

第六讲 数学排版

潘建瑜

华东师范大学数学系


jypan@math.ecnu.edu.cn

2014

本讲内容

- ① 数学宏包
- ② 行内公式与行间公式
- ③ 常见数学公式排版
- ④ 数学字体, 数学符号与数学函数
- ⑤ 数学模式中的常用参数
- ⑥ 定理类环境
- ⑦ 定界符与阵列

数学宏包

 输入数学公式时，需要在导言区加入数学宏包，常用数学宏包有：


```
\usepackage{latexsym}
\usepackage{amsmath,amssymb,amsfonts}
\usepackage{bm} % 数学公式粗体
```

几点说明：(例：41.tex)


- 数学模式中的字符都视作为数学变量，用 *italic 斜体* 显示
- 数学模式中的 **空格全部被忽略**，系统会自动安排公式中各部分的间距
- 数学公式分 **行内公式** 和 **行间公式**（或**显示公式**）
- 数学公式中的字体大小（系统自动设置各部分字体大小，也可手工调节）

```
\displaystyle → 显示样式公式中的字体大小
\textstyle → 行内公式中的字体大小
\scriptstyle → 角标中的字体大小
\scriptscriptstyle → 二级角标中的字体大小
```

数学模式中的普通文本

 在数学模式中输入普通文本:

```
\mbox{普通文本}  
\text{普通文本} → 需要加载 amsmath 宏包
```

 `\ensuremath{数学符号}`

→ 确保在文本模式和数学模式中都能正常显示数学符号, 如:

```
\newcommand{\eps}{\ensuremath{\varepsilon}}
```

普通文本中可以直接用 `\eps` 命令, 不用加数学模式.

数学模式中的空白间隔

🔗 数学模式中的空白间隔

- `\quad`, `\qquad`
- `\hspace`
- `\phantom`
- `_`
- `\,` (大小约为 $3/18 \quad$)

`\;` $\rightarrow 5/18 \quad$

`\:` $\rightarrow 4/18 \quad$

`\!` $\rightarrow -3/18 \quad$

\rightarrow 只能用于数学模式

行内公式与行间公式

☞ 行内公式：与普通文本混合排版

- 相应的数学环境为 `math`, `\(` 或 `$`

```
\begin{math} ... \end{math}
\(... \)
$ ... $ → 推荐使用这种方式
```


- 上面三种方式是等价的, 一般使用最后一种方式, 如: (例: 41.tex)

```
勾股定理 $ a^2 + b^2 = c^2 $ 也称商高定理
```

☞ 行间公式:

- 独占一行 (单行公式) 或 多行 (多行公式)
- 行间公式可以编号, 也可以不编号
- 给公式编号时, 一般采用 自动编号, 也可以人工编号
- 在输入多行公式时, 对应的源代码中 不能出现空行!

单行公式

 单行公式相应的数学环境为: `displaymath` 和 `equation`

<pre>\begin{displaymath} \end{displaymath}</pre>	不带编号的单行公式数学环境
<pre>\[... \]</pre>	<code>displaymath</code> 环境的简化形式
<pre>\$\$... \$\$</pre>	与上面的单行数学环境等价, 但可用 <code>\eqno</code> 或 <code>\leqno</code> 进行人工编号
<pre>\begin{equation} \end{equation}</pre>	带 自动编号 的单行公式数学环境

equation 环境

equation 环境

- `equation` 环境中的公式可以是普通的单行公式, 也可以是作为一个整体处理的环境或盒子, 如矩阵, 分段定义函数等, 例

$$|x| = \begin{cases} x & \text{if } x \geq 0 \\ -x & \text{if } x < 0 \end{cases} \quad (1)$$

- 若要改变公式自动编号的值, 可在公式前插入

```
\setcounter{equation}{整数}
```

这里的 `equation` 为公式计数器, 每次调用 `equation` 环境, 计数器的值会自动加 1

公式的引用: 先标记, 后引用

- 标记: `\label{公式标记}` → 公式标志必须是唯一的
- 引用: `\eqref{公式标记}` (需 `amsmath` 宏包)

多行公式

👉 多行公式数学环境: `eqnarray` 和 `eqnarray*`


👉 建议使用 `amsmath` 宏包中的 `align` 和 `align*` 环境 (后面有详细介绍)

常见数学公式排版

☞ 常见数学公式排版

- 角标: 上标和下标. 如: x^2 , a_{ij}
- 分式与根式. 如: $\frac{1}{2}$, $\frac{1}{2}$, $\sqrt{x+y}$
- 求和与积分. 如: $\sum_{i=1}^n a_i$, $\int_a^b f(x)dx$, $\sum_{i=1}^n a_i$, $\int_a^b f(x)dx$
- 数学重音符号. 如: \tilde{x} , \dot{y} , \grave{a}
- 上、下划线等. 如: \overline{abc} , $\underbrace{a+a+\cdots+a}_{20 \uparrow}$
- 堆积符号. 如: $\stackrel{\text{def}}{=}$

角标: 上标和下标

 上标: $\hat{\{\dots\}}$, 下标: $_{\{\dots\}}$

- 这两个角标命令必须在**数学模式**中使用
- 若角标由多个字符组成, 则须用大括号括起来
- 多层角标需要使用分组符号, 如

$$\$ x^{a+b}, x^{\{a+b\}}, x^{\{y^2\}} \$ \rightarrow x^a + b, x^{a+b}, xy^2$$


- 一级角标字体大小为 `\scriptsize`, 约 7pt,
二级及以上角标字体大小为 `\scriptscriptsize`, 约 5pt
- 中文角标要放入盒子, 并指定字体大小, 否则字体偏大, 如:

$$\$ x^{\{\mbox{\scriptsize 平方}\}} \$ \rightarrow x^{\text{平方}}$$

- 可以根据需要手工改变角标字体大小或层次
- 一个特殊的角标: **导数** \rightarrow 可以直接使用**右单引号** 或 `\prime`, 如

$$\$ x' \$ \text{ 或 } \$ x^{\prime} \$ \rightarrow x' \text{ 或 } x'$$

分式

 分式命令: `/` 或 `\frac{分子}{分母}`, 如

$$\$ x/y \$ \text{ 或 } \$ \backslash\text{frac}\{x\}\{y\} \$ \rightarrow x/y \text{ 或 } \frac{x}{y}$$

- 系统自动指定分子分母字体的大小:

在行间公式中为 `\displaystyle`, 在行内公式中是 `\textstyle`

$$$$ \backslash\text{frac}\{1\}\{2\}, \backslash\text{frac}\{x\}\{1+\backslash\text{frac}\{1\}\{2\}\} $$ \rightarrow \frac{1}{2}, \frac{x}{1 + \frac{1}{2}}$$

- 命令 `\displaystyle` \rightarrow 将分子分母的字体的大小都设为 `\displaystyle`

$$\$ \backslash\text{displaystyle}\backslash\text{frac}\{x\}\{y\} \$ \rightarrow \frac{x}{y}$$

根式

☞ 根式: $\sqrt{\dots}$, $\sqrt[n]{\dots}$, $\sqrt{\dots}$

$$\$ \sqrt{x+y}, \sqrt[4]{x+y} \$ \rightarrow \sqrt{x+y}, \sqrt[4]{x+y}$$

- $\sqrt{\dots}$ → 不带横线的根号

$$\$ \sqrt{(x+y)} \$ \rightarrow \sqrt{(x+y)}$$

- 根式可以嵌套
- 被开方式字符高度不同时, 根号线会在不同水平线上

$$\$ \sqrt{\sqrt{a} + \sqrt{b}} \$ \rightarrow \sqrt{\sqrt{a} + \sqrt{b}}$$

调整方法: \mathstrut → 与圆括号同高, 但没有宽度的数学支柱

$$\$ \sqrt{a\mathstrut} + \sqrt{a\mathstrut} \$ \rightarrow \sqrt{a} + \sqrt{b}$$

求和与积分

☞ 求和: `\sum`, 积分: `\int`

$$\$ \sum_{i=1}^n x^i, \int_a^b f(x) \$ \rightarrow \sum_{i=1}^n x^i, \int_a^b f(x)$$

- 上下限位于符号右上下侧, 或符号头顶和脚下
- 在行内公式和行间公式中取不同的尺寸, 上下限位置也可能不同

$$$$ \sum_{i=1}^n x^i, \int_a^b f(x) $$ \rightarrow \sum_{i=1}^n x^i, \int_a^b f(x)$$

- 可以手工指定上下限的位置: `\limits` 和 `\nolimits`

$$\$ \sum\limits_{i=1}^n x^i \$ \rightarrow \sum_{i=1}^n x^i$$

- 微分算子 `d` 为直立字体: `\mathrm{d}`, 且与积分表达式之间要留一小空

- 带圈积分: `\oint` $\rightarrow \oint$

直立的积分号

- 直立的积分号: `wasysym` 宏包

```
\usepackage{wasysym}
```

注: 该宏包需在 `amsmath` 之后调用

`\varint` \rightarrow \int 例: $\int_a^b f(x) dx$

`\iint` \rightarrow \iint

`\iiint` \rightarrow \iiint

`\varoint` \rightarrow \oint

`\oiint` \rightarrow \oiint

 更多积分符号见 `txfonts` 等宏包 (TeXFriend)

数学重音符号

数学重音符号

`\hat{a}` → \hat{a}

`\dot{a}` → \dot{a}

`\tilde{a}` → \tilde{a}

`\acute{a}` → \acute{a}

`\grave{a}` → \grave{a}

`\bar{a}` → \bar{a}

`\ddot{a}` → \ddot{a}

`\vec{a}` → \vec{a}

`\check{a}` → \check{a}

`\mathring{a}` → \mathring{a}

`\imath` → \imath

`\jmath` → \jmath

`\widehat{abc}` → \widehat{abc}

`\widetilde{abc}` → \widetilde{abc}

上、下划线等

☞ 上、下划线: `\overline{...}`, `\underline{...}`

```
$$ \overline{\overline{a}^2+\underline{abc}+\bar{b}^2} $$
```

$$\overline{\overline{a}^2 + \underline{ab} + \bar{b}^2}$$

- 对于单个字符, `\overline` 比 `\bar` 要长一些

☞ 上、下大括号: `\overbrace{...}`, `\underbrace{...}`

```
$$ \underbrace{a+\overbrace{b+\cdots+b}^m+c}_n $$
```

$$\underbrace{a + \overbrace{b + \cdots + b}^m + c}_n$$

堆积符号

☞ 堆积: `\stackrel{上位符号}{基位符号}`

```
$$ \vec{x} \stackrel{\mathrm{def}}{=} (x_1, \ldots, x_n) $$
```

$$\vec{x} \stackrel{\text{def}}{=} (x_1, \dots, x_n)$$

- 基位符号采用正常字体大小, 上位符号采用较小的字体

☞ 另外两个命令: `\atop` 和 `\choose`

```
{上位符号 \atop 下位符号}, {上位符号 \choose 下位符号}
```

- 上位与下位符号采用同样大小的字体

```
$$ \sum_{\substack{1 < i < n \\ i \neq j}} \atop i \neq j} a_{ij}, \quad \{n+1 \choose k\} $$
```

$$\sum_{\substack{1 < i < n \\ i \neq j}} a_{ij}, \quad \binom{n+1}{k}$$

数学字体

常用数学字体

`\mathrm`, `\mathit`, `\mathtt`, `\mathsf`, `\mathbf`, `\mathcal`

<code>\mathrm{ABC xyz 1234}</code>	→	ABCxyz1234
<code>\mathit{ABC xyz 1234}</code>	→	<i>ABCxyz1234</i>
<code>\mathtt{ABC xyz 1234}</code>	→	ABCxyz1234
<code>\mathsf{ABC xyz 1234}</code>	→	ABCxyz1234
<code>\mathbf{ABC xyz 1234}</code>	→	ABCxyz1234
<code>\mathcal{ABC xyz 1234}</code>	→	$\mathcal{ABC}\S\ddagger\infty\in\Delta$

其它更多数学字体需加载特殊宏包, 参见 TeXFriend, 如

<code>\mathbb{ABCR}</code>	→	\mathbb{ABCR}	(amsfonts 宏包)
<code>\mathfrak{ABCR}</code>	→	\mathfrak{ABCR}	(amsfonts 宏包)
<code>\mathscr{ABCR}</code>	→	\mathscr{ABCR}	(mathrsfs 宏包)

数学粗体

 数学粗体命令: `\boldmath`, `\unboldmath`

- 这两个命令是声明, 即对其后面的所有数学符号起作用
- 不能出现在数学模式中, 可放在数学模式前面

```
$ ABC xyz \sin\alpha $ \boldmath $ ABC xyz \sin\alpha $  
ABCxyz sin α    ABCxyz sin α
```

 `bm` 宏包: `\bm{数学符号或公式}`

```
$$ ABC xyz \sin\alpha \bm{ABC xyz \sin\alpha} $$  
  
ABCxyz sin α    ABCxyz sin α
```

数学符号

☞ 数学符号的输入

- 所有的数学符号(变量, 函数, 公式等)必须放在数学模式中
- 可直接输入的数学运算符

+	-	*	/	=	<	>	()	[]		'	!	:
---	---	---	---	---	---	---	-----	-----	--	---	---	---

- 其它符号

.	⋮	⋱	⋱
<code>\cdot</code>	<code>\ldots</code>	<code>\cdots</code>	<code>\vdots</code>	<code>\ddots</code>	<code>\reflectbox{\$\ddots\$}</code>

{	}	∞	∂
<code>\{</code>	<code>\}</code>	<code>\infty</code>	<code>\partial</code>

数学符号表

☞ 数学符号

- 二元运算符: \pm , \mp , \times , \div , ...
 - 关系运算符: \leq , \geq , \neq , \subset , ...
 - 箭头符号: \rightarrow , \leftarrow , \Rightarrow , \Leftrightarrow , ...
 - 具有两种尺寸的符号: Σ , \cup \sum , \cup , ...
 - 其它符号: \forall , \exists , \clubsuit , \spadesuit , ...
 - 函数名: \sin , \cos , \lim , \max , ...
 - 希腊字母: $\alpha, \beta, \gamma, \dots$ $\Gamma, \Omega, \Delta, \dots$
 - 直立希腊字母可以使用 `pxfont` 宏包
- ☞ 更多数学符号参见帮助文件, 或 WinEdt / TeXFriend

数学函数

- ☞ 数学公式中的函数名必须用正体, 需通过[函数名命令](#)进行输入
- ☞ 函数名命令需事先定义, 如 (更多已定义函数名参见帮助文件或 WinEdt)

```
\sin, \cos, \ln, \exp, ...
```

- ☞ 对于没有定义的函数名, 可以使用下面的命令来定义 (`amsmath` 宏包)


```
\DeclareMathOperator{\函数名命令}{函数名}
\DeclareMathOperator*{\函数名命令}{函数名}
```

- 这两个命令必须放置在[导言区](#)
- 用带星号的命令所定义的函数名, 在处理角标时, 具有类似于 `\sum` 的功能, 即可以根据需要将上下角标放置在函数名的正上方或正下方. 例如

```
\DeclareMathOperator{\abc}{abc}
\DeclareMathOperator*{\xyz}{xyz}
```

```
$ \abc_1^2, \abc\limits_1^2 $ → abc12, abc12
$ \xyz_1^2, \xyz\limits_1^2 $ → xyz12, xyz12
```

数学函数

 如果是临时使用未定义的函数名, 也可以直接在公式中使用命令


```
\operatorname{函数名}  
\operatorname*{函数名}
```

- 这两个命令使用起来比较灵活
- 带星号的命令的含义与前面一样. 如

```
$ \operatorname{newfun}\limits_1^2 $ \rightarrow \operatorname{newfun}_1^2  
$ \operatorname*{newfun}\limits_1^2 $ \rightarrow \operatorname*{newfun}_1^2
```

- 例: 42mathfun.tex

数学模式中的常用参数

 数学模式中的参数 (可使用 `\setlength` 修改)

- `\arraycolsep` → `array` 和 `eqnarray` 环境中列间距的一半, 缺省为 5.0pt
- `\jot` → `eqnarray` 中的行间隔, 缺省为 3.0pt
- `\abovedisplayskip` → 长行间公式与上方文本的间隔, 缺省为 8.5pt plus 3pt minus 4pt
- `\belowdisplayskip` → 长行间公式与下方文本的间隔, 缺省为 8.5pt plus 3pt minus 4pt
- `\abovedisplayshortskip` → 短行间公式与上方文本的间隔, 缺省为 0pt plus 2pt
- `\belowdisplayshortskip` → 短行间公式与下方文本的间隔, 缺省为 4pt plus 2pt minus 2pt
- `\mathindent` → 使用 `fleqn` 选项时, 行间公式的缩进量, 缺省为 25pt

长行间公式与短行间公式

长行间公式与短行间公式

- 行间公式的“长”与“短”并不是指公式本身的长短, 而是与其上方的文本相比较而言
- 如果行间公式上方文本的结尾处位于公式左边界的左边, 则该公式为**短公式**, 否则为**长公式** (例 71math.tex)
- 多行环境中的公式均是长公式. 如: `eqnarray`, `eqnarray*`

注意: 输入行间公式时, 数学环境中不能有空行!!!

定理类环境

 定理类环境: 定理, 引理, 推论, 命题, 定义, ...

- 需要用特定的格式显示
- 带有特定的标题
- 自动编号

Theorem 9.1 (Fermat). *There do not exist integers x , y , z and $n > 2$ such that $x^n + y^n = z^n$.*

定理类环境

☞ 定理类环境的实现

- (1) 用 `\newtheorem` 命令定义一个定理类环境

```
\newtheorem{定理环境名}{标题}[父计数器]
```

- (2) 调用该环境

```
\begin{定理环境名}[附加标题]  
... ..  
\end{定理环境名}
```

- 以上定义的定理类环境使用独立的计数器

☞ 多个定理类环境使用一个共同的计数器


```
\newtheorem {新定理环境名}[已有定理环境名] {标题}
```

- 新定义的定理环境与已有定理环境 共用一个计数器 (例 71math.tex)

☞ 定理环境缺省格式: 标题与编号用粗体, 正文用 *italic* 斜体

- 中文字体一般需要另外指定 (如正文可用直立的楷体)

定理类环境的定制

 定理类环境的定制: `amsthm` 宏包

```
\usepackage{amsthm}
```

- 新增命令 `\newtheorem*`, 创建不带编号定理环境
- `\swapnumbers`: 调用该命令后, 创建的定理类环境中编号显示在标题前
- 提供三种预定义的定理格式
 - `plain`: 标题与编号均采用粗体, 正文用斜体
 - `definition`: 标题与编号为粗体, 正文正常字体
 - `remark`: 标题与编号为斜体, 正文用正常字体
- 选择定理格式的命令: `\theoremstyle{定理格式}`
 - 在该命令后调用 `\newtheorem` 创建的定理环境, 将采用指定的格式
- 可以自己定制定理类环境: `\newtheoremstyle`
 - 具体用法参见该宏包手册 `amsthm.pdf`

定理类环境的定制

👉 证明环境: `proof`

```
\begin{proof}[可选标题]  
...  
\end{proof}
```

- 修改缺省标题: `\renewcommand{\proofname}{...}`
- 若使用了可选标题, 则输出该标题, 否则输出 *Proof*.
- 证明环境结束时, 会自动在最右边打印一个方盒子(□), 表示证明结束

👉 证明命令: `\proof`, 打印证明结束记号的命令: `\qed`

👉 修改证明结束记号: `\qedsymbol`

定制定理类环境

```

\newtheoremstyle{myth}% name
  {6pt} % Space above
  {4pt} % Space below
  {\it} % Body font
  {} % Indent amount (empty = no indent, \parindent = para indent)
  {\hei} % 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{myth}
\newtheorem{theorem}{\color{blue}定理}[chapter]
\newtheorem{lemma}{\color{blue}引理}[chapter]
\newtheorem{corollary}{\color{blue}推论}[chapter]
\newtheorem{proposition}[theorem]{\color{blue}命题}
\newtheorem{property}{\color{blue}性质}[chapter]
\newtheorem{definition}{\color{blue}定义}[chapter]
\renewcommand{\proofname}{\upshape\hei 证明}
\newtheorem{remark}{注}[section]

```

定理类环境的定制

 定理类环境的定制: `ntheorem` 宏包

```
\usepackage[选项]{ntheorem}
```

- 常用选项有
 - `thmmarks`: 自动添加“结束符号”, 如证明结束标记
 - `amsmath`: 如果文档中加载了 `amsmath` 宏包, 需加上该选项
 - `hyperref`: 兼容 `hyperref` 宏包
 - `amsthm`: 兼容 `amsthm` 宏包
 - `framed`: 带边框
 - `thref`: 设置新的引用方式
- 选择定理格式的命令: `\theoremstyle{定理格式}`
 - `plain`, `break`, `change`, `margin`, `nonumberplain`, `empty`, ...
 - 自己定制定理类环境: `\newtheoremstyle`
- 创建定理类环境: `\newtheorem`

定理类环境的定制

- 常用设置命令
 - `\theoremheaderfont`: 标题字体属性, 如大小, 形状, 颜色等
 - `\theorembodyfont`: 定理正文字体属性
 - `\theoremseparator`: 标题与定理正文之间的间隔符
 - `\theoremnumbering`: 编号样式, 如 `arabic`, `roman`, ...
 - `\theorempreskipamount`: 定理与前面文本之间的距离
 - `\theorempostskipamount`: 定理与后面文本之间的距离
- 详细用法参见该宏包手册 `ntheorem.pdf`

ntheorem

```
\usepackage[amsmath, thref, thmmarks, hyperref]{ntheorem}

\theorempreskipamount1.2em % spacing before the environment
\theorempostskipamount0em % spacing after the environment
\theoremstyle{plain}
\theoremheaderfont{\normalfont\rmfamily\bfseries\color{blue}}
\theorembodyfont{\normalfont\rmfamily\itshape\color{black}}
\theoremindent0em
\theoremseparator{\hspace{0.2em}}
\theoremnumbering{arabic}
\newtheorem{theorem}{\color{blue}定理}[chapter]
\newtheorem{definition}{\color{blue}定义}[chapter]
\newtheorem{lemma}{\color{blue}引理}[chapter]
\newtheorem{corollary}{\color{blue}推论}[chapter]

\theorembodyfont{\normalfont\rmfamily\color{black}}
\newtheorem{example}{\color{blue}例}[chapter]
```

定界符

L^AT_EX 中常用的定界符

() [] | / \ { } || L] [] < > ↑ ↓ ⇕ ↑ ↓ ⇕

- 定界符可以放大: `\big` (1.5倍), `\Big` (2倍), `\bigg` (2.5倍), `\Bigg` (3倍)

`\big`: () [] | / \ { } || L] ...

`\Big`: () [] | / \ { } || L] ...

`\bigg`: () [] | / \ { } || L] ...

`\Bigg`: () [] | / \ { } || L] ...

定界符

🔗 自适应放大命令: `\left`, `\right`

- 根据定界符所包含的内容自动放大
- 必须成对出现
- 必须出现在公式的同一行

```
$$ \left(\sum x^2\right), \left(\sum_{x=1}^n x^2\right) $$
```


$$\left(\sum x^2\right), \left(\sum_{x=1}^n x^2\right)$$

- 若只需出现一半, 则另一半须用 `\left.` 或 `\right.` 代替

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

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

阵列

 阵列环境: `array`

```
\begin{array}[竖向位置][列格式]  
  第 1 行 \\  
  ...    \\  
  第 n 行 \\  
\end{array}
```

- **竖向位置**: 阵列在垂直方向与外部文本行的相对位置, 取值可以是: `t` 或 `b`, 缺省为**垂直居中**
- **列格式**: 由若干项组成, 用于指定各列的对齐格式
 - 文本位置: `l`, `c`, `r`, 分别表示**左对齐**, **居中对齐**和**右对齐**
 - 分界线: `|`, `||`, `...`
- 每行都以强制换行符 `\\` 结束, 最后一行除外
- 每行由若干列组成, 各行中的列数通常应相同, 列与列之间用 `&` 隔开
- 该环境只能在**数学模式**中使用

阵列举例


```
$$ \begin{array}{ccc} 11 & & 12 & & 13 \\ 21 & & 22 & & 23 \end{array} $$
```

$$\begin{array}{ccc} 11 & 12 & 13 \\ 21 & 22 & 23 \end{array}$$

```
$$ \left( \begin{array}{cc} 11 & 12 \\ 21 & 22 \end{array} \right) $$
```

$$\left(\begin{array}{cc} 11 & 12 \\ 21 & 22 \end{array} \right)$$

巧妙使用阵列环境

 阵列环境: `array`

- `array` 环境只能出现在数学环境中!
- 可以使用阵列环境排版一些复杂的公式

array 环境举例

- array 环境举例

\$\$

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

\$\$

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

- 使用 amsmath 宏包: cases 环境 (不自动编号)

\$\$

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

\$\$

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

array 环境举例

- 使用 `cases` 宏包中的 `numcases` 环境 (自动编号)
(需要在导言区调用 `cases` 宏包: `\usepackage{cases}`)

```
\begin{numcases}{|x|=}
  x, & $ x \ge 0 $ \\
 -x, & $ x < 0 $ \\
\end{numcases}
```

$$|x| = \begin{cases} x, & x \geq 0 \quad (2) \\ -x, & x < 0 \quad (3) \end{cases}$$

- 使用 `cases` 宏包中的 `subnumcases` 环境 (自动编号)

```
\begin{subnumcases}{|x|=}
  x, & $ x \ge 0 $ \\
 -x, & $ x < 0 $ \\
\end{subnumcases}
```

$$|x| = \begin{cases} x, & x \geq 0 \quad (4a) \\ -x, & x < 0 \quad (4b) \end{cases}$$

👉 `numcases` 和 `subnumcases` 本身就是是一个数学环境

array 环境举例

- 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)
```

\$\$

$$\left(\begin{array}{cccc} a_{11} & a_{12} & \cdots & a_{1n} \\ & a_{22} & \cdots & a_{2n} \\ & & \ddots & \vdots \\ \mathbf{0} & & & a_{nn} \end{array} \right)$$

array 环境举例

- 分块矩阵

\$\$

\left(

\begin{array}{ccc|ccc}

```
a & \cdots & a & b & \cdots & b \\
& \ddots & \vdots & \vdots & \adots & \\
& & a & b & & \\
& & & c & \cdots & c \\
& & & \vdots & & \vdots \\
\multicolumn{3}{c|}{\raisebox{2ex}[0pt]{\Huge0}} & & & \\
& c & \cdots & c
```

\end{array}

\right)

\$\$

$$\left(\begin{array}{ccc|ccc} a & \cdots & a & b & \cdots & b \\ & \ddots & \vdots & \vdots & \ddots & \\ & & a & b & & \\ \hline & & & c & \cdots & c \\ & & & \vdots & & \vdots \\ & & & c & \cdots & c \end{array} \right)$$

array 环境举例

- 多行下标: array 环境

\$\$

```
A=\sum_{\mbox{\tiny
  $ \begin{array}{c}
    1 \le i \le m \\
    1 \le j \le n \\
    1 \le k \le p
  \end{array}
  } a_{ijk}
```

\$\$

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

- 多行下标: 堆积命令: \atop

\$\$

```
A=\sum_{
  {1\le i\le m}\atop
  {1\le j\le n \atop 1\le k\le p}
} a_{ijk}
```

\$\$

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

array 环境举例

- 多行下标: `amsmath` 宏包 (提供 `\substack` 命令和 `subarray` 环境)

\$\$

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

\$\$

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

\$\$

```
A=\sum\limits_{\begin{subarray}{l}
1 \le i \le m \\
1 \le j \le n \\
1 \le k \le p
\end{subarray}} a_{ijk}
```

\$\$

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