

kaobook 的中文支持和扩展

Latex 的学习系列内容

kaobook 文档类的中文适配模板

根据您的需要自定义此页面

Tigcat*

August 14, 2025

虚拟书社

* 一个 L^AT_EX 热爱者

免责声明

你可以编辑此页面以满足你的需求。例如，本文中包含了免责声明、书籍说明及其他一些信息。此页面基于 Ken Arroyo Ohori 的论文相应页面和 [kaobook](#) 宏包作了最小的修改。

无版权

©️基于 [kaobook](#) 宏包的中文翻译和使用已通过 CC0 许可发布至公共领域。在法律允许的范围内，我放弃对此作品的所有版权及相关或邻接权利。

要查看 CC0 许可协议，请访问：

<http://creativecommons.org/publicdomain/zero/1.0/>

书籍说明

本文件使用 [KOMA-Script](#) 和 [L^AT_EX](#) 的 [kaobook](#) 宏包进行排版。

本文件的源代码可在以下地址获取：

<https://github.com/wenhq/template-kaobook-zh> （欢迎您进行贡献！）

出版商

本文件首次发布于 2025 年 2 月，由 虚拟书社 出版。

序言

LaTeX 就像一座桥梁，将数学与艺术、形式与内容、理性与感性完美融合。它不仅展现了世界的和谐与秩序，还赋予了文字与符号生命，使自然哲学的心灵与灵魂得以深刻表达。在它的帮助下，数学之美被提升到一种诗意的境界，每一个公式、每一行文字，都蕴含着无尽的优雅与智慧，体现了人类对知识与美的追求。

1. 这段话是瞎写的；
2. 这段话是用 GPT-4.1 mini 瞎写的。

LaTeX is like a bridge, perfectly blending mathematics with art, form with content, reason with emotion. It not only showcases the harmony and order of the world but also breathes life into words and symbols, allowing the spirit and soul of natural philosophy to be deeply expressed. With its help, the beauty of mathematics is elevated to a poetic realm, where every formula, every line of text, contains infinite elegance and wisdom, reflecting humanity's pursuit of knowledge and beauty.

1. This paragraph is nonsense.
2. This paragraph is nonsense written using GPT-4.1 mini.

Tigcat Wh Q

目录

序言	i
目录	iii
1. 如何使用本项目	1
1.1. 概述	1
1.2. 注意事项	1
1.3. 快速上手	1
1.3.1. 导言区	2
1.3.2. 文档主体	2
1.4. 代码目录结构	3
2. 主要变动点	5
2.1. 原样式修改	5
2.1.1. kaobook.cls 的变动	5
2.1.2. kao.sty 的变动	6
2.1.3. kaorefs.sty 的变动	6
2.2. kao-zh.sty 中文支持	6
2.2.1. ctex 宏包	7
2.2.2. 标题文本的中文重定义	7
2.2.3. 自定义字体	7
3. 使用 kaobook 的姿势	9
3.1. 插入图片	9
3.2. 插入表格	9
3.3. 使用注释	9
3.4. 使用提示框	9
3.5. 其他	9
附录	11
A. 字体相关	13
A.1. Times New Roman 字体	13
B. 画图 tikz	15
B.1. 使用代码嵌入还是文件嵌入	15
B.1.1. 比较	15

B.1.2. 最佳实践: TikZ 外部化功能	16
致谢	17

代码列表

1. 文档导言区示例	2
2. 文档主体区域示例	3
3. 删除 scshape 后的 kaobook.cls 文件	6
4. kao-zh.sty 中引入 ctex 宏包	7
5. kao-zh.sty 中的标题文本重定义	7
6. kao-zh.sty 中使用的中文字体	8
7. 指定英文字体的写法	8
8. 引用 fontspec 宏包使用 Times New Roman 字体	13
9. 导言区 external 宏包以及配置	16

Todo list

使用 kaobook 的姿势	9
致谢部分待补充	17

如何使用本项目

1.

1.1. 概述

本项目基于 [kaobook](#) 项目进行中文汉化和二次开发。

kaobook 是一个 LaTeX 文档类，旨在为书籍、报告等长篇文档提供一个优雅、灵活的排版框架。该项目基于 KOMA-Script 构建，并融入了许多独特的设计元素，如宽边距、迷你目录、自定义章节标题等，以满足学术著作和技术文档的高排版要求。

在实际使用 Kaobook 进行文档编写的过程，发现它在处理包含大量不同尺寸图片的文档，尤其是注释说明类操作手册方面表现出色，生成的文档布局整齐、视觉效果良好，且在排版时能够合理分配空间，提升文档的可读性与专业性。

本项目是一个使用 kaobook 的文档模板，为了方便中文文档的快速使用。项目以 kaobook 文档类为基础，通过扩展宏包的方式使用 kao 样式的文档排版，以实现更高效和专业的排版效果。

1.2. 注意事项

项目的作者仅为 LaTeX 的爱好者，因此在项目的开发过程中，难免存在一些考虑不周或使用不当的地方。这些问题可能源于作者对 LaTeX 以及 kaobook 特性和功能的理解尚未完全深入，或者由于实践经验的局限性，导致某些细节未能得到充分优化。

尽管该项目能够满足基本需求，但在复杂文档处理、性能优化或特殊排版要求方面，可能会存在不足之处。用户在使用过程中应注意可能遇到的一些局限性，并根据实际需求进行适当的调整和补充。

1.3. 快速上手

项目的主文件是 `main.tex`，按导言区和文档主体两部分进行说明。

1.1 概述	1
1.2 注意事项	1
1.3 快速上手	1
1.3.1 导言区	2
1.3.2 文档主体	2
1.4 代码目录结构	3

1.3.1. 导言区

文档类 `kaobook` 被选为文档的基础类，适用于书籍和长篇文档的排版，支持双面打印（`twoside`）功能，使得排版更加符合传统书籍的格式要求。为了简化文献引用和参考文献列表的管理，使用了 `kaorefs` 宏包，它能够自动处理文献的引用格式和生成文献列表，确保文献管理的准确性和一致性。

针对中文排版，`kao-zh` 宏包提供了优化的中文排版支持，包括字体、行间距、段落以及对 `kao` 相关宏包的汉化调整，使得中文文档的排版更加规范和易读。此外，为了满足更复杂的排版需求，`kao-ext` 宏包被用于提供扩展功能，允许用户根据需要进行更多自定义，如复杂的表格、数学公式和特殊排版效果。

如 代码 1 所示，使用 `kaobook` 中文模板在导言区的宏包引用实例的写法。

代码 1: 文档导言区示例

```
1 | %! Tex program = xelatex
2 | \documentclass[twoside]{styles/kaobook}
3 | \usepackage{styles/kaorefs}
4 | \usepackage{styles/kao-zh}
5 | \usepackage{styles/kao-ext}
```

这些宏包的结合，使得本书的排版更加精美、规范，并且方便文献管理，提升了整体的排版质量和效率。

1.3.2. 文档主体

在 LaTeX 中，`\frontmatter`、`\mainmatter` 和 `\backmatter` 是用于结构化文档内容的重要命令，特别适用于书籍类文档。这些命令帮助区分文档的不同部分，并控制页码格式。具体而言，`\frontmatter` 用于定义文档的前言部分，`\mainmatter` 标志着正文部分的开始，而 `\backmatter` 则用于定义附录等后续内容。

- ▶ **`\frontmatter` 命令** 标志着文档的前言部分的开始。前言部分通常包含封面、版权信息、致谢、目录等内容，而不涉及文档的正式正文。在这一部分，LaTeX 会使用罗马数字（如 i, ii, iii 等）作为页码格式。这种格式的设置是为了区分前言部分与正文部分，并保持整洁的排版风格。
- ▶ **`\mainmatter` 命令** 标志着文档正式正文部分的开始。在执行该命令后，文档的页码将从 1 开始，并切换为阿拉伯数字（如 1, 2, 3 等）¹。这表明正文部分的内容正式开始，通常包括章节、节、小节等内容。

1: `kaobook` 文档的 `\mainmatter` 命令不仅仅恢复页码格式，还会影响章节标题的排版，提供更加专业和规范的章节排版样式。同时，正文部分的页面布局也会依据 `kaobook` 的设计进行调整，通常包括更宽的页边距和专门设计的章节封面。

- ▶ **\backmatter** 命令标志着文档的附录部分的开始。附录部分通常包括参考文献、索引、附加数据等内容。此时，文档的页码继续使用阿拉伯数字，并且不会进行新的章节编号。

为了方便管理文档，在文档主体区域的代码写法如 代码 2 所示。

```

1  \begin{document}
2  \input{infos/titlepage.tex}
3  \frontmatter
4  \input{infos/preface.tex}
5  \input{infos/toc.tex}
6
7  \mainmatter
8  \input{chapters/ch1.tex}
9  \input{chapters/ch2.tex}
10
11 \backmatter
12 \input{chapters/acks.tex}
13 \end{document}

```

代码 2: 文档主体区域示例

1.4. 代码目录结构

与文档结构对应的是代码目录的结构。在本项目中，将所有要使用的文件组织成若干子目录，确保代码结构清晰、模块化，便于管理与修改。主要的目录有 `styles`、`infos`、`chapters` 等，说明如下：

- ▶ `chapters/`: 存放文档的各个章节文件，每个章节的内容分别保存在独立的 `.tex` 文件中。
- ▶ `font/`: 用于存放文档所需的字体文件，确保文档的排版符合要求。
- ▶ `infos/`: 存放文档的信息文件，如封面 (`titlepage.tex`) 和目录 (`toc.tex`)。
- ▶ `styles/`: 存放样式文件和类文件，这些文件用于定义文档的样式和格式。
- ▶ `main.tex`: 文档的主文件，包含所有内容和结构的汇总。
- ▶ `main.pdf`: 生成的最终 PDF 文件。



主要变动点 2.

2.1. 对原始样式的修改

kaobook 项目通过提供一系列 LaTeX 文档类和宏包，为学术著作和技术文档的排版提供了强大的支持。

Kaobook 模板相关文件包括 `kaobook.cls`、`kao.sty`、`kaobiblio.sty`、`kaorefs.sty` 和 `kaotheorems.sty`，这些文件可以灵活的引用。它们的主要作用为：

- ▶ **kaobook.cls** 作为主文档类文件定义了文档的整体排版框架,包括页边距、章节样式、多级标题等。它支持分页控制、多语言兼容等书籍排版特性，为文档提供统一且专业的外观。
- ▶ **kao.sty** 基础样式包提供了颜色方案、自定义命令和工具函数等底层支持。这些功能被广泛应用于文档中的代码高亮、侧边注释等场景，是其他扩展样式包的基础依赖。
- ▶ **kaobiblio.sty** 是参考文献处理模块，集成并扩展了 `biblatex` 的功能，预设了符合 **kaobook** 风格的引用格式（如作者-年份或数字标号）。它支持多文献库管理，简化了参考文献的插入和格式化过程。
- ▶ **kaorefs.sty** 交叉引用增强工具优化了图表、章节等元素的智能引用显示。它支持超链接跳转和动态标签生成，提高了文档中交叉引用的可读性和易用性。
- ▶ 数学环境定制包，**kaotheorems.sty** 提供了统一风格的定理、引理、证明等数学环境框。这些框支持可折叠功能，便于读者浏览和隐藏数学证明等细节内容。同时，它还支持自定义编号规则和边注标记，满足了复杂数学文档的排版需求。

本项目引入了 `kaobook.cls`、`kao.sty` 和 `kaorefs.sty` 三个文件。在引入原始 `.cls` 或 `.sty` 文件时，尽量避免修改原文件内容，以便在原项目升级时能够方便地进行集成。尽管如此，仍然不可避免地会遇到一些不适应中文排版的警告或错误，需要针对这些问题进行修复和调整。

2.1.1. kaobook.cls 的变动

由于要引入 `styles` 文件夹下的对应文件，因此在第 28 行，将 `\ProvidesClass{kaobook}` 改为 `\ProvidesClass{styles/kaobook}`；

- 2.1 原样式修改 5
 - 2.1.1 kaobook.cls 的变动 5
 - 2.1.2 kao.sty 的变动 6
 - 2.1.3 kaorefs.sty 的变动 6
- 2.2 kao-zh.sty 中文支持 6
 - 2.2.1 ctex 宏包 7
 - 2.2.2 标题文本的中文重定义 .. 7
 - 2.2.3 自定义字体 7

在第 44 行,将 `\RequirePackage{kao}` 改为 `\RequirePackage{styles/kao}`。

另外,因为中文无法支持“小型大写字母”这样的字体,为避免编译告警故删除了第 254 和 255 行的 `\scshape` 命令,如 代码 3 所示。

代码 3: 删除 `scshape` 后的 `kaobook.cls` 文件

```
\addtokomafont{part}{\normalfont\bfseries}
\addtokomafont{partentry}{\normalfont\bfseries}
```

2.1.2. kao.sty 的变动

`kao.sty` 文件同样要指定到 `styles` 文件夹下,修改文件第 1 行为 `\ProvidesPackage{styles/kao}`。

一些兼容性的修改,如第 19 行删除了 `usenames`,改为 `\RequirePackage[dvipsnames,table]{xcolor}`;
第 29 行增加了 `listings=false`,改为 `\AtEndPreamble{\RequirePackage[listings=false]{scrhack}}`;
第 240 行增加了 `singlespacing=true`,改为 `\RequirePackage[singlespacing=true]{scrlayer-scrpage}`。

调整了目录的深度,从 `section` 调整到 `subsection`,文件第 1208 行改为 `bookmarksdepth=subsection`。

2.1.3. kaorefs.sty 的变动

`kaorefs.sty` 文件类似地修改到 `styles` 文件夹下,修改文件第 1 行为 `\ProvidesPackage{styles/kaorefs}`。

注释掉第 41 行,以及从 48 行到 148 行这段多语言支持的部分。

2.2. 新增 kao-zh.sty 中文支持文件

新增的 `kao-zh.sty` 宏包文件是通过 `ctex` 包来支持中文的显示,并且通过这个文件对 `kaobook` 中的关键字展示采用中文重定义,并规定了文档的字体。

2.2.1. ctex 宏包

在 kao-zh.sty 文件中引入 ctex 宏包来支持中文，引用时不指定具体字体，后续使用自定义的开源字体代替。

在 代码 4 里，如果不准备自定义字体，可以将 fontset=none 改为对应的 mac 或 windows。

```
\RequirePackage[UTF8, fontset=none,  
  scheme=chinese]{ctex}
```

代码 4: kao-zh.sty 中引入 ctex 宏包

ctex 是一个专门为支持中文排版而设计的 LaTeX 宏包¹，它是 LaTeX 中是中文排版里最常用的标准方案，为 LaTeX 提供了对中文字符、字体、段落、编码等的良好支持。

1: 关于 ctex 宏包的详细文档，可以参考 [CTeX 官方文档](#)

2.2.2. 标题文本的中文重定义

对于 kao 样式里定义的一些标题文本，在本项目中进行了重新定义²。如 代码 5 里的重新定义命令，可以修改以适配自己需要的中文。

2: 在英文环境中更改特定标签（如目录、插图、表格等）的名称，**应该有更好的修改方式**。

代码 5: kao-zh.sty 中的标题文本重定义

```
\RequirePackage[english]{babel}  
\renewcaptionname{english}{\contentsname}{目录}  
\renewcaptionname{english}{\listfigurename}{插图}  
\renewcaptionname{english}{\listtablename}{表格}  
\renewcaptionname{english}{\indexname}{索引}  
\renewcaptionname{english}{\bibname}{参考文献}  
\renewcaptionname{english}{\figurename}{图}  
\renewcaptionname{english}{\tablename}{表}  
  
\renewcommand{\lstlistlistingname}{代码列表}
```

2.2.3. 自定义字体

为了避免使用系统默认字体可能带了的商业使用法律问题，本项目采用了开源免费商用字体。这部分内容可以根据需要自行修改。

中文字体

本项目采用 [WenYuanFonts](#) 项目提供的文源字体³作为文档字体，另采用 [maple-font](#) 项目提供的 Maple Mono 字体⁴作为文档的等宽字体。所有使用到的字体文件参考 1.4 (代码目录结构) 的内容都放在了 font 目录下，且注意，**字体文件不保证为最新**。

3: 基于思源字体二次开发, 开源免费商用。

4: 开源等宽字体，中英文宽度完美 2:1。

代码 6: kao-zh.sty 中使用的中文字体

```

\setCJKmainfont [Path={font/}, AutoFakeBold,
  ↳ AutoFakeSlant] {WenYuanMincho-Regular.ttf}
% 中文无衬线字体
\setCJKsansfont [Path={font/},
  ↳ BoldFont={WenYuanGothic-Medium.ttf}, AutoFakeSlant]
  ↳ {WenYuanGothic-Regular.ttf}
% 中文等宽字体
\setCJKmonofont [Path={font/},
  ↳ BoldFont={MapleMonoNormalNL-NF-CN-Bold.ttf},
  ↳ ItalicFont={MapleMonoNormalNL-NF-CN-Italic.ttf}]
  ↳ {MapleMonoNormalNL-NF-CN-Light.ttf}

```

5: 有些地方配置了 AutoFakeBold 和 AutoFakeSlant, 没有什么原因, 只是懒得去下载字体了。

如 代码 6 所示⁵, 中文有衬线字体使用文源宋体 (WenYuanMincho)、中文无衬线字体使用文源黑体 (WenYuanGothic)、等宽字体使用 Maple Mono 字体。

英文字体

文档默认的英文字体为 Times New Roman 字体, 参考 A.1 (Times New Roman 字体) 章节内容。可以修改为其他英文字体, 曾经比较喜欢的 TeX Gyre 系列字体如 代码 7 所示。

代码 7: 指定英文字体的写法

```

\setmainfont{TeX Gyre Termes}
\setsansfont{TeX Gyre Heros}
\setmonofont{TeX Gyre Cursor}
\setmathfont{TeX Gyre Schola Math}

```

使用 kaobook 的姿势

3.

使用 kaobook 的姿势

3.1. 插入图片

3.2. 插入表格

3.3. 使用注释

3.4. 使用提示框

3.5. 其他

3.1 插入图片	9
3.2 插入表格	9
3.3 使用注释	9
3.4 使用提示框	9
3.5 其他	9

附录

A. 字体相关

字体的使用是 \LaTeX 里绕不过去的话题，本篇附录用于记录字体使用的一些内容。

A.1. Times New Roman 字体

本文来源 [LaTeX 技巧 885: LaTeX 下使用 Times New Roman 字体的正确姿势](#)，引用为附录内容进行了简化和修改。

Times New Roman 字体由 Monotype 公司于 1932 年发表，并为英国的《泰晤士报》首次采用。这是一个经典的、中规中矩的字体，所以经常被选为西文的标准字体之一。

- ▶ Windows 中使用的 Times New Roman 是 Monotype 在最初字形上稍加修改（字宽方面）得到的。
- ▶ macOS 系统中使用的 Times Roman 是 Linotype 公司出品的，它与 Monotype 家的 Times New Roman 除了个别字形稍有区别之外（相信你看不出来），几乎完全相同。
- ▶ 开源系统中对应的字体，则是 URW 的 Nimbus Roman No9 L 字体，它在 GPL 许可下发布。

它们几乎没有差别。

最推荐的方法是使用 fontspec 系列宏包，直接选取系统中的字体，如 [代码 8](#) 所示。

代码 8: 引用 fontspec 宏包使用 Times New Roman 字体

```
1 \documentclass{article}
2 \usepackage{fontspec}
3 \setmainfont{Times New Roman}
4 \usepackage{mathspec}
5 \setmathfont{Times New Roman}
6 \begin{document}
7 This is the typeface Times New Roman.
8
9 Enjoy!
10 \end{document}
```


B. 画图 tikz

B.1. 使用代码嵌入还是文件嵌入

在 \LaTeX 中使用 TikZ 绘图有两种主要方式：直接嵌入代码和外部生成后嵌入。两种方法各有优缺点。

B.1.1. 比较

先说结论：两种方式都能保留矢量效果；外部生成后嵌入的方式编译速度明显更快。

实践建议：简单图形 - 直接嵌入 \LaTeX 文档；复杂图形 - 使用 `externalize` 功能；超复杂图形 - 考虑完全外部化，使用单独的文件生成后导入；需要频繁修改调试的图形 - 先在单独文件中开发和测试，完成后再整合到文档中。

直接嵌入 TikZ 代码

优点

- 保留完整矢量图质量，缩放不失真
- 图形字体与文档保持一致性
- 便于随时修改图形
- 可引用文档中的变量和参数
- 与文档完全集成，便于版本控制

缺点

- 编译速度较慢，每次都需重新生成图形
- 复杂图形会消耗大量内存
- 调试可能较为困难

外部生成 PDF 后嵌入

优点

- 编译速度快，图形只需生成一次
- 降低主文档复杂度
- 图形可在多个文档中重用
- 可单独调试图形

缺点

- 工作流程较复杂，需维护额外文件
- 修改图形需重新编译外部文件
- 无法直接引用主文档中的变量

B.1.2. 最佳实践：TikZ 外部化功能

可使用 `externalize` 库结合两种方法的优点：

代码 9: 导言区 `external` 宏包以及配置

```
1 \usepackage{tikz}
2 \usetikzlibrary{external}
3 \tikzexternalize[prefix=tikz/]
```

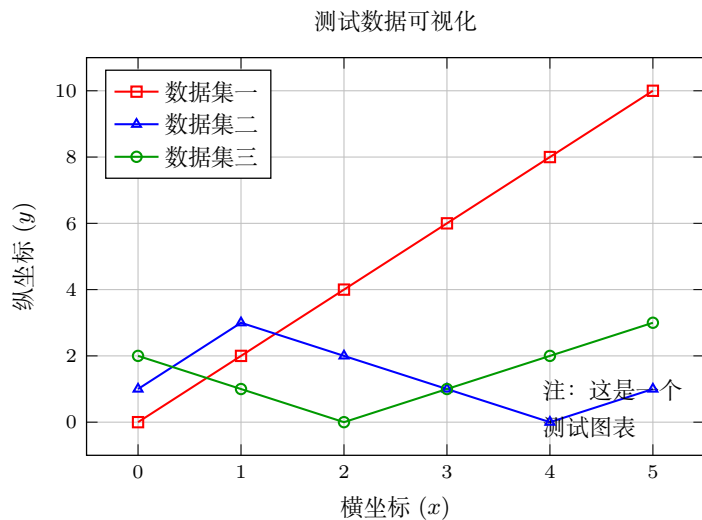


图 B.1.: 这是一个通过独立编译生成的测试数据可视化图表。

致谢

本文源于 2024 年从 [L^AT_EX 工作室](https://github.com/JimRou/template_kaobook) 看到了有关
https://github.com/JimRou/template_kaobook 的介绍。

致
谢
部
分
待
补
充