggggggg    longggggggggggg
longggggggggggg    longggggggggggg    longggggggggggg    longggggggggggg
```

有一种特殊的橡皮长度 `\fill`，`\fill` 可以从零开始无限延伸，此时橡皮长度就真的像一个弹簧，可以用来把几个内容均匀排列在一行之中：

```
left                middle                left
```

left middle ..... eft  
left middle \_\_\_\_\_ left

`\hfill` 是命令 `\hspace{\fill}` 的简写，还可以用 `\stretcg<times>` 产生具有指定“弹力”的橡皮长度，如 `\stretch{2}` 相当于两倍的 `\fill`。

`\hrulefill` 和 `\dotfill` 与 `\hfill` 功能类似，只是中间填充内容不一样（横线和点线）

- 自定义长度变量 (`setlength`)

L<sup>A</sup>T<sub>E</sub>X 预定义了一些长度变量控制排版的参数，可以通过 `\setlength` 命令来设置，如段首缩进：

```
\setlength\parindent{8em}
```

注：导言处定义

- 长度累加 (`addtolength`)

还可以通过 `\addtolength` 命令在长度变量上做累加，如：

```
Para
  Para
    Para
```

### 3.3.2 盒子

盒子 (box) 处理 T<sub>E</sub>X 中的基本处理单位，一个字符、一行文字、一个页面、一张表格在 T<sub>E</sub>X 中都是一个盒子。

- `mbox`

最简单的命令是 `\box<text>`，它产生一个内容以左右模式排列的盒子，可以用它表示不允许断行的内容，如果不在行末，其与普通内容无异，如：

```
\mbox{cannot be broken} 表示 cannot be broken
```

- `makebox`

`\makebox` 与 `\mbox` 类似，但可以带两个可选参数，指定盒子的宽度和对齐方式 (`c`(center, default), `l`(left), `r`(right), `s`(scatter))：

```
\makebox[<width>][<loc>]{<context>}
```

还可以使用 `\makebox` 产生宽度为 0 的盒子，产生重叠 (overlap) 的效果：

文字



- overlap

LaTeX 已经提供了两个命令来专门生成重叠的效果，即 `\llap` 和 `\rlap`，分别表示把参数中的内容向当前位置的左侧和右侧重叠：

语言文字

语言文字

- frame

命令 `\fbox` 和 `\framebox` 产生带边框的盒子，语法与 `\mbox` 和 `\makebox` 类似：

`\fbox`      `\framebox`      `\box`

边框与内容的距离由长度变量 `\fboxsep` 控制（默认为 3pt），边框线的粗细则由长度变量 `\fboxrule` 控制（默认为 0.4 pt）。

`\tight`      `\loose`

在 `\makebox`、`\framebox` 等盒子命令的参数中，可以使用 `\width`、`\height`、`\depth`、`\totalheight` 来分别表示盒子内容的自然宽度、深度、以及高度和深度之和。如下产生的盒子总宽度恰好是文字自然宽度的 3 倍：

带边框

### 3.3.3 正文段落

- 分段和缩进

LaTeX 使用空行表示分段，在自定义命令中，也常用 `\par` 命令分段。每个自然段在第一行有一个固定的缩进，可以由长度变量 `\parindent` 控制；可在段前用 `\noindent` 命令临时禁止使用缩进；也可用 `\indent` 命令产生一个长为 `\parindent` 的缩进。

- 段间距

除了段落首行缩进，还有一个重要参数是段与段之间的垂直距离，由变量 `\parskip` 控制，默认值是橡皮长度 0pt plus 1 pt，可用

`\setlength\textbackslash parskip{0pt}`

把段间距定义为固定长度，禁止段落间距离伸长。

- 对齐方式

段落默认两端均匀对其，可用 `\raggedright` 命令（右边参差不齐）、`\raggedleft` 命令、`\centering` 命令选择左对齐、右对齐和居中对齐。这三个环境会在段落前后增加一小段垂直距离表示强调。

L<sup>A</sup>T<sub>E</sub>X 还提供了三个环境来排版不同对齐方式的文字：`flushleft`（左对齐）、`flushright`（右对齐）、`center`（居中）。

- 断字和形状

见 L<sup>A</sup>T<sub>E</sub>X 入门 P<sub>93-95</sub>

### 3.3.4 文本环境

- `quote` 和 `quotation` 环境

`quote` 环境在段前没有首行缩进，左右边距比正文大，适合小段内容引用，如：

学而时习之，不亦说乎？有朋自远方来，不亦乐乎？

而 `quotation` 则在每段首行缩进，适合多段的文字引用，如：

学而时习之，不亦说乎？有朋自远方来，不亦乐乎？人不知而不愠，不亦君子乎？

默而识之，学而不厌，诲人不倦，何有于我哉？

- `verse` 环境

段内使用 `\\` 换行，  
分段仍用空行。

过长的行会在折行时悬挂缩进，就像这一段内容。学而时习之，不亦说乎？有朋自远方来，不亦乐乎？人不知而不愠，不亦君子乎？

- `abstract` 环境

摘要环境时 `article` 和 `report` 文档类（包括中文的 `ctexart` 和 `ctexrep`）定义的，它产生一个类似 `quotation` 的小号字环境，并添加标题：

#### 摘要

本文介绍 L<sup>A</sup>T<sub>E</sub>X 的使用。

摘要的标题由 `\abstractname` 定义，英文默认是 “Abstract”，中文默认是 “摘要”，可通过重定义 `\abstractname` 来设置，如：

```
\CTEXoptions[abstractname={摘\quad 要}]
```

### 3.3.5 列表环境

列表是常用的文本格式， $\text{\LaTeX}$  标准文档提供了三种列表环境，编号的 `enumerate`，不编号的 `itemize` 和使用关键字的 `description` 环境。

- 基本列表环境

- `enumerate` 环境

1. 中文
2. English
3. Franch

- `itemize` 环境

- `description` 环境

**中文** 中国的语言文字

**English** The language of England

**Franch** Language of France

上述三种列表环境可以嵌套使用（最多四层）， $\text{\LaTeX}$  会自动处理不同层次的缩进和编号，如 `enumerate` 环境：

1. 中文

- (a) 古代汉语

- (b) 现代汉语

- i. 口语

- A. 普通话

- B. 方言

- ii. 书面语

2. English

3. Franch

- `item` 可选参数所有条目都以 `\item` 命令开头，`\item` 命令的可选参数可以为 `enumerate` 或 `itemize` 环境临时手动设置编号或标志符号，如

1. Chinese

- 1a. English

2. Franch

以及:

† Chinese

– English

– Franch

- 计数器与编号

enumerate 环境的编号由计数器 (counter) 控制, 每遇到一个没有可选参数的 `\item` 命令, 计数器值加 1, 4 个不同嵌套层次的 enumerate 环境使用不同的计数器, 分别是 `enumi`, `enumii`, `enumiii` 和 `enumiv`。L<sup>A</sup>T<sub>E</sub>X 的计数器都有一个对应的命令 `\the` 计数器名, 用来输出计数器的值。同时还定义了 `\labelnumi` 来输出条目标签, 一般是在编号后加一个标点。

1. This is No. 1 and label is 1.(there is no punctuation)

2. This is No. 2 and label is 2.(there is no punctuation)

计数器的值可以使用命令 `\arabic`, `\roman`, `\Roman`, `\alph`, `\Alph`, `\Fnsymbol` 带上计数器参数输出, 分别表示阿拉伯数字、小大写罗马数字、西澳大学字母、特殊符号, 见表8

表 8: enumerate 环境的编号和标签

嵌套层数	计数器	计数器输出		条目标签	
		命令	默认值	命令	默认值
1	<code>enumi</code>	<code>\theenumi</code>	<code>\arabic{enumi}</code>	<code>\labelenumi</code>	<code>\theenumi.</code>
2	<code>enumii</code>	<code>\theenumii</code>	<code>\alph{enumi}</code>	<code>\labelenumii</code>	<code>\(theenumi)</code>
3	<code>enumiii</code>	<code>\theenumiii</code>	<code>\roman{enumi}</code>	<code>\labelenumiii</code>	<code>\theenumiii.</code>
4	<code>enumiv</code>	<code>\theenumiv</code>	<code>\Alph{enumi}</code>	<code>\labelenumiv</code>	<code>\theenumiv.</code>

具体情况如下:

1. 编号 1, i, I, a, A, \*

2. 编号 2, ii, II, b, B, †

3. 编号 3, iii, III, c, C, ‡

可以通过重定义 `\theenumi` (默认为 `\arabic{enumi}`) 和 `\labelenumi` (默认为 `\theenumi.`) 等命令, 控制 enumerate 环境的编号:

- 定制列表环境

itemize 环境也有命令控制标签，见表9

表 9: itemize 环境的标签

嵌套层次	命令	默认值	效果
1	<code>\labelitemi</code>	<code>\textbullet</code>	•
2	<code>\labelitemii</code>	<code>\normalfont\bfseries\textendash</code>	—
3	<code>\labelitemiii</code>	<code>\textasteriskcentered</code>	*
4	<code>\labelitemiv</code>	<code>\textperiodcenterd</code>	·

description 环境则较简单，使用`\descriptionlabel` 命令来控制标签输出格式。

#1 Chinese

#2 English

可以通过使用 `enumitem` 宏包在 `enumerate` 等环境后加可选参数来实现对各种列表环境的标签、尺寸进行定制，如：

(1) Chinese

(2) English

(3) Franch

其中标签的参数 `label` 用星号 `*` 表示计数器。对于非标准计数器输出（如`\chinese`）还要进行单独设置，如

一、内容清晰

二、格式美观

### 3.3.6 定理类环境

- 概述

定理类环境是 L<sup>A</sup>T<sub>E</sub>X 总一类重要的文本环境，可以产生一个标题、编号和特定格式的文本，如：

**定理 2** 直交三角形斜边的平方等于两腰的平方和。

- `\newtheorem` 声明新的定理类环境

命令 `\newtheorem` 用来声明一个新的定理类环境，两个参数分别是定理类环境名和定理输出的标题名：

`\newtheorem{thm}{定理}`

即 `thm` 环境，效果是输出标题为“定理”的一段文字，且该环境允许带有可选参数表示定理的小标题，如：

**定理 3 (勾股定理)** 直角三角形斜边的平方等于两腰的平方和。

`\newtheorem` 命令可以在最后使用一个可选的计数器参数，如：

`\newtheorem{lemma}{定理}[section]`

用来表示定理编号是 `<Ctr>` 的下一级编号，通常用来让定理按章节编号：

**定理 3.1** 偏序集可良序化。

**定理 3.2** 实数集不可数。

也可以在两个参数之间使用可选的计数器参数，用来让几个不同（样式、字体等不同）的定理类环境使用相同的编号，如：

**命题 4** 直角三角形的斜边大于直角边。

**命题 5** 直角三角形的斜边大于直角边。

**命题 6** 直角三角形的斜边大于直角边。

- 自定义定理类环境

默认的定理类环境的格式是固定的，`theorem` 宏包扩展了定理类环境的格式，可以修改定理类环境的格式。它提供了 `\theoremstyle{< 格式 >}` 命令来选择定理类环境点格式，可用的预定义格式有：

<code>plain</code>	默认格式；
<code>break</code>	定理头换行；
<code>marginbreak</code>	编号在页边，定理头换行；
<code>changebreak</code>	定理头编号在前文字在后，换行；
<code>change</code>	定理头编号在前文字在后，不换行；
<code>margin</code>	编号在页边，定理头不换行。

### 3.3.7 抄录环境

- `\verb` 命令

`\verb` 命令可用来表示行文中的抄录，其语法格式如下：

`\verb<符号><抄录内容><符号>`

注意其实的符号和末尾的符号相同，两个符号之间的部分将逐字原样输出。

- `\verb*` 命令

使用带星号的命令 `\verb*` 则可以将空格输出为可见的，如：

显示空格：1 2 3 4

- `verbatim` 环境

```
#!/usr/bin/env perl
$name = "guy";
print "Hello, $name!\n"
```

- `verbatim*` 环境（空格可见）

```
#include <studio.h>
main()
{
    printf("hello world.\n");
}
```

### 3.3.8 代码抄录

可以使用 `verbatim` 环境排版程序代码，但若想在程序代码中增加语法高亮功能，那么 `verbatim` 环境就捉襟见肘了，可以使用 `listings` 宏包进行语法高亮排版。

```
1      /* hello.c */
2      #include <stdio.h>
3      main()
4      {
5          print ("Hello.\n")
6      }
```

要想得到漂亮的排版效果，必须仔细设置 `lstlisting` 环境的格式。可以使用 `lstlisting` 环境的可选参数设置格式，也可以使用 `\lstset{< 设置 >}` 进行全局设置。

`listings` 宏包还提供了 `\verb` 命令的对应物 `\lstinline`，这样可以直接在行文段落中使用语法高亮的代码：

语句 `typedef char byte`

`listings` 宏包对汉字等非 ASCII 字符的支持不是很好，一般需要对汉字进行转移字符进行转义处理（`xeCJK` 从 v3.2.2 版本后不需要再做转义），`listings` 将转义字符之间的内容当作普通  $\text{\LaTeX}$  代码处理，所以也可对数学公式进行转义处理，如：

```
int n; // 一个整数
int m; // 一个整数
double x = 1/sin(x); //  $\frac{1}{\sin x}$ 
```

### 3.3.9 tabbing 环境

`tabbing` 环境用来排版制表位，即让不同的行在指定的地方对齐。行与行之间用 `\\` 分隔，使用 `\=` 命令来设置指标为，`\>` 命令跳到下一个已设置的制表位，该环境适合做比较简单的无线表格：

格式	作者
Plain $\text{\TeX}$	高德纳
$\text{\LaTeX}$	Leslie Lamport

`\kill` 命令与 `\\` 类似，但它会忽略这一行的内容，只保留制表位的设置：

Plain $\text{\TeX}$	高德纳
$\text{\LaTeX}$	Leslie Lamport

### 3.3.10 脚注与边注

#### • 概述

$\text{\LaTeX}$  使用 `\footnote{<footnote content>}` 产生脚注，例如<sup>2</sup>。`\footnote` 可以带一个可选参数，表示手工数字编号，如 `\footnote[2]{又是 2?}` 的效果<sup>2</sup>，它不改变原来的脚注编号<sup>3</sup>。

---

<sup>2</sup>脚注默认以阿拉伯数字的上标为编号，内容出现在页面底部，以 `\footnotesize` 的字号输出。

<sup>2</sup>又是 2?

<sup>3</sup>继续之前的编号



- minipage 中的脚注

脚注的使用有一些限制，通常只能用在一般的正文段落中，不能用于表格、左右盒子（如 `\mbox` 的参数中）等受限的场合。但可以把脚注用在 `minipage` 环境中（见 3.3.11 节）。此时脚注出现在 `minipage` 环境产生的盒子底部，并使用局部编号（默认按字母编号）：

This is a footnote in minipage environment<sup>a</sup>.

---

<sup>a</sup>footnote content.

This is a footnote in minipage environment<sup>a</sup>.

---

<sup>a</sup>footnote content.

- 计数器

脚注使用的计数器是 `footnote`，`minipage` 环境中则使用 `mpfootnote` 计数器，可以通过修改 `\thefootnote` 和 `\thempfootnote` 改变编号的格式，常见的种是：

```
\renewcommand\thefootnote{\fnsymbol{footnote}}
```

```
\renewcommand\thefootnote{\textcircled{\arabic{footnote}}}
```

得到的效果分别是<sup>§</sup>，以及带圈的数字脚注<sup>⑤</sup>。

使用 `pifont` 宏包提供的带圈数字符号效果要比 `\textcircled` 效果更好，使用 `pifont` 宏包的 `\ding` 命令可以输出符号表中的符号，如：

```
\usepackage{pifont} % preamble
```

```
\renewcommand\thefootnote{\ding{\numexpr171+\value{footnote}}}
```

此时得到的脚注是<sup>⑥</sup>。

在不能使用脚注的位置，可以使用命令 `\footnotemark` 和 `\footnotetext` 分开输入脚注的编号和内容。

自变量 <sup>7</sup>	因变量 <sup>8</sup>
$x$	$y$

- 边注

$\text{\LaTeX}$  还提供了边注的命令 `\marginpar{<content>}`，用来给文档添加页边的旁注。边注的使用与脚注差不多，只是边注不编号，内容出现的位置与正文更接近。单页模式 `onecolumn` 下，边注在页面的右侧，双面模式 `twocolumn` 下，边注在页面的外侧（奇数页在右，偶数页在左）。

---

<sup>§</sup>符号编号的脚注

<sup>⑤</sup>带圈数字编号脚注

<sup>⑥</sup>更好的带圈数字脚注

<sup>7</sup> $x = \sqrt{y}$ 。

<sup>8</sup> $y = x^2$ 。

### 3.3.11 垂直间距与垂直盒子

- 水平间距与垂直间距

`\hspace` 和 `\hspace*` 生成水平间距, `\vspace{<width>}` 和 `\vspace*{<width>}` 生成垂直距离, 垂直距离也是弹性距离, 可以用 `\vfill` 表示 `\vspace{\fill}`。

- 子段盒子: `\parbox` 和 `minipage`

在 3.3.2 节介绍的是水平盒子, 文字不能在其中分行分段, 使用 `\parbox` 命令或 `minipage` 环境生成的垂直盒子没有这种限制, 垂直盒子在  $\text{\LaTeX}$  中也称为子段盒子 (`parbox`)。格式如下:

```
\parbox[<位置>][<高度>][<内容位置>]{<width>}{<box text>}

\begin{minipage}[position]{width}
    <盒子内容>
\end{minipage}
```

`\parbox` 和 `minipage` 环境必须带一个宽度参数, 表示内容的宽度, 超出宽度的内容会自动换行:

	不搭		不搭
前言	不搭。	前言不搭。	前言后语。
	后语	后语	

`\parbox` 和 `minipage` 都可以带三个可选参数, 分别表示盒子的基线位置、盒子的高度以及盒子内容在盒子内的位置。**位置参数**可以使用 `c`、`t`、`b`, 默认为 `c` (居中); **内容位置参数**可以使用 `c`、`t`、`b`、`s` (垂直分散对齐)。

可以在 `verbatim` 环境外套一层 `minipage`, 将整个抄录环境的内容看作一个整体:

```
#!/bin/sh
cat ~/${file}
```

```
#!/bin/sh
cat ~/${file}
```

- 标尺 (`rule`) 盒子一个标尺盒子就是一个实心的举行盒子, 不过通常只是使用一个西昌的标尺盒子化纤, 这正是“表示”一词的由来, 其命令为:

```
\rule[<升高距离>][<宽度>]{<高度>}
```

这里的参数 `< 升高距离 >` 默认为 `0pt`, 效果就像把盒子做了位移一样, 如:

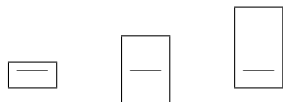
| Middle |

Left ————— Right

■ 也可以用作证毕符号

- 支架 `\strut`

类似于幻影 `\phantom`， $\text{\LaTeX}$  有一种宽或高为零的盒子，专门用来占位，称为“支架”（`\strut`），其占当前字号大小的高度和深度；有时也可用标尺盒子来表示支架，如：



- `\raisebox` 升降的水平盒子

可以用 `\raisebox` 造成升降的（水平）盒子：

```
\raise{<距离>}[<高度>][<深度>]{<内容>}
```

其中距离为正使盒子里面的内容上升，距离为负使下降，例如可以自己定义  $\text{\TeX}$  的标志：

$\text{\TeX}$

- `varwidth` 环境

`\parbox` 命令和 `minipage` 环境产生的垂直盒子必须实现确定宽度，但有时需要盒子的内容能手动分行，却又希望它保持按手工分行的宽度，这是可以使用 `varwidth` 宏包提供的 `varwidth` 环境代替 `minipage` 环境。

```
\usepackage{varwidth} % preamble
\begin{quote}[position]{max width}
    <盒子内容>
\end{minipage}
```

`varwidth` 对应 `minipage` 的宽度参数，表示盒子的最大宽度，两种环境的区别如下：

自然 宽度	自然 宽度
----------	----------

- $\text{\TeX}$  标志

表 10 给出了一些常见标志在  $\text{\LaTeX}$  中的命令：

表 10:  $\text{\TeX}$  及部分相关软件所使用的标志

标志	命令	额外的宏包
$\text{\TeX}$	<code>\TeX</code>	

标志	命令	额外的宏包
$\LaTeX$	<code>\LaTeX</code>	
$\LaTeX 2_{\epsilon}$	<code>\LaTeXe</code>	
METAFONT	<code>\MF</code>	mflogo
METAPOST	<code>\MP</code>	mflogo
$\mathcal{A}\mathcal{M}\mathcal{S}$	<code>\AmS</code>	amsmath
$\XeTeX$	<code>\XeTeX</code>	metalogo 或 hologo
$\XeLaTeX$	<code>\XeLaTeX</code>	metalogo 或 hologo
$\BibTeX$	<code>\BibTeX</code>	doc 或 hologo

## 3.4 文档的结构层次

### 3.4.1 划分章节

- 章节层次

$\LaTeX$  的标准文档类可以划分多层章节，如表 11 所示：

表 11: 章节层次

层次	名称	命令	说明
-1	part（部分）	<code>\part</code>	可选的最高层次
0	chapter（章）	<code>\chapter</code>	report, book 或 ctexrep 文档类的最高层
1	section（节）	<code>\section</code>	article 或 ctexart 类最高层
2	subsection （小节）	<code>\subsection</code>	
3	subsubsection （小小节）	<code>\subsubsection</code>	report, book 或 ctexrep, ctex-book 类，默认不编号、不编目录
4	paragraph （段）	<code>\paragraph</code>	默认不编号、不编目录
5	subparagraph （小段）	<code>\subparagraph</code>	默认不编号、不编目录

文档的最高层章节可以使 `\part`，也可以直接用 `\chapter`（book 和 report 等）或 `\sectionn`（article 等）；除 `\part` 外，只有上一层存在时才能使用下一层，否则编号会出现错误。可以使用带星号的章节命令（如 `\chapter*`）表示不编号、不编目的章节。

- 长标题

有时希望在正文中的章节题目与目录、页眉中的不同——正文使用长标题，目录和页眉使用短标题。可以通过给章节命令调价可选参数来实现：

```
\chapter[short title]{long title}
\chapter[展望与未来]{展望与未来：畅想新时代的计算机排版软件}
```

### 3.4.2 多文件编译

L<sup>A</sup>T<sub>E</sub>X 提供 `\included{<filename>}` 命令导入另一个文件作为一个章节，文件名不用带后缀名。

使用 `\include` 命令可以把不要的章节注释掉来达到目的，但更好的办法是在导言区使用 `\includeonly{<filelist>}` 命令，其中 `filelist` 用英文逗号相隔。

### 3.4.3 定制章节格式

L<sup>A</sup>T<sub>E</sub>X 标准文档类 `book`, `report`, `article` 的章节格式是固定的，章节标题用什么字体、字号、前后间距、编号方式等都是预定好的，可以通过 `ctex` 宏包进行定制。`ctex` 宏包提供了 `\CTEXsetup` 命令来设置章节标题的格式，语法如下：

```
\CTEXsetup[<opt1>=<value1>, <opt2>=<value2>, ...]{<obj>}
```

其中 `<obj>` 可以是 `part`, `chapter`, `section`, `subsection`, 素不素吧 `section`, `paragraph`, `subparagraph`; `<opt1>` 和 `<opt2>` 等是设置的选项，包括 `name`, `number`, `format`, `nameformat`, `numberformat`, `titleformat`, `aftername`, `beforekip`, `afterskip`, `indent`; 而 `<value1>` 和 `<value2>` 等式对应选项的设置内容。

- `name={<前名>,<后名>}`

用来设置章节的名字，包括章节编号签好和后面的词语，前后两个名字之间用西文逗号分开，如：

```
\CTEXsetup[name={第,节}]{section}
```

将设置 `\section` 的标题名字类似“第 1 节”。

- `number={<\textit{编号格式}>}`

相当于设置章节的计数器的 `\the<counter>` 命令，例如：

```
\CTEXsetup[<number={\chinese[section]}]{section}
```

如果配合前面章节名的设置，将产生类似“第一节”的效果。

- `format={<格式>}`

用于控制章节标题的全局格式，作用域为章节名字和随后的标题内容。常用于控制章节标题的对齐形式，如设置节标题左对齐，整体粗体：

```
=\CTEXsetup[format={\raggedright\bfseries}]{section}
```

- `nameformat={<格式>}`

类似 `format`，控制**章节名和编号**的格式，不包括后面的**章节标题**。

- `numberformat={<格式>}`

类似`format`，仅控制**编号**的格式，不包括前后**章节名和章节标题**。

- `titleformat={<格式>}`

类似`format`，进控制**章节标题**的格式，不包括前面的**章节名和编号**。

- `aftername={<代码>}`

控制章节名和编号（如“第一章”）与后面标题字间的内容。通常设置为一定间距的空格，或换行（换行仅对 `chapter` 和 `part` 有效），如：

```
\CTEXsetup[aftername={\\vspace{2ex}}]{part}
```

- `beforeskip={<长度>}`

控制章节标题前的垂直距离。

- `afterskip={<长度>}`

控制章节标题后的垂直距离。

- `indent={<长度>}`

控制章节标题的缩进长度。

## 3.5 文档类与整体格式设置

### 3.5.1 页面格式

- `\pagenumbering` 页码控制

页码的计数器是 `page`，它会随着文档自动计数。`LaTeX` 提供了一个简单的命令 `\pagenumbering{<格式>}` 来设置页码的编号方式，这个命令会令页码重新从 1 开始按 `<格式>` 参数计数，如：

```
\pagenumbering{roman}
\pagenumbering{arabic}
```

相当于设置 `\thepage` 为 `\roman{page}`，并设置计数器值为 1。

- `\pagestyle` 页面风格

L<sup>A</sup>T<sub>E</sub>X 提供了多种预定义的页面风格（page style），它们控制页眉页脚的整体风格设置：

<code>empty</code>	没有页眉页脚；
<code>plain</code>	没有页眉，页脚是居中的页码；
<code>headings</code>	没有页脚，页眉是章节名称和页码；
<code>myheadings</code>	没有页脚，页眉是页码和用户自定义的内容。

可以用 `\pagestyle{< 风格 >}` 整体设置页面风格，也可以使用 `\thispagestyle{< 风格 >}` 单独设置当前页的风格。如：

```
\pagestyle{headings}
\thispagestyle{headings}
```

也可以在插入大幅图片的页面使用 `plain` 风格，取消负载的页眉：

```
\begin{figure}[p]
  \thispagestyle{plain}
  .....
\end{figure}
```

- 页眉页脚线

`fancyhdr` 宏包提供了新的页面风格 `fancy`，作为标准 L<sup>A</sup>T<sub>E</sub>X 的 `myheadings` 及 `\markboth`，`\markright` 机制的扩展，具体见《L<sup>A</sup>T<sub>E</sub>X 入门》P<sub>147-149</sub>。

`fancy` 页面风格还会给页眉和页脚加一条横线。可以重定义宏 `\headrulewidth` 和 `\footrulewidth` 来修改页眉页脚线的宽度，宽度为零则表示无，注意它们只是文本宏而不是长度变量，如：

```
\usepackage{fancyhdr} % preamble
\renewcommand\headrulewidth{0.6pt} % 默认为 0.4pt
\renewcommand\footrulewidth{0.6pt} % 默认为 0pt
```

- 重定义页面风格

fancyhdr 宏包还提供了 `\fancypagestyle` 命令重定义原油的页面风格，通常可以用它来重定义 plain 风格，如：

```
\fancypagestyle{plain}{
  \fancyhf{}
  \cfoot{--\textit{\thepage}--} % 改变页码形状
  \renewcommand\headrulewidth{0pt} % 无页眉线
  \renewcommand\footrulewidth{0pt} % 无页脚线
}
```

### 3.5.2 分栏控制

见《L<sup>A</sup>T<sub>E</sub>X 入门》页 P<sub>149-151</sub>。

### 3.5.3 定义命令

- newcommand 命令

可以使用 `\newcommand` 来新定义一个命令，其语法格式如下：

```
\newcommand<command>[<参数个数>][<首参数默认值>]{<具体定义>}
```

- 无参数 newcommand

最简单的用法就是定义没有参数的命令（此时不需要 `< 参数个数 >` 和 `< 首参数默认值 >`），其功能就是简单的字符替换，如：

```
\newcommand\PRC{People's Republic of \emph{China}}
```

定义了一个 `\PRC` 命令，用来替换内容。注意在定义宏时，`< 具体定义 >` 的外面必须要有花括号界定。

- 带参 newcommand

如果定义了参数的个数（从 1 到 9），则可以在宏的定义中使用参数。在 `< 具体定义 >` 中，宏的参数一次编号，用 `#1`, `#2`, ..., `#9` 表示，使用宏时它们会替换为参数的字符串，例如：

```
\newcommand\loves[2]{#1喜欢#2}
\newcommand\hatedby[2]{#2不受#1喜欢}
```



此时, `\loves{猫儿}{鱼}` 表示猫儿喜欢鱼

如果在指定参数个数的同时还指定了首个参数的默认值, 那么这个命令的第一个参数就成为可选参数 (使用中括号界定)。如:

```
\newcommand\loves[3][喜欢]{\#2\#1\#3}
```

此时 `\loves{猫儿}{鱼}` 的作用与前面无异, 还是表示猫儿喜欢鱼; 同时也可以使用可选参数覆盖默认值, 如: `\loves[最爱]{猫儿}{鱼}` 表示猫儿最爱鱼

- 重定义 `renewcommand` 命令

重定义一个 `LaTeX` 命令可以使用:

```
\renewcommand<命令>[<参数个数>][<首选参数默认值>]{<具体定义>}
```

它的用法与 `\newcommand` 完全相同, 只是 `\renewcommand` 用于改变已有命令的定义, 而 `\newcommand` 只能用于定义新命令。如重定义摘要名:

```
\renewcommand\abstractname{内容简介}
```

- `\providecommand` 命令

`\providecommand` 的语法与 `\newcommand`、`\renewcommand` 相同, 也可用来定义命令。不同的时, 它检查命令是否已经定义后, 对已定义的命令不会报错, 也不重定义, 而是保留旧定义, 忽略相信定义。这种方式可以用来保证一个命令的存在性。如:

```
\providecommand\url[1]{\texttt{#1}}
```

这样就可以在后面放心使用 `\url` 命令, 保证命令可用。`\providecommand` 也可以与 `\renewcommand` 连用, 表示无论之前是否定义过这个命令, 都改用 `\renewcommand` 的新定义。

### 3.5.4 定义环境

定义环境与定义命令方式类似, 分别使用如下命令:

```
\newenvironment{环境名}[参数个数][<首参数默认值>]
  {<环境前定义>}{<环境后定义>}
```

```
\renewenvironment{环境名}[参数个数][<首参数默认值>]
  {<环境前定义>}{<环境后定义>}
```

例如, `book` 类中没有显示摘要的 `abstract` 环境, 故可以仿照 `article` 类中的格式利用 `quotation` 环境定义一个:

```

\newenvironment{myabstract}[1][摘要] % 默认标题为摘要值
{
  \small
  \begin{center}\bfseries #1\end{center}
  \begin{quotation}} % 环境前定义
\end{quotation}} % 环境后定义

```

需要注意的时，在定义有参数的环境时，只有 < 环境前定义 > 中可以使用参数，在 < 环境后定义 > 中不能使用环境参数。如果需要在环境后使用环境参数，可以先把前面得到的参数保存在一个命令中，在后面使用，如：

```

\newenvironment{Quotation}[1]
{
  \newcommand\quotesource{#1} % 保存参数
  \begin{quotation}} % 环境前定义
{
  \par\hfill —— 《\textit{\quotesource}》
  \end{quotation}} % 环境后定义
\begin{Quotation}{易\textbullet 乾}
  初九，潜龙勿用。
\end{Quotation}

```

得到效果如下：

初九，潜龙勿用。

—— 《易 • 乾》

## 4 自动化工具

### 4.1 目录

#### 4.1.1 目录和图表目录

目录时最基本的自动化工具。L<sup>A</sup>T<sub>E</sub>X 会自动收集章节命令所定义的各章节标题，用 `\tableofcontents` 命令输出。

类似的，命令 `\listoffigures` 和 `\listoftables` 则会收集 figure 环境和 table 环境中 `\caption` 命令的图表标题，产生图表的目录，其格式与章节目录类似。

### 4.1.2 控制目录内容

除 `\part` 外，标准文档类默认输出 3 级目录：article 是 section、subsection、subsubsection；而 report 和 book 是 chapter、section、subsection 3 级。可以通过修改计数器 `tocdepth` 的值来控制输出目录的深度。`tocdepth` 通常与 `secnumdepth` 一起修改，如：

```
\setcounter{secnumdepth}{4} % 增加编号深度
\setcounter{tocdepth}{4} % 增加目录深度
```

## 4.2 交叉引用

交叉引用（cross reference）是 L<sup>A</sup>T<sub>E</sub>X 中另一个常用的自动化工具，它可以通过一个符号标签引用文档中某个对象的编号、页码或标题等信息，而步滨知道这个对象具体在什么地方。如，这一章的标题在 29 页，编号第 4 章，标题是“自动化工具”，输出这段话使用的代码是：

```
这一章的标题在 ~\pageref{sec:autotool} 页，编号第 ~\ref{sec:autotool} 章，
标题是 “\nameref{sec:autotool}” ……
```

### 4.2.1 标签与引用

交叉引用的使用可以分为两个部分：定义标签和引用标签。

针对定义标签，如果对象是用一条命令产生的（如 `\section`，`\item`），标签通常直接放在命令的后面；如果对象是由环境产生的（如 `table`，`equation` 环境），则放在环境里面。不过也有例外：脚注的标签应该放在脚注内容里面，即 `\footnote{\label{fn:foot}}`；多行编号的公式中，标签应加在每行公式的后面（`\\` 命令之前），如：

```
\begin{align}
c^2 &= a^2 + b^2
&\label{eq:gougu-formula}\\
5^2 &= 3^2 + 4^2 + 5^2
&\label{eq:gougu-example}
\end{align}
```

$$c^2 = a^2 + b^2 \quad (4.1)$$

$$5^2 = 3^2 + 4^2 + 5^2 \quad (4.2)$$

引用标签则是在文档的另一个地方，利用已有的标签获得对象的编号、页码等信息。L<sup>A</sup>T<sub>E</sub>X 提供了两个引用的命令：`\ref` 和 `\pageref`，它们分别产生被引用对象的编号和所在页码，例如：勾股定理公式 (4.4.1) 出现在第 30 页，代码为：

```
勾股定理公式 (\ref{eq:gougu-formula}) 出现在第 ~\pageref{eq:gougu-formula} 页
```

### 4.2.2 更多交叉引用

- `\eqref` 命令

除了 L<sup>A</sup>T<sub>E</sub>X 标准的 `\label`, `\ref`, `\nameref` 和 `\pageref`, 很多宏包还提供了各种功能的拓展。如: `amsmath` 提供了 `\eqref` 命令可以自动为引用加上括号,

```
(\ref{eq:gougu-formula})
\eqref{eq:gougu-formula}
\newcommand\eqref[1]{(\ref{#1})} % amsmath 中\eqref 的简易实现
```

- `\autoref` 命令

`hyperref` 宏包提供了自动识别编号类型的 `\autoreff` 命令, 如:

```
See \autoref{tab:Length_unit} % 默认为 Table 7 的效果
See \autoref{fig:WTI}} % 默认为 figure 1 的效果
```

可以通过如下命令对显示方式进行重定义:

```
\renewcommand\figureautorefname{图}
\renewcommand\tableautorefname{表}
```

此时, 上述命令得到的效果是: 表 7;图 1

但是, `\autoref` 只能对引用的数字加前缀名称, 并不能产生“第 4 章”张扬的引用效果。

- LastPage 标签

排版中有时需要只当整个文档内的总页数, 直接在文档末定义 `\label{LastPage}` 往往并不能正确使用, 因为可能还要输出一些浮动体, 尾注等。

`lastpage` 宏包提供了一个特殊的标签 `LastPage`, 它是保证在文档末尾的标签, 可以得到最后一页的页码, 如本页是在: 全文 47 页中的第 31 页:

```
\usepackage{lastpage} % preamble
全文\pageref{LastPage}页中的第\thepage页
```

### 4.2.3 电子文档与超链接

可以给 `hyperref` 设置许多宏包选项来控制其行为, 可以使用 `\hypersetup` 命令单独设置。`hyperref` 的选项大多使用 `<选项>=<值>`。常见的一些选项见表 12。

表 12: hyperref 宏包的常用选项

选项	类型	默认值	说明
colorlinks	<boolean>	false	超链接用彩色显示。相关的彩色选项包括 linkcolor, anchorcolor, citecolor, filecolor, urlcolor 等, 颜色参见 6.4 节
bookmarks	<boolean>	true	生成 PDF 目录书签书签
bookmarksopen	<boolean>	false	在 PDF 阅读器中自动打开书签
bookmarksnumbered	<boolean>	false	目录书签带编号
pdfborder	<num><num><num>	0 0 1	当 colorlink 为假时超链接由彩色边框包围(不会被打印)。默认值表示 1pt 宽的边框, 可以设置为 0 0 0 表示没有边框
pdfpagemode	<text>		在 PDF 阅读器中的页面显示方式, 常用值是 FullScreen, 表示全屏显示
pdfstartview	<text>	Fit	在 PDF 阅读器中的页面缩放大小, 默认值 Fit 表示“适合页面”; 常用取值有适合宽度 FitH、适合高度 FitV
pdftitle	<text>		文档标题, 会在 PDF 文档属性中显示
pdfauthor	<text>		文档作者, 会在 PDF 文档属性中显示
pdfsubject	<text>		文档主题, 会在 PDF 文档属性中显示
pdfkeywords	<text>		文档关键字, 会在 PDF 文档属性中显示

例如, 在制作 PDF 格式的电子书时, 为了方便屏幕上阅读, 可能需要特殊的页面、彩色超链接、全屏显示, 并有何使的文档信息, 那么, 很可能有如下定义:

```

\documentclass[hyperref,UTF8]{ctexbook}
\usepackage{geometry}
\geometry{screen}
\hypersetup{
colorlinks=true,bookmarks=true,
bookmarksopen=false,pdfpagemode=FullScreen,
pdftitle={初等几何教程(电子版)},
pdfauthor={YangSu}
}

```

除了自动的超链接功能，hyperref 还提供了许多命令，用来生成书签、超链接或网址等。

- `\url` 和 `\nolinkurl`

`\url` 命令用来输出 URL 网址，同时也具有超链接功能。与排版村文本不同，`\url` 命令的参数中，网址允许使用的合法符号（如 `&`，`%` 等）直接输入，并且默认以打字机字体输出；如果 URL 地址不需要超链接的效果，也可以用 `\nolinkurl` 命令：

```
\url{https://github.com/YangSuoly/GitStudy1}
\nolinkurl{https://github.com/YangSuoly/GitStudy1}
```

效果分别为：`https://github.com/YangSuoly/GitStudy1`  
`https://github.com/YangSuoly/GitStudy1`

- `\path` 命令

`\path` 命令可以用来排版文件路径，它的格式和 `\url` 差不多

- `\href` 命令

`\href{<URL>}{<Text>}` 命令可以用来是文字产生指向 URL 地址的超链接效果，这里的 URL 地址中的 `#`，`&` 等符号也不需要做特别处理，如：

```
\href{https://yangsuoly.com}{ 独孤诗人的学习驿站}
```

效果为： 独孤诗人的学习驿站

- `\hyperref` 命令

`\hyperref[<Label>]{<Text>}` 命令可以用来使文字产生指向标签的超链接效果，这里的 label 与 `\ref` 使用的 label 相同，不能省略：

```
\hyperref[eq:gougu-formula]{勾股定理}
```

效果为：勾股定理

- `\hypertarget` 和 `\hyperlink`

`\hypertarget{<名称>}{<文字>}` 用来给给文字定义带有名称的链接点，在文档的其他地方使用命令 `\hyperlink{<名称>}{<文字>}` 让另一段文字链接到指定名称的链接点。

- `\pdfbookmark`

`\pdfbookmark[level]{text}{name}` 命令可用来手工添加 PDF 书签，同时定义一个带有名称的链接点（可用于 `\hyperlink` 等命令）。例如建立一个与 `\subsection` 同级的书签，可以用：

```
\pdfbookmark[2]{勾股定理}{gouguproof}
\begin{proof} ..... \end{proof}
```

- `\texorpdfstring` 命令

章节标题中，有时会出现一些数学式、特殊符号等命令，在 PDF 书签中不方便展示，此时可以使用命令 `\texorpdfstring{TEXstring}{PDFstring}` 分别定义在  $\text{T}_\text{E}\text{X}$  文档输出的代码和 PDF 书签等处的纯文本形式，如：

```
\section{\texorpdfstring{\$\frac{1}{\pi}$}{[1/  $\pi$ ] 的计算}}
```

## 4.3 Bib $\text{T}_\text{E}\text{X}$ 与文献数据库

### 4.3.1 Bib $\text{T}_\text{E}\text{X}$ 基础

文献数据库的录入，在 `.tex` 源文件中需要做好如下三件事：

1. 使用 `\bibliographystyle` 命令设定参考文献的格式，通常在导言区完成。基本的 Bib $\text{T}_\text{E}\text{X}$  文献格式包括 `plain`、`unsrt`、`alpha` 和 `abbrv`（见表 13）。前两种使用一般的数字编号文献，`plain` 格式按作者、日期、标题排序，`unsrt` 不排序（保持引用的次数）；`alpha` 则使用一种三字母缩写的方式编号并按作者排序；`abbrv` 格式与 `plain` 基本相同，只是定义了一些缩写。

```
\bibliographystyle{alpha}
```

表 13: 几种基本 Bib $\text{T}_\text{E}\text{X}$  文献格式的属性比较

格式	字母编号	条目排序	不缩写任命	月份全称	期刊全称
<code>plain</code>	✗	✓	✓	✓	✓
<code>unsrt</code>	✗	✗	✓	✓	✓
<code>alpha</code>	✓	✓	✓	✓	✓
<code>abbrv</code>	✗	✓	✗	✗	✗

2. 在正文中用标签使用 `\cite` 命令引用需要的文献，或是使用 `\nocite` 命令指明不引用但仍需要列出的文献标签。`\cite` 命令引用的位置会出现文献的编号，同时将提示 L<sup>A</sup>T<sub>E</sub>X 列出所引用的文献。
3. 使用 `\bibliography` 命令指明要使用的文献数据库。

```
\bibliography{./bibliography/Yangsu_test} % 指明所使用的数据库是 Yangsu_test.bib
```

指明数据库文件时不带 `.bib` 后缀，可以同时从多个文件数据库中提取文献，中间用英文逗号隔开。

### 4.3.2 用 natbib 定制文献格式

- `\citet` 和 `\citep`

`natbib` 宏包提供一种“自然”的文献引用方式，即按文献作者和年带显示文献的引用方式。其提供了许多新的引用命令，最常用的是 `\citeth` 和 `\citep` 命令，分别对应于正文的 (textual) 引用和带括号 (parenthetical) 引用。`\citet` 将产生 Kumar et al. (2020) 这样的引用，适合作为正文的一部分，而 `\citep` 将产生 (Kumar et al., 2020) 这样的引用，作为括号中的补充说明

- 重定义 `\cite`

`natbib` 重定义了原来的 `\cite` 命令，使他在作者年代引用模式下与 `\citet` 等价，而在数字编号引用下与 `\citep` 等价。此外每一中引用还可以带一个 `*` 号，表示当文献作者有两人以上时不缩略显示，如：

```
\citet{Kumar2020} 效果为：Kumar et al. (2020)
\citet*{Kumar2020} 效果为：Kumar, Gandhi, Zhou, Kumar, and Xiang (2020)
```

标准的 `\cite` 只能带一个可选参数，而 `natbib` 中引用命令（通常是 `\citep`，或数字引用模式的 `\cite`）则可以进一步带两个可选参数，分别表示引用前后增加的说明文字：

```
\citep[\S 4.3~节]{Kumar2020} 效果为：(Kumar et al., 2020, § 4.3 节)
\citep[可见][第 13~章]{Kumar2020} 效果为：(可见 Kumar et al., 2020, 第 13 章)
```

- 单独引用

引用的作者、年代和数字不编号也有自己单独的命令可以直接得到：



`\citeauthor{Kumar2020}` 效果为: Kumar et al.

`\citefullauthor{Kumar2020}` 效果为: Kumar, Gandhi, Zhou, Kumar, and Xiang % 或加 \*

`\citeyear{Kumar2020}` 效果为: 2020

`\citeyearpar{Kumar2020}` 效果为: (2020)

`\citenum{Kumar2020}` 效果为: 1

## 5 玩转数学公式

### 5.1 数学模式概说

- 行内公式

在  $\text{T}_{\text{E}}\text{X}$  中, 行内公式一般在前后加个美元符号  $\$ \dots \$$ , 例如:

交换律是  $a + b = b + a$ , 如  $1 + 2 = 2 + 1 = 3$

在数学模式下, 符号会使用单独的字体, 字母通常是倾斜的意大利体, 数字和符号则是直立体。

除了美元符号,  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  中还额外定义了命令格式与环境格式的方式输入行内公式, 即使用命令 `\a+b` 和 `\)` 或是 `\begin{math} a+b \end{math}`。

- 显示数学公式 (不带编号)

使用两个连续美元  $\$ \$ \dots \$ \$$  界定, 同样的,  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  也定义了命令形式和环境形式的输入, 即用 `\[`和`\]` 或是 `displaymath` 环境, 例如:

`\[ a+b=b+a, \]` 效果是:

$$a + b = b + a,$$

`\begin{displaymath} a+b=b+a. \end{displaymath}` 效果是:

$$a + b = b = a.$$

推荐使用 `\[ \dots \]`,  $\$ \$ \dots \$ \$$  会产生不良间距。

- 显示数学公式 (带编号)

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  还提供了自动不编号的数学公式, 可以用 `equation` 环境表示, 公式后还可带引用的标签:

$$a + b = b + a \tag{5.1}$$

amsmath 宏包提供了带星号的 equation\* 环境，功能与 displaymath 环境相同，表示不编号的显示公式，此时 `\[ ... \]` 就称为 equation\* 环境的简写。

- amsmath 宏包

amsmath 提供了 `\text` 命令可以用来在数学公式中插入文字，如：被减数 - 减数 = 差。

amsmath 能识别文档类的 leqno（左侧编号）、reqno（右侧编号）或 fleqn（公式固定缩进不居中）选项的功能：

## 5.2 数学结构

### 5.2.1 上下标

- 上下标

$$A_{ij} = 2^{i+j}, A_i^j = B_i^k, K_{ni} = K_{2i} = 2^{n_i}, 3^{3^{3^{\cdot^3}}}$$

- 撇号 '

数学公式中的撇号 ' 就是一种特殊的商标，表示用符号 `\prime`（即 '）作上标。撇号可以与下标混用，也可以连续使用（普通上标不能连续使用），但不能与上标直接混用：

$$a = a' = a', b'_0 = b''_0, c'^2 = (c')^2$$

- 角度 °

$$\angle a = \angle BAC = 90^\circ = \pi/2$$

- 求和上下标

多数数学算子的上下标，位置时正上或正下方，但行内公式仍在角标位置：

$$\text{行内公式: } \max_n f(n) = \sum_{i=0}^n A_i$$

显示公式：

$$\max_n f(n) = \sum_{i=0}^n A_i$$

不过在行内公式中，为了避免产生难看的行距，所有算子的上下标都在角标位置，如： $\max_n f(n) = \sum_{i=0}^n A_i$

此外，amsmath 还提供了 `\overset`, `\underset` 命令，用来给任意符号的上下方添加标记，这种命令有点像是加了 `\limits` 的巨算符的上下标： $\overset{*}{X}$ ,  $\underset{*}{X}$ ,  $\overset{*}{X}_{\dagger}$

TeX 中上下标是互补影星的，因此想到得到  $A_m^n$  可以使用幻影（phantom）来处理：

$$A_m^n \quad (5.2)$$

$$A_m^n \quad (5.3)$$

- 积分上下标

对于积分等个别算子，显示公式中的上下标在右上右下角：

$$\int_o^1 f(t) dt = \iint_D g(x, y) dx dy \quad (5.4)$$

在上下标前面使用 `\limits` 会使上下标在正上正下方，这是通常上下限（limits）的排版方式，而使用 `\nolimits` 则会使上下标在角标：

$$\iiint_D df = \max_D g \quad (5.5)$$

`\sum\limits_{i=0}^n A_i` 不如用 `\sum\nolimits_{i=0}^n A_i` 更适合文本段落，得到的结果是：

$$\sum_{i=0}^n A_i \text{ 不如用 } \sum_{i=0}^n A_i \text{ 更适合文本段落。}$$

- 左上左下标

有时需要在符号的左上、左下加角标，此时可以使用 `\mathop{H}_m^n_i^j` 的形式得到  ${}_m^j H_i^n$ ，但这种方式得到的效果不尽人意，间距和对齐都不合理，此时可以使用 `amsmath` 包提供的 `\sideset` 命令，如：

$$\sum_{m=0}^n \sum_c^d \quad (5.6)$$

或者是 `mathtools` 宏包的 `\prescript{<up><down><element>}` 来处理：

$${}_m^j H_i^n < L. \quad (5.7)$$

有时需要使用好几列的内容，此时可以用 `amsmath` 提供的 `\substack` 命令排版，效果相当于只有一列的无括号的矩阵：

$$\sum_{\substack{0 \leq i < n \\ 0 \leq j < i}} A_{ij} \quad (5.8)$$

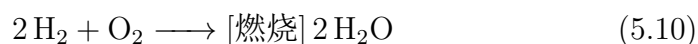
subarray 环境也可使用在上述情况，但可以指定对齐方式为 l、c、r：

$$\sum_{\substack{l]i < 10 \\ j < 100 \\ k < 1000}} X_{i,j,k} \quad (5.9)$$

### • 化学式

将化学式直接作为数学式输入看起来十分笨拙，可以使用专业的化学宏包 mhchem（使用最为广泛的化学宏包）。醋中主要有  $\text{H}_2\text{O}$ ，含有  $\text{CH}_3\text{COO}^-$ 。

${}^{227}_{90}\text{Th}$  元素具有强放射性



## 5.2.2 上下划线和花括号

### • 上下划线

`\overline` 和 `\underline` 命令可以用来在公式的上方和下方划横线：

$$\overline{a + b} = \overline{a} + \underline{b} \quad (5.11)$$

### • 箭头

amsmath 还提供了在公式上下加箭头的命令，使用方法与 `\overline` 和 `\underline` 类似：

```
\overleftarrow{a+b}$,
\overrightarrow{a+b}$,
\overleftrightharrow{a+b}$,
\underleftarrow{a+b}$,
\underrightarrow{a+b}$,
\underleftrightharrow{a+b}$
```

结果是： $\overleftarrow{a+b}$ ,  $\overrightarrow{a+b}$ ,  $\overleftrightharrow{a+b}$ ,  $\underleftarrow{a+b}$ ,  $\underrightarrow{a+b}$ ,  $\underleftrightharrow{a+b}$

### • 花括号

使用 `\overbrace{a+b+c}=\underbrace{1+2+3}` 带上花括号，效果为：

$$\overbrace{a+b+c} = \underbrace{1+2+3} \quad (5.12)$$

使用上下标在花括号上下作标注：

$$\overbrace{a_0, a_1, \dots, a_n}^{n+1} = \underbrace{1+2+3}_n \quad (5.13)$$

- 方括号

mathtools 宏包还提供了在数学公式上下加方括号的命令：

```
\usepackage{mathtools} % preamble
\underbracket[<线宽>][<伸出高度>]{<内容>}
\overbracket[<线宽>][<伸出高度>]{<内容>}
```

如： $\overbracket[1pt][1pt]{1+2+3}_3$

### 5.2.3 分式

- `\frac`

`\frac`< 分子>< 分母>：

$$\frac{1}{2} + \frac{1}{a} = \frac{2+a}{2a} \quad (5.14)$$

行内公式会用较小的字号排版，以免超出文本高度： $\frac{1}{2} + \frac{1}{a} + \frac{2+a}{2a}$

已经在分子或分母中的分式，也会按行内公式的大小排版：

$$\frac{1}{\frac{1}{2}(a+b)}$$

- `\dfrac` 和 `\tfrac`

有时需要指定格式，可以使用 `amsmath` 提供的 `\dfrac` 和 `\tfrac` 分别指定显示格式（display style）和正文格式（text style）的分式：

$$\frac{1}{2}f(x) = \frac{1}{\frac{1}{a} + \frac{1}{b} + c} \quad (5.15)$$

- 连分式 `\cfrac`

连分式 (continued fraction), `amsmath` 提供的 `\cfrac` 专用于输入连分式, `\cfrac` 可以带一个可选参数 `l` 或 `r`, 分别表示左、右对齐, 默认居中对齐:

$$\frac{1}{1 + \frac{2}{1 + \frac{3}{1+x}}} = \frac{1}{1 + \frac{2}{1 + \frac{3}{1+x}}} \quad (5.16)$$

- 小分式 `\sfrac`

`xfrac` 宏包提供了 `\sfrac` 命令可以得到小分式:

```
\usepackage{xfrac} % preamble
```

`\sfrac{1a + b}{1/(a+b)}`, `\sfrac{1}{a+b}` 和 `\sfrac{1}{a+b}` 的效果是  $\frac{1}{a+b}$ ,  $\frac{1}{a+b}$  和  $\frac{1}{a+b}$ 。

- 二项式 `\binom`

还有一些类似分数分成上下两半的数学结构, 如二项式系数  $\binom{n}{k}$ 。 `amsmath` 提供了 `\binom` 来输入二项式系数, 其用法与 `\frac` 类似:

`\binom{n}{k}` 的效果是:  $\binom{n}{k}$

与 `\frac` 类似, `\binom` 也有指定大小的形式 `tbinom` 和 `dbinom`, 分别表示正文格式和显示格式大小的二项式系数。

- 广义分式命令

其他类似分数的形式, 可以由 `amsmath` 提供的更一般的广义分式命令:

```
\genfrac{< 左括号>}{< 右括号>}{< 线宽>}{< 大小>}{< 分子>}{< 分母>}
```

其中 `< 线宽 >` 和 `< 大小 >` 如果为空表示默认值; `< 大小 >` 可以是 0, 1, 2, 3, 分别表示 `\displaystyle`, `\textstyle`, `\scriptstyle`, `\scriptscriptstyle` 四种数字字号。如:

$$\left[ \begin{matrix} n \\ 1 \end{matrix} \right] = (n-1)!, \quad n > 0. \quad (5.17)$$

通常并不直接使用广义分式命令, 而是用它来定义新的分数形式, 如第一类 Stirling 数:

```
\newcommand\stiring[2]{\genfrac{}{}{0pt}{}{#1}{#2}}
\newcommand\dstiring[2]{\genfrac{}{}{0pt}{0}{#1}{#2}}
\newcommand\tstiring[2]{\genfrac{}{}{0pt}{1}{#1}{#2}}
```

### 5.2.4 根式

- `\sqrt` 根式可以由 `\sqrt` 命令得到，同时可以带一个可选参数，表示开方的次数，如：

`\sqrt[4]{} = \sqrt[3]{8} = 2` \$ 效果为： $\sqrt{4} = \sqrt[3]{8} = 2$

如果开方的次数不是建党的整数，或者被开方的内容过长，甚至超过一行）。通常就不使用根数的形式，而改用等价的指数形式：

$$(x_1^p + y_1^p + z_1^p + x_1^q + y_1^q + z_1^q + x_2^p + y_2^p + z_2^p + x_2^q + y_2^q + z_2^q)^{\frac{1}{1/p+1/q}} \quad (5.18)$$

- 调整排版

有时可能对开方次数的排版位置不太满意，可以用 `amsmath` 提供的 `\uproot` 和 `\leftroot` 命令进行调整，命令参数是证书，移动单位是很小的一段距离，如 `\sqrt[n]{\frac{x^2 + \sqrt{2}}{x+y}}` 的原始效果是：

$$\sqrt[n]{\frac{x^2 + \sqrt{2}}{x + y}} \quad (5.19)$$

经过微调 `\sqrt[\uproot{16}\leftroot{-2}n]{\frac{x^2 + \sqrt{2}}{x+y}}` 的效果是：

$$\sqrt[n]{\frac{x^2 + \sqrt{2}}{x + y}} \quad (5.20)$$

- 根式高度

根式的高度是随内容而改变的，但当几个根式并列时，有时需要它们有统一的高度，此时可以使用 `\vphantom` 占位，如：

`\sqrt{\frac{1}{2}} < \sqrt{\vphantom{\frac{1}{2}}2}` \$ 的效果是

$$\sqrt{\frac{1}{2}} < \sqrt{2} \quad (5.21)$$

特别的，数学直接 `\mathstrut` 表示一个圆括号高度和深度的支架，它通常用来平衡不同高度和深度的字母：

`\sqrt b \sqrt y \text{和} \sqrt{\mathstrut y} \sqrt{\mathstrut b}` 效果是：

$$\sqrt{b}\sqrt{y} \text{ 和 } \sqrt{y}\sqrt{b} \quad (5.22)$$

## 5.3 符号与类型

### 5.3.1 字母表与普通符号

数字字母可以使用多种字体，数字字体命令与正文字体命令语法类似，如 `\mathbf{X}` 可以得到直立粗体的 **X**。数学公式总，只有变量使用默认的意大利体，数字常数 *e* 和虚数单位 *i* 通常使用罗马体。标准 L<sup>A</sup>T<sub>E</sub>X 提供默认数学字母见 表 14。

表 14: 标准 L<sup>A</sup>T<sub>E</sub>X 默认提供的数学字母字体

类别	字体命令	输出效果
数学环境的默认字体	<code>\mathnormal</code>	<i>ABCHIJXYZabchijxyz12345</i>
意大利体	<code>\mathit</code>	<i>ABCHIJXYZabchijxyz12345</i>
罗马体	<code>\mathrm</code>	ABCHIJXYZabchijxyz12345
粗体	<code>\mathbf</code>	<b>ABCHIJXYZabchijxyz12345</b>
无衬线体	<code>\mathsf</code>	ABCHIJXYZabchijxyz12345
打字机体	<code>\mathtt</code>	ABCHIJXYZabchijxyz12345
手写体（花体） <sup>†</sup>	<code>\mathcal</code>	<i>ABCHIJXYZ</i>

<sup>†</sup>L<sup>A</sup>T<sub>E</sub>X 默认只支持大写，使用专业字体包（如 `amssymb` 的子包 `amsfonts`）可能支持小写字母

默认的数学字母字体由 `\mathnormal` 产生，它使用与正文不同的数学字体，字体间距比正文大一些。因此使用 `$xyz$` 通常表示三个字母的乘积 *xyz*（距离较大），而如果要表示一个多字母的长变量名 *xyz*，则应该使用 `\mathit{xyz}` 等形式。

利用数学字体包，还可以得到其他一些数学字母字体，常见的字体包见图 3。



表 4.3 常见的数学字母字体包

类别	字体命令	输出效果	宏包及说明
黑板粗体	<code>\mathbb</code>	ABCXYZ	amssymb, 仅大写字母 <sup>†</sup>
	<code>\mathbb</code>	ABCXYZabcdefghijklmnopqrstuvwxyz123890	bbold
	<code>\mathbbm</code>	ABCXYZabcdefghijklmnopqrstuvwxyz12	bbm, 数字仅有 1 和 2
花体	<code>\mathscr</code>	<i>ABCXYZ</i>	mathrsfs, 仅大写字母
	<code>\mathcal</code>	ABCXYZ	euscript 加 eucal 选项, 仅大写字母
哥特体	<code>\mathfrak</code>	A B C X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z	amssymb 或 eufrak

<sup>†</sup>此外 `\Bbbk` 可以得到  $\mathbb{k}$ 。

图 3 常见的数学字母字体包

除了拉丁字母，数学公式中还经常使用希腊字母和少量其他类型的字母，见表 15。

表 15: 特殊字符

小写希腊字母							
$\alpha$	<code>\alpha</code>	$\beta$	<code>\beta</code>	$\gamma$	<code>\gamma</code>	$\delta$	<code>\delta</code>
$\epsilon$	<code>\epsilon</code>	$\zeta$	<code>\zeta</code>	$\eta$	<code>\eta</code>	$\theta$	<code>\theta</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>	$\pi$	<code>\pi</code>	$\rho$	<code>\rho</code>
$\sigma$	<code>\sigma</code>	$\tau$	<code>\tau</code>	$\upsilon$	<code>\upsilon</code>	$\phi$	<code>\phi</code>
$\chi$	<code>\chi</code>	$\psi$	<code>\psi</code>	$\omega$	<code>\omega</code>		
$\varepsilon$	<code>\varepsilon</code>	$\vartheta$	<code>\vartheta</code>	$\varkappa$	<code>\varkappa</code> <sup>†</sup>	$\varpi$	<code>\varpi</code>
$\varrho$	<code>\varrho</code>	$\varsigma$	<code>\varsigma</code>	$\varphi$	<code>\varphi</code>	$F$	<code>\digamma</code> <sup>†</sup>
大写希腊字母							
$\Gamma$	<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>		
$\varGamma$	<code>\varGamma</code>	$\varDelta$	<code>\varDelta</code>	$\varTheta$	<code>\varTheta</code>	$\varLambda$	<code>\varLambda</code>
$\varXi$	<code>\varXi</code>	$\varPi$	<code>\varPi</code>	$\varSigma$	<code>\varSigma</code>	$\varUpsilon$	<code>\varUpsilon</code>
$\varPhi$	<code>\varPhi</code>	$\varPsi$	<code>\varPsi</code>	$\varOmega$	<code>\varOmega</code>		

注：小写中前面带 var 的命令是原来字母的变体。大写中则是原来字母的变体（倾斜）形式。

<sup>†</sup>  $\mathcal{A}\mathcal{M}\mathcal{S}$  符号，需要 amssymb 或类似的包。

也可用 `\mathnormal` 得到倾斜形式的大写希腊字母。

更多字母见《 $\text{\LaTeX}$  入门》P<sub>238–241</sub>。

5.3.2 L<sup>A</sup>T<sub>E</sub>X 数学符号对应表表 16: L<sup>A</sup>T<sub>E</sub>X 数学符号对应表

运算符符号							
$\pm$	<code>\pm</code>	$\mp$	<code>\mp</code>	$\times$	<code>\times</code>	$\div$	<code>\div</code>
$\cdot$	<code>\cdot</code>	$*$	<code>\ast</code>	$\star$	<code>\star</code>	$\dagger$	<code>\dagger</code>
$\ddagger$	<code>\ddagger</code>	$\amalg$	<code>\amalg</code>	$\cap$	<code>\cap</code>	$\cup$	<code>\cup</code>
$\uplus$	<code>\uplus</code>	$\sqcap$	<code>\sqcap</code>	$\sqcup$	<code>\sqcup</code>	$\vee$	<code>\vee</code>
$\wedge$	<code>\wedge</code>	$\setminus$	<code>\setminus</code>	$\wr$	<code>\wr</code>	$\circ$	<code>\circ</code>
$\bullet$	<code>\bullet</code>	$\oslash$	<code>\oslash</code>	$\odot$	<code>\odot</code>	$\bigcirc$	<code>\bigcirc</code>
$\oplus$	<code>\oplus</code>	$\ominus$	<code>\ominus</code>	$\otimes$	<code>\otimes</code>	$\oint$	<code>\oint</code>
$\nabla$	<code>\bigtriangledown</code>	$\triangleup$	<code>\bigtriangleup</code>	$\perp$	<code>\bot</code>	$\top$	<code>\top</code>
$\triangleleft$	<code>\triangleleft</code>	$\triangleright$	<code>\triangleright</code>	$\sum$	<code>\sum</code>	$\prod$	<code>\prod</code>
$\int$	<code>\int</code>	$\biguplus$	<code>\biguplus</code>	$\bigoplus$	<code>\bigoplus</code>	$\bigvee$	<code>\bigvee</code>
$\bigcap$	<code>\bigcap</code>	$\bigotimes$	<code>\bigotimes</code>	$\bigwedge$	<code>\bigwedge</code>	$\bigcup$	<code>\bigcup</code>
$\bigodot$	<code>\bigodot</code>	$\bigsqcup$	<code>\bigsqcup</code>				
关系符号							
$<$	<code>&lt;</code>	$>$	<code>&gt;</code>	$\leq$	<code>\leq</code>	$\geq$	<code>\geq</code>
$\ll$	<code>\ll</code>	$\gg$	<code>\gg</code>	$\neq$	<code>\neq</code>	$\doteq$	<code>\doteq</code>
$\sim$	<code>\sim</code>	$\simeq$	<code>\simeq</code>	$\approx$	<code>\approx</code>		
$\asymp$	<code>\asymp</code>	$\cong$	<code>\cong</code>	$\equiv$	<code>\equiv</code>	$\subset$	<code>\subset</code>
$\supset$	<code>\supset</code>	$\subseteq$	<code>\subseteq</code>	$\supseteq$	<code>\supseteq</code>	$\sqsubseteq$	<code>\sqsubseteq</code>
$\sqsupseteq$	<code>\sqsupseteq</code>	$\models$	<code>\models</code>	$\vdash$	<code>\vdash</code>	$\dashv$	<code>\dashv</code>
$\perp$	<code>\perp</code>	$\mid$	<code>\mid</code>	$\parallel$	<code>\parallel</code>	$\propto$	<code>\propto</code>
$\bowtie$	<code>\bowtie</code>	$\preceq$	<code>\preceq</code>	$\succ$	<code>\succ</code>	$\succeq$	<code>\succeq</code>
$\nless$	<code>\nless</code>	$\ngtr$	<code>\ngtr</code>	$\nleqslant$	<code>\nleqslant</code>	$\ngeqslant$	<code>\ngeqslant</code>
$\not\equiv$	<code>\not\equiv</code>	$\not\approx$	<code>\not\approx</code>	$\not\cong$	<code>\not\cong</code>	$\not\sim$	<code>\not\sim</code>
$\not\simeq$	<code>\not\simeq</code>	$\not\models$	<code>\not\models</code>	$\not\ni$	<code>\not\ni</code>	$\not\succ$	<code>\not\succ</code>
$\sqsupseteq$	<code>\sqsupseteq</code>	$\not\succeq$	<code>\not\succeq</code>	$\not\prec$	<code>\not\prec</code>	$\not\preceq$	<code>\not\preceq</code>
$\nparallel$	<code>\nparallel</code>	$\not\subset$	<code>\not\subset</code>	$\not\supset$	<code>\not\supset</code>	$\not\subseteq$	<code>\not\subseteq</code>
$\not\supseteq$	<code>\not\supseteq</code>						
括号符号							
$\{$	<code>\lbrace</code>	$\}$	<code>\rbrace</code>	$[$	<code>\lbrack</code>	$]$	<code>\rbrack</code>
$\langle$	<code>\rangle</code>	$\rangle$	<code>\rangle</code>	$\lceil$	<code>\lceil</code>	$\rceil$	<code>\rceil</code>
$\lfloor$	<code>\lfloor</code>	$\rfloor$	<code>\rfloor</code>	$\lvert$	<code>\lvert</code>	$\rVert$	<code>\rVert</code>

(续表)

\ \backslash				其他符号			
$\infty$	<code>\infty</code>	$\exists$	<code>\exists</code>	$\forall$	<code>\forall</code>	$\neg$	<code>\neg</code>
$\nabla$	<code>\nabla</code>	$\triangle$	<code>\triangle</code>	$\angle$	<code>\angle</code>	$\partial$	<code>\partial</code>
$\emptyset$	<code>\emptyset</code>	$'$	<code>\prime</code>	$:$	<code>\colon</code>	$\Re$	<code>\Re</code>
$\Im$	<code>\Im</code>	$\dots$	<code>\ldots</code>	$\cdots$	<code>\cdots</code>	$\vdots$	<code>\vdots</code>
$\ddots$	<code>\ddots</code>	$\sqrt{\phantom{x}}$	<code>\surd</code>	$\cdot$	<code>\ldotp</code>	$\rightarrow$	<code>\to</code>
$\leftarrow$	<code>\gets</code>	$\aleph$	<code>\aleph</code>	$\hbar$	<code>\hbar</code>	$\wp$	<code>\wp</code>
$\ell$	<code>\ell</code>	$\imath$	<code>\imath</code>	$j$	<code>\jmath</code>		

注：小写中前面带 var 的命令是原来字母的变体. 大写中则是原来字母的变体（倾斜）形式.

<sup>†</sup>  $\mathcal{M}\mathcal{S}$  符号，需要 amssymb 或类似的包.

也可用 `\mathnormal` 得到倾斜形式的大写希腊字母。

表 17: L<sup>A</sup>T<sub>E</sub>X 数学符号 (箭头符号) 对应表

箭头符号			
$\leftarrow$	<code>\leftarrow</code>	$\Leftarrow$	<code>\Leftarrow</code>
$\Longleftarrow$	<code>\Longleftarrow</code>	$\rightarrow$	<code>\rightarrow</code>
$\longrightarrow$	<code>\longrightarrow</code>	$\Rightarrow$	<code>\Rightarrow</code>
$\Leftrightarrow$	<code>\Leftrightarrow</code>	$\Longrightarrow$	<code>\Longrightarrow</code>
$\hookrightarrow$	<code>\hookrightarrow</code>	$\Longleftrightarrow$	<code>\Longleftrightarrow</code>
$\rightharpoonup$	<code>\rightharpoonup</code>	$\hookleftarrow$	<code>\hookleftarrow</code>
$\uparrow$	<code>\uparrow</code>	$\leftharpoonup$	<code>\leftharpoonup</code>
$\Downarrow$	<code>\Downarrow</code>	$\rightarrowtail$	<code>\rightarrowtail</code>
$\swarrow$	<code>\swarrow</code>	$\Uparrow$	<code>\Uparrow</code>
$\searrow$	<code>\searrow</code>	$\updownarrow$	<code>\updownarrow</code>
		$\nearrow$	<code>\nearrow</code>
		$\nwarrow$	<code>\nwarrow</code>
		$\mapsto$	<code>\mapsto</code>
		$\longmapsto$	<code>\longmapsto</code>

### 5.3.3 数学算子

## 6 绘制图表

### 6.1 L<sup>A</sup>T<sub>E</sub>X 中的表格

### 6.2 插图与变换

### 6.3 浮动体与标题控制

### 6.4 使用彩色

## 参考文献

Anil Kumar, C. P. Gandhi, Yuqing Zhou, Rajesh Kumar, and Jiawei Xiang. Variational mode decomposition based symmetric single valued neutrosophic cross entropy measure for the identification of bearing defects in a centrifugal pump. *Applied Acoustics*, 165: UNSP 107294, August 2020. doi: 10.1016/j.apacoust.2020.107294.

卢琳珍, 徐定华, and 徐映红. 应用三层热防护服热传递改进模型的皮肤烧伤度预测. 纺织学报, 39(01):111–118+125, 2018. ISSN 0253-9721. URL <http://kns.cnki.net/KCMS/detail/detail.aspx?dbcode=CJFQ&dbname=CJFDLAST2018&filename=FZXB201801020&v=MjAwMdB1WDFMdXhZUzdEaDFUM3FUCldNMUZyQ1VSTEtmYnVkcUZDbm1VTDNJSXpmVGJMRzRIOW5Ncm85SFp>