

1. 排版基础

1.1 行内和行间公式

你好公式： $a^2 + b^2 = c^2$

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

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

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

1.2 数学模式

特点：

- 输入的空格被忽略。数学符号的间距默认由符号的性质决定。需要人为引入时，使用 `\quad` 和 `\qquad` 等命令。
- 不允许有空行（分段）。
- 所有的字母被当作数学公式中的变量处理。如果现在数学公式中输入正体的文本，简单情况下可用 `\mathrm`。或者用amsmath提供的 `\text` 命令。

2. 数学符号

2.1 一般符号

```
1 a_1,a_2,\dots,a_n\\
2 a_1+a_2+a_3+\dots+a_n
```

$$a_1, a_2, \dots, a_n$$
$$a_1 + a_2 + a_3 + \dots + a_n$$

2.2 指数、上下标和导数

`^` 和 `_` 指明上下标，一般需要将上下标的把内容用 `{}` 括起来，否则只能对后面一个符号起作用

```
1 p^3_{ij} \qquad
2 m_{\text{Kunth}} \qquad
3 m_{Kunth} \qquad
4 \sum_{k=1}^3 k \quad [5pt]
5 a^{x+y} \neq a^{x+y} \qquad
6 e^{x^2} \neq {e^x}^2
```

$$p_{ij}^3 \quad m_{\text{Kunth}} \quad m_{Kunth} \quad \sum_{k=1}^3 k$$
$$a^x + y \neq a^{x+y} \quad e^{x^2} \neq e^{x^2}$$

`\text` 命令仅适合在公式中穿插少量的文字

2.3 分式和根式

分式使用 `\frac{分子}{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。

```
1 3/8 \qquad
2 \frac{3}{8} \qquad
3 \tfrac{3}{8}\qquad
4 \dfrac{3}{8}
```

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

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

```
1 \sqrt{x} \Leftrightarrow x^{1/2} \qquad
2 \sqrt[3]{2} \qquad
3 \sqrt{x^2} + \sqrt{y}
```

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

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

```
1 \binom{n}{k}
```

$\binom{n}{k}$

2.4 关系符

LATEX 常见的关系符号除了可以直接输入的 `=`，`>`，`<`，其它符号用命令输入，常用的有不等号 `\neq`、大于等于号 `\geq` 和小于等于号 `\leq`、约等号 `\approx`、等价 `\equiv`、正比 `\propto`、相似 `\sim` 等等。更多符号命令可参考表。

LATEX 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：

```
1 \stackrel{*}{\approx} \qquad \stackrel{=}{\leq}
```

$\overset{*}{\approx} \qquad \overset{=}{\leq}$

2.5 算符

LATEX 中的算符大多数是二元算符，除了直接用键盘可以输入的 `+`、`-`、`*`、`=`，其它符号用命令输入，常用的有乘号 `\times`、除号 `\div`、点乘 `\cdot`、加减号 `\pm` / 干(`\mp`) 等等。更多符号命令可参考表。

∇ (`\nabla`) 和 ∂ (`\partial`) 也是常用的算符，虽然它们不属于二元算符。LATEX 将数学函数的名称作为一个算符排版，字体为直立字体。其中有一部分符号在上下位置可以书写一些内容作为条件，类似于后文所叙述的巨算符。

Latex作为算符的函数名字一览(不带上下限的算符)							
函数	公式	函数	公式	函数	公式	函数	公式
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				

Latex作为算符的函数名字一览(带上下限的算符)							
函数	公式	函数	公式	函数	公式	函数	公式
lim	\lim	lim sup	\limsup	lim inf	\liminf	sup	\sup
inf	\inf	min	\min	max	\max	det	\det
Pr	\Pr	gcd	\gcd				

```
1 \lim_{x \rightarrow 0}
2 \frac{\sin x}{x}=1
```

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

对于求模表达式， LATEX 提供了 `\bmod` 和 `\pmod` 命令，前者相当于一个二元运算符，后者作为同余表达式的后缀：

```
1 a \bmod b\\
2 x \equiv a \pmod{b}
```

$a \bmod b$

$x \equiv a \pmod{b}$

如果算符不够用的话， amsmath 允许用户用 `\DeclareMathOperator` 定义自己的算符，其中带星号的命令定义带上下限的算符：

```
1 \DeclareMathOperator{\argh}{argh}
2 \DeclareMathOperator*{\Nut}{Nut}
3 \argh 3 = \Nut_{x=1} 4x
```

$\argh 3 = \Nut_{x=1} 4x$

2.6 巨算符

积分号 \int (`\int`)、求和号 \sum (`\sum`) 等符号称为巨算符。巨算符在行内公式和行间公式的大小和形状有区别。

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

行间：

```
1 \sum_{i=1}^n \quad \int_0^{\frac{\pi}{2}} \quad \prod_{\epsilon}
```

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

巨算符的上下标位置可由 `\limits` 和 `\nolimits` 调整，前者令巨算符类似 \lim 或求和算符 \sum ，上下标位于上下方；后者令巨算符类似积分号，上下标位于右上方和右下方。

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

行间：

```
1 \sum\limits_{i=1}^n \quad \int\limits_0^{\frac{\pi}{2}} \quad \prod\limits_{\epsilon}
```

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

amsmath 宏包还提供了 `\substack`，能够在下限位置书写多行表达式； subarray 环境更进一步，令多行表达式可选择居中 (c) 或左对齐 (l)：

```
1 \sum_{\substack{0\leq i\leq n \\ j\in \mathbb{R}}} P(i, j) = Q(n) \\
2 j\in \mathbb{R}} \\
3 P(i, j) = Q(n) \\
4
5 \sum_{\begin{subarray}{l} 0\leq i \leq n \\ j \in \mathbb{R} \end{subarray}} P(i,j) = Q(n) \\
6 0\leq i \leq n \\
7 j \in \mathbb{R} \\
8 \end{subarray}} \\
9 P(i,j) = Q(n) \\
10
11 \sum_{\begin{subarray}{r} 0\leq i \leq n \\ j \in \mathbb{R} \end{subarray}} P(i,j) = Q(n) \\
12 0\leq i \leq n \\
13 j \in \mathbb{R} \\
14 \end{subarray}} \\
15 P(i,j) = Q(n)
```

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

$\sum_{\begin{subarray}{r} 0\leq i\leq n \\ j\in \mathbb{R} \end{subarray}} P(i, j) = Qn$

$\sum_{\begin{subarray}{l} 0\leq i\leq n \\ j\in \mathbb{R} \end{subarray}} P(i, j) = Qn$

2.7 数学重音和上下括号

数学符号可以像文字一样加重音，比如求导符号 \dot{r} (`\dot{r}`)、 \ddot{r} (`\ddot{r}`)、表示向量的箭头 \vec{r} (`\vec{r}`)、表示单位向量的符号 $\hat{\mathbf{e}}$ (`\hat{\mathbf{e}}`) 等，详见表。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：

```
1 \bar{x}_0 \quad \bar{x}_0 \\
2 \vec{x}_0 \quad \vec{x}_0 \\
3 \hat{\mathbf{e}}_x \quad \hat{\mathbf{e}}_x
```

\bar{x}_0

\vec{x}_0

$\hat{\mathbf{e}}_x$

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。后两者详见表。

```
1 0.\overline{3} =
2 \underline{\underline{1/3}} \\
3 $\hat{XY}$ \quad \widehat{XY} \\
4 $\vec{AB}$ \quad \overrightarrow{AB}
```

$0.\overline{3} = \underline{\underline{1/3}}$

$\hat{XY} \quad \widehat{XY}$

$\vec{AB} \quad \overrightarrow{AB}$

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

```
1 \underbrace{\overbrace{(a+b+c)}^6}^7
2 \cdot \overbrace{(d+e+f)}^7
3 _\text{meaning of file} = 42
```

$$\overbrace{(a+b+c)}^6 \cdot \underbrace{(d+e+f)}_{\text{meaning of file}}^7 = 42$$

2.8 箭头

常用的箭头包括 `\rightarrow` (\rightarrow ，或 `\to`)、`\leftarrow` (\leftarrow ，或 `\gets`) 等。更多箭头详见表。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：

```
1 a \xleftarrow{x+y+z} b \\
2 c \xrightarrow[x<y]{a*b*c} d
```

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

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

2.9 括号和定界符

LATEX 提供了多种括号和定界符表示公式块的边界，如小括号()`\{ \}` (`\{ \}`)、尖括号`\langle \rangle` (`\langle \rangle`) 等。更多的括号/定界符命令见表。

```
1 {a,b,c} \neq \{a,b,c\}
```

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

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

$1 + \left(\frac{1}{1-x^2} \right)^3$

$\left. \frac{\partial f}{\partial t} \right|_{t=0}$

有时我们不满足于 LATEX 为我们自动调节的定界符大小。这时我们还可以用 `\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）。

```
1 \Bigl((x+1)(x+2)\Bigr)^{2}
```

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

```
1 \bigl( \Bigl( \biggl( \Biggl( \quad
2 \bigr) \Bigr) \biggr) \Biggr) \quad
3 \big\| \Big\| \bigg\| \Bigg\| \quad
4 \big\Downarrow \Big\Downarrow \bigg\Downarrow \Bigg\Downarrow
```

$$\left(\left(\left(\left(\right)\right)\right)\right) \quad \big\| \Big\| \bigg\| \Bigg\| \quad \big\Downarrow \Big\Downarrow \bigg\Downarrow \Bigg\Downarrow$$

使用 `\big` 和 `\bigg` 等命令的另外一个好处是：用 `\left` 和 `\right` 分界符包裹的公式块是不允许断行的（下文提到的 `array` 或者 `aligned` 等环境视为一个公式块），所以也不允许在多行公式里跨行使用，而 `\big` 和 `\bigg` 等命令不受限制。

3. 多行公式

3.1 长公式折行

通常来讲应当避免写出超过一行而需要折行的长公式。如果一定要折行的话，习惯上优先在等号之前折行，其次在加号、减号之前，再次在乘号、除号之前。其它位置应当避免折行。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。

```
1 \begin{multline}
2 a + b + c + d + e + f + g + h + i \\
3 = j + k + l + m + n \\
4 = o + p + q + r + s \\
5 = t + u + v + x + z
6 \end{multline}
```

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

3.2 多行公式

更多的情况是，我们需要罗列一系列公式，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：

```
1 \begin{align}
2 a &= b + c \\
3 &= d + e
4 \end{align}
```

$$\begin{aligned} a &= b + c \\ &= d + e \end{aligned}$$

`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：

```
1 \begin{align}
2 a &= 1 & b &= 2 & c &= 3 \\
3 d &= -1 & e &= -2 & f &= -5
4 \end{align}
```

$$\begin{aligned} a &= 1 & b &= 2 & c &= 3 \\ d &= -1 & e &= -2 & f &= -5 \end{aligned}$$

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：

```
1 \begin{gather}
2 a = b + c \\
3 d = e + f + g \\
4 h + i = j + k \notag \\
5 l + m = n
6 \end{gather}
```

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

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

不用 `gather` 则行间距略大：

```
1 a = b + c \\
2 d = e + f + g \\
3 h + i = j + k \notag \\
4 l + m = n
```

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

3.3 公式编号的多行公式

另一个常见的需求是将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。为此，`amsmath` 宏包提供了诸如 `aligned`、`gathered` 等环境，与 `equation` 环境（可用 `\tags` 替代）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——对应。我们仅以 `aligned` 举例：

```
1 \begin{aligned}
2 a &= b + c \\
3 d &= e + f + g \\
4 h + i &= j + k \\
5 l + m &= n
6 \end{aligned}
7 \tag{123}
```

$$\begin{aligned} a &= b + c \\ d &= e + f + g \\ h + i &= j + k \\ l + m &= n \end{aligned} \tag{123}$$

4. 数组和矩阵

为了排版二维数组，`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：

```
1 \mathbf{X} = \left(
2 \begin{array}{cccc}
3 x_{11} & x_{12} & \ldots & x_{1n} \\
4 x_{21} & x_{22} & \ldots & x_{2n} \\
5 \vdots & \vdots & \ddots & \vdots \\
6 x_{n1} & x_{n2} & \ldots & x_{nn}
7 \end{array} \right)
```

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nn} \end{pmatrix}$$

值得注意的是，上一节末尾介绍的 `aligned` 等环境也可以用定界符包裹。

我们还可以利用空的定界符排版出这样的效果：

```
1 |x| = \left\{
2 \begin{array}{rl}
3 -x & \text{if } x < 0, \\
4 0 & \text{if } x = 0, \\
5 x & \text{if } x > 0.
6 \end{array} \right.
```

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

不过上述例子可以用 `amsmath` 提供的 `cases` 环境更轻松地完成：

```
1 |x| =
2 \begin{cases}
3 -x & \text{if } x < 0, \\
4 0 & \text{if } x = 0, \\
5 x & \text{if } x > 0.
6 \end{cases}
```

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

我们当然也可以用 `array` 环境排版各种矩阵。 `amsmath` 宏包还直接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` (`{}`)、`vmatrix` (`|`)、`Vmatrix` (`||`)。使用这些环境时，无需给定列格式：

```
1 \begin{matrix}
2 1 & 2 & \ldots & 3 & 4 \\
3 \end{matrix} \quad
4 \begin{bmatrix}
5 x_{11} & x_{12} & \ldots & x_{1n} \\
6 x_{21} & x_{22} & \ldots & x_{2n} \\
7 \vdots & \vdots & \ddots & \vdots \\
8 x_{n1} & x_{n2} & \ldots & x_{nn} \\
9 \end{bmatrix}
```

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

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，这时要用到 `\[8pt]` 的方法来调节间距：

```
1 \mathbf{H}=
2 \begin{bmatrix}
3 \dfrac{\partial^2 f}{\partial x^2} & \\
4 \dfrac{\partial^2 f}{\partial x \partial y} & \\
5 \dfrac{\partial^2 f}{\partial y^2} & \\
6 \end{bmatrix}
7 \dfrac{\partial^2 f}{\partial x \partial y} & \\
8 \dfrac{\partial^2 f}{\partial y^2} & \\
9 \end{bmatrix}
```

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

5. 公式中的间距

前文提到过，绝大部分时候，数学公式中各元素的间距是根据符号类型自动生成的，需要我们手动调整的情况极少。我们已经认识了两个生成间距的命令 `\quad` 和 `\qquad`。在公式中我们还可能用到的间距包括 `\,`、`\:`、`\;` 以及负间距 `\!`，其中 `\quad`、`\qquad` 和 `\,` 在文本和数学环境中可用，后三个命令只用于数学环境。

```
1 \begin{aligned}
2 & \quad \quad \\
3 & a \quad b \\
4 & a \quad \quad b \\
5 & a \backslash a \\
6 & a \backslash, a \\
7 & a \backslash: a \\
8 & a \backslash; a \\
9 & a \backslash! a \\
10 \end{aligned}
```

aa
 $a \quad b$
 $a \quad \quad b$
 $a \, a$
 $a \, a$
 $a \, a$
 $a \, a$
 $\alpha\alpha$

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。注意微元里的 d 用的是直立体：

```
1 \int_a^b f(x)\mathrm{d}x
2 \qquad
3 \int_a^b f(x)\,,\mathrm{d}x
```

$$\int_a^b f(x)dx \qquad \int_a^b f(x) \, dx$$

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。不过 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 `\newcommand\diff{\,,\mathrm{d}}` 相当于C语言宏。

```
1 \newcommand\diff{\,,\mathrm{d}}
2
3 \begin{gather*}
4 \int\int f(x)g(y)
5 \diff x \diff y \\
6 \int\!\!\!\!\!\int\int
7 f(x)g(y) \diff x \diff y \\
8 \iint f(x)g(y) \diff x \diff y \\
9 \iint\quad\iint\quad\idotsint
10 \end{gather*}
```

$$\int\int f(x)g(y) \, dx \, dy$$
$$\iint f(x)g(y) \, dx \, dy$$
$$\iint f(x)g(y) \, dx \, dy$$
$$\iiint \quad \iiint \quad \int \cdots \int$$

6. 数学符号的字体控制

6.1 数学字母字体

6.2 数学符号的尺寸

数学符号按照符号排版的位置规定尺寸，从大到小包括行间公式尺寸、行内公式尺寸、上下标尺寸、次级上下标尺寸。除了字号有别之外，行间和行内公式尺寸下的巨算符也使用不一样的大小。LATEX 为每个数学尺寸指定了一个切换的命令。

数学字母字体

示例	命令
$ABCDEabcde1234$	<code>\mathnormal{...}</code>
$ABCD\mathrm{E}abcde1234$	<code>\mathrm{...}</code>
$ABCDEabcde123\mathit{4}$	<code>\mathit{...}</code>
$\mathbf{ABCDEabcde1234}$	<code>\mathbf{...}</code>
$ABCD\mathrm{E}abcde1234$	<code>\mathsf{...}</code>
$ABCD\mathrm{E}abcde1234$	<code>\mathtt{...}</code>
$\mathcal{ABCDEabcde1234}$	<code>\mathcal{...}</code>
$\mathscr{ABCDEabcde1234}$	<code>\mathscr{...}</code>
$\mathfrak{ABCDEabcde1234}$	<code>\mathfrak{...}</code>
$\mathbb{ABCDEabcde1234}$	<code>\mathbb{...}</code>

数数字号尺寸

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

我们通过以下示例对比行间公式和行内公式的区别。在分式中，分子分母默认为行内公式尺寸，示例中将分母切换到行间公式尺寸：

```
1 r = \frac
2 {\sum_{i=1}^n (x_i- x)(y_i- y)}
3 {\displaystyle \left[
4 \sum_{i=1}^n (x_i-x)^2
5 \sum_{i=1}^n (y_i-y)^2
6 \right]^{1/2} }
```

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

6.3 加粗的数学符号

在 LATEX 中为符号切换数学字体并不十分自由，只能通过 `\mathbf` 等有限的命令切换字体。比如想得到粗斜体的符号，就没有现成的命令；再比如 `\mathbf` 只能改变拉丁字母和大写希腊字母，小写希腊字母就没有用。LATEX 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（前提是数学字体宏包本身支持粗体符号）。但这个命令只能在公式外使用：

```
1 \mu, M \quad
2 \boldsymbol{\mu}, \boldsymbol{M}
```

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

7. 模板

```
1 \left(x-1\right)\left(x+3\right)
```

$$(x-1)(x+3)$$

```
1 \sqrt{a^2+b^2}
```

$$\sqrt{a^2+b^2}$$

```
1 \left(\frac{a}{b}\right)^n=\frac{a^n}{b^n}
```

$$\left(\frac{a}{b}\right)^n=\frac{a^n}{b^n}$$

```
1 x =\frac{-b\pm\sqrt{b^2-4ac}}{2a}
```

$$x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$$

```
1 a>b,b>c\Rightarrow a>c
```

$$a>b,b>c\Rightarrow a>c$$

```
1 \int \frac{1}{1+x^{2}}\mathrm{d}x= \arctan x +C
```

$$\int \frac{1}{1+x^2}\mathrm{d}x=\arctan x+C$$

```
1 f(x) = \int_{-\infty}^{\infty} \hat{f}(x)\mathrm{d}x\mathrm{e}^{2\pi i x} \mathrm{d}x
```

$$f(x)=\int_{-\infty}^{\infty}\hat{f}(x)\xi e^{2\pi i \xi x}\mathrm{d}\xi$$

```
1 \begin{pmatrix}
2 a_{11} & a_{12} & a_{13} \\
3 a_{21} & a_{22} & a_{23} \\
4 a_{31} & a_{32} & a_{33}
5 \end{pmatrix}
```

$$\begin{pmatrix}a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33}\end{pmatrix}$$

```
1 \begin{pmatrix}
2 a_{11} & \cdots & a_{1n} \\
3 \vdots & \ddots & \vdots \\
4 a_{m1} & \cdots & a_{mn}
5 \end{pmatrix}
```

$$\begin{pmatrix}a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn}\end{pmatrix}$$

```
1 \mathbf{A}_{m\times n}=
2 \begin{bmatrix}
3 a_{11}&a_{12}&\cdots &a_{1n} \\
4 a_{21}&a_{22}&\cdots &a_{2n} \\
5 \vdots &\vdots &\ddots &\vdots \\
6 a_{m1}&a_{m2}&\cdots &a_{mn}
7 \end{bmatrix}
8 =\left[ a_{ij}\right]
```

$$\mathbf{A}_{m\times n}=\begin{bmatrix}a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn}\end{bmatrix}=[a_{ij}]$$

```
1 X_1, \cdots, X_n
```

$$X_1,\cdots,X_n$$

```
1 \frac{\sin A}{a}=\frac{\sin B}{b}=\frac{\sin C}{c}=\frac{1}{2R}
```

$$\frac{\sin A}{a}=\frac{\sin B}{b}=\frac{\sin C}{c}=\frac{1}{2R}$$

```
1 \begin{array}{c}
2 \text{若}P\left( AB \right) =P\left( A \right) P\left( B \right) \\
3 \text{则}P\left( A\left| B \right. \right) =P\left( B \right) \\
4 \end{array}
```

$$\begin{array}{l} \text{若}P(AB)=P(A)P(B) \\ \text{则}P(A|B)=P(B) \end{array}$$

1 (1+x)^{n} =1 + \frac{nx}{1!} + \frac{n(n-1)x^2}{2!} + \cdots

$$(1+x)^n = 1 + \frac{nx}{1!} + \frac{n(n-1)x^2}{2!} + \cdots$$

1 \lambda
2 = \frac{\displaystyle{\frac{c^2}{v}}}{\displaystyle{\frac{mc^2}{h}}}
3
4 = \frac{h}{mv}
5 = \frac{h}{p}

$$\lambda = \frac{\frac{c^2}{v}}{\frac{mc^2}{h}} = \frac{h}{mv} = \frac{h}{p}$$