

这是一个 L^AT_EX 模板

目录

| | | |
|----------|------------------------|----------|
| 1 | 标题 · 章 | 2 |
| 1.1 | 标题 · 节 | 2 |
| 1.1.1 | 小节 1 | 2 |
| 1.1.2 | 小节 2 | 2 |
| 2 | 字体、符号和排版 | 3 |
| 2.1 | 文本字体 | 3 |
| 2.2 | 数学 | 3 |
| 2.2.1 | 公式环境 | 3 |
| 2.2.2 | 字体 | 4 |
| 2.2.3 | 其他符号 | 5 |
| 2.2.4 | 定理环境 | 5 |
| 2.3 | 排版 | 5 |
| 3 | 图表，代码和其他环境 | 6 |
| 3.1 | 浮动体位置参数 | 6 |
| 3.2 | 图片 | 6 |
| 3.2.1 | 插入单个图片 | 6 |
| 3.2.2 | 插入偏左或偏右的图片 | 6 |
| 3.2.3 | 插入两个并排的图片 | 7 |
| 3.2.4 | 解决图片/表格过宽的问题 | 7 |
| 3.3 | 表格 | 8 |
| 3.4 | 算法 | 9 |
| 3.5 | 代码 | 9 |
| 3.6 | 列表 | 10 |
| 3.6.1 | 无序列表 | 10 |
| 3.6.2 | 编号列表 | 10 |
| 3.6.3 | 描述列表 | 10 |

Chapter 1

标题 · 章

这是第一段话。

1.1 标题 · 节

1.1.1 小节 1

小小节 1

小小节 2

无序号小小节

这是一段正文。

1.1.2 小节 2

无序号小节

这是新的一段正文。

Chapter 2

字体、符号和排版

2.1 文本字体

标题字体、`\textbf{}`和`\textsf{}`在 Windows 和 Linux 环境下表现不同。本文档默认在 WSL 环境编译。区别如表 2.1 所示。

表 2.1: 两种系统下的字体区别

| | Windows | Linux |
|------------------------|---------|-------|
| 标题默认字体 | 黑体 | 宋体 |
| <code>\textsf{}</code> | 较大的等线字体 | 黑体 |
| <code>\textbf{}</code> | 黑体 | 宋体加粗 |

一般情况下强调字体使用`\textbf{}`即可，在 Linux 环境下如果需要黑体需换成`\textsf{}`。

2.2 数学

2.2.1 公式环境

数学公式环境可以使用：

- `align`: 多行公式，默认居中，多列使用 `&` 对齐。
- `flalign`: 多行公式，默认左对齐，多列使用 `&` 对齐。
- `gather`: 多行公式，居中，无法多列及使用 `&`。
- `equation`: 单行公式。

下面是一个多栏公式的例子，使用`flalign*`环境，注意第一列前用 `&`，其他列前使用 `&&`，换行前再使用一次 `&`。

$$\begin{array}{lll} (1) \frac{1}{5}e^{5t} + C & (2) -\frac{(3-2x)^4}{8} + C & (3) -\frac{1}{2}\ln|1-2x| + C \\ (4) -\frac{1}{2}(2-3x)^{\frac{2}{3}} + C & (5) -\frac{\cos ax}{a} - be^{\frac{x}{b}} + C & (6) -2\cos\sqrt{t} + C \end{array}$$

$$(7) -\frac{1}{2}e^{-x^2} + C \qquad (8) \frac{1}{2}\sin(x^2) + C \qquad (9) -\frac{\sqrt{2-3x^2}}{3} + C$$

$$\begin{array}{ll} (1) \frac{1}{5}e^{5t} + C & (2) -\frac{(3-2x)^4}{8} + C \\ (3) -\frac{1}{2}\ln|1-2x| + C & (4) -\frac{1}{2}(2-3x)^{\frac{2}{3}} + C \\ (5) -\frac{\cos ax}{a} - be^{\frac{x}{b}} + C & (6) -2\cos\sqrt{t} + C \\ (7) -\frac{1}{2}e^{-x^2} + C & (8) \frac{1}{2}\sin(x^2) + C \\ (9) -\frac{\sqrt{2-3x^2}}{3} + C & \end{array}$$

在模板中有`\allowdisplaybreaks`[4] 允许较长的公式在中间换页。

2.2.2 字体

正文默认使用正体，数学环境默认使用斜体。

使用正体的情况：物理量的单位，单词缩写表示的下（上）标，一些常数（ π, e, i ），微分算符 d ，特殊函数（ \sin, \ln ）。向量和矩阵使用粗斜体。

希腊字母的正体，使用`\unicode-math`宏包（`upgreek`宏包会改变字体），希腊字母加粗使用`\symbf{}`，`\symbfit{}`或`\symbfup{}`，效果如下：

$\alpha\beta\gamma\delta\epsilon\zeta\eta\theta\iota\kappa\lambda\mu\nu\xi\omicron\pi\rho\sigma\tau\upsilon\phi\chi\psi\omega AB\Gamma\Delta EZH\Theta IK\Lambda MN\xi\omicron\pi\rho\sigma\tau\upsilon\phi\chi\psi\omega$
 $(\text{up-})\alpha\beta\gamma\delta\epsilon\zeta\eta\theta\iota\kappa\lambda\mu\nu\xi\omicron\pi\rho\sigma\tau\upsilon\phi\chi\psi\omega\Gamma\Theta\Lambda\xi\pi\sigma\tau\upsilon\phi\chi\psi\omega$
 $(\text{bfit})\alpha\beta\gamma\delta\epsilon\zeta\eta\theta\iota\kappa\lambda\mu\nu\xi\omicron\pi\rho\sigma\tau\upsilon\phi\chi\psi\omega AB\Gamma\Delta EZH\Theta IK\Lambda MN\xi\omicron\pi\rho\sigma\tau\upsilon\phi\chi\psi\omega$
 $(\text{bf})\alpha\beta\gamma\delta\epsilon\zeta\eta\theta\iota\kappa\lambda\mu\nu\xi\omicron\pi\rho\sigma\tau\upsilon\phi\chi\psi\omega AB\Gamma\Delta EZH\Theta IK\Lambda MN\xi\omicron\pi\rho\sigma\tau\upsilon\phi\chi\psi\omega$
 $(\text{bfup})\alpha\beta\gamma\delta\epsilon\zeta\eta\theta\iota\kappa\lambda\mu\nu\xi\omicron\pi\rho\sigma\tau\upsilon\phi\chi\psi\omega AB\Gamma\Delta EZH\Theta IK\Lambda MN\xi\omicron\pi\rho\sigma\tau\upsilon\phi\chi\psi\omega$

小结：对小写希腊字母，`up`前缀正体，`\symbf{}`加粗斜体同`\symbfit{}`，`\symbfup{}`加粗正体；对大写希腊字母，`up`前缀正体同默认，`var`前缀斜体，`\symbf{}`加粗正体同`\symbfup{}`，`\symbfit{}`加粗斜体。

英文字母：

$abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ$
 $(\text{rm})abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ$
 $(\text{bf})abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ$
 $(\text{bf})abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ$
 $(\text{bfit})abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ$
 $(\text{bb})\mathfrak{A}\mathfrak{B}\mathfrak{C}\mathfrak{D}\mathfrak{E}\mathfrak{F}\mathfrak{G}\mathfrak{H}\mathfrak{I}\mathfrak{J}\mathfrak{K}\mathfrak{L}\mathfrak{M}\mathfrak{N}\mathfrak{O}\mathfrak{P}\mathfrak{Q}\mathfrak{R}\mathfrak{S}\mathfrak{T}\mathfrak{U}\mathfrak{V}\mathfrak{W}\mathfrak{X}\mathfrak{Y}\mathfrak{Z}$
 $(\text{cal})abcdefghijklmnopqrstuvwxyz\mathcal{A}\mathcal{B}\mathcal{C}\mathcal{D}\mathcal{E}\mathcal{F}\mathcal{G}\mathcal{H}\mathcal{I}\mathcal{J}\mathcal{K}\mathcal{L}\mathcal{M}\mathcal{N}\mathcal{O}\mathcal{P}\mathcal{Q}\mathcal{R}\mathcal{S}\mathcal{T}\mathcal{U}\mathcal{V}\mathcal{W}\mathcal{X}\mathcal{Y}\mathcal{Z}$

对于常用正体的字母，在配置文件中增加快捷键：

$$e^{i\pi} + 1 = 0 \qquad \frac{dy}{dx} = \frac{d}{dx}f(x) \qquad \delta x$$

积分式的表示：

$$\int_a^b f(x) \, dx = F(x) \Big|_a^b = F(x) \Big|_a^b = F(x) \Big|_a^b = F(x) \Big|_a^b$$

复数实部和虚部的表示： $z = a + bi$, $\Re(z) = a$, $\Im(z) = b$ 。使用正体字母可以使用自定义命令`\re`和`\im`：
 $\Re(z) = a$, $\Im(z) = b$ 。

2.2.3 其他符号

文本中的分式 $1/2$ ，虚位 命令； ℓ ∇ 30° 。

2.2.4 定理环境

定理 2.2.1 (定理名字). 这是一个定理。

定义 2.2.1 (定义名字). 这是一个定义。

证明. 这是一个证明。□

例 2.2.1. 这是一个例子。

解. 这是一个解。□

2.3 排版

取消新段落前空格

加入一个水平线：

空格

| 代码 | 效果 | 长度 | 是否需要 <code>amsmath</code> |
|--|-----|----------|---------------------------|
| <code>a\quad b</code> | a b | 1 em | 否 |
| <code>a\qquad b</code> | a b | 2 em | 否 |
| <code>a\enspace b</code> | a b | 0.5 em | 否 |
| <code>a\;b</code> | a b | 5/18 em | 是 |
| <code>a\:b</code> | a b | 4/18 em | 是 |
| <code>a\,b</code> 或者 <code>a\thinspace b</code> | a b | 3/18 em | 否 |
| <code>a\!b</code> 或者 <code>a\negthinspace b</code> | a b | -3/18 em | 是 |

图 2.1: 不同空格及其宽度

使用`a~b`:a b 不可分断空格，在此处不会换行。

使用 (同高宽) 或 (宽为 0) 或 (高为 0) 产生指定占位字符长度的空白。

Chapter 3

图表，代码和其他环境

3.1 浮动体位置参数

h 将浮动体放置在当前位置，如果当前位置无法容纳，则放置在页面顶部。

t 将浮动体放置在页面顶部。

b 将浮动体放置在页面底部。

p 将浮动体放置在单独一页中。

H 强制将浮动体放置在当前位置，而不管当前位置是否能容纳。

此外,还可以使用叹号! 来增加强制性,例如[!t]表示强制将浮动体放置在页面顶部。一般用[ht]或[htbp]。

3.2 图片

3.2.1 插入单个图片



图 3.1: Enter Caption

3.2.2 插入偏左或偏右的图片

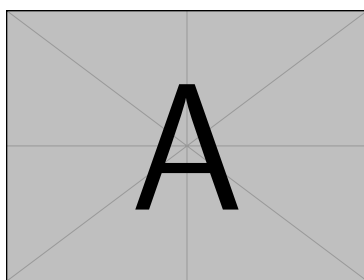


图 3.2: 示例图片标题

如图 3.3 所示，有一个边长为 $2b$ 的正三角形，从中挖去一个倒放的正三角形，重复这个步骤，被挖去的面积之和形成一个无穷级数。

(a) 求这个无穷级数；

(b) 求这个级数和；

(c) 是否原来三角形中的每个点都被挖去了？解释为什么。

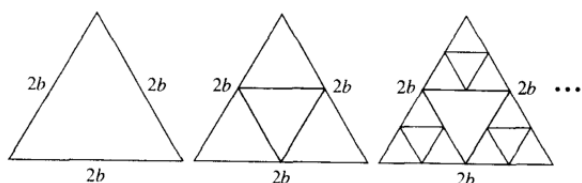


图 3.3: 一个三角形图形序列

```

1 \begin{wrapfigure}
2   [行数]{位置}{超出长度}{宽度}<图形> % 注意，行数两边是方括号，不是花括号
3 \end{wrapfigure}

```

3.2.3 插入两个并排的图片

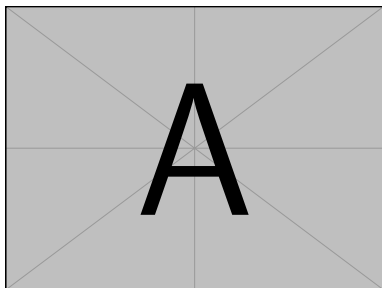


图 3.4: 标题 1

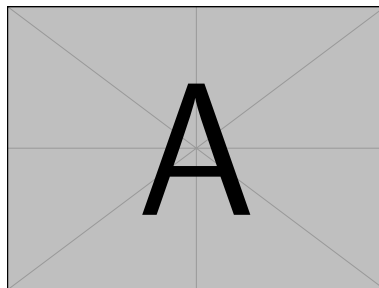


图 3.5: 标题 2

3.2.4 解决图片/表格过宽的问题

```

1 % \usepackage{graphicx}
2 \resizebox{\textwidth}{!}{
3   %图片/表格内容
4 }

```


3.3 表格

三线表，如表 3.1。

表 3.1: 三线表示例

| 文件名 | 描述 |
|-----------------|------------------|
| thuthesis.dtx | 模板的源文件，包括文档和注释 |
| thuthesis.cls | 模板文件 |
| thuthesis-*.bst | BibTeX 参考文献表样式文件 |

表格如果有附注，尤其是需要在表格中进行标注时，可以使用`threeparttable`宏包。

表 3.2: 带附注的表格示例

| 文件名 | 描述 |
|----------------------------|------------------|
| thuthesis.dtx ^a | 模板的源文件，包括文档和注释 |
| thuthesis.cls ^b | 模板文件 |
| thuthesis-*.bst | BibTeX 参考文献表样式文件 |

^a 可以通过 `xelatex` 编译生成模板的使用说明文档。

^b 更新模板。

如某个表需要转页接排，可以使用`longtable` 宏包，需要在随后的各页上重复表的编号。编号后跟表题（可省略）和“（续）”，置于表上方。续表均应重复表头。

表 3.3: 跨页长表格的表题

| 表头 1 | 表头 2 | 表头 3 | 表头 4 |
|--------|------|------|------|
| Row 1 | | | |
| Row 2 | | | |
| Row 3 | | | |
| Row 4 | | | |
| Row 5 | | | |
| Row 6 | | | |
| Row 7 | | | |
| Row 8 | | | |
| Row 9 | | | |
| Row 10 | | | |
| Row 11 | | | |
| Row 12 | | | |
| Row 13 | | | |
| Row 14 | | | |
| Row 15 | | | |

续表 3.3 跨页长表格的表题

| 表头 1 | 表头 2 | 表头 3 | 表头 4 |
|--------|------|------|------|
| Row 16 | | | |
| Row 17 | | | |
| Row 18 | | | |
| Row 19 | | | |
| Row 20 | | | |

3.4 算法

Algorithm 1 Calculate $y = x^n$

输入: $n \geq 0$

输出: $y = x^n$

```
 $y \leftarrow 1$   
 $X \leftarrow x$   
 $N \leftarrow n$   
while  $N \neq 0$  do  
  if  $N$  is even then  
     $X \leftarrow X \times X$   
     $N \leftarrow N/2$   
  else { $N$  is odd}  
     $y \leftarrow y \times X$   
     $N \leftarrow N - 1$   
  end if  
end while
```

3.5 代码

```
1  #include<stdio.h>  
2  
3  int main(){  
4      printf("Hello!\n"); // 输出你好  
5      return 0;  
6  }
```

```
1  import random  
2  import collections  
3  Card = collections.namedtuple('Card', ['rank', 'suit'])  
4  # 一个叫做 FrenchDesk 的类。a class named FrenchDesk.
```

```

5 class FrenchDesk:
6     ranks = [str(n) for n in range(2, 11)] + list('JQKA')
7     suits = 'spades_diamonds_clubs_hearts'.split()
8
9     def __init__(self):
10         self._cards = [Card(rank, suit) for rank in self.ranks for suit in self.suits]
11
12     def __len__(self):
13         return len(self._cards)
14
15     def __getitem__(self, position):
16         return self._cards[position]
17 desk = FrenchDesk()

```

3.6 列表

3.6.1 无序列表

- 文本 1
- 文本 2

3.6.2 编号列表

1. 文本 1
2. 文本 2

3.6.3 描述列表

First 文本 1

Second 文本 2