

学 号： 20XXXXXXXX

密 级： 密级

合肥工业大学

Hefei University of Technology

# 本科毕业设计（论文）

UNDERGRADUATE THESIS



类 型：	【设计或者论文】
题 目：	合肥工业大学学位论文 L <sup>A</sup> T <sub>E</sub> X 模板
专业名称：	【专业全称】
入校年份：	【20XX 级】
学生姓名：	Liangliang
指导教师：	【姓名】【职称】
学院名称：	【系全称】
完成时间：	2023 年 11 月

合 肥 工 业 大 学

本科毕业设计（论文）

合肥工业大学学位论文 L<sup>A</sup>T<sub>E</sub>X 模板

学生姓名：Liangliang

学生学号：20xxxxxxxx

指导教师：【姓名】【职称】

专业名称：【专业全称】

学院名称：【系全称】

2023 年 11 月

**A Dissertation Submitted for the Degree of Bachelor**

**LaTeX-based HFUT Thesis Template LaTeX-based  
HFUT Thesis Template**

By

Liangliang

Hefei University of Technology

Hefei, Anhui, P.R.China

November, 2023

本人郑重声明：所呈交的毕业设计（论文）是本人在指导教师指导下进行独立研究工作所取得的成果。据我所知，除了文中特别加以标注和致谢的内容外，设计（论文）中不包含其他人已经发表或撰写过的研究成果，也不包含为获得合肥工业大学或其他教育机构的学位或证书而使用过的材料。对本文成果做出贡献的个人和集体，本人已在设计（论文）中作了明确的说明，并表示谢意。

毕业设计（论文）作者签名： 签名日期： 年 月 日

本学位论文作者完全了解 合肥工业大学 有关保留、使用毕业设计（论文）的规定，即：除保密期内的涉密设计（论文）外，学校有权保留并向国家有关部门或机构送交设计（论文）的复印件和电子光盘，允许设计（论文）被查阅或借阅。本人授权 合肥工业大学 可以将本毕业设计（论文）的全部或部分内容编入有关数据库，允许采用影印、缩印或扫描等复制手段保存、汇编毕业设计（论文）。

签名日期:      年      月      日

## 摘 要

本项目致力于开发一个高效的  $\text{\LaTeX}$  模板，旨在为合肥工业大学的同学们撰写毕业设计论文提供便利，从而免去繁琐的格式调整工作。相比于 Word， $\text{\LaTeX}$  在处理数学公式、实现精确严格的格式控制，以及管理参考文献方面展现出显著优势。重要的是， $\text{\LaTeX}$  的源文件为纯文本格式，这不仅方便使用版本控制工具进行管理，还极大地简化了协作写作过程。采用  $\text{\LaTeX}$  编写文档意味着可以将注意力集中在内容创作上，而无需担心格式问题。

此外，本项目不仅提供一个模板，还包括一个全面的教程。我们将在模板中加入详尽的注释，以辅助同学们更好地理解 and 掌握  $\text{\LaTeX}$  的使用技巧。我们希望通过这个综合性的模板，同学们不仅能够学习到  $\text{\LaTeX}$  的操作方法，还能掌握一系列有价值的论文写作技巧。我们的目标是让这个模板成为一个完善的学习资源，帮助同学们高效地完成他们的毕业设计论文，同时提升写作和排版能力。

**关键词：**  $\text{\LaTeX}$  模版；合肥工业大学；教程

## ABSTRACT

This project is committed to developing an efficient  $\text{\LaTeX}$  template, specifically designed to facilitate the students of Hefei University of Technology in writing their graduation design theses, thereby eliminating the need for tedious format adjustments. Compared to Word,  $\text{\LaTeX}$  offers significant advantages in handling mathematical formulas, achieving precise and strict format control, and in the management of references. Importantly, the source files of  $\text{\LaTeX}$  are in plain text format, which not only makes it convenient to manage using version control tools but also greatly simplifies the collaborative writing process. Writing documents with  $\text{\LaTeX}$  means that one can concentrate on content creation without the concern for formatting issues.

In addition, this project offers not just a template but also an extensive tutorial. We plan to incorporate detailed annotations within the template to assist students in better understanding and mastering the use of  $\text{\LaTeX}$ . Through this comprehensive template, we aim for the students to not only learn the operational methods of  $\text{\LaTeX}$  but also to acquire a range of valuable thesis writing skills. Our goal is to make this template a complete learning resource, helping students efficiently complete their graduation design theses while enhancing their writing and typesetting skills.

**KEYWORDS:**  $\text{\LaTeX}$  Template, HFUT, Tutorial

# 目 录

毕业设计（论文）独创性声明 .....	I
毕业设计（论文）版权使用授权书 .....	I
摘 要 .....	III
Abstract .....	IV
插图清单 .....	V
表格清单 .....	VI
1 绪论 .....	1
2 使用本模板 .....	1
3 安装 LaTeX .....	2
3.1 Windows .....	2
3.2 MacOS .....	3
3.3 Linux .....	4
4 LaTeX 编辑器 .....	4
4.1 TeXstudio .....	4
4.2 Visual Studio Code .....	4
5 编写 LaTeX 文档 .....	6
5.1 章节和段落 .....	6
5.2 公式 .....	6
5.3 图片 .....	8
5.4 表格 .....	10
5.5 伪代码 .....	11
5.6 引用 .....	11
5.7 参考文献 .....	13
参考文献 .....	14
致 谢 .....	15
附 录 .....	16

## 插图清单

图 2.1 字体放置目录示意 .....	3
图 4.1 TeXstudio 界面展示 .....	4
图 4.2 Visual Studio Code 界面展示 .....	5
图 5.1 示例图片 1 .....	8
图 5.2 示例图片 2 .....	10



## 表格清单

表 5.1 示例表格.....	10
-----------------	----

## 1 绪论

LaTeX（通常发音为”Lay-tech”或”Lah-tech”）是一种高质量排版系统，它特别适用于技术和科学文档的制作。LaTeX 基于 TeX 排版系统，由 Donald Knuth 开发，而 LaTeX 本身则是由 Leslie Lamport 在 20 世纪 80 年代创建的。LaTeX 的主要特点包括：

- **高度专业的排版质量：**LaTeX 能够生成极为专业和美观的文档，特别是对数学公式和科学文档的处理。
- **内容与样式分离：**用户主要关注内容的输入，而文档的格式和样式由 LaTeX 的内部算法处理，确保了文档的一致性和专业性。
- **广泛的定制和扩展性：**通过不同的宏包（packages），LaTeX 可以用于生成各种类型的文档，如学术论文、书籍、幻灯片等。
- **强大的数学公式支持：**LaTeX 提供了丰富的数学符号和公式排版功能，这也是其在学术界广泛使用的主要原因。
- **跨平台：**LaTeX 可以在多种操作系统上运行，如 Windows、Mac OS 和各种 Linux 系统。
- **开源免费：**LaTeX 是免费软件，用户可以自由使用和修改。

虽然 LaTeX 拥有许多优点，但它也有一定的学习曲线。在下面的章节中，我们将介绍如何使用 LaTeX 撰写毕业设计论文。具体而言，我们将从如何使用本模板开始，然后介绍如何安装 LaTeX 编译环境，最后介绍如何使用 LaTeX 撰写文档。

## 2 使用本模板

本模版需使用 XeLaTeX 编译，如果你使用的是 Visual Studio Code，并按照我的教程配置了 LaTeX 环境，那么只需保存即可编译，编译结果会输出到 `./tmp` 文件夹下。如果你使用的是其他编辑器，那么你需要自行配置编译环境。或者使用本项目提供的 `build.bat` (windows) 和 `build.sh` (linux&mac) 进行编译。

使用本模版的同学首先要修改 `info.tex` 文件，将里面的信息修改为自己的信息，并完善 `tex/pages` 文件夹中的文件，包括摘要、致谢、附录等。如果需要添加新的章节，只需要在 `tex` 文件夹下添加对应的 `.tex` 文件，并在 `Thesis.tex` 中引入即可。

本模版还提供了两种不同的编译模式，方便同学们和导师之间沟通，分别是 `preprint` 模式和 `final` 模式。其中，`preprint` 模式下会在文档左侧显示行号，方便沟通时定位问题，并且在这个模式下导师批阅的内容也会显示出来，方便修改。使用的是 `changes` 宏包，有需求的老师可自行学习相关用法，非常简单。

而 `final` 模式下则不会显示行号，也不会显示导师批阅的内容，这是最终的打印版本。默认情况下，本模版使用 `final` 模式，如果需要切换到 `final` 模式，只需要在 `Thesis.tex` 中将 `mode` 设置为 `final` 即可，如下所示：

```
% final 模式
\documentclass{hfut}
% preprint 模式
\documentclass[mode=preprint]{hfut}
% final 模式
\documentclass[mode=final]{hfut}
```

此外，本模版默认使用 `fandol` 字体，如果需要使用 `adobe` 字体，只需要在 `Thesis.tex` 中将 `font` 设置为 `adobe` 即可，如下所示：

```
% fandol 字体
\documentclass{hfut}
% adobe 字体
\documentclass[font=fandol]{hfut}
% adobe 字体
\documentclass[font=adobe]{hfut}
```

`adobe` 字体需要下载，可以从本项目的“`README.md`”文件中找到下载链接，下载完成后解压到当前目录下即可，具体见图 2.1。

尽管默认字体使用比较方便，但推荐使用 `adobe` 字体，更加美观。

## 3 安装 LaTeX

各个平台的安装方法不同，我们将分别介绍 Windows、Mac OS 和 Linux 系统下的安装方法。

### 3.1 Windows

对于 Windows 用户，我们推荐使用 TeX Live。TeX Live 是一个跨平台的 TeX 发行版，它包含了大量的宏包和工具，可以满足大多数用户的需求。

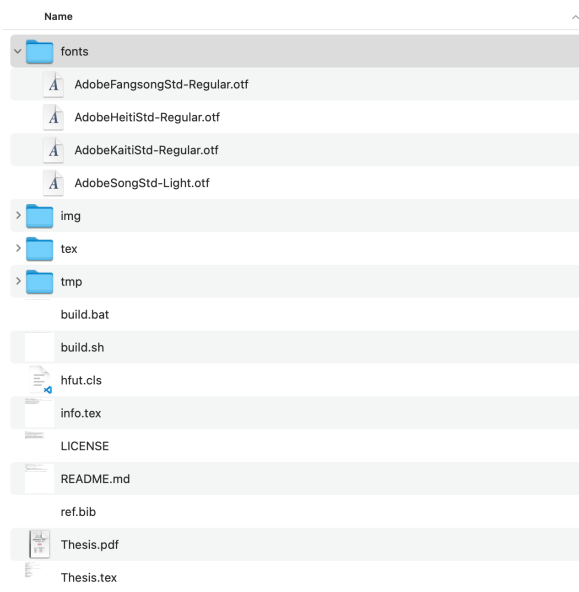


图 2.1 字体放置目录示意

TeX Live 有两种安装方式：网络安装和镜像安装。网络安装需要联网，它会从网络上下载安装文件，因此安装过程会比较慢。镜像安装则不需要联网，它会从本地镜像文件安装，因此安装过程会比较快。我们推荐使用镜像安装，因为它不仅安装速度快，而且可以避免网络问题导致的安装失败。TeX Live 的镜像文件可以从[中科大开源镜像站](#)下载。我们推荐下载 `texlive.iso`，它是一个 ISO 镜像文件，可以使用 Windows 自带的虚拟光驱软件直接挂载并安装。如果你不知道如何挂载 ISO 镜像文件，请自行搜索。

### 3.2 MacOS

对于 MacOS 用户，我们推荐使用 MacTeX。MacTeX 是一个专门为 MacOS 用户准备的 TeX 发行版，它和 TeX Live 是同一个项目的不同版本，两者除了名字不同外，其他都是一样的。MacTeX 的安装方式和 TeX Live 一样，也有网络安装和镜像安装两种方式。我们推荐使用镜像安装，因为它不仅安装速度快，而且可以避免网络问题导致的安装失败。MacTeX 的镜像文件可以从[中科大开源镜像站](#)下载。我们推荐下载 `MacTeX.pkg`，它是一个 pkg 安装包，可以直接双击安装。

### 3.3 Linux

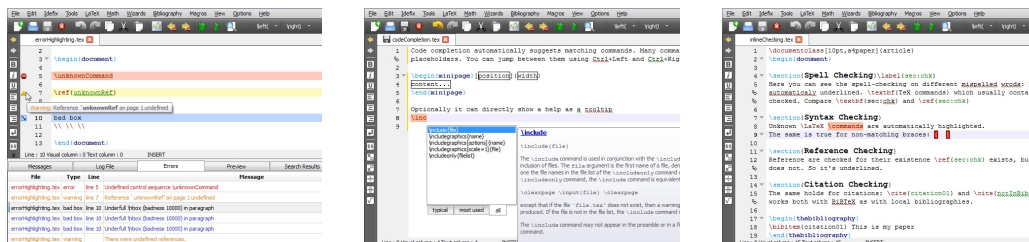
对于 Linux 用户，我们同样推荐使用 TeX Live，并且两者的镜像文件是相同的，都可以从[中科大开源镜像站](#)下载。此外，在 Linux 下还可以使用命令行进行安装，只需要在挂载镜像后在目录下执行 `sudo ./install-tl` 命令即可。当然，在基于 Debian 的 Linux 发行版（如 Ubuntu）下，也可以使用 `sudo apt-get install texlive-full` 命令进行安装，但是这种方式安装的 TeX Live 版本较旧，不推荐使用。

## 4 LaTeX 编辑器

LaTeX 是一种标记语言，它的源文件是纯文本格式，因此可以使用任何文本编辑器进行编辑。由于本人主要使用 Visual Studio Code，因此本教程主要介绍如何在 Visual Studio Code 中使用 LaTeX，其他编辑器只进行简单的介绍。

### 4.1 TeXstudio

TeXstudio 是一款跨平台的 LaTeX 编辑器，它是 TeXmaker 的一个分支，提供了丰富的功能，如语法高亮、智能补全、编译预览等。TeXstudio 的安装包可以从[官网](#)下载，一般来说该软件会随着 TeX Live 的安装一并安装到系统中。



(a) 错误提示

(b) 代码补全

(c) 代码检查

图 4.1 TeXstudio 界面展示

### 4.2 Visual Studio Code

Visual Studio Code 是一款跨平台的轻量级代码编辑器，它支持多种编程语言，如 C/C++、Python、Java、JavaScript 等。最值得一提的是，Visual Studio Code 有着丰富的插件生态系统，用户可以根据自己的需求安装各种插件，从而实现各种功能，比如 LaTeX Workshop 这个插件针对 LaTeX 提供了丰富的功能：语法高亮、智能补全、

编译预览等, Visual Studio Code 的安装包可以从[官网](#)下载。

至于如何在 Visual Studio Code 中配置 LaTeX 环境, 本人早已写过一份教程, 感兴趣的同学可以参考[这里](#)。该教程中搭建的环境支持如下功能:

1. 保存文件时自动编译
2. 支持 XeLaTeX 和 PdLaTeX 编译 (中英文)
3. 编译结果输出到特定文件夹./tmp
4. 英文单词补全, 以及中文翻译
5. LaTeX 语法自动补全
6. 支持快速输入公式, 比如输入 @a 会自动补全为 \alpha
7. 自动补全路径
8. 自动生成矩阵和图片环境
9. 实时预览公式、图片
10. 自动格式化 tex 文件

再加上如今语言模型的支持, 编写 LaTeX 文档将会变得更加轻松。

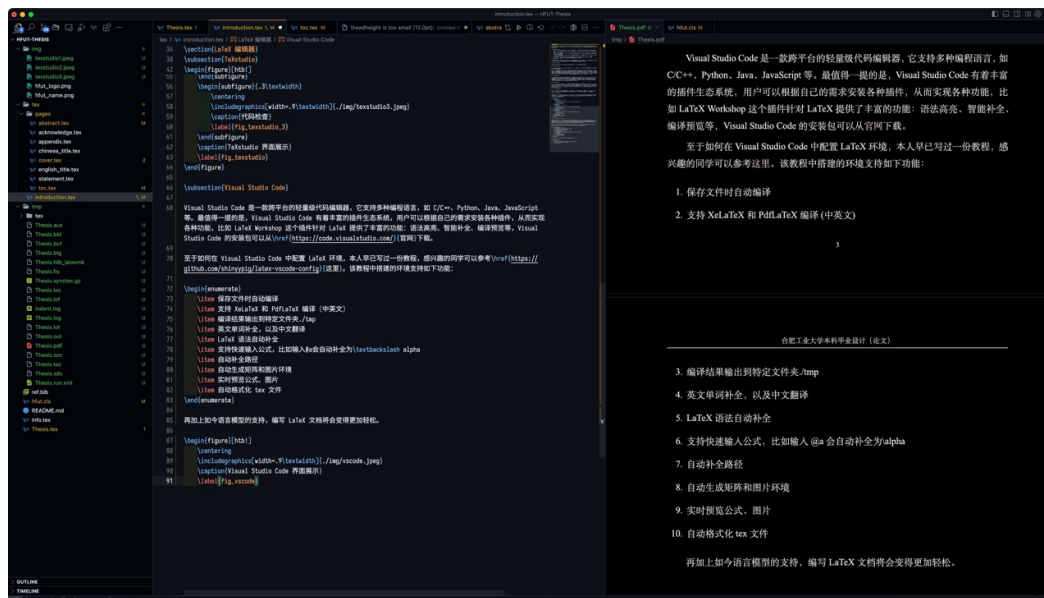


图 4.2 Visual Studio Code 界面展示

## 5 编写 LaTeX 文档

本小节将介绍如何使用 LaTeX 编写文档，包括一些基本知识，以及如何插入公式、插入图片、插入表格、插入参考文献等。

### 5.1 章节和段落

在 LaTeX 中，我们可以使用 `\section`、`\subsection` 和 `\subsubsection` 命令来创建章节。比如，本小节标题对应的代码为

```
\subsection{章节和段落}
```

此外，在 LaTeX 中换行并不会产生新的段落，如果需要新的段落，我们需要在段落之间插入一个空行，即

```
这是第一个段落。
```

```
这是第二个段落。
```

会生成如下效果：

这是第一个段落。

这是第二个段落。

你可以使用 `%` 符号来注释掉一行内容，比如

```
% 这是一行注释
```

```
这是一行正文。
```

会生成如下效果：

这是一行正文。

可以看到，注释掉的内容不会被编译。

### 5.2 公式

在 LaTeX 中，我们可以使用 `\(` 和 `\)` 命令来创建一个行内公式，比如

```
著名的欧拉公式 
$$e^{i\pi} = -1$$
 是数学中最美丽的公式之一。
```

会生成如下效果：

著名的欧拉公式  $e^{i\pi} = -1$  是数学中最美丽的公式之一。

我们可以使用 `\begin{equation}` 和 `\end{equation}` 命令来创建一个带有编号的公式，比如

```
\begin{equation}
    \sum_{i=1}^n i = \frac{n(n+1)}{2}.
\end{equation}
```

会生成如下效果：

$$\sum_{i=1}^n i = \frac{n(n+1)}{2}. \quad (1)$$

如果不需要编号，我们可以使用 `\[` 和 `\]` 命令来创建一个不带编号的公式，比如

```
\[
    \sum_{i=1}^n i = \frac{n(n+1)}{2}.
\]
```

会生成如下效果：

$$\sum_{i=1}^n i = \frac{n(n+1)}{2}.$$

我们可以使用 `\begin{aligned}` 和 `\end{aligned}` 命令来创建一个带有多行的公式，比如

```
\begin{equation}
    \begin{aligned}
        a &= b + c \quad \backslash\backslash \\
        d &= e + f + g
    \end{aligned}
\end{equation}
```

会生成如下效果：

$$\begin{aligned} a &= b + c \\ d &= e + f + g \end{aligned}. \quad (2)$$

其中，`&` 用于对齐，`\backslash\backslash` 用于换行。

还可以使用 `\begin{matrix}` 和 `\end{matrix}` 命令来创建一个矩阵，比如



```

\begin{equation}
\begin{bmatrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{bmatrix}
\end{equation}

```

会生成如下效果：

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad (3)$$

更多的公式用法同学们可以根据自己的需求自行搜索。

### 5.3 图片

在 LaTeX 中，我们可以使用 `\includegraphics` 命令来插入图片，比如

```

\begin{figure}[htb!]
\centering
\includegraphics[width=.9\textwidth]{img/eg1.png}
\caption{示例图片1}
\label{fig_eg1}
\end{figure}

```

会生成如下效果：



图 5.1 示例图片 1

其中，`[htb!]` 用于设置图片的位置，`h` 表示将图片放在当前位置，`t` 表示将图片放在页面顶部，`b` 表示将图片放在页面底部，`!` 表示忽略一些限制。如果不设置图片位置，LaTeX 会自动选择一个合适的位置放置图片，一般来说推荐使用 `htb!`。此外，`\centering` 命令用于将图片居中，`\caption` 命令用于设置图片标题，`\label`

命令用于设置图片标签。`width=.4\textwidth` 表示将图片的宽度设置为当前页面宽度的 0.4 倍，当然你也可以设置图像的高度，比如 `height=.4\textwidth`。最后，`img/eg1.png` 表示图片的路径。通常，LaTeX 支持的图片格式有 `.png`、`.jpg`、`.pdf` 等。如果需要插入多张图片，我们可以使用 `\begin{subfigure}` 和 `\end{subfigure}` 命令来插入子图，比如

```
\begin{figure}[htb!]
  \centering
  \begin{subfigure}{.4\textwidth}
    \centering
    \includegraphics[height=.15\textheight]{./img/cat1.jpeg}
    \caption{小猫 1}
    \label{fig_cat_1}
  \end{subfigure}
  \begin{subfigure}{.25\textwidth}
    \centering
    \includegraphics[height=.15\textheight]{./img/cat2.jpeg}
    \caption{小猫 2}
    \label{fig_cat_2}
  \end{subfigure}
  \begin{subfigure}{.25\textwidth}
    \centering
    \includegraphics[height=.15\textheight]{./img/cat3.jpeg}
    \caption{小猫 3}
    \label{fig_cat_3}
  \end{subfigure}
  \caption{示例图片 2}
  \label{fig_cat}
\end{figure}
```

会生成如下效果：

从中我们可以看到，`\begin{subfigure}` 和 `\end{subfigure}` 命令用于创建子图，`\caption` 命令用于设置子图标题，`\label` 命令用于设置子图标签，`.4\textwidth` 表示子图的宽度为当前页面宽度的 0.4 倍，`.15\textwidth` 表示子图的高度为当前页面高度的 0.15 倍。



图 5.2 示例图片 2

## 5.4 表格

在 LaTeX 中，我们可以使用 `\begin{tabular}` 和 `\end{tabular}` 命令来创建一个表格，比如

```
\begin{table}[htb!]  
  \centering  
  \begin{tabular}{|c|c|c|}  
    \hline  
    1 & 2 & 3 \\  
    \hline  
    4 & 5 & 6 \\  
    \hline  
    7 & 8 & 9 \\  
    \hline  
  \end{tabular}  
  \caption{示例表格}  
  \label{tab_eg}  
\end{table}
```

会生成如下效果：

1	2	3
4	5	6
7	8	9

表 5.1 示例表格

推荐使用 **Tables Generator** 可视化编辑表格，并生成对应的 LaTeX 代码。

## 5.5 伪代码

在 LaTeX 中，我们可以使用 `\begin{algorithm}` 和 `\end{algorithm}` 命令来创建一个伪代码，比如

```
\begin{algorithm}[htb!]
  \caption{示例伪代码}
  \label{alg_eg}
  \begin{algorithmic}[1]
    \REQUIRE $n \geq 0$
    \ENSURE $y = x^n$
    \STATE $y \gets 1$
    \STATE $X \gets x$
    \STATE $N \gets n$
    \WHILE{$N \neq 0$}
      \IF{$N$ is even}
        \STATE $X \gets X \times X$
        \STATE $N \gets N / 2$
      \ELSE[$N$ is odd]
        \STATE $y \gets y \times X$
        \STATE $N \gets N - 1$
      \ENDIF
    \ENDWHILE
  \end{algorithmic}
\end{algorithm}
```

会生成如下效果：

其中，`\caption` 命令用于设置伪代码标题，`\label` 命令用于设置伪代码标签，`\REQUIER` 命令用于设置输入，`\ENSURE` 命令用于设置输出，`\STATE` 命令用于设置状态，`\WHILE` 命令用于设置循环，`\IF` 命令用于设置条件，`\ELSE` 命令用于设置否定条件，`\ENDWHILE` 命令用于结束循环，`\ENDIF` 命令用于结束条件。

## 5.6 引用

在 LaTeX 中，我们可以使用 `\label` 命令来设置标签，使用 `\ref` 命令来引用标签，比如

```
\begin{equation}\label{eq_eg}
  \sum_{i=1}^n i = \frac{n(n+1)}{2}.
\end{equation}
```

**算法 1** 示例伪代码**输入:**  $n \geq 0$ **输出:**  $y = x^n$ 

```

1:  $y \leftarrow 1$ 
2:  $X \leftarrow x$ 
3:  $N \leftarrow n$ 
4: while  $N \neq 0$  do
5:   if  $N$  is even then
6:      $X \leftarrow X \times X$ 
7:      $N \leftarrow N/2$ 
8:   else  $\{N \text{ is odd}\}$ 
9:      $y \leftarrow y \times X$ 
10:     $N \leftarrow N - 1$ 
11:   end if
12: end while

```

式(\ref{eq\_eg}) 是一个示例公式。

会生成如下效果：

$$\sum_{i=1}^n i = \frac{n(n+1)}{2}. \quad (4)$$

式 (4) 是一个示例公式。

不过，本模版内置了 `cleveref` 宏包，因此我们可以使用 `\cref` 命令来引用标签，并且会自动识别标签的类型，比如

```

\begin{equation}\label{eq_eg2}
\int_{-\infty}^{+\infty} e^{-x^2} dx = \sqrt{\pi}.
\end{equation}
\cref{eq_eg2} 是一个示例公式。

```

$$\int \frac{1}{x^2} dx = -\frac{1}{x}. \quad (5)$$

式 (5) 是一个示例公式。

类似地，我们可以引用图片，表格，伪代码等，比如

这是`\cref{fig_eg1}`，这是`\cref{tab_eg}`，这是`\cref{alg_eg}`。

这是图 5.1，这是表 5.1，这是算法 1。

我们还可以一次性引用多个标签，比如

这些是`\cref{fig_cat_1,fig_cat_2,fig_cat_3}`，那些是`\cref{fig_eg1,tab_eg,alg_eg}`。

这些是图 5.2a 到 5.2c，那些是图 5.1、表 5.1 和 算法 1。

## 5.7 参考文献

在 LaTeX 中，我们可以使用 `\cite` 或 `\supercite` 命令来引用参考文献，比如

这是一个示例参考文献`\cite{hfut-thesis}`，而这是一个上标形式的引用`\supercite{hfut-thesis}`。

这是一个示例参考文献[1]，而这是一个上标形式的引用<sup>[1]</sup>。

当然，类似于`\cref`，我们也可以一次性引用多个参考文献。所有的参考文献都保存在主目录的 `ref.bib` 文件中，如果需要添加参考文献，只需要在该文件中添加即可。一般来说，我们可以在 Google Scholar 中找到对应的参考文献，然后点击引用，选择 BibTeX 格式，复制粘贴到 `ref.bib` 文件中即可，其格式如下：

```
@article{hfut-thesis,
  author    = {Liangliang},
  journal   = {LaTeX Template},
  publisher = {Github},
  title     = {HFUT Thesis Template based on LaTeX},
  url       = {https://github.com/shinyypig/HFUT-Thesis},
  year      = {2023}
}
```

## 参考文献

- [1] Liangliang. HFUT Thesis Template based on LaTeX[J/OL]. LaTeX Template, 2023.  
<https://github.com/shinyypig/HFUT-Thesis>.

## 致 谢

这里是致谢。

作者：Liangliang

2023 年 11 月 30 日



## 附 录

这里是附录。如果不需要，请在 Thesis.tex 中注释掉对应代码。