

# 第九章

# 数学公式排版技巧

## 第一部分

华东师范大学数学系

2011.04

# 本讲内容

- ① 数学排版的国际标准
- ② 数学模式中的参数
- ③ 定理定义的排版
- ④ 巧妙使用阵列环境
- ⑤ 多行公式左列问题

# ISO 数学排版公认标准

- ❑ 用粗斜体表示向量, 如:  $B, u, \omega$
- ❑ 用无衬线斜体表示矩阵与 2 阶张量, 如:  $M, D, I$
- ❑ 特殊常数与微分算子用直立体, 如:  $e, i, d$
- ❑ 数与单位构成的度量是一个整体, 不可分割  
数与单位之间留一小间隙, 单位用直立体  
如:  $5.3 \text{ km}, 62 \text{ kg}$

# 数学模式中的字体尺寸

## □ 数学模式中的四种字体尺寸

`\displaystyle` → 简记为 D

`\textstyle` → 简记为 T

`\scriptstyle` → 简记为 S

`\scriptscriptstyle` → 简记为 SS

- ☞ 对于变量, D 尺寸和 T 尺寸是一样的
- ☞ 数学模式中的变量和数学符号等自动选择尺寸
- ☞ `eqnarray` 中基本字体尺寸为 D
- ☞ `array` 中基本字体尺寸为 T

① 数学排版的国际标准

② 数学模式中的参数

③ 定理定义的排版

④ 巧妙使用阵列环境

⑤ 多行公式左列问题

# 数学模式中的参数

## □ 数学模式中的参数

☞ `\arraycolsep`

→ `array/eqnarray` 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

# 数学模式中的参数

## □ 数学模式中的参数(续)

☞ `\abovedisplayskip`

- 短行间公式与上方文本的间隔
- 缺省为 0pt plus 2pt

☞ `\belowdisplayskip`

- 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

- 👉 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在公式左边界的左边，则该公式为短公式，否则为长公式
- 👉 例 901.tex
- 👉 多行环境中的公式均是长公式  
如：`eqnarray`, `eqnarray*`
- 输入行间公式时，数学环境中不能有空行！

① 数学排版的国际标准

② 数学模式中的参数

③ 定理定义的排版

④ 巧妙使用阵列环境

⑤ 多行公式左列问题

# 定理类环境

- **定理类环境**: 定理, 引理, 推论, 命题, 定义等
  - ☞ 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  $x^n + y^n = z^n$ .*

# 定理类环境的实现

## □ 定理类环境的实现方法

👉 首先用 `\newtheorem` 命令定义一个定理类环境

```
\newtheorem{定理环境名}{标题}[主计数器]
```

# 定理类环境的实现

## □ 定理类环境的实现方法

- ☞ 首先用 `\newtheorem` 命令定义一个定理类环境

```
\newtheorem{定理环境名}{标题}[主计数器]
```

- ☞ 调用该环境

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

# 定理类环境的实现

## □ 定理类环境的实现方法

☞ 首先用 `\newtheorem` 命令定义一个定理类环境

```
\newtheorem{定理环境名}{标题}[主计数器]
```

☞ 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

# 定理类环境的实现

## □ 多个定理类环境使用同一个计数器

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

- ☞ 新定义的定理环境与已定义的旧定理环境共用一个计数器（例 901.tex）

## 定理类环境的实现

### □ 多个定理类环境使用同一个计数器

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

- ☞ 新定义的定理环境与已定义的 旧定理环境 共用一个计数器 (例 901.tex)

### □ 定理环境缺省格式

- ☞ 标题与编号用黑体, 正文用 *italic* 斜体
- ☞ 含中文时, 一般需指定字体

## 定理环境宏包 amsthm

- ❑  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$  提供了一个加强定理环境功能的宏包

```
\usepackage{amsthm}
```

- 👉 新增命令 `\newtheorem*`, 创建不带编号定理环境

- 👉 提供 3 中预定义的定理格式

`plain`: 标题与编号均采用黑体, 正文用斜体

`definition`: 标题与编号为黑体, 正文正常字体

`remark`: 标题与编号为斜体, 正文用正常字体

- 👉 选择格式: `\theoremstyle{格式}`

→ 在该命令后调用 `\newtheorem` 命令创建的定理环境, 将采用指定的格式

# 定理环境宏包 amsthm

👉 自定义定理类环境格式: `\newtheoremstyle`

→ 具体用法参见该宏包手册 `amsthm.pdf`

→ 详细示例参见 `thmtest.tex`

→ 例 `901.tex`

## 定理环境宏包 amsthm

- 👉 自定义定理类环境格式: `\newtheoremstyle`
  - 具体用法参见该宏包手册 `amsthm.pdf`
  - 详细示例参见 `thmtest.tex`
  - 例 `901.tex`
- 👉 也可是使用 `ntheorem` 宏包

## 定理环境宏包 amsthm

👉 自定义定理类环境格式: `\newtheoremstyle`

→ 具体用法参见该宏包手册 `amsthm.pdf`

→ 详细示例参见 `thmtest.tex`

→ 例 `901.tex`

👉 也可是使用 `ntheorem` 宏包

👉 `\swapnumber`

调用该命令后, 创建的定理类环境中编号显示在标题前

# 定理环境宏包 amsthm

☞ 证明环境: `proof`

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

- 若使用了**标题**, 则输出该标题, 否则输出 *Proof*.
- 证明环境结束时自动输出**证毕**记号 → □

# 定理环境宏包 amsthm

☞ 证明环境: `proof`

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

→ 若使用了**标题**, 则输出该标题, 否则输出 *Proof*.

→ 证明环境结束时自动输出**证毕**记号 → □

☞ 证明命令: `\proof`

☞ 证毕记号: `\qed`

# 定理环境宏包 amsthm

☞ 证明环境: `proof`

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

→ 若使用了**标题**, 则输出该标题, 否则输出 *Proof*.

→ 证明环境结束时自动输出**证毕**记号 → □

☞ 证明命令: `\proof`

☞ 证毕记号: `\qed`

→ 修改证毕记号: `\qedsymbol`

- ① 数学排版的国际标准
- ② 数学模式中的参数
- ③ 定理定义的排版
- ④ 巧妙使用阵列环境
- ⑤ 多行公式左列问题

# 巧妙使用阵列环境

## □ 阵列环境: `array`

- 👉 `array` 环境只能在数学模式中使用!
- 👉 可以使用阵列环境排版一些复杂的公式

## array 环境举例

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

 使用 array 环境(不自动编号)

```
|x|=\left\{
\begin{array}{rl}
x, & x\ge 0\\[2pt]
-x, & x<0
\end{array}
\right.
```

## array 环境举例

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

👉 使用 `amsmath` 宏包: `cases` 环境(不自动编号)

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

## array 环境举例

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

(2)

 使用 `cases` 宏包: `numcases` 环境(自动编号)

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

## □ 三角形矩阵

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

---

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

## array 环境举例

## □ 分块和加边矩阵

$$\left( \begin{array}{ccc|ccc}
 a & \cdots & a & b & \cdots & b \\
 & \ddots & \vdots & \vdots & \ddots & \\
 & & a & b & & \\
 \hline
 & & 0 & c & \cdots & c \\
 & & & \vdots & & \vdots \\
 & & & c & \cdots & c
 \end{array} \right) \left. \begin{array}{l} \\ \\ \\ \\ \\ \\ \\ \end{array} \right\} \begin{array}{l} p \\ \\ \\ \\ q \\ \\ \\ \end{array}$$

$$\underbrace{\hspace{10em}}_m \quad \underbrace{\hspace{10em}}_n$$

(例 901.tex)

## array 环境举例

□ 多行下标问题  $A = \sum_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n \\ 1 \leq k \leq p}} a_{ijk}$

→ 使用 array 环境

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

## array 环境举例

□ 多行下标问题  $A = \sum_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n \\ 1 \leq k \leq p}} a_{ijk}$

---

→ 使用堆积命令: `\atop`

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

## array 环境举例

👉 amsmath 宏包: `\substack` 命令或 `subarray` 环境

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

或

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

- ① 数学排版的国际标准
- ② 数学模式中的参数
- ③ 定理定义的排版
- ④ 巧妙使用阵列环境
- ⑤ 多行公式左列问题

## 多行公式左列问题

### □ `\lefteqn{公式}`

- 👉 通常出现在 `eqnarray` 环境中  
用于解决左列太长的
- 👉 输出公式, 但将其的长度视为零  
(类似于创建一个宽度为零的盒子)
- 👉 例 901.tex