

# 目 录

<b>7 数学公式排版的一些技巧</b>	<b>2</b>
7.1 数学排版的国际标准(略)	2
7.2 数学模式中的字体尺寸	2
7.3 数学模式中的参数	3
7.4 定理类环境的排版	4
7.5 巧妙使用阵列环境	6
7.6 多行公式左列问题	8
7.7 amsmath 宏包介绍	9
7.8 公式中的文本(amsmath)	10
7.9 单个公式(amsmath)	11
7.10 方程组	12
7.10.1 gather 环境	12
7.10.2 align 环境	12
7.10.3 flalign 环境	13
7.10.4 alignat 环境	13
7.10.5 gathered, aligned, alignedat 环境	14
7.10.6 cases 环境	14
7.11 amsmath 环境中的矩阵	15
7.12 多重数学符号	16
7.12.1 多重脚标	16
7.12.2 多重积分	16

---

7.12.3 重叠重音符号 . . . . .	16
7.12.4 省略号 . . . . .	16
7.13 分式 . . . . .	17
7.13.1 普通分式 . . . . .	17
7.13.2 连分式 . . . . .	17
7.13.3 二项式系数 . . . . .	17
7.13.4 自定义分式类命令 . . . . .	18
7.14 amsmath 中的函数(算子)名 . . . . .	19
7.14.1 已定义的函数名 . . . . .	19
7.14.2 定义新的函数名 . . . . .	19
7.15 amsmath 的其他功能 . . . . .	20
7.15.1 公式中的空白 . . . . .	20
7.15.2 调整根式指数的位置 . . . . .	20
7.15.3 调整公式编号的垂直位置 . . . . .	20
7.15.4 特殊的上下标(上下限) . . . . .	20
7.15.5 不可断行的区间符 . . . . .	21

## 第七章 数学公式排版的一些技巧

§ 7.1 数学排版的国际标准(略)

§ 7.2 数学模式中的字体尺寸

四种: `\displaystyle`, `\textstyle`, `\scriptstyle`, `\scriptscriptstyle`.

## § 7.3 数学模式中的参数

参数	含义	缺省值
<code>\arraycolsep</code>	<code>array</code> 环境中列间距的一半	5.0pt
<code>\jot</code>	在 <code>eqnarray</code> 和 <code>eqnarray*</code> 中增加或减小行间距	3.0pt
<code>\abovedisplayskip</code>	在长行间公式与上方文本之间插入的垂直距离	8.5pt plus 3.0pt minus 4.0pt
<code>\belowdisplayskip</code>	在长行间公式与下方文本之间插入的垂直距离	8.5pt plus 3.0pt minus 4.0pt
<code>\abovedisplayshortskip</code>	在短行间公式与上方文本之间插入的垂直距离	0.0pt plus 2.0pt
<code>\belowdisplayshortskip</code>	在短行间公式与下方文本之间插入的垂直距离	4.0pt plus 2.0pt minus 2.0pt
<code>\mathindent</code>	选用文档类选项 <code>fleqn</code> 后行间公式的缩进量	25.0pt

## 几点说明:

1. 以上长度都可以通过命令 `\setlength` 修改;
2. 行间公式的“长”“短”并不是指公式本身的长短, 是与其上方的文本相比较而言; 如果上方文本的结尾处在公式左边界的左边, 则该公式为短公式, 否则为长公式; 如下面的公式(7.1)是长公式, 而(7.2)是短公式.
3. 多行环境(如: `eqnarray` 和 `eqnarray*`)中的公式均是长公式;

---

There do not exist integers  $x, y, z$  and  $n > 2$  such that

$$x^n + y^n = z^n. \quad (7.1)$$

不存在使得

$$x^n + y^n = z^n \quad (7.2)$$

成立的整数  $x, y, z$  and  $n > 2$ .

---

## § 7.4 定理类环境的排版

定理类环境包括：定理、引理、推论、命题、定义等。

它们的共同特点：(1) 需要用特定的格式显示；(2) 自动编号。

实现方法：先使用 `\newtheorem` 命令定义一个定理类环境，然后调用。

`\newtheorem{定理环境名}{标题}[主计数器]`

---

```
\newtheorem{theorem}{Theorem}[chapter]
\begin{theorem}[Fermat]
There do not exist integers
 $x$ ,  $y$ ,  $z$  and  $n > 2$ 
such that  $x^n + y^n = z^n$ .
\end{theorem}
```

**Theorem 7.1** (Fermat). *There do not exist integers  $x$ ,  $y$ ,  $z$  and  $n > 2$  such that  $x^n + y^n = z^n$ .*

---

```
\newtheorem{dingli}{\hei 定理}[chapter]
\begin{dingli}{\hei 费马}{\upshape\kai
不存在使得  $x^n + y^n = z^n$  成立的整数
 $x$ ,  $y$ ,  $z$  and  $n > 2$ . }
\end{dingli}
```

**定理 7.1** (费马). 不存在使得  $x^n + y^n = z^n$  成立的整数  $x$ ,  $y$ ,  $z$  and  $n > 2$ .

若想让不同的定理环境使用同一个计数器编号，则可以使用 `\newtheorem` 的另一种用法：

`\newtheorem{定理环境名}[已定义的定理环境]{标题}`

该命令使得新定义的定理环境和选项中的“已定义的定理环境”使用同一个计数器。

---

```
\newtheorem{proposition}[theorem]{Proposition}
\begin{proposition}[Fermat]
There do not exist integers
 $x$ ,  $y$ ,  $z$  and  $n > 2$ 
such that  $x^n + y^n = z^n$ .
\end{proposition}
```

**Proposition 7.2** (Fermat). *There do not exist integers  $x$ ,  $y$ ,  $z$  and  $n > 2$  such that  $x^n + y^n = z^n$ .*

$\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ 提供了一个加强定理类环境的宏包 `amsthm`，增加了以下功能：

1. 添加了命令 `\newtheorem*`，用来创建不带编号的定理类环境；
2. 提供 3 中预定义的定理格式供调用：
  - `plain`: 标题和编号均为黑体，定理正文用斜体 (即为默认格式)
  - `definition`: 标题和编号均为黑体，定理正文用正常字体
  - `remark`: 标题和编号均为斜体，定理正文用正常字体
 选用格式的命令为: `\theoremstyle{格式}`
3. 用户可以用 `\newtheoremstyle` 创建自己的定理环境格式，具体用法参见 `thmtest.tex` 或 `amsthm.dtx`；

- 4. 命令 `\swapnumbers` 可使其后创建的定理环境中的序号打印在标题前面;
- 5. 提供了 `proof` 环境和 `\proof`, `\qed` 命令

```
\begin{proof}
Now we begin to prove the above theorem.
First,  $\cdots\ \ \cdots$ 
\end{proof}
```

*Proof.* Now we begin to prove the above theorem. First, ... .. □

其中 □ 是“证毕”记号. 对于中文证明, 可以使用中文标题“证”, 并利用 `\qedsymbol` 重新定义证毕记号.

```
\renewcommand{\qedsymbol}{[[证毕]]}
\begin{proof}[[\hei证]]
现在我们来证明上面的定理.
首先,  $\cdots\ \ \cdots$ 
\end{proof}
```

证. 现在我们来证明上面的定理. 首先, ... .. [证毕]

自定义定理类环境格式举例

```
\newtheoremstyle{mystyle}
  {3pt} % Space above
  {3pt} % Space below
  {\kai} % Body font
  {2em} % Indent amount (empty = no indent, \parindent = para indent)
  {\hei} % Thm head font
  {} % Punctuation after thm head
  {.5em} % Space after thm head: " " = normal interword space;
         % \newline = linebreak
  {} % Thm head spec (can be left empty, meaning 'normal')
```

```
\theoremstyle{mystyle}
\newtheorem{mydl}{定理\ }
\begin{mydl}[费马]
不存在使得  $x^n+y^n=z^n$  成立的整数
 $x$ ,  $y$ ,  $z$  and  $n>2$ .
\end{mydl}
```

定理 1 (费马) 不存在使得  $x^n + y^n = z^n$  成立的整数  $x, y, z$  and  $n > 2$ .

## § 7.5 巧妙使用阵列环境

这里主要通过一些例子来说明 `array` 环境的一些使用技巧。

注意：`array` 环境只能在数学模式中使用！

```

$$
|x|=\left\{\begin{array}{r}
x, & x\geq 0\\
-x, & x<0
\end{array}\right.
\right.
$$

```

$$|x| = \begin{cases} x, & x \geq 0 \\ -x, & x < 0 \end{cases}$$

- 三角形矩阵：

```

$$
\left(\begin{array}{cccc}
a_{11} & a_{12} & \cdots & a_{1n} \\
& a_{22} & \cdots & a_{2n} \\
& & \ddots & \vdots \\
\multicolumn{2}{c}{\raisebox{1.3ex}[0pt]{\Huge 0}} & & \\
& & & a_{nn}
\end{array}\right)
\end{array}\right)
$$

```

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ & a_{22} & \cdots & a_{2n} \\ & & \ddots & \vdots \\ \multicolumn{2}{c}{0} & & \\ & & & a_{nn} \end{pmatrix}$$

- 用命令 `\eqno` 编号 或 `\leqno` 编号 对方程进行人工编号：  
(如果想要自动编号，则应该使用 `equation` 或 `eqnarray` 环境)

```

$$
\left(\begin{array}{lcr}
lll & ccc & rrr \\
l & c & r
\end{array}\right) \eqno(5.3)
$$

```

$$\begin{pmatrix} lll & ccc & rrr \\ l & c & r \end{pmatrix} \quad (5.3)$$

```

$$
\left(\begin{array}{lcr}
lll & ccc & rrr \\
l & c & r
\end{array}\right) \leqno(*)
$$

```

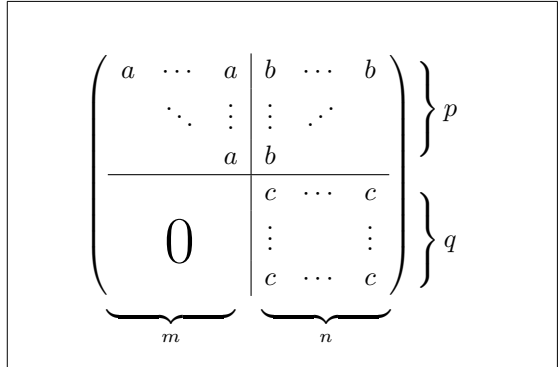
$$(*) \quad \begin{pmatrix} lll & ccc & rrr \\ l & c & r \end{pmatrix}$$

• 分块和加边矩阵:

```


$$\begin{array}{c}
\begin{array}{c}
\left( \begin{array}{ccc|ccc}
a & \cdots & a & b & \cdots & b \\
& & \vdots & & & \\
& & a & b & & \\
\hline
& & c & \cdots & c & \\
& & & \vdots & & \\
& & & c & \cdots & c
\end{array} \right) \\
\begin{array}{c}
\left. \begin{array}{c}
0 \\
\vdots \\
c
\end{array} \right\} q \\
\begin{array}{c}
\underbrace{\hspace{1.7cm}}_m \\
\underbrace{\hspace{1.7cm}}_n
\end{array}
\end{array}
\end{array}$$


```

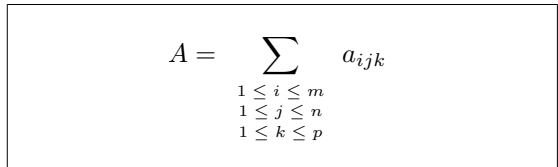


• 多行下标问题: 使用 array 环境

```


$$A = \sum_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n \\ 1 \leq k \leq p}} a_{ijk}$$


```

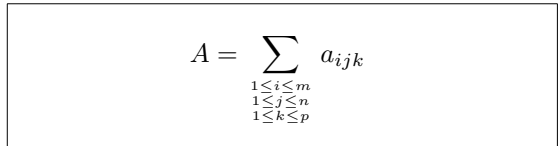


也可以使用堆积命令 \atop 实现:

```


$$A = \sum_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n \\ 1 \leq k \leq p}} a_{ijk}$$


```





- 多行下标问题: 若调用了 `amsmath` 宏包, 则可使用 `\substack` 命令

```
\begin{equation}
\sum_{\substack{1\leq i\leq m\\1\leq j\leq n\\1\leq k\leq p}} a_{ijk}
\end{equation}
```

$$\sum_{\substack{1\leq i\leq m \\ 1\leq j\leq n \\ 1\leq k\leq p}} a_{ijk} \quad (7.3)$$

或 `\subarray` 环境:

```
\begin{equation*}
\sum_{\begin{subarray}{l}1\leq i\leq m\\1\leq j\leq n\\1\leq k\leq p\end{subarray}} a_{ijk}
\end{equation*}
```

$$\sum_{\begin{subarray}{l}1\leq i\leq m \\ 1\leq j\leq n \\ 1\leq k\leq p\end{subarray}} a_{ijk}$$

## § 7.6 多行公式左列问题

通常是在第一行公式中使用 `\lefteqn{公式}` 命令, 它显示作为参数的公式, 但却被认为所占的宽度为零.

```
\begin{eqnarray*}
\lefteqn{w+x+y+z = } \\\
& & a+b+c+d+e+f+g+ \\\
& & h+i+j+k+l+m+n
\end{eqnarray*}
```

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

## § 7.7 amsmath 宏包介绍

这里介绍的 `amsmath` 宏包为 2.13 版本(2000/07/18)。它提供下列数学环境：

```
equation   align   gather   flalign   multline   alignat   split
equation*  align*  gather*  flalign*  multline*  alignat*
```

几点说明：

1. 其中带 `*` 的环境不自动编号；
2. 自动编号的多行公式环境中每行都编号，在 `\\` 之前加入 `\notag` 可使该行不带编号；
3. 可用 `\tag{编号}` 人为指定带圆括号的标号，`\tag*{编号}` 产生不带圆括号的标号；
4. 除 `split` 外，其它环境都进入行间模式；`split` 必须放在其他数学环境中使用；
5. `eqnarray` 仍然有效，也可以用 `align` 或 `equation+split` 代替，后两者效果更好。

调用 `amsmath` 宏包的方法是在导言区使用命令：`\usepackage[选项]{amsmath}`

其中选项通常不用写，即使用默认选项。具体选项有（每项中的前者为默认选项）：

- `centertags`, `tbtags`: 设置 `split` 环境中标号的位置, 默认为居中, 使用选项 `tbtags` 时为: 左 `top` 右 `bottom`.
  - `sumlimits`, `nosumlimits`: 求和号的上下限的默认排版方式.
  - `nointlimits`, `intlimits`: 积分号的上下限的默认排版方式.
  - `namelimits`, `nonamelimits`: 行间公式中函数名上下限的默认排版方式.  
以上三个选项中, 上下限的排版方式可通过命令 `\limits` 和 `\nolimits` 修改
  - `reqno`, `lenqo`: 行间公式编号的位置
  - `fleqn`: 使行间公式左对齐, 缩进量由 `\mathindent` 确定。不使用该选项时, 行间公式默认为居中对齐。
-

## § 7.8 公式中的文本(amsmath)

命令: `\text{文本}` 和 `\intertext{文本}`

作用: 在数学公式中插入文本, 能根据位置(如角标等)自动调整大小。

`\intertext{文本}` 是在多行公式中的两行之间插入文本, 位于其上下方的两个公式仍在同一个数学环境中。

```
\newcommand{\mi}{\mathrm{i}}
\begin{align*}
(x+\mi y)(x-\mi y)
&= x^2 + \mi xy - \mi xy - \mi^2 y^2 \\
&= x^2 + y^2 \\
\intertext{利用 $\mi^2=-1$, 还可得到}
(x+\mi y)^2 &= x^2 + 2\mi xy - y^2 \\
(x-\mi y)^2 &= x^2 - 2\mi xy - y^2
\end{align*}
```

$$\begin{aligned} (x + iy)(x - iy) &= x^2 + ixy - ixy - i^2 y^2 \\ &= x^2 + y^2 \end{aligned}$$

利用  $i^2 = -1$ , 还可得到

$$\begin{aligned} (x + iy)^2 &= x^2 + 2ixy - y^2 \\ (x - iy)^2 &= x^2 - 2ixy - y^2 \end{aligned}$$

## § 7.9 单个公式(amsmath)

单个公式可以只有一行,也可以多行, L<sup>A</sup>T<sub>E</sub>X 将其作为一个整体看待. 最多只有一个自动编号,也可以不带编号(使用带\* 环境).

将单个公式排成多行的实现方法有两种:

(1) 在 `equation` 中使用 `split` 环境. (2) 使用 `multline` 环境

```
\begin{equation}
\begin{split}
(a+b)^2 &= a^2 + b^2 + 2ab \\
(a-b)^2 &= a^2 + b^2 - 2ab \\
\frac{a^2-b^2}{a-b} &= a+b
\end{split}
\end{equation}
```

$$\begin{aligned} (a+b)^2 &= a^2 + b^2 + 2ab \\ (a-b)^2 &= a^2 + b^2 - 2ab \\ \frac{a^2-b^2}{a-b} &= a+b \end{aligned} \quad (7.4)$$

```
\begin{multline}
\framebox[.5\columnwidth]{%
  第一行(自动靠左)} \\
\framebox[.4\columnwidth]{自动居中} \\
\boxed{\sum_{k=1}^{100} k = 5050} \\
\shoveleft{\framebox[.7\columnwidth]{%
  强制靠左}} \\
\shoveright{\framebox[.7\columnwidth]{%
  强制靠右}} \\
\framebox[.5\columnwidth]{%
  末行(自动靠右)} \\
\end{multline}
```

### 几点说明:

- `split` 环境的作用与 `array` 类似,都必须放在数学模式中使用. 但 `split` 至多只能排两列,第一列居右对齐,第二列居左对齐,只有一列时为右对齐. `split` 环境中使用的是 `displaystyle` 字体尺寸,而 `array` 环境中使用的是 `textstyle`. 带编号时,编号垂直居中.
- `multline` 环境中第一行靠左,最后一行靠右,中间的行居中. 首末两行与左右边界的距离由 `\multlinegap` 的值确定. 公式编号位于首行左边,或末行右边. 使用命令 `\shoveleft` 或 `\shoveright` 可以使中间的公式靠左或靠右对齐. 如果使用 `fleqn` 选项,则中间的公式都靠左.
- `\boxed{公式}` 命令是给一个公式加边框.

## § 7.10 方程组

方程组由多个公式组成, 每个公式可占一行或多行.

单个公式和方程组的区分: 自动编号时只有一个编号的是单个公式.

### 7.10.1 gather 环境

每行只有一列, 不能指定对齐方式, 通常为居中对齐. 若在加载 `amsmath` 宏包时, 使用了 `fleqn` 选择, 则所以公式靠左对齐.

每个公式都自动编号, 可以在换行符 `\\` 前使用命令 `\notag` 或 `\nonumber` 使得某些行不带编号.

```
\begin{gather}
(a+b)^2 = a^2 + b^2 + 2ab\\
(a-b)^2 = a^2 + b^2 - 2ab\notag\\
\frac{a^2-b^2}{a-b} = a+b
\end{gather}
```

$$(a+b)^2 = a^2 + b^2 + 2ab \quad (7.6)$$

$$(a-b)^2 = a^2 + b^2 - 2ab$$

$$\frac{a^2 - b^2}{a - b} = a + b \quad (7.7)$$

### 7.10.2 align 环境

每行可以有多列, 对齐方式为: 奇数列靠右对齐, 偶数列靠左对齐. 左右紧靠在一起的两列合称为一个“列对”.

```
\begin{align*}
(x^n)' &= nx^{n-1} & (\sin x)' &= \cos x \\
(a^x)' &= a^x \ln a & (\cos x)' &= -\sin x \\
&& (\tan x)' &= \frac{1}{\cos^2 x}
\end{align*}
```

$$(x^n)' = nx^{n-1} \quad (\sin x)' = \cos x$$

$$(a^x)' = a^x \ln a \quad (\cos x)' = -\sin x$$

$$(\tan x)' = \frac{1}{\cos^2 x}$$

## 7.10.3 flalign 环境

使用方法与 `align` 完全一样, 区别在于: `flalign` 环境在输出时每行对应于偶数个 `&` 的位置上 (即相邻列对之间) 自动插入弹性长度, 以充满整行.

<pre>\begin{flalign*} (x^n)'&amp;= nx^{n-1} &amp; (\sin x)' &amp;= \cos x \\ (a^x)'&amp;= a^x \ln a &amp; (\cos x)' &amp;= -\sin x \\ &amp;&amp; (\tan x)' &amp;= \frac{1}{\cos^2 x} \\ \end{flalign*}</pre>	$(x^n)' = nx^{n-1}$ $(a^x)' = a^x \ln a$	$(\sin x)' = \cos x$ $(\cos x)' = -\sin x$ $(\tan x)' = \frac{1}{\cos^2 x}$
--	--	---

## 7.10.4 alignat 环境

同 `align`, 区别是: (1) 相邻列对之间的间距默认为 0, 这样就可以通过插入一定的空白长度, 使得各列对之间保持指定的间隔. (2) `alignat` 环境需要一个参数, 用来指定同一行中列对的个数. (可以大于实际列对个数, 但不能小于实际个数)

<pre>\begin{alignat}{2} (x^n)'&amp;= nx^{n-1} &amp; \hspace{20pt} &amp; (\sin x)' &amp;= \cos x \\ (\sin x)'&amp;= \cos x \\ (a^x)'&amp;= a^x \ln a &amp; (\cos x)' &amp;= -\sin x \\ &amp;&amp; (\tan x)' &amp;= \frac{1}{\cos^2 x} \\ &amp;&amp; &amp; \label{eq:x} \\ \end{alignat}</pre>	$(x^n)' = nx^{n-1} \quad (\sin x)' = \cos x \quad (7.8)$ $(a^x)' = a^x \ln a \quad (\cos x)' = -\sin x \quad (7.9)$ $(\tan x)' = \frac{1}{\cos^2 x} \quad (7.10)$
--	---

<pre>\begin{alignat}{2} y &amp;= f(x)+g(x) &amp; \quad \quad &amp; \text{(由 引理)} \\ &amp;= \sec^2 x &amp; \quad \quad &amp; \text{(由\eqref{eq:x}式)}</pre>	$y = f(x) + g(x) \quad \text{(由引理)} \quad (7.11)$ $= \sec^2 x \quad \text{(由(7.10)式)} \quad (7.12)$
---	---

**注:** 这里的 `\eqref` 与 `\ref` 的作用相同, 但输出时会自动被圆括号括起来.

## 7.10.5 gathered, aligned, alignat 环境

这些环境的语法和效果与不带 `ed` 的相应环境相同, 但整个公式只占有公式本身的实际宽度, 而不是整行. 他们必须放在其他数学模式中使用, 而且也不再自动编号.

```
\begin{equation}
\begin{aligned}
a &= b + c \quad d = bb + cc \end{aligned}
\Longrightarrow
\begin{gathered}[b]
A = aa + bb \quad D = c + f \end{gathered}
\Longrightarrow
\begin{aligned}[t]
X &= A + aa \quad Y = D + d \end{aligned}
\end{equation}
```

$$\begin{array}{l}
 A = aa + bb \\
 a = b + c \quad \implies \quad D = c + f \quad \implies \quad X = A + aa \\
 d = bb + cc \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad Y = D + d \\
 \hspace{15em} (7.13)
 \end{array}$$

```
\begin{equation*}
\left. \begin{aligned}
B' &= -\partial \times E \\
E' &= \partial \times B - 4\pi j \end{aligned}
\right\} \quad \text{Maxwell 方程}
\end{equation*}
```

$$\left. \begin{array}{l}
 B' = -\partial \times E \\
 E' = \partial \times B - 4\pi j
 \end{array} \right\} \quad \text{Maxwell 方程}$$

**注:** 旧版本中, `aligned` 环境中每行只能有两列, 且没有 `alignat` 环境.

## 7.10.6 cases 环境

`cases` 环境专门用于排版左侧带有花括号的方程组. 使用时必须放在数学模式中.

```
\begin{equation}
f(x) =
\begin{cases}
1 & -1 < x < 1 \\
0 & \text{其他 } x \end{cases}
\end{equation}
```

$$f(x) = \begin{cases} 1 & -1 < x < 1 \\ 0 & \text{其他 } x \end{cases} \quad (7.14)$$

## § 7.11 amsmath 环境中的矩阵

专门用于输出矩阵的环境: `matrix`, `pmatrix`, `bmatrix`, `Bmatrix`, `vmatrix`, `Vmatrix`.

\$\$

`\begin{matrix}`

`\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix}` &

`\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}` &

`\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}`

`\\[12pt]`

`\begin{Bmatrix} 1 & 2 \\ 3 & 4 \end{Bmatrix}` &

`\begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix}` &

`\begin{Vmatrix} 1 & 2 \\ 3 & 4 \end{Vmatrix}`

`\end{matrix}`

\$\$

$$\begin{array}{ccc} \begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix} & \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} & \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \\ \begin{Bmatrix} 1 & 2 \\ 3 & 4 \end{Bmatrix} & \begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix} & \begin{Vmatrix} 1 & 2 \\ 3 & 4 \end{Vmatrix} \end{array}$$

**几点说明:**

1. 这些环境必须放在数学模式中使用.
2. 默认矩阵最多有10列, 可以通过修改 `MaxMatrixCols` 的值来改变最大列数的默认值.
3. 矩阵各列居中对齐, 如果要改变对齐方式, 需使用 `array` 环境.
4. 对于行内矩阵, 可以使用 `smallmatrix` 环境.
5. 本环境的例子为 [9-11-1.tex](#)

这个矩阵 `$$\big(\begin{smallmatrix} a & b & c \\ x & y & z \end{smallmatrix}\big)$$`

`a & b & c \\ x & y & z`

`\end{smallmatrix}\big)$$`

是行内矩阵.

这个矩阵  $\begin{pmatrix} a & b & c \\ x & y & z \end{pmatrix}$  是行内矩阵.



## § 7.12 多重数学符号

### 7.12.1 多重脚标

(见前面的[巧妙使用阵列环境](#), 例子见 [9-12-1.tex](#))

### 7.12.2 多重积分

有: `\iint`, `\iiint`, `\iiint`, `\idotsint`, 它们与普通积分命令 `\int` 一样具有两种尺寸, 上下限可以放在右侧, 也可以放在上下方.

```
\newcommand{\dif}{\mathrm{d}}
\begin{gather*}
\iint_D f(x,y)\,,\,\dif x\,\dif y\!\!\!/[3pt]
\iiint\limits_V f(x,y,z)\,,\,\dif V
\end{gather*}
```

$$\iint_D f(x,y) \, dx dy$$

$$\iiint_V f(x,y,z) \, dV$$

### 7.12.3 重叠重音符号

`amsmath` 环境提供了一组专门用于重叠的重音符号:

```
\Hat      \Bar      \dot      \Tilde    \Breve
\Check    \Acute    \Grave    \Vec      \Ddot
```

试比较 `$$\hat{\tilde{A}}$$`  
与 `$$\hat{\hat{A}}$$` 的区别

试比较  $\hat{\tilde{A}}$  与  $\hat{\hat{A}}$  的区别

### 7.12.4 省略号

1. `amsmath` 增加的三个点的省略号: `\dots`, `\dotssb`, `\dotsc`, `\dotsm`, `\dotsi`
2. 用于矩阵中跨越多列的长省略号: `\hdotsfor{列数}`

```
\begin{gather*}
x_1 + \dots + x_n\!\\
x_1, \dots, x_n\!\\
x_1 + \dots + x_k + \dots
\end{gather*}
```

$$x_1 + \cdots + x_n$$

$$x_1, \dots, x_n$$

$$x_1 + \cdots + x_k + \dots$$

```


$$\begin{pmatrix}
a_{11} & a_{12} & \dots & a_{1n} \\
a_{21} & a_{22} & \dots & a_{2n} \\
\hdotsfor{4} \\
a_{n1} & a_{n2} & \dots & a_{nn}
\end{pmatrix}$$


```

$$\begin{pmatrix}
a_{11} & a_{12} & \dots & a_{1n} \\
a_{21} & a_{22} & \dots & a_{2n} \\
\hdotsfor{4} \\
a_{n1} & a_{n2} & \dots & a_{nn}
\end{pmatrix}$$

### § 7.13 分式

#### 7.13.1 普通分式

行内分式: `\tfrac{分子}{分母}`, 行间分式: `\dfrac{分子}{分母}`.

```

放在行内公式中显示为:
 $\tfrac{a}{b}$ , \quad  $\dfrac{a}{b}$ ,
\quad 放在行间公式中显示为:
 $\tfrac{a}{b}$ , \quad  $\dfrac{a}{b}$ 

```

放在行内公式中显示为:  $\frac{a}{b}$ ,  $\frac{a}{b}$ , 放在  
行间公式中显示为:

$$\frac{a}{b}, \quad \frac{a}{b}$$

#### 7.13.2 连分式

`\cfrac[位置]{分子}{分母}`: 选项“位置”可以取 l(与分数线左对齐) 或 r(与分数线右对齐), 缺省为居中. 分子分母自动使用相同大小的字体.

```


$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{a_4}}}}$$


```

$$a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \frac{1}{a_4}}}}$$

#### 7.13.3 二项式系数

TEX中可以使用 `\atop` 和 `\choose` 命令, 其中前者不带圆括号.

amsmath 提供了代替 `\choose` 的命令: `\binom`, `\tbinom`, `\dbinom`, 其中前者自动选择字体大小, 中间的始终使用 `\textstyle` 字体尺寸, 最后一个始终使用 `\displaystyle` 字体尺寸.

```

\begin{gather}
{n+1 \atop k}, \quad \quad \quad \\
{n+1 \choose k}, \quad \quad \quad \\
\binom{n+1}{k}, \quad \quad \quad \\
\mathrm{tbinom}{n+1}{k}, \quad \quad \quad \\
\mathrm{dbinom}{n+1}{k}. \\
\end{gather}

```

$$\frac{n+1}{k}, \quad \binom{n+1}{k}, \quad (7.15)$$

$$\binom{n+1}{k}, \quad \binom{n+1}{k}, \quad \binom{n+1}{k}. \quad (7.16)$$

#### 7.13.4 自定义分式类命令

`\genfrac{左定界符}{右定界符}{线的粗细}{字体尺寸}{分子}{分母}`

其中:

1. 线的粗细可以使用空白, 表示使用 L<sup>A</sup>T<sub>E</sub>X 标准的分数线.
2. 字体尺寸可以是空白或 0 到 3 之间的一个数, 其中空白表示自动选择, 0,1,2,3 分别表示 `\displaystyle`, `\textstyle`, `\scriptstyle`, `\scriptscriptstyle`.

```

\begin{gather*}
\genfrac{}{}{}{n+1}{k}, \\
\quad \genfrac{}{}{1}{n+1}{k} \\
\quad \genfrac{}{}{2}{n+1}{k} \\
\genfrac{}{0pt}{}{n+1}{k} \\
\quad \genfrac{()}{0pt}{0}{n+1}{k} \\
\quad \genfrac{()}{0pt}{1}{n+1}{k} \\
\end{gather*}

```

$$\frac{n+1}{k}, \quad \frac{n+1}{k} \quad \frac{n+1}{k}$$

$$\frac{n+1}{k} \quad \binom{n+1}{k} \quad \binom{n+1}{k}$$

### § 7.14 amsmath 中的函数(算子)名

数学公式中的函数名通常使用正体, 并在函数名的左右两边带有适当的空白.

#### 7.14.1 已定义的函数名

<code>arccos \arccos</code>	<code>coth \coth</code>	<code>hom \hom</code>	<code>ln \ln</code>	<code>sinh \sinh</code>
<code>arcsin \arcsin</code>	<code>csc \csc</code>	<code>inf \inf</code>	<code>log \log</code>	<code>sup \sup</code>
<code>arctan \arctan</code>	<code>deg \deg</code>	<code>ker \ker</code>	<code>max \max</code>	<code>tan \tan</code>
<code>arg \arg</code>	<code>det \det</code>	<code>lg \lg</code>	<code>min \min</code>	<code>tanh \tanh</code>
<code>cos \cos</code>	<code>dim \dim</code>	<code>lim \lim</code>	<code>Pr \Pr</code>	<code>mod \bmod</code>
<code>cosh \cos</code>	<code>exp \exp</code>	<code>lim inf \liminf</code>	<code>sec \sec</code>	<code>(mod) \pmod</code>
<code>cot \cot</code>	<code>gcd \gcd</code>	<code>lim sup \limsup</code>	<code>sin \sin</code>	<code>injlim \injlim</code>
<code>projlim \projlim</code>	<code>\overline{\lim} \varlimsup</code>	<code>\varinjlim</code>	<code>\varliminf</code>	<code>\varprojlim</code>

#### 7.14.2 定义新的函数名

`\DeclareMathOperator{新的函数命令}{函数名}`

`\DeclareMathOperator*{新的函数命令}{函数名}`

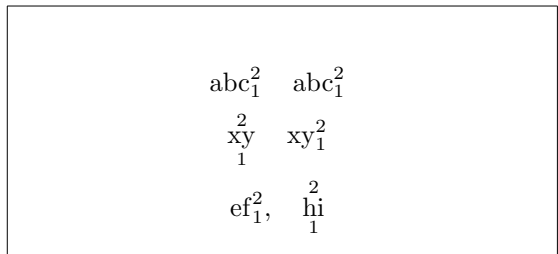
##### 几点说明:

1. 这两个命令必须放在导言区.
2. 带星号形式定义的函数名, 会根据环境或使用命令 `\limits` 将上下限放置在函数名的上方或下方.
3. 可以用命令 `\operatorname{函数名}` 或 `\operatorname*{函数名}` 定义临时使用的函数名.

```

\begin{gather*}
% 下面两个命令放在导言区
% \DeclareMathOperator{abc}{abc}
% \DeclareMathOperator*{xy}{xy}
\abc_1^2\quad \abc\limits_1^2 \quad
\xy_1^2\quad \xy\nolimits_1^2 \quad
\operatorname{ef}_1^2,\quad
\operatorname*{hi}_1^2
\end{gather*}

```



## § 7.15 amsmath 的其他功能

### 7.15.1 公式中的空白

`\mspace{数mu}`: 其中单位 `mu` 是固定的,  $1\text{ mu}=1/18\text{ em}$ , 不能使用其它单位, 也不能省略.

其他空白命令:

```

\, (\thinspace)
\! (\negthinspace)
\: (\medspace)      \negmedspace
\; (\thickspace)    \negthickspace
\lrcorner
\quad                \qquad

```

### 7.15.2 调整根式指数的位置

`\leftroot`数: 指数的左右移动, 不带单位, 正值为左移, 负值为右移.

`\uproot`数: 指数的上下移动, 不带单位, 正值为上移, 负值为下移.

试比较下面两种显示的差别:

```

$$
\sqrt[\beta]{f(x)}\quad
\sqrt[\leftroot{-2}\uproot{2}\beta]{f(x)}
$$

```

试比较下面两种显示的差别:

$$\sqrt[\beta]{f(x)} \quad \sqrt[\beta]{f(x)}$$

### 7.15.3 调整公式编号的垂直位置

`\raisetag`高度: 高度要带单位, 正值为上移, 负值为下移. 主要用于当一个公式太长时, 公式编号将独占一行被排在公式的下方, 此时通过该命令可以减少公式编号行与相应公式行之间的间隔.

```

\begin{gather}
f(x)=a_0 +a_1x +a_2x^2 +\dots +a_ny\
f(x)=a_0 +a_1x +a_2x^2 +\dots +a_ny
\raisetag{3pt}
\end{gather}

```

$$f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_ny \quad (7.17)$$

$$f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_ny \quad (7.18)$$

### 7.15.4 特殊的上下标(上下限)

`\sideset{左侧脚标}{右侧脚标}主体符号`: 在主体符号两侧放置上下标.

`\overset{上标}主体符号`: 在主体符号的正上方放置上标.

`\underset{下标}主体符号`: 在主体符号的正下方放置下标.

\$\$

```
\sideset{^a\dag}{^*_b}\sum\qqquad
\overset{abc}{XY}\qqquad
\underset{*}{Z}
```

$$\overset{a}{\underset{\dagger}{\sum}}^* \quad \overset{abc}{XY} \quad \underset{*}{Z}$$

\$\$

下面的命令分别用于在表达式的上方或下方放置能够自动伸缩的箭头.

`\overleftarrow{表达式}` `\underleftarrow{表达式}` (7.19)

`\overrightarrow{表达式}` `\underrightarrow{表达式}` (7.20)

`\overleftrightarrow{表达式}` `\underleftrightarrow{表达式}` (7.21)

命令

`\xleftarrow[下方表达式]{上方表达式}` 和 `\xrightarrow[下方表达式]{上方表达式}`

则在中间位置画水平箭头, 把上方表达式放在箭头上, 如果存在下方表达式, 则将它放置在箭头下方.

\$\$

```
\overleftrightarrow{ABCDE} =
  \underrightarrow{ABC}
+ \overleftarrow{uvwxyz}
```

$$\overleftarrow{ABCDE} = \underrightarrow{ABC} + \overleftarrow{uvwxyz}$$

\$\$

### 7.15.5 不可断行的区间符

为了避免区间符与其后面符号之间可能产生的断行, 应在区间符前加上命令 `\nobreakdash`.

调查 10\nobreakdash--15 岁的学生

调查 10–15 岁的学生