

一份简短的 L^AT_EX 数学指南

译者: 八一

排版协作: <https://www.latexstudio.net>

American Mathematical Society

版本 2.0 (2019/8/14), 目前可从以下链接获得

<https://www.ams.org/tex/amslatex>

<http://mirrors.sjtug.sjtu.edu.cn/ctan/info/short-math-guide/>

目录

1 背景	4
2 行内数学符号和行间等式	4
2.1 数学环境	4
2.2 自动编号和交叉引用	6
3 数学符号和数学字体	8
3.1 数学符号类	8
3.2 符号的有意省略	8
3.3 字母和数字	9
3.3.1 拉丁字母和阿拉伯数字	9
3.3.2 希腊字母	9
3.3.3 其他“基本”字母符号	10
3.3.4 数学字体切换	10
3.3.5 黑板粗体字母	11
3.3.6 手写体字母	12
3.3.7 花体字母	12
3.3.8 哥特体字母 (eufm)	12
3.4 其他简单符号	12
3.5 运算符号	13
3.6 关系符号: $< = > \succsim$ 与变式	14
3.7 关系符号: 箭头	15
3.8 关系符号: 杂项	16
3.9 累积运算符	16
3.10 标点符号	17
3.11 配对分隔符	17

一份简短的 \LaTeX 数学指南, 版本 2.0 (2019/8/14)	2
3.12 非配对分隔符	17
3.13 可扩展的垂直箭头	17
3.14 数学重音	18
3.15 命名操作符	18
4 符号	19
4.1 顶部和底部装饰	19
4.2 可扩展箭头	19
4.3 符号粘贴	19
4.4 矩阵	19
4.5 数学间距命令	20
4.6 点	20
4.7 不中断的破折号	21
4.8 根	21
4.9 盒装公式	21
5 分数及相关结构	21
5.1 <code>\frac</code> , <code>\dfrac</code> , 与 <code>\tfrac</code> 命令	21
5.2 <code>\binom</code> , <code>\dbinom</code> , 和 <code>\tbinom</code> 命令	22
5.3 <code>\genfrac</code> 命令	22
5.4 连分数	22
6 分隔符	23
6.1 分隔符大小	23
6.2 竖条符号	23
7 <code>\text</code> 命令	24
7.1 与 <code>\mod</code> 相关	24
8 积分与求和	24
8.1 改变限制的位置	24
8.2 多重积分	24
8.3 多行下标和上标	25
8.4 <code>\sideset</code> 命令	25
9 改变公式中元素的大小	25
10 其它宏包组合	26

致谢及未来工作计划

感谢所有提供建议、援助和鼓励的世卫组织。特别感谢 David Carlisle 修复了相关旧的宏, 以及 Jennifer Wright Sharp 在 AMS 风格中应用了一致的编辑。

未来版本的计划包括添加索引。

关于错误的报告和改进建议应发送到

tech-support@ams.org.

1. 背景

这是对 \LaTeX 和两个用于**数学公式书写**的扩展包推荐特性的简要总结, 需要更深入细节的读者可以参考书目中列出的资源, 特别是 [Lam]、[AMUG] 和 [LFG]。假设您对标准 \LaTeX 术语有一定的熟悉程度, 如果你需要对命令、可选参数、环境、包等的含义进行刷新, 请参见 [Lam]。

如果您使用美国数学协会发布的 \LaTeX 和 `amssymb` 以及 `amsmath` 两个扩展包, 那么这里描述的大多数特性都是可用的。因此此文档的源文件从

```
\documentclass{article}
\usepackage{amssymb,amsmath}
```

对于数学符号使用相对较少的文档, 可能会遗漏 `amssymb` 包; 在 Section 3 中, 需要 `amssymb` 的符号用^a 或^b (`font msam` 或 `msbm`) 标记。在 Section 3.3 中, 包含了一些额外的字体; 在那里标识了必要的包。

在其他包中发现的许多值得注意的特性没有在这里介绍; 请参见 Section 10。关于数学符号, 请特别注意, 这里给出的列表并不是全面的, 而是为了说明用户通常会发现在他们的 \LaTeX 系统中已经存在的、无需安装任何其他字体或做其他设置工作就可以使用。

如果您需要这里没有显示的符号, 您可能想咨询 *The Comprehensive \LaTeX Symbol List* [CLSL]。如果您的 \LaTeX 安装是基于 \TeX Live, 并包括文档, 列表也可以通过输入 `texdoc comprehensive` 在系统提示。

2. 行内数学符号和行间等式

2.1. 数学环境. 在 \LaTeX 中输入和退出数学模式通常使用以下命令和环境完成。

行内公式	行间等式	
$\$ \dots \$$	<code>\[\dots \]</code>	unnumbered
$\backslash(\dots \backslash)$	<code>\begin{equation*}</code>	unnumbered
	...	
	<code>\end{equation*}</code>	
	<code>\begin{equation}</code>	automatically
	...	numbered
	<code>\end{equation}</code>	

Note 1. 不要在文本和显示的等式之间留下空行。这允许在该位置进行分页, 这是一种糟糕的样式。它还会导致文本与显示之间的间距不正确, 一般情况下会比实际值大。如果需要在输入中插入一个可视的断点, 请在开头插入一行只包含一个 % 的行。仅当要显示新段落时, 才在显示文本和后续文本之间留出空行。

Note 2. 不要在输入 (`\[\dots \]`, `equation`, etc.) 将多个显示结构分组。相反使用带有子结构 (`split`, `aligned` 等) 的多行结构。

Note 3. 另一种环境 `\begin{math} \dots \end{math}` 与 `\begin{displaymath} \dots \end{displaymath}` 在练习中很少需要, 强烈反对使用默认的 \TeX 符号 `$$ \dots $$` 来显示方程。尽管在 \LaTeX 中并没有明确禁止它, 但在 \LaTeX 书中并没有将它作为 \LaTeX 命令集的一部分进行记录, 并且它干扰了各种特性的正确操作, 比如 `fleqn` 选项。

表 1: 多线方程和方程组 (竖线表示标称边缘).

<code>\begin{equation}\label{xx}</code>			
<code>\begin{split}</code>			
<code>a& =b+c-d\\</code>		$a = b + c - d$	
<code>& \quad +e-f\\</code>		$+ e - f$	(1.1)
<code>& =g+h\\</code>		$= g + h$	
<code>& =i</code>		$= i$	
<code>\end{split}</code>			
<code>\end{equation}</code>			
<code>\begin{multline}</code>			
<code>a+b+c+d+e+f\\</code>	$a + b + c + d + e + f$		
<code>+i+j+k+l+m+n\\</code>	$+ i + j + k + l + m + n$		
<code>+o+p+q+r+s</code>	$+ o + p + q + r + s$	(1.2)	
<code>\end{multline}</code>			
<code>\begin{gather}</code>			
<code>a_1=b_1+c_1\\</code>	$a_1 = b_1 + c_1$	(1.3)	
<code>a_2=b_2+c_2-d_2+e_2</code>	$a_2 = b_2 + c_2 - d_2 + e_2$	(1.4)	
<code>\end{gather}</code>			
<code>\begin{align}</code>			
<code>a_1& =b_1+c_1\\</code>	$a_1 = b_1 + c_1$	(1.5)	
<code>a_2& =b_2+c_2-d_2+e_2</code>	$a_2 = b_2 + c_2 - d_2 + e_2$	(1.6)	
<code>\end{align}</code>			
<code>\begin{align}</code>			
<code>a_{11}& =b_{11}&</code>			
<code> a_{12}& =b_{12}\\</code>	$a_{11} = b_{11}$	$a_{12} = b_{12}$	(1.7)
<code>a_{21}& =b_{21}&</code>	$a_{21} = b_{21}$	$a_{22} = b_{22} + c_{22}$	(1.8)
<code> a_{22}& =b_{22}+c_{22}</code>			
<code>\end{align}</code>			
<code>\begin{alignat}{2}</code>			
<code>a_1& =b_1+c_1& & +e_1-f_1\\</code>	$a_1 = b_1 + c_1$	$+ e_1 - f_1$	(1.9)
<code>a_2& =b_2+c_2&{}-d_2+e_2</code>	$a_2 = b_2 + c_2 - d_2 + e_2$	(1.10)	
<code>\end{alignat}</code>			
<code>\begin{flalign}</code>			
<code>a_{11}& =b_{11}&</code>			
<code> a_{12}& =b_{12}\\</code>	$a_{11} = b_{11}$	$a_{12} = b_{12}$	(1.11)
<code>a_{21}& =b_{21}&</code>	$a_{21} = b_{21}$	$a_{22} = b_{22} + c_{22}$	(1.12)
<code> a_{22}& =b_{22}+c_{22}</code>			
<code>\end{flalign}</code>			

Note 1. 将 `*` 应用于任何主环境都会抑制等式编号的赋值。但是, `\tag` 可以用来应用一个可见的标签, 而 `\eqref` 可以用来引用这些手工标记的行。在从属环境中使用 `*` 或 `\tag` 是错误的。

Note 2. 这里的 `split` 环境是一种特殊的情况。它是一个从属环境, 可作为 `equation` 环境的内容, 也可以作为一个“line”在多个等式结构 (如 `align` 或 `gather`) 中的内容。

Note 3. 主要环境 `gather`、`align` 和 `alignat` 具有从属的“-ed”同 (`-ed`、`aligned` 和 `aligned dat`), 它们可以用作更复杂显示的组件, 或者在行内数学环境下使用。这些“-ed”环境可以使用可选参数 `[t]`、`[c]` 或 `[b]`。垂直定位。

Note 4. `flalign` 环境是全长度即 `full length`, 而不是左对齐, 这个环境经常被误解。主要是因为这个环境是从左边来开始计算全长度的。他的缩进值是由 `\multlinegap` 来设置。

Note 4. 强烈反对 `eqnarray` 和 `eqnarray*` 引用 [Lam] 中描述的环境是因为它们产生了不一致的等号间距, 并且不会试图通过等式编号来防止方程组的重叠。

处理方程组和多行方程的环境显示在 Table 1 中。

2.2. 自动编号和交叉引用. 使用 `equation` 环境可以得到一个自动编号的等式; 若要为交叉引用指定一个标签请使用 `\label` 命令:

```
\begin{equation}\label{reio}
...
\end{equation}
```

要得到对自动编号方程的交叉引用, 请使用 `\eqref` 命令:

```
... using equations~\eqref{ax1} and~\eqref{bz2}, we
can derive ...
```

上面的示例将生成如下内容

using equations (3.2) and (3.5), we can derive

换句话说, `\eqref{ax1}` 相当于 (`\ref{ax1}`), 但是 `\eqref` 生成的圆括号总是直立的。

给方程编号的形式是 $m.n$ (*section-number.equation-number*), 在你的导言区中使用 `\numberwithin` 命令:

```
\numberwithin{equation}{section}
```

有关自定义编号方法的详细信息, 请参考 [Lam, §6.3, §C.8.4].

这 `subequations` 环境提供了一种使用从属编号方案对组中的方程编号的方便方法。例如假设当前的方程编号是方程, 可写成

```
\begin{equation}\label{first}
a=b+c
\end{equation}
some intervening text
\begin{subequations}\label{grp}
\begin{align}
a&=b+c\label{second}\\
d&=e+f+g\label{third}\\
h&=i+j\label{fourth}
\end{align}
\end{subequations}
```

可得到

$$a = b + c \tag{2.1}$$

some intervening text

$$a = b + c \tag{2.2a}$$

$$d = e + f + g \tag{2.2b}$$

$$h = i + j \tag{2.2c}$$

在 `\begin{subequations}` 之后添加 `\label` 命令, 你可以得到对一级编号的引用; 以上例子中的 `\eqref{grp}` 将产生(2.2), 而 `\eqref{second}` 将产生(2.2a)。

这个例子见链接 <https://tex.stackexchange.com/questions/220001/> 显示上述示例的一个变式, 使得该编号是 (2.1), (2.1a), ..., 而不是 (2.1), (2.2a), 这是通过使用 `\tag` 等长成分的交叉引用来实现的。

3. 数学符号和数学字体

3.1. 数学符号类. 数学公式中的符号可分不同类别, 这些类别如果用单词来表示公式中每个符号的词性, 对于不同的符号类, 传统上使用一定的间距和位置提示来增加公式的可读性。

分类	记忆符	描述 (发音部分)	例子
0	Ord	simple/ordinary (“noun”)	$A \ 0 \ \Phi \ \infty$
1	Op	prefix operator	$\sum \ \prod \ f$
2	Bin	binary operator (conjunction)	$+ \cup \ \wedge$
3	Rel	relation/comparison (verb)	$= \ < \ \subset$
4	Open	left/opening delimiter	$(\ [\ \{ \ \langle$
5	Close	right/closing delimiter	$) \] \ } \ \rangle$
6	Punct	postfix/punctuation	$. \ ; \ !$

Note 1. 在 \TeX 中, 类 0 和其他类 7 之间的区别只与字体选择问题有关, 且无关紧要。

Note 2. class 2 (Bin) 的符号, 尤其是减号 $-$, 如果它们没有合适的左操作数 $-$, 例如在数学公式的开头或在开始的分隔符之后, 这样就会被 \TeX 自动打印为 class 0(无空格)。

一些符号的间距遵循传统而不是一般规则: 虽然 $/$ (从语义上讲) 是 class 2 的, 但是我们在斜杠左右没有空格的情况下写入 $k/2$, 而不是 $k / 2$ 。比较 $p|q \ p|q$ (无空格) 与 $p\mid q \ p \mid q$ (class-3 空格)。

在 $\TeX 2_{\epsilon}$ 字体选择 [LFG] 中讨论了定义新数学符号的正确方法。这里不可能给出有用的概要是因为首先需要理解字体规范的分支。但假设你知道有一种名为 `wncyr10` 的西里尔字体可用, 下面是一个简单的示例来演示如何定义 \TeX 命令来将该字体中的一个字母打印为数学符号:

```
% Declare that the combination of font attributes OT2/wncyr/m/n
% should select the wncyr font.
\DeclareFontShape{OT2}{wncyr}{m}{n}{<->wncyr10}{}
% Declare that the symbolic math font name "cyr" should resolve to
% OT2/wncyr/m/n.
\DeclareSymbolFont{cyr}{OT2}{wncyr}{m}{n}
% Declare that the command \Sh should print symbol 88 from the math font
% "cyr", and that the symbol class is 0 (= alphabetic = Ord).
\DeclareMathSymbol{\Sh}{\mathalpha}{cyr}{88}
```

3.2. 符号的有意省略. 下面的数学符号是在 \TeX 书 [Lam] 中提到的。本讨论中有意省略是因为当 `amssymb` 包被加载时它们会被等价的符号所替代。无论何时如果你正在使用 `amssymb` 包, 那么通过使用替代名你可能获得的唯一益处就不必要增加内容中所使用的字体数量。

`\Box`, see `\square` \square
`\Diamond`, see `\lozenge` \diamond
`\leadsto`, see `\rightsquigarrow` \rightsquigarrow
`\Join`, see `\bowtie` \bowtie
`\lhd`, see `\vartriangleleft` \triangleleft
`\unlhd`, see `\trianglelefteq` \trianglelefteq
`\rhd`, see `\vartriangleright` \triangleright
`\unrhd`, see `\trianglerighteq` \trianglerighteq

此外除了上表 \TeX 符号外, 还有很多符号可供 \TeX 使用。这个列表并不全面, 有关更全面的符号列表, 包括非数学符号如音节字母或 dingbats 字体, 请参见全面 \TeX 符号列表 [CLSL]。(完整字体表, 按字体名称排序, 包含在 \TeX Live: `texdoc rawtables` 提供的文档中。这些表不包括符号名称) 符号信息的另一个来源是 `unicode-math` 包; 详细看 [UCM]。

3.3. 字母和数字

3.3.1. 拉丁字母和阿拉伯数字

拉丁字母是简单符号, 属于类 0。在数学公式中它们的默认字体是斜体。

$A B C D E F G H I J K L M N O P Q R S T U V W X Y Z$
 $a b c d e f g h i j k l m n o p q r s t u v w x y z$

当在数学字母中为 i 或 j 添加重音时, 可以使用 `\imath` 和 `\jmath` 得到无点变体:

\imath `\imath` \jmath `\jmath` $\hat{\jmath}$ `\hat{\jmath}`

阿拉伯数字 0–9 也属于类 0。它们的默认字体是直立/罗马字体。

0 1 2 3 4 5 6 7 8 9

3.3.2. 希腊字母

如拉丁字母一样, 希腊字母是简单的符号, 属于类 0。出于模糊的历史原因, 在数学公式中小写希腊字母的默认字体是斜体, 而大写希腊字母的默认字体是直立/罗马字体。(然而在物理和化学等其他领域, 排印的传统是有些不同。) 在这个列表中没有出现的大写希腊字母与某些拉丁字母可得到相同的效果: A 表示 Alpha, B 表示 Beta 等等; 在小写字母列表中没有 `omicron` 命令, 因为它显示出来的效果与拉丁语 o 是相同。实际上

在数学公式中, 带有拉丁字母的希腊字母很少使用以避免混淆。

Γ \Gamma	α \alpha	μ \mu	ν \nu	F \digamma
Δ \Delta	β \beta		ξ \xi	ε \varepsilon
Λ \Lambda	γ \gamma		π \pi	\varkappa \varkappa
Φ \Phi	δ \delta		ρ \rho	φ \varphi
Π \Pi	ϵ \epsilon		σ \sigma	ϖ \varpi
Ψ \Psi	ζ \zeta		τ \tau	ϱ \varrho
Σ \Sigma	η \eta		υ \upsilon	ς \varsigma
Θ \Theta	θ \theta		ϕ \phi	ϑ \vartheta
Υ \Upsilon	ι \iota		χ \chi	
Ξ \Xi	κ \kappa		ψ \psi	
Ω \Omega	λ \lambda		ω \omega	

3.3.3. 其他“基本”字母符号

这些也是类 0。

\aleph \aleph ^a	ℓ \ell	∂ \partial ^a	\Game \Game
\beth \beth	\eth \eth	\wp \wp	\Im \Im
\daleth \daleth	\hbar \hbar	$\text{\textcircled{S}}$ \text{\textcircled{S}}	\Re \Re
\gimel \gimel	\hslash \hslash	\Bbbk \Bbbk	
\complement \complement	\mho \mho	\Finv \Finv	

Note 1. Labels ^{a,b} indicate `amssymb` package, font `msam` or `msbm`.

3.3.4. 数学字体切换

在传统的 \LaTeX 设置中, 并不是支持全面的数学字体切换所必需的所有字体都是通用的。下面是在使用标准的计算机现代字体集时, 将各种字体转换应用于广泛的数学符号的结果。可以看到对所有字体切换作出正确反应的唯一符号是大写拉丁字母。事实上除了拉丁字母之外, 几乎所有的数学符号都不受字体转换的影响; 尽管小写的拉丁字母、大写的希腊字母和数字对某些字体切换做出了正确的反应, 但它们对其他字体切换产生了奇怪的结果。(使用其他的数学字体集比如 `Lucida New math` 可能会在一定程度上改善这种情况。)

default	<code>\mathbf</code>	<code>\mathrm</code>	<code>\mathsf</code>	<code>\mathit</code>	<code>\mathcal</code>	<code>\mathbb</code>	<code>\mathfrak</code>
X	X	X	X	X	\mathcal{X}	\mathbb{X}	\mathfrak{X}
x	x	x	x	x	\mathcal{X}	\mathbb{X}	\mathfrak{X}
0	0	0	0	0	\mathcal{O}	\mathbb{O}	\mathfrak{O}
$[]$	$[]$	$[]$	$[]$	$[]$	\mathcal{O}	\mathbb{O}	\mathfrak{O}
$+$	$+$	$+$	$+$	$+$	\mathcal{O}	\mathbb{O}	\mathfrak{O}
$-$	$-$	$-$	$-$	$-$	\mathcal{O}	\mathbb{O}	\mathfrak{O}
$=$	$=$	$=$	$=$	$=$	\mathcal{O}	\mathbb{O}	\mathfrak{O}
Ξ	Ξ	Ξ	Ξ	Ξ	\mathcal{O}	\mathbb{O}	\mathfrak{O}
ξ	ξ	ξ	ξ	ξ	\mathcal{O}	\mathbb{O}	\mathfrak{O}
∞	∞	∞	∞	∞	\mathcal{O}	\mathbb{O}	\mathfrak{O}
\aleph	\aleph	\aleph	\aleph	\aleph	\mathcal{O}	\mathbb{O}	\mathfrak{O}
Σ	Σ	Σ	Σ	Σ	\mathcal{O}	\mathbb{O}	\mathfrak{O}
Π	Π	Π	Π	Π	\mathcal{O}	\mathbb{O}	\mathfrak{O}
\Re	\Re	\Re	\Re	\Re	\mathcal{O}	\mathbb{O}	\mathfrak{O}

一个常见的命令给数学符号加粗粗, 对于那些 `\mathbf` 不适用的符号, 可以使用 `\boldsymbol` 或 `\pmb` 命令。

$$A_\infty + \pi A_0 \sim \mathbf{A}_\infty + \pi \mathbf{A}_0 \sim \mathbf{A}_\infty + \pi \mathbf{A}_0 \quad (3.1)$$

```
A_\infty + \pi A_0
\sim \mathbf{A}_{\boldsymbol{\infty}} \boldsymbol{+}
\boldsymbol{\pi} \mathbf{A}_{\boldsymbol{0}}
\sim \pmb{A}_{\pmb{\infty}} \pmb{+} \pmb{\pi} \pmb{A}_{\pmb{0}}
```

这个 `\boldsymbol` 命令是需要使用 `bm` 包获得的, 它提供了比 `amsmath` 包更新、更强大的版本。通常不建议同时对多个符号应用 `\boldsymbol`; 如果出现这种需求, 更有可能意味着有另一种更好的方式来解决它。

3.3.5. 黑板粗体字母

(`msbm`; 无小写) 使用: `\mathbb{R}`. 需要 `amsfonts` 宏包.

ABCDEFGHIJKLMNOPQRSTUVWXYZ

此外: `\Bbbk` 可以得到 `k`

3.3.6. 手写体字母

(cmsy; 无小写) 使用: `\mathcal{M}`.

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

3.3.7. 花体字母

(rsfs; 无小写) 使用: `\usepackage{mathrsfs} \mathscr{B}`.

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

(eusm; 无小写) 使用: `\usepackage{euscript} \mathscr{E}`.

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

3.3.8. 哥特体字母 (eufm)

使用: `\mathfrak{S}`. 需要 `amsfonts`.

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

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

3.4. 其他简单符号. 这些符号也是类 0(普通), 这说明了它们没有任何内置的间距。

<code># \#</code>	<code>\diamondsuit</code>	<code>/ \prime</code>
<code>& \&</code>	<code>\emptyset</code>	<code>\sharp^b</code>
<code>\angle^b</code>	<code>\exists \exists</code>	<code>\spadesuit</code>
<code>\backprime</code>	<code>\flat^b</code>	<code>\sphericalangle^b</code>
<code>\bigstar^a</code>	<code>\forall \forall</code>	<code>\square</code>
<code>\blacklozenge</code>	<code>\heartsuit</code>	<code>\surd</code>
<code>\blacksquare</code>	<code>\infty</code>	<code>\top</code>
<code>\blacktriangle^a</code>	<code>\lozenge</code>	<code>\triangle</code>
<code>\blacktriangledown^a</code>	<code>\measuredangle^b</code>	<code>\triangledown^a</code>
<code>\bot</code>	<code>\nabla</code>	<code>\varnothing</code>
<code>\clubsuit</code>	<code>\natural^b</code>	
<code>\diagdown</code>	<code>\neg</code>	
<code>\diagup</code>	<code>\nexists^a</code>	

Note 1. Labels ^{a,b} indicate `amssymb` package, font `msam` or `msbm`.

Note 2. A common mistake in the use of the symbols `\square` and `\sharp` is to try to make them serve as binary operators or relation symbols without using a properly defined math symbol command. If you merely use the existing commands `\square` or `\sharp` the intersymbol spacing will be incorrect because those commands produce a class-0 symbol.

Note 3. Synonyms: `\neg \not`

3.5. 运算符号

$**$	$\ominus \backslash \text{circleddash}^a$	$\oslash \backslash \text{oslash}$
$++$	$\cup \backslash \text{cup}$	$\otimes \backslash \text{otimes}$
$--$	$\cup \backslash \text{Cup}^a$	$\pm \backslash \text{pm}$
\amalg	$\curlyvee \backslash \text{curlyvee}^a$	$\times \backslash \text{rightthreetimes}^a$
\ast	$\curlywedge \backslash \text{curlywedge}^a$	$\rtimes \backslash \text{rtimes}^b$
$\bar{\wedge} \backslash \text{barwedge}^a$	$\dagger \backslash \text{dagger}$	$\setminus \backslash \text{setminus}$
$\bigcirc \backslash \text{bigcirc}$	$\ddagger \backslash \text{ddagger}$	$\smallsetminus \backslash \text{smallsetminus}^b$
$\bigtriangledown \backslash \text{bigtriangledown}$	$\diamond \backslash \text{diamond}$	$\sqcap \backslash \text{sqcap}$
$\bigtriangleup \backslash \text{bigtriangleup}$	$\div \backslash \text{div}$	$\sqcup \backslash \text{sqcup}$
$\boxdot \backslash \text{boxdot}^a$	$\divideontimes \backslash \text{divideontimes}^b$	$\star \backslash \text{star}$
$\boxminus \backslash \text{boxminus}^a$	$\dotplus \backslash \text{dotplus}^a$	$\times \backslash \text{times}$
$\boxplus \backslash \text{boxplus}^a$	$\overline{\wedge} \backslash \text{doublebarwedge}^a$	$\triangleleft \backslash \text{triangleleft}$
$\boxtimes \backslash \text{boxtimes}^a$	$\gtrdot \backslash \text{gtrdot}^b$	$\triangleright \backslash \text{triangleright}$
$\bullet \backslash \text{bullet}$	$\intercal \backslash \text{intercal}^a$	$\uplus \backslash \text{uplus}$
$\cap \backslash \text{cap}$	$\leftthreetimes \backslash \text{leftthreetimes}^a$	$\vee \backslash \text{vee}$
$\Cap \backslash \text{Cap}^a$	$\lessdot \backslash \text{lessdot}^b$	$\veebar \backslash \text{veebar}^a$
$\cdot \backslash \text{cdot}$	$\ltimes \backslash \text{ltimes}^b$	$\wedge \backslash \text{wedge}$
$\cdot \backslash \text{centerdot}^a$	$\mp \backslash \text{mp}$	$\wr \backslash \text{wr}$
$\circ \backslash \text{circ}$	$\odot \backslash \text{odot}$	
$\circledast \backslash \text{circledast}^a$	$\ominus \backslash \text{ominus}$	
$\circledcirc \backslash \text{circledcirc}^a$	$\oplus \backslash \text{oplus}$	

Note 1. Labels ^{a,b} indicate `amssymb` package, font `msam` or `msbm`.

Synonyms: $\wedge \backslash \text{land}$, $\vee \backslash \text{lor}$, $\cup \backslash \text{doublecup}$, $\Cap \backslash \text{doublecap}$

3.6. 关系符号: $< = > \sim$ 与变式

$<$	\gtrsim <code>\gnsim^b</code>	\nleqslant <code>\nleqslant^b</code>
$=$	\gtrapprox <code>\gtrapprox^a</code>	\nless <code>\nless^b</code>
$>$	\gtreqless <code>\gtreqless^a</code>	\nprec <code>\nprec^b</code>
\approx <code>\approx</code>	\gtreqqless <code>\gtreqqless^a</code>	\npreceq <code>\npreceq^b</code>
\approxq <code>\approxq^b</code>	\gtrless <code>\gtrless^a</code>	\nsim <code>\nsim^b</code>
\asymp <code>\asymp</code>	\gtrsim <code>\gtrsim^a</code>	\nsucc <code>\nsucc^b</code>
\backsim <code>\backsim^a</code>	\gvertneqq <code>\gvertneqq^b</code>	\nsucceq <code>\nsucceq^b</code>
\backsimeq <code>\backsimeq^a</code>	\leq <code>\leq</code>	\prec <code>\prec</code>
\bumpeq <code>\bumpeq^a</code>	\leqq <code>\leqq^a</code>	\precapprox <code>\precapprox^b</code>
\Bumpeq <code>\Bumpeq^a</code>	\leqslant <code>\leqslant^a</code>	\preccurlyeq <code>\preccurlyeq^a</code>
\circeq <code>\circeq^a</code>	\lessapprox <code>\lessapprox^a</code>	\preceq <code>\preceq</code>
\cong <code>\cong</code>	\lesseqgtr <code>\lesseqgtr^a</code>	\precnapprox <code>\precnapprox^b</code>
\curlyeqprec <code>\curlyeqprec^a</code>	\lesseqqgtr <code>\lesseqqgtr^a</code>	\precneqq <code>\precneqq^b</code>
\curlyeqsucc <code>\curlyeqsucc^a</code>	\lessgtr <code>\lessgtr^a</code>	\precnsim <code>\precnsim^b</code>
\doteq <code>\doteq</code>	\lessssim <code>\lessssim^a</code>	\precsim <code>\precsim^a</code>
\doteqdot <code>\doteqdot^a</code>	\ll <code>\ll</code>	\risingdotseq <code>\risingdotseq^a</code>
\eqcirc <code>\eqcirc^a</code>	\lll <code>\lll^a</code>	\sim <code>\sim</code>
\eqsim <code>\eqsim^b</code>	\lnapprox <code>\lnapprox^b</code>	\simeq <code>\simeq</code>
\eqslantgtr <code>\eqslantgtr^a</code>	\lneq <code>\lneq^b</code>	\succ <code>\succ</code>
\eqslantless <code>\eqslantless^a</code>	\lneqq <code>\lneqq^b</code>	\succapprox <code>\succapprox^b</code>
\equiv <code>\equiv</code>	\lnsim <code>\lnsim^b</code>	\succcurlyeq <code>\succcurlyeq^a</code>
\fallingdotseq <code>\fallingdotseq^a</code>	\lvertneqq <code>\lvertneqq^b</code>	\succeq <code>\succeq</code>
\geq <code>\geq</code>	\ncong <code>\ncong^b</code>	\succnapprox <code>\succnapprox^b</code>
\geqq <code>\geqq^a</code>	\neq <code>\neq</code>	\succneqq <code>\succneqq^b</code>
\geqslant <code>\geqslant^a</code>	\ngeq <code>\ngeq^b</code>	\succnsim <code>\succnsim^b</code>
\gg <code>\gg</code>	\ngeqq <code>\ngeqq^b</code>	\succsim <code>\succsim^a</code>
\ggg <code>\ggg^a</code>	\ngeqslant <code>\ngeqslant^b</code>	\thickapprox <code>\thickapprox^b</code>
\gnaprox <code>\gnaprox^b</code>	\ngtr <code>\ngtr^b</code>	\thicksim <code>\thicksim^b</code>
\gneq <code>\gneq^b</code>	\nleq <code>\nleq^b</code>	\triangleq <code>\triangleq^a</code>
\gneqq <code>\gneqq^b</code>	\nleqq <code>\nleqq^b</code>	

Note 1. Labels ^{a,b} indicate `amssymb` package, font `msam` or `msbm`.

Synonyms: \neq `\ne`, \leq `\le`, \geq `\ge`, \doteq `\Doteq`, \lll `\lllless`, \ggg `\gggtr`

3.7. 关系符号: 箭头. 注意看 Section 4.

\circlearrowleft^a	\Lleftarrow^a	\nwarrow
\circlearrowright^a	\longleftarrow	\rightarrow
\curvearrowleft^b	\Longleftarrow	\Rightarrow
\curvearrowright^b	\longleftrightarrow	\rightarrowtail^a
\downdownarrows^a	\Longleftrightarrow	\rightarrowharpoonright
\downharpoonleft^a	\mapsto	\rightarrowharpoonleft
\downharpoonright^a	\longrightarrow	\rightleftarrows^a
\hookleftarrow	\Longrightarrow	\rightleftharpoons^a
\hookrightarrow	\looparrowleft^a	\rightarrowtail^a
\leftarrow	\looparrowright^a	\rightsquigarrow^a
\Leftarrow	\Lsh^a	\Rightarrow^a
\leftarrowtail^a	\mapsto	\Rsh^a
\leftharpoonright	\multimap^a	\searrow
\leftharpoonup	\nLeftarrow^b	\swarrow
\leftleftarrows^a	\nLeftrightarrow^b	\twoheadleftarrow^a
\leftrightharpoonup	\nRightarrow^b	\twoheadrightarrow^a
\Leftrightarrow	\nearrow	\upharpoonleft^a
\leftrightsquigarrow^a	\nleftarrow^b	\upharpoonright^a
\Rightarrow	\nleftrightarrow^b	\upuparrows^a
\rightsquigarrow^a	\nrightarrow^b	

Note 1. Labels ^{a,b} indicate amssymb package, font msam or msbm.

Synonyms: \leftarrow `\gets`, \rightarrow `\to`, \upharpoonright `\restriction`

3.8. 关系符号: 杂项

\backslash backepsilon ^b	\ntrianglerighteq ^b	\supset
\because ^a	\nvdash ^b	\Supset ^a
\between ^a	\nVDash ^b	\supseteq
\blacktriangleleft ^a	\nvDash ^b	\supseteqq ^a
\blacktriangleright ^a	\nVDash ^b	\supsetneq ^b
\bowtie	\parallel	\supsetneqq ^b
\dashv	\perp	\therefore ^a
\frown	\pitchfork ^a	\trianglelefteq ^a
\in	\propto	\trianglerighteq ^a
\mid	\shortmid ^b	\varpropto ^a
\models	\shortparallel ^b	\varsubsetneq ^b
\ni	\smallfrown ^a	\varsubsetneqq ^b
\nmid ^b	\smallsmile ^a	\varsupsetneq ^b
\notin	\smile	\varsupsetneqq ^b
\nparallel ^b	\sqsubset ^a	\vartriangle ^a
\nshortmid ^b	\sqsubseteq	\vartriangleleft ^a
\nshortparallel ^b	\sqsupset ^a	\vartriangleright ^a
\nsubseteq ^b	\sqsupseteq	\vdash
\nsubseteqq ^b	\subset	\Vdash ^a
\nsupseteq ^b	\Subset ^a	\vDash ^a
\nsupseteqq ^b	\subseteq	\Vvdash ^a
\ntriangleleft ^b	\subseteqq ^a	
\ntrianglelefteq ^b	\subsetneq ^b	
\ntriangleright ^b	\subsetneqq ^b	

Note 1. Labels ^{a,b} indicate amssymb package, font msam or msbm.

Synonyms: \ni \owns

3.9. 累积运算符

\int	\bigodot	\bigoplus	\prod
\oint	\bigoplus	\bigvee	\int (smallint)
\bigcap	\bigotimes	\bigwedge	\sum
\bigcup	\bigsqcup	\coprod	

3.10. 标点符号

<code>..</code>	<code>\dotsb</code>
<code>//</code>	<code>\dotsc</code>
<code> </code>	<code>\dotsi</code>
<code>,,</code>	<code>\dotsm</code>
<code>;;</code>	<code>\dotso</code>
<code>:</code> <code>\colon</code>	<code>\ddots</code>
<code>::</code>	<code>\vdots</code>
<code>!!</code>	
<code>??</code>	

Note 1. The `:` by itself produces a colon with class-3 (relation) spacing. The command `\colon` produces special spacing for use in constructions such as $f \colon A \to B$ or $f: A \rightarrow B$.

Note 2. Although the commands `\cdots` and `\ldots` are frequently used, we recommend the more semantically oriented commands `\dotsb` `\dotsc` `\dotsi` `\dotsm` `\dotso` for most purposes (see Section 4.6).

3.11. 配对分隔符. 有关更多信息, 请参见 6 一节。

<code>()</code> <code>()</code>	<code>\langle \rangle</code> <code>\langle \rangle</code>
<code>[]</code> <code>[]</code>	<code>\lceil \rceil</code> <code>\lceil \rceil</code>
<code>\{ \}</code> <code>\lbrace \rbrace</code>	<code>\lfloor \rfloor</code> <code>\lfloor \rfloor</code>
<code> </code> <code>\lvert \rvert</code>	<code>()</code> <code>\lgroup \rgroup</code>
<code> </code> <code>\lVert \rVert</code>	<code>\int \int</code> <code>\lmoustache \rmoustache</code>

3.12. 非配对分隔符

`\vert` `||` `\Vert` `/ /` `\` `\backslash` `|` `\arrowvert` `||` `\Arrowvert` `|` `\bracevert`

Note 1. Using `\vert`, `|`, `\Vert`, or `\|` for paired delimiters is not recommended (see Section 6.2). Instead, use delimiters from the list in Section 3.11.

Synonyms: `||` `\|`

3.13. 可扩展的垂直箭头

`\uparrow` `\Uparrow` `\downarrow` `\Downarrow` `\updownarrow` `\Updownarrow`

3.14. 数学重音

\acute{x} <code>\acute{x}</code>	\bar{x} <code>\bar{x}</code>	\vec{x} <code>\vec{x}</code>	\mathring{x} <code>\mathring{x}</code>
\grave{x} <code>\grave{x}</code>	\breve{x} <code>\breve{x}</code>	\dot{x} <code>\dot{x}</code>	\widetilde{xxx} <code>\widetilde{xxx}</code>
\ddot{x} <code>\ddot{x}</code>	\check{x} <code>\check{x}</code>	\ddot{x} <code>\ddot{x}</code>	\widehat{xxx} <code>\widehat{xxx}</code>
\tilde{x} <code>\tilde{x}</code>	\hat{x} <code>\hat{x}</code>	\dddot{x} <code>\dddot{x}</code>	

3.15. 命名操作符. 这些操作符由一个多字母缩写表示。

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

使用 `\DeclareMathOperator` 命令定义上述列表之外的其他命名操作符; 例如在

```
\DeclareMathOperator{\rank}{rank}
\DeclareMathOperator{\esssup}{ess\,sup}
```

可以写成

$$\backslash\operatorname{rank}(x) \quad \operatorname{rank}(x)$$

$$\backslash\operatorname{esssup}(y,z) \quad \operatorname{ess\,sup}(y,z)$$

这星形形式 `\DeclareMathOperator*` 创建一个在行间公式中显示限制的操作符, 例如 `sup` 或 `max`。

当预先定义这样的命名运算符有问题时 (例如在标题或文章摘要中使用某个运算符时), 可以直接使用另一种形式:

$$\backslash\operatornamename{rank}(x) \quad \rightarrow \quad \operatorname{rank}(x)$$

4. 符号

4.1. 顶部和底部装饰. 它们看上去都类似于重音, 但通常跨多个符号而不是应用于单个基本符号。为了便于参考, `\widetilde` 和 `\widehat` 应用下面数学重音表中。

\widetilde{xxx} <code>\widetilde{xxx}</code>	\overleftarrow{xxx} <code>\overleftarrow{xxx}</code>
\widehat{xxx} <code>\widehat{xxx}</code>	\overrightarrow{xxx} <code>\overrightarrow{xxx}</code>
\overline{xxx} <code>\overline{xxx}</code>	\underrightarrow{xxx} <code>\underrightarrow{xxx}</code>
\underline{xxx} <code>\underline{xxx}</code>	$\overleftrightharpoonrightarrow{xxx}$ <code>\overleftrightharpoonrightarrow{xxx}</code>
\overbrace{xxx} <code>\overbrace{xxx}</code>	$\underleftrightharpoonrightarrow{xxx}$ <code>\underleftrightharpoonrightarrow{xxx}</code>
\underbrace{xxx} <code>\underbrace{xxx}</code>	
\overleftarrow{xxx} <code>\overleftarrow{xxx}</code>	

4.2. 可扩展箭头. `\xleftarrow` 和 `\xrightarrow` 生成箭头, 自动扩展以适应异常宽的下标或上标。这些命令使用一个可选参数 (下标) 和一个强制参数 (上标, 可能是空的):

$$A \xleftarrow{n+\mu-1} B \xrightarrow[T]{n\pm i-1} C \quad (4.1)$$

`\xleftarrow{n+\mu-1}` `\xrightarrow[T]{n\pm i-1}`

4.3. 符号粘贴. 除了标准的重音符号 (Section 3.14), 其他符号也可以用 `\overset` 和 `\underset` 命令放在基本符号的上面或下面。例如, 写 `\overset{*}{X}` 会在 X 上面放一个超文本大小的 $*$, 因此为 $\overset{*}{X}$ 。参考 `secret sideset` 中对 `\sideset` 的描述。

4.4. 矩阵. 该环境 `pmatrix`, `bmatrix`, `matrix`, `vmatrix`, `vmatrix` 和 `matrix` 有 (分别) `()`, `[\]`, `\{ \}`, `\| \|` 和 `\| \|` 分隔符。还有一个没有分隔符的 `matrix` 环境和一个 `array` 环境, 可以用来得到规范列中的左对齐或其他变化。

```
\begin{pmatrix}
\alpha & \beta^* \\
\gamma^* & \delta
\end{pmatrix}
```

为了生成一个适合在文本中使用的小矩阵, 需要一个 `smallmatrix` 环境 (例如, $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$) 这比普通矩阵更接近于在单个文本行中进行匹配, 这个例子是由

```
\bigl( \begin{smallmatrix}
a&b \\
c&d
\end{smallmatrix} \bigr)
```

默认情况下, 矩阵中的所有元素都是水平居中的。 `mathtools` 包提供了所有矩阵环境的星号版本以方便其他对齐, 该包还提供了带星号和非星号名称的 `smallmatrix` 的命令作用。

在矩阵给定的列数, 使用 `\hdotsfor`。例如, 在四列矩阵的第二列中 `\hdotsfor{3}` 将在最后三列中打印一行圆点。

分段函数定义有一个 `case` 环境:

```
P_{r-j}=\begin{cases}
0& \text{\if $r-j$ is odd},\!
r!\,(-1)^{(r-j)/2}& \text{\if $r-j$ is even}.
\end{cases}
```

注意 `\text` 和嵌入数学的使用。

Note. 简单的 \TeX 形式 `\matrix{... \cr ... \cr}` 应当避免使用相关 `\pmatrix` 与 `\cases` 等 \TeX 命令(当加载 `amsmath` 宏包时, 它们是被禁用).

4.5. 数学间距命令. 当使用 `amsmath` 包时, 可以在数学模式中使用或之外使用所有这些数学间距命令。

简写	详细说明	例子	简写	详细说明	例子
	<code>no space</code>	34		<code>no space</code>	34
<code>\,</code>	<code>\thinspace</code>	3 4	<code>\!</code>	<code>\negthinspace</code>	3 4
<code>\:</code>	<code>\medspace</code>	3 4		<code>\negmedspace</code>	3 4
<code>\;</code>	<code>\thickspace</code>	3 4		<code>\negthickspace</code>	3 4
	<code>\quad</code>	3 4			
	<code>\qquad</code>	3 4			

为了更好地控制数学间距, 可使用 `\mspace` and ‘math units’. 一个数学单元, 或 `\mu`, 等于 $1/18$ em. 因此得到负半 `\quad` 可写成 `\mspace{-9.0\mu}`.

这里也有三个命令, 留下的空间等于一个给定的 \TeX 片段的高度与宽度。

Example	Result
<code></code>	space as wide and high as three X’s
<code>\hphantom{XXX}</code>	space as wide as three X’s; height 0
<code>\vphantom{X}</code>	space of width 0, height = height of X

4.6. 点. 在各种情况下, 对于省略号的首选位置 (凸点或连线点) 没有普遍的共识。因此, 这可能被认为是品味的问题。在大多数情况下, `\dots` 是通用的和 `amsmath` 将按照 AMS 喜欢的方式解释它, 即逗号之间的低点 (`\ldots`) 或二进制运算符和关系之间的高点 (`\cdots`) 等, 但是通过使用面向语义的命令。

- `\dotsc` for “逗号点”
- `\dotsb` for “带有二进制操作符/关系的点”
- `\dotsm` for “乘法点”
- `\dotsi` for “积分点”
- `\dotso` for “其它点” (以上皆非)

你使文档尽可能动态地适应不同的约定而不是只使用 `\ldots` 和 `\cdots`, 以防 (例如) 你不得将其提交给坚持遵循这方面传统的出版商, 对各种类型的默认处理遵循美国数学协会的惯例:

我们就有级数 A_1, A_2, \dots ,
 部分和 $A_1 + A_2 + \dots$,
 正交积 $A_1 A_2 \dots$,
 和无穷积分
 $\int_{A_1} \int_{A_2} \dots$.

我们就有级数 A_1, A_2, \dots , 部分和 $A_1 + A_2 + \dots$, 正交积 $A_1 A_2 \dots$, 和无穷积分

$$\int_{A_1} \int_{A_2} \dots$$

4.7. 不间断的破折号. 命令 `\nobreakdash` 禁止在以下连字符或破折号之后使用换行符。例如, 如果你将 ‘pages 1–9’ 写成 `pages 1\nobreakdash--9`, 那么破折号和 9 之间永远不会出现换行符。您还可以使用 `\nobreakdash` 来防止像 `p-adic` 这样的组合中出现不需要的连字符。为了经常使用, 建议使用缩写, 例如,

```
\newcommand{\p}{p\nobreakdash}% for "p-adic" ("p-adic")
\nobreakdash\textendash}% for "pages 1\ndash 9"
% For "n-dimensional" ("n-dimensional"):
\nobreakdash-\hspace{0pt}}
```

最后一个例子展示了如何禁止连字符后的换行符, 但允许在下面的单词中使用正常的连字符。(在连字符后面加一个零宽度的空格。)

4.8. 根. 命令 `\sqrt` 生成一个平方根。要指定显式基数, 请将其作为可选参数。

$$\sqrt[n]{S} \quad \sqrt{\frac{n}{n-1} S} \quad \sqrt[3]{2} \quad \sqrt[3]{2}$$

4.9. 盒装公式. 命令 `\boxed` 在它的参数周围放一个框, 就像 `\fbox` 一样, 只是内容是在数学模式下:

$$\boxed{\eta \leq C(\delta(\eta) + \Lambda_M(0, \delta))} \quad (4.2)$$

```
\boxed{\eta \leq C(\delta(\eta) + \Lambda_M(0, \delta))}
```

如果你需要框住一个方程, 包括方程编号, 这可能是困难的, 这取决于上下文; AMS 作者常见问题解答中有一些建议; 查看页面上用红色标出的条目 https://www.ams.org/faq?faq_id=290.

5. 分数及相关结构

5.1. `\frac`, `\dfrac`, 与 `\tfrac` 命令. `\frac` 命令接受两个参数—分子和分母—, 并将它们以正分数的形式排版。使用 `\dfrac` 或 `\tfrac` 来否决 \LaTeX 对分数内容的正确大小的猜测 (`t = text style`, `d = display style`).

$$\frac{1}{k} \log_2 c(f), \quad \frac{1}{k} \log_2 c(f), \quad \frac{1}{k} \log_2 c(f) \quad (5.1)$$

```
\begin{equation}
\frac{1}{k} \log_2 c(f), \quad \dfrac{1}{k} \log_2 c(f),
\quad \tfrac{1}{k} \log_2 c(f)
\end{equation}
```

$$\Re z = \frac{n\pi \frac{\theta + \psi}{2}}{\left(\frac{\theta + \psi}{2}\right)^2 + \left(\frac{1}{2} \log \left| \frac{B}{A} \right| \right)^2}. \quad (5.2)$$

```
\begin{equation}
\Re{z} = \frac{n\pi \dfrac{\theta + \psi}{2}}{\left(\dfrac{\theta + \psi}{2}\right)^2 + \left(\dfrac{1}{2} \log \left| \dfrac{B}{A} \right| \right)^2}.
\end{equation}
```

5.2. `\binom`, `\dbinom`, 和 `\tbinom` 命令. 对于像 $\binom{n}{k}$ 这样的二项式表达式, 有 `\binom`、`\dbinom` 和 `\tbinom` 命令:

$$2^k - \binom{k}{1} 2^{k-1} + \binom{k}{2} 2^{k-2} \quad (5.3)$$

```
2^k - \binom{k}{1} 2^{k-1} + \binom{k}{2} 2^{k-2}
```

5.3. `\genfrac` 命令. `\frac`、`\binom` 及其变体的功能由一个带六个参数的通用分数命令 `\genfrac` 包含. 后两者对应于 `\frac` 的分子和分母; 前两个是可选的分隔符 (如 `\binom` 中所示); 第三个是行厚度覆盖 (`\binom` 使用它将分数行厚度设置为 0 pt—, 看不见的); 第四个参数是 `mathstyle` 覆盖: 整数值 0–3 select, 分别是 `\displaystyle`、`\textstyle`、`\scriptstyle` 和 `\scriptscriptstyle`. 如果第三个参数为空, 则行厚度默认为 “normal”。

```
\genfrac{left-delim}{right-delim}{thickness}{mathstyle}
{numerator}{denominator}
```

为了说明这一点, 下面介绍如何定义 `\frac`、`\frac` 和 `\binom`。

```
\newcommand{\frac}[2]{\genfrac{}{}{}{}{#1}{#2}}
\newcommand{\tfrac}[2]{\genfrac{}{}{1}{#1}{#2}}
\newcommand{\binom}[2]{\genfrac{()}{()}{0pt}{}{#1}{#2}}
```

Note. 出于技术原因, 不建议在 \LaTeX 文档中使用原始分数命令 `\over`、`\top`、`\above` (例如 https://www.ams.org/faq?faq_id=288, 用红色标出的条目).

5.4. 连分数. 连分数

$$\frac{1}{\sqrt{2} + \frac{1}{\sqrt{2} + \frac{1}{\sqrt{2} + \dots}}} \quad (5.4)$$

可以通过编写得到

```
\cfrac{1}{\sqrt{2}+
\cfrac{1}{\sqrt{2}+
\cfrac{1}{\sqrt{2}+\dotsb
}}}
```

这比直接使用 `\frac` 产生更好的效果. 任何一个分子的左或右放置都是通过使用 `\frac[l]` 或 `\frac[r]` 而不是 `\cfrac` 来完成的。

6. 分隔符

6.1. 分隔符大小. 除非另有说明, 否则数学公式中的分隔符将保持标准大小, 而不考虑所包含材料的高度。要获得更大的尺寸, 你可以使用 `\big...` 前缀 (见下面), 或者你可以使用 `\left` 和 `\right` 前缀来自动调整大小。

`\left` 和 `\right` 执行的自动分隔符大小有两个限制: 首先它被机械地应用于生成足够大的分隔符, 以包含所包含的最大项; 其次大小的范围具有相当大的量子跳跃, 这意味着对于给定的分隔符大小来说, 如果表达式非常非常大, 则会得到下一个更大的大小, 在正常大小的文本中跳转约 6pt(顶部和底部为 3pt)。分隔符大小通常在两种或三种情况下进行调整。这些调整是使用以下命令完成:

分隔符 大小	no size specified	<code>\left</code> <code>\right</code>	<code>\bigl</code> <code>\bigr</code>	<code>\Bigl</code> <code>\Bigr</code>	<code>\biggl</code> <code>\biggr</code>	<code>\Biggl</code> <code>\Biggr</code>
结果	$(b)\left(\frac{c}{d}\right)$	$(b)\left(\frac{c}{d}\right)$	$(b)\left(\frac{c}{d}\right)$	$(b)\left(\frac{c}{d}\right)$	$(b)\left(\frac{c}{d}\right)$	$(b)\left(\frac{c}{d}\right)$

第一种调整是对带极限的累加运算符进行的, 比如求和符号。使用 `\left` 和 `\right` 分隔符通常会比需要的大, 而使用 `Big` 或 `bigg` 大小会得到更好的结果:

$$\left[\sum_i a_i \left| \sum_j x_{ij} \right|^p \right]^{1/p} \quad \text{versus} \quad \left[\sum_i a_i \left| \sum_j x_{ij} \right|^p \right]^{1/p}$$

`\biggl[\sum_i a_i\Bigl|\lvert\sum_j x_{ij}\Bigr\rvert^p\biggr]^{1/p}`

第二种情况是聚集的分隔符对, 其中 `\left` 和 `\right` 使它们的大小相同 (因为这足以覆盖所包含的内容), 但是你真正想要的是使一些分隔符稍微大一些, 以便更容易看到嵌套。

$$((a_1 b_1) - (a_2 b_2))((a_2 b_1) + (a_1 b_2)) \quad \text{versus} \quad ((a_1 b_1) - (a_2 b_2))((a_2 b_1) + (a_1 b_2))$$

`\left((a_1 b_1) - (a_2 b_2)\right)`

`\left((a_2 b_1) + (a_1 b_2)\right)`

`\quad\text{versus}\quad`

`\bigl((a_1 b_1) - (a_2 b_2)\bigr)`

`\bigl((a_2 b_1) + (a_1 b_2)\bigr)`

第三种情况是在运行文本时对象的大小稍大, 比如 $\left|\frac{b'}{a'}\right|$, 其中 `\left` 和 `\right` 产生的分隔符会导致过多的行扩展。在这种情况下, 可以使用 `\bigl` 和 `\bigr` 来生成比基本大小大但仍然能够适应正常行间距的分隔符: $\left|\frac{b'}{a'}\right|$ 。

`mathtools` 包提供了一个可以简化大小设置的功能 `\DeclarePairedDelimiter`; 有关详细信息, 请参阅包文档。

6.2. 竖条符号. 不建议使用 `|` 字符来生成成对的分隔符。符号的方向性存在歧义, 在极少数情况下会产生不正确的空格, 例如, `|k|=|-k|` 生成 $|k| = |-k|$, 而 `|\sin x|` 生成 $|0 \sin x|$, 而不是正确的 $|\sin x|$ 。将 `cnlvert` 用于“left vert bar”和 `\rvert` 用于“right vert bar”, 只要它们成对使用, 就可以防止这个问题; 比较由 `\lvert -k\rvert` 生成的 $|-k|$ 。

对于双杠, 有类似的 \lVert 、 \rVert 命令。推荐的做法是在文档序言中为使用垂直条符号的任何配对分隔符定义适当的命令:

```
\providecommand{\abs}[1]{\lvert#1\rvert}
\providecommand{\norm}[1]{\lVert#1\rVert}
```

于是 $\text{\abs}\{z\}$ 会产生 $|z|$ 和 $\text{\norm}\{v\}$ 会生成 $\|v\|$.

7. \text 命令

命令 \text 的主要用途是用于显示中的单词或短语 显示中的数学片段。它的效果类似于 \mbox , 但是不同于 \mbox , 如果在下标中使用, 它会自动生成订阅大小的文本。

$$f_{[x_{i-1}, x_i]} \text{ is monotonic, } i = 1, \dots, c + 1 \quad (7.1)$$

```
f_{[x_{i-1}, x_i]} \text{\text{ is monotonic,}}
\quad i = 1, \dots, c+1
```

7.1. 与 \mod 相关. 命令 \mod 、 \bmod 、 \pmod 、 \pod 处理“mod”符号的特殊间距约定。 \mod 和 \pod 是一些作者喜欢的 \pmod 的变体; \mod 省略括号, 而 \pod 省略“mod”并保留括号。

$$\gcd(n, m \text{ mod } n); \quad x \equiv y \pmod{b}; \quad x \equiv y \text{ mod } c; \quad x \equiv y \pmod{d} \quad (7.2)$$

```
\gcd(n, m \bmod n) ; \quad x \equiv y \pmod b
; \quad x \equiv y \text{ mod } c ; \quad x \equiv y \pod d
```

8. 积分与求和

8.1. 改变限制的位置. 根据惯例和上下文, 对积分、求和与类似符号的限制要么放在基符号一侧, 要么放在基符号上下两侧。 \int 有自动选择一个或另一个的规则, 大多数时候结果是令人满意的。如果没有, 有三个 \int 命令可以用来影响限制的位置: \limits 、 \nolimits 、 \displaylimits . 比较

$$\int_{|x-x_z(t)| < X_0} z^6(t) \phi(x) \quad \text{and} \quad \int_{|x-x_z(t)| < X_0} z^6(t) \phi(x)$$

```
\int_{\abs{x-x_z(t)} < X_0} ... \int \limits_{\abs{x-x_z(t)} < X_0} ...
\limits 命令应该紧跟在它所应用的基本符号之后, 意思是: 将下面的下标和/或上标移到 limit 位置, 而不考虑这个符号的通常约定。 \nolimits 的意思是将它们移到一边, 而 \displaylimits 可能用于定义一种新的基符号, 意思是使用 \sum 命令的标准定位。
```

还请参阅 [AMUG] 中的选项 `intlimits` 和 `nosumlimits` 的描述。

8.2. 多重积分. \iint 、 \iiint 和 \iiint 给出多个积分符号之间的间距很好地调整, 在文本和显示风格。 \idotsint 是同一个概念的扩展, 它给出了两个带点的积分符号。注意在 dx 和之前使用更小空间 (\backslash , $)$ 来阐明其含义。

$$\iint_A f(x, y) dx dy \quad \iiint_A f(x, y, z) dx dy dz \quad (8.1)$$

$$\iiint_A f(w, x, y, z) dw dx dy dz \quad \int \cdots \int_A f(x_1, \dots, x_k) \quad (8.2)$$


```
\iint\limits_A f(x,y)\,dx\,dy\qquad\iiint\limits_A
f(x,y,z)\,dx\,dy\,dz\qquad\iiint\limits_A
f(w,x,y,z)\,dw\,dx\,dy\,dz\qquad\idotsint\limits_A f(x_1,\dots,x_k)
```

8.3. 多行下标和上标. 可以使用 `\substack` 命令生成多行下标或上标:

$$\sum_{\substack{0 \leq i \leq m \\ 0 < j < n}} P(i, j)$$

8.4. `\sideset` 命令. 还有一个名为 `\sideset` 的命令, 用于一个非常特殊的目的: 将符号放在下标和上标在符号 \sum 或 \prod 的角落上. *Note: The `\sideset` command is only designed for use with large operator symbols; with ordinary symbols the results are unreliable.* 使用 `\sideset`, 你可以编写

$$\sum_{n < k, n \text{ odd}} n E_n$$

额外的一对空大括号是由 `\sideset` 能够在大型运算符的每个角上放置一个或多个符号来解释的; 要在累乘符号的每个角上加上星号, 你需要键入

$$\prod_{*}^{*} P(i, j)$$

9. 改变公式中元素的大小

用于更改数学公式内部字体大小的 \LaTeX 机制与数学公式外部使用的机制完全不同。如果您试图在公式中使用 `\large` 或 `\huge` 等文本命令将内容变大

```
# \large \#
```

你将收到一条报错警告消息.

命令 `\large` 在数学模式下无效

然而这样的尝试常常表明对 \LaTeX 数学符号工作原理的误解, 如果你希望在它的排版属性中有一个类似于求和符号的 `a#` 符号, 那么在原则上, 实现这一点的最佳方法是使用标准 \LaTeX `\DeclareMathSymbol` 命令将它定义为“`mathop`”类型的符号(请参阅 [LFG]).(然而这需要获得具有合适文本大小/显示, 而事实可能并非如此简单.)

考虑表达式:

$$\frac{\sum_{n>0} z^n}{\prod_{1 \leq k \leq n} (1 - q^k)}$$

在这种情况下, 使用 `\dfrac` 而不是 `\frac` 不会改变任何东西; 如果您希望求和累乘的数学符号显示完整的大小, 你需要 `\displaystyle` 命令:

$$\frac{\sum_{n>0} z^n}{\prod_{1 \leq k \leq n} (1 - q^k)}$$

如果你想要全尺寸的符号, 但是在边上有限制, 也可以使用 `\nolimits` 命令:

$$\frac{\sum_{n>0} z^n}{\prod_{1\leq k\leq n} (1-q^k)} \quad \frac{\{\{\displaystyle\sum\nolimits_{n>0} z^n\}}{\{\{\displaystyle\prod\nolimits_{1\leq k\leq n} (1-q^k)\}}\}}$$

类似的命令 `\textstyle`、`\scriptstyle` 和 `\scriptstyle` 强制 \LaTeX 使用 (分别) 内联数学、一阶下标或二阶下标中应用的符号大小和间距, 即使当前上下文通常也会产生一些其他大小。

Note: 这些命令属于 \LaTeX book 中称为“声明”的一类特殊命令。特别需要注意的是分隔命令效果的大括号落在哪里:

Right: `\{\displaystyle ...}` **Wrong:** `\displaystyle{...}`

10. 其它宏包组合

许多其他处理数学公式某些方面的 \LaTeX 包可以从 CTAN(全面的 \TeX 存档网络) 获得。推荐几个例子:

mathtools `amsmath` 附加功能扩展; 加载 `amsmath`.

amsthm 一般定理和证明的设置;

amssymb 定义 `\mathbb` 和 `\mathfrak`, 并提供对许多附加符号的使用 (without names; `amssymb` provides the names).

accents 在重音符号和使用任意符号的重音符号下.

bm Bold math 包, 提供了 `\boldsymbol` 使数学公式以粗体的方式来显示.

mathrsfs Ralph Smith's Formal Script, 花体设置.

cases 将大括号应用于两个或多个方程, 而不丢失单个方程的编号.

delarray 跨越数组多行的分隔符.

xypic 交换图和其他图.

TikZ 图形工具, 包括各种绘制图表的功能.

如果你知道这个宏包的名称, \TeX 目录是一个快速查找的捷径.

[http:](http://mirror.ctan.org/help/Catalogue/alpha.html)

[//mirror.ctan.org/help/Catalogue/alpha.html](http://mirror.ctan.org/help/Catalogue/alpha.html),

有关 \TeX 问题与解答的论坛

<https://tex.stackexchange.com/questions>

检索归类后已有解答; 按关键词主题进行搜索

<https://tex.meta.stackexchange.com/a/2425#2425>

如果没有什么有用的东西出现, 那就自我解答.

参考文献

- [AMUG] American Mathematical Society and the \LaTeX 3 Project: *User's Guide for the amsmath package*, Version 2.+, <http://mirror.ctan.org/macros/latex/required/amsmath/amslatex.tex> and <http://mirror.ctan.org/macros/latex/required/amsmath/amslatex.pdf>, 2017.
- [AFUG] American Mathematical Society: *User's Guide, AMSFonts*, <http://mirror.ctan.org/fonts/amsfonts/amsfont.pdf>, 2002.
- [CLSL] Scott Pakin: *The Comprehensive \LaTeX Symbol List*, <http://mirror.ctan.org/tex-archive/info/symbols/comprehensive/>, January 2017. Raw font tables, without symbol names, are shown alphabetically by font name in the rawtables*.pdf files in the same area of CTAN and from \TeX Live with texdoc rawtables.
- [Lam] Leslie Lamport: *\LaTeX : A document preparation system*, 2nd edition, Addison-Wesley, 1994.
- [LC] Frank Mittelbach and Michel Goossens, with Johannes Braams, David Carlisle, and Chris Rowley: *The \LaTeX Companion*, 2nd edition, Addison-Wesley, 2004.
- [LFG] \LaTeX 3 Project Team: *\LaTeX 2_ε font selection*, <http://mirror.ctan.org/macros/latex/doc/fntguide.pdf>, 2005.
- [LGC] Michel Goossens, Frank Mittelbach, Sebastian Rahtz, Denis Roegel, and Herbert Voß: *The \LaTeX Graphics Companion*, 2nd edition, Addison-Wesley, 2008.
- [LGG] D. P. Carlisle, \LaTeX 3 Project: *Packages in the 'graphics' bundle*, <http://mirror.ctan.org/macros/latex/required/graphics/grfguide.pdf>, 2017.
- [LUG] \LaTeX 3 Project Team: *\LaTeX 2_ε for authors*, <http://mirror.ctan.org/macros/latex/doc/usrguide.pdf>, 2015.
- [MML] George Grätzer: *More Math into \LaTeX* , 5th edition, Springer, New York, 2016.
- [UCM] Will Robertson: *Every symbol (most symbols) defined by unicode-math*, <http://mirror.ctan.org/macros/latex/contrib/unicode-math/unimath-symbols.pdf>, 2017; and Will Robertson, Philipp Stephani, Joseph Wright, and Khaled Hosny: *Experimental Unicode mathematical typesetting: The unicode-math package*, <http://mirror.ctan.org/macros/latex/contrib/unicode-math/unicode-math.pdf>, 2017.