

# LaTeX 公式

## 版本号: V1.6.6

### 编者

中山大学 易鹏

华南理工大学 关舒文

### 目录

- LaTeX 公式
  - 1 公式环境
    - 1.1 行间公式
    - 1.2 行内公式
  - 2 公式的基本输入
    - 2.1 上下标
    - 2.2 括号和分隔符
    - 2.3 分数
    - 2.4 开方
    - 2.5 省略号与点
    - 2.6 矢量与方向
    - 2.7 积分
    - 2.8 极限
    - 2.9 累加、累乘运算
    - 2.10 希腊字母
    - 2.11 其它特殊字符
      - 1. 关系运算符
      - 2. 集合运算符
      - 3. 对数运算符
      - 4. 三角运算符
      - 5. 微积分运算符
      - 6. 逻辑运算符
      - 7. 戴帽符号
      - 8. 连线符号
      - 9. 箭头符号
  - 4 矩阵
    - 4.1 矩阵的基本用法
  - 5 方程组

- 6 公式优化
  - 6.1 数学字体
  - 6.2 自动编号与手动编号
  - 6.3 公式断行
  - 6.4 连分数
  - 6.5 公式中输入中文
  - 6.6 粗斜体

## 1 公式环境

无论使用哪种公式,都请在文档的导言区加上 `amsmath` 宏包,即 `\usepackage{amsmath}` .

### 1.1 行间公式

行间公式是指单独开一行来专门放置公式. 有三种方式:

- 使用 `$$` `$$`
- 使用 `\[` `\]`
- 使用 `equation` 环境

```
$$ x^2+2x+1=0 $$  
\[ x^2+2x+1=0 \]
```

其编译结果为

$$x^2 + 2x + 1 = 0$$

$$x^2 + 2x + 1 = 0$$

```
\begin{equation}  
x^2+2x+1=0  
\end{equation}
```

其编译结果为

$$x^2 + 2x + 1 = 0 \tag{1}$$

从上面两个编译结果来看,明显的, `equation` 环境比直接使用 `$$` `$$` 多了自动编号功能,如果后面习惯了用 `equation` 环境而不想某个公式自动编号,则使用 `equation*` 环境,例如

```
\begin{equation*}
x^2+2x+1=0
\end{equation*}
```

其编译结果为

$$x^2 + 2x + 1 = 0$$

对于 `equation` 环境还可以对公式进行标签和引用,例如

```
\begin{equation} \label{eq1} %设置标签,方便引用
x^2+2x+1=0
\end{equation}
公式\eqref{eq1}是自动编号的.
公式\ref{eq1}是自动编号的.
```

$$x^2 + 2x + 1 = 0 \tag{2}$$

公式(2)是自动编号的.

公式2是自动编号的.

公式的引用需要在 `equation` 环境中预先标签 `\label{XXX}`,后面引用的时候

用 `\eqref{XXX}` 或 `\ref{XXX}`,它们的区别在于有没有 `()`,后者常用于定理环境的引用,前者常用于公式环境,这两个 `xxx` 要一致(一个空格也不能差).

对于使用 `$$` `$$` 和使用 `\[ \]`,在LaTeX2e中我们较为推荐使用中括号的写法,这对间隔控制更友好.后者在定义上基本与 `equation*` 环境等价/

## 1.2 行内公式

顾名思义,行内公式就是公式和文字混排在一起,这个时候只需使用 `$` `$`,例如

物体的动能为 $E=\frac{1}{2}mv^2$ .

其编译结果为

物体的动能为 $E = \frac{1}{2}mv^2$ .

你会发现 $\frac{1}{2}$ 变小了,这是LaTeX在排版时为了让公式美观,更贴合文字的大小,所以调小了 $\frac{1}{2}$ 的大小,如果你想使得行内公式的显示和行间公式一样,则在公式的最前面添加 `\displaystyle` 即可,例如

物体的动能为 $\displaystyle E=\frac{1}{2}mv^2$ .

其编译结果为

物体的动能为 $E = \frac{1}{2}mv^2$ .

## 2 公式的基本输入

### 2.1 上下标

^ 表示上标, \_ 表示下标.如果上下标的内容多于一个字符,需要用 {} 将这些内容括成一个整体.上下标可以嵌套,也可以同时使用.例如

```
$$ x^{y^z}=(1+{\rm e}^x)^{-2xy^w} $$
```

其编译结果为

$$x^{y^z} = (1 + e^x)^{-2xy^w}$$

如果要在左右两边都有上下标,可以用 \sideset 命令.

例如

```
$$ \Delta S=S_2-S_1=\sideset{_R}{_{1}^{2}}{\int}\frac{{\rm d} Q}{T} $$
```

其编译结果为

$$\Delta S = S_2 - S_1 = \int_1^2 \frac{{\rm d} Q}{T}$$

从这个例子我们可以看出来, \sideset 的用法为

```
\sideset{左边的上下标}{右边的上下标}{被标号的部分}
```

### 2.2 括号和分隔符

()、[] 和 | 表示符号本身,使用 \{\} (转义字符) 来表示 {}.当要显示大号的括号或分隔符时,要用 \left 和 \right 命令使得括号的大小与公式高度匹配.

一些特殊的括号：

输入	显示	输入	显示
\angle	⟨	\rangle	⟩
\lceil	⌈	\rceil	⌋
\lfloor	⌊	\rfloor	⌋

输入	显示	输入	显示
<code>\lbrace</code>	{	<code>\rbrace</code>	}

例如：

```
$$
f(x,y,z) = 3y^2z \left( 3+\frac{7x+5}{1+y^2} \right)
$$
```

其编译结果为

$$f(x,y,z) = 3y^2z \left( 3 + \frac{7x + 5}{1 + y^2} \right)$$

**注意:** `\left` 和 `\right` 要同时使用，所以当使用单边括号的时候要用 `\left.` 或 `\right.` 作为不显示括号的一侧.

例如：

```
$$
\left. \frac{{\rm d}u}{{\rm d}x} \right|_{x=0}
$$
$$
\left[ 1,\frac{1}{2} \right)
$$
```

其编译结果为

$$\left. \frac{du}{dx} \right|_{x=0}$$

$$\left[ 1, \frac{1}{2} \right)$$

如果你需要将行内显示的分隔符也根据高度自动调整,可以使用 `\middle` 命令,例如

```


$$\left\langle q \left| \frac{\frac{x}{y}}{\frac{u}{v}} \right| p \right\rangle$$


```

其编译结果为

$$\left\langle q \left| \frac{\frac{x}{y}}{\frac{u}{v}} \right| p \right\rangle$$

$$\left( \frac{1}{2} \middle/ \frac{1}{3} \right)$$

## 2.3 分数

通常使用 `\frac {分子} {分母}` 命令产生一个分数,分数可嵌套.  
 便捷情况 (当分子和分母都仅有一个字符的情况) 可直接输入 `\frac ab` 来快速生成一个  $\frac{a}{b}$  (这种方法不仅适用于 `\frac` ).  
 如果分式很复杂,亦可使用 `{分子 \over 分母}` 命令 (不建议),此时分数仅有一层.

例如：

```


$$\frac{a-1}{b-1} \quad \text{and} \quad \frac{a+1}{b+1}$$


```

其编译结果为

$$\frac{a-1}{b-1} \quad and \quad \frac{a+1}{b+1}$$

对于行间公式环境中的分数,我们不仅可以使用 `\displaystyle` 以产生跨行的分数,还可以时使用 `\dfrac`  
 例如:

小的`\frac 12`和大的`\dfrac 12`

其编译结果为

小的 $\frac{1}{2}$ 和大的 $\frac{1}{2}$

注意在行间公式环境中,第一层分数将会以 `\displaystyle` 显示,第二层将以 `\textstyle` ,字体大小逐级递减, 故若需要创建大小一致的分式形式,请使用不会改变字体显示样式的 `\cfrac` (详见下文连分数)

$$x = a_0 + \frac{1^2}{a_1 + \frac{2^2}{a_2 + \frac{3^2}{a_3 + \frac{4^2}{a_4 + \cdots}}}}$$

## 2.4 开方

使用 `\sqrt[n]{被开方数}` 命令输入开方. 同时 `\sqrt` 命令亦可以使用简略的书写方法,但根指数依然要使用中括号.

例如:

```
$$
\sqrt{2} \quad \text{and} \quad \sqrt[n]{3}
$$
\sqrt[n]{m}
$$
```

其编译结果为

$$\sqrt{2} \quad \text{and} \quad \sqrt[n]{3}$$

$$\sqrt[n]{m}$$

## 2.5 省略号与点

省略号常用于矩阵环境和列举环境.

输入	显示	说明
<code>\cdot</code>	$\cdot$	点乘

输入	显示	说明
<code>\cdots</code>	...	水平省略号
<code>\vdots</code>	⋮	竖直省略号
<code>\ddots</code>	⋱	对角省略号
<code>\ldots</code>	...	跟文本底线对齐

例如：

```


$$
f(x_1,x_2,\underbrace{\ldots}_{\rm ldots} ,x_n) = x_1^2 + x_2^2 + \underbrace{\cdots}_{\rm cdots}_{\rm rdots}


```

其编译结果为

$$f(x_1,x_2,\underbrace{\ldots}_{\rm ldots},x_n) = x_1^2 + x_2^2 + \underbrace{\cdots}_{\rm cdots} + x_n^2$$

## 2.6 矢量与方向

使用 `\vec{矢量}` 来自动产生一个矢量.也可以使用 `\overrightarrow` 生成自动根据内容调节长度的向量箭头

例如：

```


$$
\vec{a} \cdot \vec{b}=0
$$

$$\overrightarrow{ABCD}, \vec{ABCD}$$



```

其编译结果为

$$\vec{a} \cdot \vec{b} = 0$$

$$\overrightarrow{ABCD}, \vec{ABCD}$$



例如：

```
$$
\overleftarrow{xy} \quad \text{and} \quad \overleftrightharpoonup{xy} \quad \text{and} \quad \overrightarrow{xy}
$$
```

其编译结果为

$\overleftarrow{xy} \quad \text{and} \quad \overleftrightharpoonup{xy} \quad \text{and} \quad \overrightarrow{xy}$

2.7 积分

使用 `\int_{积分下限}^{积分上限} {被积表达式}` 来输入一个积分.多重积分使用 `\iint` 或者其他.

例如：

```
$$
\int_0^{m^2} {x^2 \, \rm d} x
$$
```

其编译结果为

$$\int_0^{m^2} x^2 \, \mathrm{d} x$$

本例中 `{\rm}` 表示对括号部分的公式取消斜体, `\,` 表示一个小空格, 根据IEEE出版物规定,建议使用正体微分符号,被积变量与被积函数之间以空格相隔.

2.8 极限

使用 `\lim_{变量 \to 表达式} 表达式` 来输入一个极限.如有需求,可以更改 `\to` 符号至任意符号.

例如：

```
$$
\lim_{n \to +\infty} \frac{1}{n(n+1)} \quad \text{and} \quad \lim_{x \leftarrow \text{示例}} \frac{1}{n(
$$
```

其编译结果为

$$\lim_{n \rightarrow +\infty} \frac{1}{n(n+1)} \quad \text{and} \quad \lim_{x \leftarrow \text{示例}} \frac{1}{n(n+1)}$$

有时候我们可能会用到多重极限或者具有层叠结构的数学表达式,这时候我们可以使用 `\atop` 命令来构造层叠结构

```
$$ \lim_{z \rightarrow 0} \atop z \in D} f(x,y) $$
```

其编译结果为

$$\lim_{\substack{z \rightarrow 0 \\ z \in D}} f(x,y)$$

对于积分符号也同样适用

```
$$ \iint_{\limits_{-1 < x < 1} \atop -1 < y < 1}} f(x,y) \, \mathrm{d}x \times $$
```

$$\iint_{\substack{-1 < x < 1 \\ -1 < y < 1}} f(x,y) \, \mathrm{d}x$$

`\limits` 命令可以实现将下标放在符号正下方的功能.

## 2.9 累加、累乘运算

使用 `\sum_{下标表达式}^{上标表达式} {累加表达式}` 来输入一个累加.

与之类似,使用 `\prod` `\bigcup` `\bigcap` 来分别输入累乘、并集和交集.

此类符号在行内显示时上下标表达式将会移至右上角和右下角.

- 例子：

```
$$ \sum_{i=1}^n \frac{1}{i^2} \quad \text{and} \quad \prod_{i=1}^n \frac{1}{i^2} \quad \text{and} \quad \bigcup_{i=1}^2 R: $$
```

- 显示：

$$\sum_{i=1}^n \frac{1}{i^2} \quad \text{and} \quad \prod_{i=1}^n \frac{1}{i^2} \quad \text{and} \quad \bigcup_{i=1}^2 R$$

## 2.10 希腊字母

输入 `\小写希腊字母英文全称` 和 `\首字母大写希腊字母英文全称` 来分别输入小写和大写希腊字母。  
对于大写希腊字母与现有字母相同的,直接输入大写字母即可。

输入	显示	输入	显示	输入	显示	输入	显示
<code>\alpha</code>	$\alpha$	<code>A</code>	$A$	<code>\beta</code>	$\beta$	<code>B</code>	$B$
<code>\gamma</code>	$\gamma$	<code>\Gamma</code>	$\Gamma$	<code>\delta</code>	$\delta$	<code>\Delta</code>	$\Delta$
<code>\epsilon</code>	$\epsilon$	<code>E</code>	$E$	<code>\zeta</code>	$\zeta$	<code>Z</code>	$Z$
<code>\eta</code>	$\eta$	<code>H</code>	$H$	<code>\theta</code>	$\theta$	<code>\Theta</code>	$\Theta$
<code>\iota</code>	$\iota$	<code>I</code>	$I$	<code>\kappa</code>	$\kappa$	<code>K</code>	$K$
<code>\lambda</code>	$\lambda$	<code>\Lambda</code>	$\Lambda$	<code>\mu</code>	$\mu$	<code>M</code>	$M$
<code>\nu</code>	$\nu$	<code>N</code>	$N$	<code>\xi</code>	$\xi$	<code>\Xi</code>	$\Xi$
<code>o</code>	$o$	<code>O</code>	$O$	<code>\pi</code>	$\pi$	<code>\Pi</code>	$\Pi$
<code>\rho</code>	$\rho$	<code>P</code>	$P$	<code>\sigma</code>	$\sigma$	<code>\Sigma</code>	$\Sigma$
<code>\tau</code>	$\tau$	<code>T</code>	$T$	<code>\upsilon</code>	$\upsilon$	<code>\Upsilon</code>	$\Upsilon$
<code>\phi</code>	$\phi$	<code>\Phi</code>	$\Phi$	<code>\chi</code>	$\chi$	<code>X</code>	$X$
<code>\psi</code>	$\psi$	<code>\Psi</code>	$\Psi$	<code>\omega</code>	$\omega$	<code>\Omega</code>	$\Omega$

部分字母有变量专用形式,以 `\var-` 开头。

小写形式	大写形式	变量形式	显示
<code>\epsilon</code>	<code>E</code>	<code>\varepsilon</code>	$\epsilon \mid E \mid \varepsilon$
<code>\theta</code>	<code>\Theta</code>	<code>\vartheta</code>	$\theta \mid \Theta \mid \vartheta$
<code>\rho</code>	<code>P</code>	<code>\varrho</code>	$\rho \mid P \mid \varrho$
<code>\sigma</code>	<code>\Sigma</code>	<code>\varsigma</code>	$\sigma \mid \Sigma \mid \varsigma$
<code>\phi</code>	<code>\Phi</code>	<code>\varphi</code>	$\phi \mid \Phi \mid \varphi$

## 2.11 其它特殊字符

若需要显示更大或更小的字符,在符号前插入 `\large` 或 `\small` 命令。

若找不到需要的符号,使用 [Detexify](#) 来画出想要的符号.

也可以在LaTeX自带的宏包说明中找到相应的符号,见附件.

## 1. 关系运算符

输入	显示	输入	显示	输入	显示	输入	显示
<code>\pm</code>	$\pm$	<code>\times</code>	$\times$	<code>\div</code>	$\div$	<code>\mid</code>	$ $
<code>\nmid</code>	$\nmid$	<code>\cdot</code>	$\cdot$	<code>\circ</code>	$\circ$	<code>\ast</code>	$*$
<code>\bigodot</code>	$\bigodot$	<code>\bigotimes</code>	$\bigotimes$	<code>\bigoplus</code>	$\bigoplus$	<code>\leq</code>	$\leq$
<code>\geq</code>	$\geq$	<code>\neq</code>	$\neq$	<code>\approx</code>	$\approx$	<code>\equiv</code>	$\equiv$
<code>\sum</code>	$\sum$	<code>\prod</code>	$\prod$	<code>\coprod</code>	$\coprod$	<code>\backslash</code>	$\backslash$

## 2. 集合运算符

输入	显示	输入	显示	输入	显示
<code>\emptyset</code>	$\emptyset$	<code>\in</code>	$\in$	<code>\notin</code>	$\notin$
<code>\subset</code>	$\subset$	<code>\supset</code>	$\supset$	<code>\subseteq</code>	$\subseteq$
<code>\supseteq</code>	$\supseteq$	<code>\bigcap</code>	$\bigcap$	<code>\bigcup</code>	$\bigcup$
<code>\bigvee</code>	$\bigvee$	<code>\bigwedge</code>	$\bigwedge$	<code>\biguplus</code>	$\biguplus$

## 3. 对数运算符

输入	显示	输入	显示	输入	显示
<code>\log</code>	$\log$	<code>\lg</code>	$\lg$	<code>\ln</code>	$\ln$

## 4. 三角运算符

输入	显示	输入	显示	输入	显示
<code>30^\circ</code>	$30^\circ$	<code>\bot</code>	$\bot$	<code>\angle A</code>	$\angle A$
<code>\sin</code>	$\sin$	<code>\cos</code>	$\cos$	<code>\tan</code>	$\tan$
<code>\csc</code>	$\csc$	<code>\sec</code>	$\sec$	<code>\cot</code>	$\cot$

## 5. 微积分运算符

输入	显示	输入	显示	输入	显示
<code>\int</code>	$\int$	<code>\iint</code>	$\iint$	<code>\iiint</code>	$\iiint$
<code>\iiint</code>	$\iiint$	<code>\oint</code>	$\oint$	<code>\prime</code>	$'$
<code>\lim</code>	$\lim$	<code>\infty</code>	$\infty$	<code>\nabla</code>	$\nabla$

6. 逻辑运算符

输入	显示	输入	显示	输入	显示
<code>\because</code>	$\because$	<code>\therefore</code>	$\therefore$		
<code>\forall</code>	$\forall$	<code>\exists</code>	$\exists$	<code>\not\subset</code>	$\not\subset$
<code>\not&lt;</code>	$\not<$	<code>\not&gt;</code>	$\not>$	<code>\not=</code>	$\neq$

7. 戴帽符号

输入	显示	输入	显示
<code>\hat{xy}</code>	$\hat{xy}$	<code>\widehat{xyz}</code>	$\widehat{xyz}$
<code>\tilde{xy}</code>	$\tilde{xy}$	<code>\widetilde{xyz}</code>	$\widetilde{xyz}$
<code>\check{x}</code>	$\check{x}$	<code>\breve{y}</code>	$\breve{y}$
<code>\grave{x}</code>	$\grave{x}$	<code>\acute{y}</code>	$\acute{y}$

8. 连线符号

输入	显示
<code>\fbox{a+b+c+d}</code>	<div style="border: 1px solid black; padding: 2px; display: inline-block;"><math>a+b+c+d</math></div>
<code>\overleftarrow{a+b+c+d}</code>	$\overleftarrow{a+b+c+d}$
<code>\overrightarrow{a+b+c+d}</code>	$\overrightarrow{a+b+c+d}$
<code>\overleftrightarrow{a+b+c+d}</code>	$\overleftrightarrow{a+b+c+d}$
<code>\underleftarrow{a+b+c+d}</code>	$\underleftarrow{a+b+c+d}$
<code>\underrightarrow{a+b+c+d}</code>	$\underrightarrow{a+b+c+d}$

输入	显示
<code>\underleftrightharrow{a+b+c+d}</code>	$\underleftrightharrow{a + b + c + d}$
<code>\overline{a+b+c+d}</code>	$\overline{a + b + c + d}$
<code>\underline{a+b+c+d}</code>	$\underline{a + b + c + d}$
<code>\overbrace{a+b+c+d}^{\text{Sample}}</code>	$\overbrace{a + b + c + d}^{\text{Sample}}$
<code>\underbrace{a+b+c+d}_{\text{Sample}}</code>	$\underbrace{a + b + c + d}_{\text{Sample}}$
<code>\overbrace{a+\underbrace{b+c}_{1.0}+d}^{2.0}</code>	$\overbrace{a + \underbrace{b + c}_{1.0} + d}^{2.0}$
<code>\underbrace{a\cdot a\cdots a}_{b\text{ times}}</code>	$\underbrace{a \cdot a \cdots a}_{b \text{ times}}$
<code>\underrightarrow{1^{\circ}\text{C}/\text{min}}</code>	$\underrightarrow{1^{\circ}\text{C}/\text{min}}$

9. 箭头符号

推荐使用符号：

输入	显示	输入	显示	输入	显示
<code>\to</code>	$\rightarrow$	<code>\mapsto</code>	$\mapsto$		
<code>\implies</code>	$\implies$	<code>\iff</code>	$\iff$	<code>\impliedby</code>	$\impliedby$

其它可用符号：

输入	显示	输入	显示
<code>\uparrow</code>	$\uparrow$	<code>\Uparrow</code>	$\Uparrow$
<code>\downarrow</code>	$\downarrow$	<code>\Downarrow</code>	$\Downarrow$
<code>\leftarrow</code>	$\leftarrow$	<code>\Leftarrow</code>	$\Leftarrow$
<code>\rightarrow</code>	$\rightarrow$	<code>\Rightarrow</code>	$\Rightarrow$
<code>\leftrightarrow</code>	$\leftrightarrow$	<code>\Leftrightarrow</code>	$\Leftrightarrow$

输入	显示	输入	显示
<code>\longleftarrow</code>	$\longleftarrow$	<code>\Longleftarrow</code>	$\Longleftarrow$
<code>\longrightarrow</code>	$\longrightarrow$	<code>\Longrightarrow</code>	$\Longrightarrow$
<code>\longleftrightarrow</code>	$\longleftrightarrow$	<code>\Longleftrightarrow</code>	$\Longleftrightarrow$

## 4 矩阵

### 4.1 矩阵的基本用法

矩阵有很多的打法,这里只讲述最常用的一种,先举个例子.

```
$$
\left[ %左括号
\begin{array}[cccc] %[cccc]表示的是一行有4列,c代表的是居中
a_{11} & a_{12} & a_{13} & a_{14} \\ %每行的&分割列,\\代表换行
a_{21} & a_{22} & a_{23} & a_{24} \\
a_{31} & a_{32} & a_{33} & a_{34} \\
a_{41} & a_{42} & a_{43} & a_{44}
\end{array}
\right] %右括号
$$
```

其编译结果为

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

我们分析上面的代码, `\left[ \right]` 就是我们所看到的两个打括号,而中间的 `array` 环境是表格环境, `[cccc]` 表示的是一行有4列,c代表的是居中,c可以替换为l(左对齐)、r(右对齐).结合省略号,可以构造一个一般矩阵,例如

```


$$\left[
\begin{array}{cccc}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & \cdots & a_{nn}
\end{array}
\right]$$


```

其编译结果为

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

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：

```


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


```

其编译结果为

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

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



```


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


```

其编译结果为

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

## 5 方程组

方程组可以直接用 `cases` 环境直接打出,例如

```


$$\begin{cases} x+y=1 \\ x-y=1 \end{cases}$$


```

其编译结果如下

$$\begin{cases} x + y = 1 \\ x - y = 1 \end{cases}$$

## 6 公式优化

### 6.1 数学字体

若要对公式的某一部分字符进行字体转换,可以用 `\字体 {需转换的部分字符}` 命令,其中 `\字体` 部分可以参照下表选择合适的字体.一般情况下,公式默认为意大利体 *italic*.

示例中 **全部大写** 的字体仅大写可用.

输入	说明	显示	输入	说明	显示
<code>\mathrm</code>	罗马体	Sample	<code>\cal</code>	花体	<i>SAMPLE</i>
<code>\mathit</code>	意大利体	<i>Sample</i>	<code>\Bbb</code>	黑板粗体	SAMPLE
<code>\mathbf</code>	粗体	<b>Sample</b>	<code>\mit</code>	数学斜体	<i>SAMPLE</i>
<code>\mathsf</code>	等线体	Sample	<code>\scr</code>	手写体	<i>SAMPLE</i>
<code>\mathtt</code>	打字机体	Sample			
<code>\mathfrak</code>	旧德式字体	Sample			

转换字体十分常用,例如在积分中：

例如：

```

\begin{array}{cc}
\mathrm{Bad} & \mathrm{Better} \\
\hline
\int_0^1 x^2 dx & \int_0^1 x^2 \,d{x}
\end{array}

```

其编译结果为

BadBetter

$\int_0^1 x^2 dx$ 
 $\int_0^1 x^2 \,d{x}$

注意比较两个式子间  $dx$  与  $d{x}$  的不同.  
 使用 `\operatorname` 命令也可以达到相同的效果.

## 6.2 自动编号与手动编号

手动编号仅需在打完的公式后边加上 `\tag{编号}` ,例如

```


$$f\left(\left[\frac{1+\left\{x,y\right\}}{\left(\frac{x}{y}+\frac{y}{x}\right)\left(u+1\right)}+a\right]^{3/2}\right)$$

\tag{行标}

```

其编译结果为

$$f\left(\left[\frac{1+\{x,y\}}{\left(\frac{x}{y}+\frac{y}{x}\right)(u+1)}+a\right]^{3/2}\right) \tag{行标}$$

对于自动编号,前面已经提过,在 `equation` 环境中会自动编号,有时我们会将其与章节相关联起来,我们可以在 `tex` 文件的导言区加上`amsmath`宏包后再添加命令 `\numberwithin{equation}{section}` , 这样我们的公式就可以变为如下的形式

$$f\left(\left[\frac{1+\{x,y\}}{\left(\frac{x}{y}+\frac{y}{x}\right)(u+1)}+a\right]^{3/2}\right) \tag{1.1}$$

### 6.3 公式断行

公式有的时候太长或是连等式需要手动断行,这个时候可以使用 `split` 环境(需要和 `equation` 环境同时使用).例如,

```


$$\begin{split} 1+2+3+4+5+6+7+\\ +8+9+10+11+12+13=91. \end{split}$$


```

其编译结果为

$$1 + 2 + 3 + 4 + 5 + 6 + 7 \\ + 8 + 9 + 10 + 11 + 12 + 13 = 91.$$

当式子为连等式的时候需要对齐,这个时候我们会用上 & 来匹配每行的位置以达到对齐的效果.例如

```


$$\begin{split}
&1+2+3+4+5+6+7 \\
&\&=3+3+4+5+6+7 \ \backslash\backslash \\
&\&=6+4+5+6+7 \ \backslash\backslash \\
&\&=10+5+6+7 \ \backslash\backslash \\
&\&=15+6+7 \ \backslash\backslash \\
&\&=21+7 \ \backslash\backslash \\
&\&=28
\end{split}$$


```

其编译结果为

$$\begin{aligned}
1 + 2 + 3 + 4 + 5 + 6 + 7 &= 3 + 3 + 4 + 5 + 6 + 7 \\
&= 6 + 4 + 5 + 6 + 7 \\
&= 10 + 5 + 6 + 7 \\
&= 15 + 6 + 7 \\
&= 21 + 7 \\
&= 28
\end{aligned}$$

当然也可以写成这样

```


$$\begin{split}
&\&\backslash,\backslash,\backslash,\backslash,\backslash,\backslash,\backslash,\backslash,1+2+3+4+5+6+7 \\
&\&=3+3+4+5+6+7 \ \backslash\backslash \\
&\&=6+4+5+6+7 \ \backslash\backslash \\
&\&=10+5+6+7 \ \backslash\backslash \\
&\&=15+6+7 \ \backslash\backslash \\
&\&=21+7 \ \backslash\backslash \\
&\&=28
\end{split}$$


```

其编译结果为

$$\begin{aligned}
 &1 + 2 + 3 + 4 + 5 + 6 + 7 \\
 &= 3 + 3 + 4 + 5 + 6 + 7 \\
 &= 6 + 4 + 5 + 6 + 7 \\
 &= 10 + 5 + 6 + 7 \\
 &= 15 + 6 + 7 \\
 &= 21 + 7 \\
 &= 28
 \end{aligned}$$

以上稍微引入了多行公式的概念,下面我们详细讲讲多行公式.目前我们最常用的是 align 多行公式环境,它不需要外套 equation 环境,并且可以实现自动编号.

```

\begin{align}
a &= {} & b + c \\
&= {} & d + e + f + g + h + i \\
&+ j + k + l \notag \\
&+ m + n + o \\
&= {} & p + q + r + s
\end{align}

```

其编译结果为

$$a = b + c \tag{4.5}$$

$$\begin{aligned}
 &= d + e + f + g + h + i + j + k + l \\
 &\quad + m + n + o
 \end{aligned} \tag{4.6}$$

$$= p + q + r + s \tag{4.7}$$

若只需要居中对齐公式我们可以选择 gather 环境

```

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

```

其编译结果为

$$a = b + c \quad (4.10)$$

$$d = e + f + g \quad (4.11)$$

$$h + i = j + k$$

$$l + m = n \quad (4.12)$$

若需要对同一个align环境使用一个编号而不是多个编号,我们可以使用 `aligned` , `gathered` 环境(此时需要和 `equation` 环境同时使用).

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

其编译结果为

$$a = b + c$$

$$d = e + f + g$$

$$h + i = j + k$$

$$l + m = n$$

(4.13)

## 6.4 连分数

就像输入分式时使用 `\frac` 一样,使用 `\cfrac` 来创建一个连分数.例如,

```


$$x = a_0 + \cfrac{1^2}{a_1 + \cfrac{2^2}{a_2 + \cfrac{3^2}{a_3 + \cfrac{4^4}{a_4 + \cdots}}}}$$


```

其编译结果为

$$x = a_0 + \frac{1^2}{a_1 + \frac{2^2}{a_2 + \frac{3^2}{a_3 + \frac{4^4}{a_4 + \cdots}}}}$$

不要使用普通的 `\frac` 或 `\over` 来创建,否则会看起来 **很奇怪**.例如,

```


$$x = a_0 + \frac{1^2}{a_1 + \frac{2^2}{a_2 + \frac{3^2}{a_3 + \frac{4^4}{a_4 + \cdots}}}}$$


```

其编译结果为

$$x = a_0 + \frac{1^2}{a_1 + \frac{2^2}{a_2 + \frac{3^2}{a_3 + \frac{4^4}{a_4 + \cdots}}}}$$

当然,你可以使用 `\frac` 来表达连分数的 **紧缩记法**.例如,

```


$$x = a_0 + \frac{1^2}{a_1 + \frac{2^2}{a_2 + \frac{3^2}{a_3 + \frac{4^4}{a_4 + \cdots}}}}$$


```

其编译结果为

$$x = a_0 + \frac{1^2}{a_1 + \frac{2^2}{a_2 + \frac{3^2}{a_3 + \frac{4^4}{a_4 + \cdots}}}}$$

连分数通常都太大以至于不易排版,所以建议在连分数前后声明 `$$` 符号,或使用像 `[a0;a1,a2,a3,...]` 一样的紧缩记法.

# 6.5 公式中输入中文

使用 `\mbox{中文}` 命令输入中文.例如,

```
$$
\sqrt{4} \,, \mbox{或} \,, 2
$$
```

其编译结果为

$\sqrt{4}$  或 2

# 6.6 粗斜体

在给公式加粗体的时候,我们会发现字体变成正体了而不是斜体了.例如,

```
$$
\bf{xyz}
$$
```

其编译结果为

**xyz**

如果想导出粗斜体,则需要在导言区的部分导入宏包 `amsmath` 的时候再导入 `amsmath` 自带的加粗宏包 `bm` ,即 `\usepackage{amsmath,bm}` ,这个时候使用命令 `\bm{}` 即可,例如

```
\bm{xyz}
```

其编译结果为

***xyz***