

# 南 开 大 学

## 本 科 生 毕 业 论 文 （ 设 计 ）

中文题目： 南开大学本科毕业生毕业论文模板

外文题目： English Title

学 号： 18xxxxxx

姓 名： XX

年 级： 2018 级

学 院： 软件学院

系 别： 软件工程

专 业： 软件工程

完成日期： 2022 年 5 月 \*\* 日

指导教师： 邢树松

## 关于南开大学本科生毕业论文（设计）的声明

本人郑重声明：所呈交的学位论文，是本人在指导教师指导下，进行研究工作所取得的成果。除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人创作的、已公开发表或没有公开发表的作品内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律责任由本人承担。

学位论文作者签名：

年 月 日

本人声明：该学位论文是本人指导学生完成的研究成果，已经审阅过论文的全部内容，并能够保证题目、关键词、摘要部分中英文内容的一致性和准确性。

学位论文指导教师签名：

年 月 日

## 摘 要

摘要

**关键词：**南开大学，毕业论文，模板

## **Abstract**

abstract

**Key Words:** English; Language

## 目录

第一章 南开大学学位论文格式宏包 NKThesis 使用说明	1
第一节 系统要求	1
第二节 NKThesis 使用说明	1
第三节 引用章节号	2
第四节 中英文间隔	2
第五节 NKThesis 预调用的宏包	3
第六节 图表	3
第七节 字体	3
第八节 参考文献	3
1.8.1 录入参考文献	4
第九节 一些建议	4
1.9.1 关于分数的写法	4
1.9.2 标点	5
1.9.3 列举环境: <code>enumerate</code>	6
第十节 $\text{\TeX}$ 简介	8
1.10.1 什么是 $\text{\TeX}$ / $\text{\LaTeX}$ , 我是否应该选择它	8
1.10.2 我该用什么编辑器?	9
1.10.3 我该去哪里寻找答案?	9
1.10.4 我应该看什么 $\text{\LaTeX}$ 读物?	9
1.10.5 什么知识会过时? 什么不会?	10
1.10.6 插图格式	12
1.10.7 $\text{\LaTeX}$ 作图	12
第二章 说明	13
参考文献	14
致 谢	16

## 第一章 南开大学学位论文格式宏包 NKThesis 使用说明

$\text{\LaTeX}$  由于其专业的排版效果, 擅长处理复杂的数学公式, 免费等优势而得到了广泛的使用, 成为许多西文期刊接收投稿的主流格式; 而  $\text{\LaTeX}$  环境的安装, 编辑器的选择以及编译  $\text{\LaTeX}$  文件的设置等增加了使用者的使用困难, 并且在此过程中出现的问题有难有易, 种类繁多, 不易找到解决方案。Overleaf 作为一档在线  $\text{\LaTeX}$  编辑网站, 使用简单, 支持在线和跨平台编译修改毕业论文, 并且支持多人协作以方便老师和项目组其他人修改。因此, 我们参考南开大学 08 级本科生卓小杨依据《南开大学本科生毕业论文(设计)管理规定(2011 年修订)》修改制成的本科生毕业论文模板, 将南开大学孙文昌老师制作的南开大学研究生论文  $\text{\LaTeX}$  模板移植到了 Overleaf 上, 借鉴未知人士对其的修改, 针对该模板的若干问题和版本旧的不足进行了修改及更新。

本模板参考《南开大学本科生学位论文排版规范(试行)(2016 修正) 版本 1.2.21》修改而成, 更新了封面格式、添加了所需字体、更新参考文献到最新的 bibtex 标准。作者希望能给使用者写作论文带来方便。然而, 作者不保证本模板完全符合学校要求, 也不对由此带来的风险和损失承担任何责任。如果您不接受这项条款, 请不要使用本模板。

### 第一节 系统要求

模板仅在 CTeX2.9.2 下测试  $\text{\LaTeX}$  通过。对于其它 TeX 发行版可能需要做个别修改。本模板改编自孙老师 (<http://222.30.48.141/~sunwch>) 的版本, 格式参照南开大学 2019 年毕业论文格式。

### 第二节 NKThesis 使用说明

本模板可以使用以下两种方式编译:

#### 1. PDF $\text{\LaTeX}$

## 2. X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X[推荐]

例如,

```
pdflatex main
bibtex main      % 处理参考文献
pdflatex main    % 连续编译两遍
pdflatex main    % 以生成正确的文献引用。
```

本模板用到宋体、楷体、仿宋、黑体四种字体. 若需重新配置字体, 请修改 NKTfonts.cfg. 对于 Linux/Mac 下的 TeX Live 2009, 可能需要设置环境变量 OSFONTDIR, 具体内容请参考 texmf.cnf.

我们建议您使用 X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X 编译. 与前两种方法相比, X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X 编译长文档的速度更快, 编译一篇一百多页的论文只需几秒的时间 (SL9400 @ 1.86GHz)。

在改变编译方式前应先删除 \*.toc 和 \*.aux 文件, 因为不同编译方式产生的辅助文件格式可能并不相同。

注意: 使用 X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X 编译时, X<sub>Y</sub>T<sub>E</sub>X 的版本应不低于 0.9995.0(MiKTeX 2.8 或者 TeXLive 2009)。

## 第三节 引用章节号

引用章节号请参考如下格式: 第一章第三节.

## 第四节 中英文间隔

使用 X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X 编译时, 会自动在中英文转换时添加必要的空格. 使用 [PDF]L<sup>A</sup>T<sub>E</sub>X 编译时仅忽略中文之间的空格, 而中英文之间的空格予以保留. 因此, 不管何种编译方式, 您都不需要在中英文间添加~以获得额外的空格. 例如,

这是 English 中文  $x = y$  测试

这是 English 中文  $x = y$  测试

可以看出, 以上两行用 X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X 编译的结果是相同的。

## 第五节 NKThesis 预调用的宏包

NKThesis 已经调用以下宏包，您无须重新调用。

表 1.1 NKThesis 预调用的宏包

编译方式	调用的宏包
X <sub>Y</sub> L <sup>A</sup> T <sub>E</sub> X	xeCJK, CJKnumb, graphicx, mathptmx
[PDF]L <sup>A</sup> T <sub>E</sub> X	CJK, CJKnumb, uniGBK, graphicx, mathptmx

## 第六节 图表

插图例子：



图 1.1 南开大学

## 第七节 字体

一般情况下，您不需要显式地设置字体。如果确实需要，请使用以下命令

宋体： `\rmfamily\upshape` 或 `\songti`

黑体： `\bfseries` 或 `\heiti`

楷体： `\itshape` 或 `\kaiti`

仿宋： `\ttfamily` 或 `\fangsong`

加粗： `\jiacu`

## 第八节 参考文献

参考文献引用： [1–3] 文献<sup>[4, 5]</sup> [6–8] [4] 测试 [9] 或者 [10]

第一次编译 [11][12][13]



### 1.8.1 录入参考文献

本模板采用 `biblatex` 宏包管理参考文献。如果你对此不熟悉,可以参考 `biblatex` 使用手册。

## 第九节 一些建议

### 1.9.1 关于分数的写法

$\text{\LaTeX}$  提供宏命令 `\frac`, 用以打印分数. 为使得版面整齐, 该命令的使用应遵循以下原则:

1. 仅在分行表达式中使用,
2. 不嵌套使用,
3. 不在上下标中使用.

也就是说, 行内表达式和上下标中出现分数时一律用  $a/b$  表示, 如  $(x+2)/((3x^2+4)(7+y))$ . 下面是居中表达式:

$$x^2 = y^{1/2} + 3.$$

多行表达式: 尽量在加、减、乘、等号前换行. 在乘号前换行时, 下一行首用

\times:

$$\begin{aligned}
& |(W_{\psi_1} f)(a, b) - (W_{\psi_1} f)(a_j, b_{j,k})|^2 \\
&= \frac{1}{C_\varphi^2} \left| \iint_{\mathcal{G}} (W_\varphi f)(s, t) \right. \\
&\quad \times \left( (W_{\psi_1} \varphi) \left( \frac{a}{s}, \frac{b-t}{s} \right) - (W_{\psi_1} \varphi) \left( \frac{a_j}{s}, \frac{b_{j,k}-t}{s} \right) \right) \frac{ds dt}{s^{d+1}} \Big|^2 \\
&\leq \frac{1}{C_\varphi^2} \iint_{\mathcal{G}} |(W_\varphi f)(s, t)|^2 \\
&\quad \times \left| (W_{\psi_1} \varphi) \left( \frac{a}{s}, \frac{b-t}{s} \right) - (W_{\psi_1} \varphi) \left( \frac{a_j}{s}, \frac{b_{j,k}-t}{s} \right) \right| \frac{ds dt}{s^{d+1}} \\
&\quad \times \iint_{\mathcal{G}} \left| (W_{\psi_1} \varphi) \left( \frac{a}{s}, \frac{b-t}{s} \right) - (W_{\psi_1} \varphi) \left( \frac{a_j}{s}, \frac{b_{j,k}-t}{s} \right) \right| \frac{ds dt}{s^{d+1}} \\
&= \frac{1}{C_\varphi^2} \dots
\end{aligned} \tag{1.1}$$

### 1.9.2 标点

科技文献中一般用半角标点, 请参考《中国科学》发表的论文.

如果使用全角标点, 可以使用

`\punctstyle{<style>}`

选择标点样式, 有效值为

`quanjiao` (所有标点符号占一个汉字宽度,  
相邻两个标点占一个半汉字宽度)

`banjiao` (所有标点符号占半个汉字宽度)

`hangmobanjiao` (所有标点符号占一个汉字宽度, 行末行首半角)

`kaiming` (句号、叹号、问号占一个汉字宽度, 其他标点占半个汉字宽度)

缺省为全角式。注意: 不论选择哪种样式, 都提供行末对齐 (`margin kerning`) 功能。

**定理 1.1** *LaTeX* 的输出是最完美的。

先证明一个引理

引理 1.2  $T_E X$  文件在不同操作系统下的排版结果完全一致.

证明. 这是证明. □

定理 1.1 的证明. 显然是错的. □

单个带编号的表达式

$$x = y + z \tag{1.2}$$

单个不带编号的表达式

$$y = x - z.$$

不带编号的多行表达式

$$\begin{aligned} x &= y + z \\ &= z - s \\ &< 3. \end{aligned}$$

一些注释

带编号的多行表达式

$$x = y - z, \tag{1.3}$$

$$y = x + z,$$

$$z = y - x. \tag{1.4}$$

引用: 定理1.1的推论是什么呢? 方程式编号: 由 (1.2)(1.4) 式.

### 1.9.3 列举环境: **enumerate**

环境 `enumerate` 已经被改写, 增加了一个可选参数 [字符串], 用以控制所进。  
例如,

```
\begin{enumerate}
  \item This is an example.
  \item This is an example.
```

```
\begin{enumerate}
  \item This is an example.
  \item This is an example.
\end{enumerate}
\end{enumerate}
\begin{enumerate}[Mn]% 字符串"Mn"的宽度为增加的缩进。
    % 缺省值为 [M]
  \item This is an example.
  \item This is an example.
    \begin{enumerate}[Mnn]% 字符串"Mnn"的宽度为增加的缩进。
      \item This is an example.
      \item This is an example.
    \end{enumerate}
  \end{enumerate}
\end{enumerate}
```

的输出为

1. This is an example.
2. This is an example.
  - (a) This is an example.
  - (b) This is an example.
1. This is an example.
2. This is an example.
  - (a) This is an example.
  - (b) This is an example.

## 第十节 $\text{T}_\text{E}\text{X}$ 简介

以下内容是 [milksea@bbs.ctex.org](mailto:milksea@bbs.ctex.org) 撰写的关于  $\text{T}_\text{E}\text{X}$  的简单介绍。注意这不是一个入门教程，不讲  $\text{T}_\text{E}\text{X}$  系统的配置安装，也不讲具体的  $\text{L}^{\text{A}}\text{T}_\text{E}\text{X}$  代码。这里仅仅试图以一些只言片语来解释：进入这个门槛之前新手应该知道的注意事项，以及遇到问题以后该去如何解决问题。

### 1.10.1 什么是 $\text{T}_\text{E}\text{X}/\text{L}^{\text{A}}\text{T}_\text{E}\text{X}$ ，我是否应该选择它

$\text{T}_\text{E}\text{X}$  是最早由高德纳 (Donald Knuth) 教授创建的一门标记式宏语言，用来排版科技文章，尤其擅长处理复杂的数学公式。 $\text{T}_\text{E}\text{X}$  同时也是处理这一语言的排版软件。 $\text{L}^{\text{A}}\text{T}_\text{E}\text{X}$  是 Leslie Lamport 在  $\text{T}_\text{E}\text{X}$  基础上按内容/格式分离和模块化等思想建立的一集  $\text{T}_\text{E}\text{X}$  上的格式。

$\text{T}_\text{E}\text{X}$  本身的领域是专业排版（即方正书版、InDesign 的领域），但现在  $\text{TeX}/\text{LaTeX}$  也被广泛用于生成电子文档甚至幻灯片等， $\text{T}_\text{E}\text{X}$  语言的数学部分偶尔也在其他一些地方使用。但注意  $\text{T}_\text{E}\text{X}$  并不适用于文书处理（MS Office 的领域，以前和现在都不是）。

选择使用  $\text{T}_\text{E}\text{X}/\text{L}^{\text{A}}\text{T}_\text{E}\text{X}$  的理由包括：

- 免费软件；
- 专业的排版效果；
- 是事实上的专业数学排版标准；
- 广泛的西文期刊接收甚或只接收  $\text{LaTeX}$  格式的投稿；
- .....

不选择使用  $\text{T}_\text{E}\text{X}/\text{L}^{\text{A}}\text{T}_\text{E}\text{X}$  的理由包括：

- 需要相当精力学习；
- 图文混合排版能力弱；
- 仅流行于数学、物理、计算机等领域；
- 中文期刊的支持较差；
- .....

请尽量清醒看待网上经常见到的关于  $\text{T}_\text{E}\text{X}$  与其他软件的优劣比较和口水战。在选择使用或离开之前，请先考虑  $\text{T}_\text{E}\text{X}$  的应用领域，想想它是否适合你的需要。

### 1.10.2 我该用什么编辑器？

编辑器功能有简有繁，特色不一，从简单的纯文本编辑器到繁复的 Emacs，因人而异。基本功能有语法高亮、方便编译预览就很好了，扩充功能和定制有无限的可能。初学者可以使用功能简单、使用方便的专用编辑器，如 TeXWorks、Kile、WinEdt 等，或者类似所见即所得功能的 LyX；熟悉的人可以使用定制性更强的 Notepad++、SciTE、Vim、Emacs 等。这方面的介绍很多，一开始不妨多试几种，找到最适合自己的才是最好的。

另外提醒一句，编辑器只是工作的助力，不必把它看得太重。一些编辑器有极为繁杂的功能，一些编辑器常常会引来黑客们的论战（如 Emacs 与 Vim）。为工作，别为这些浪费太多精力，适用即可。

### 1.10.3 我该去哪里寻找答案？

- 0、绝对的新手，先读完一本入门读物，了解基本的知识。
- 1、无论如何，先读文档！绝大部分问题都是文档可以解决的。
- 2、再利用 Google 搜索，利用 (bbs.ctex.org) 版面搜索。
- 3、清楚、聪明地提出你的问题。

### 1.10.4 我应该看什么 L<sup>A</sup>T<sub>E</sub>X 读物？

这不是一个容易回答的问题，因为有许多选择，也同样有许多不合适的选择。这里只是选出一个比较好的答案。更多更详细的介绍可以在版面和网上寻找（注意时效）。

近两年 T<sub>E</sub>X 的中文处理发展很快，目前没有哪本书在中文处理方面给出一个最新进展的合适综述，因而下面的介绍也不主要考虑中文处理。

1. 我可以阅读英文
  - (a) 我要迅速入门：ltxprimer.pdf (LaTeX Tutorials: A Primer, India TUG)
  - (b) 我要系统学习：A Guide to LaTeX, 4th Edition, Addison-Wesley 有机工业出版社的影印版（《LaTeX 实用教程》）
  - (c) 我要深入学习：要读许多书和文档，TeXbook 是必读的
  - (d) 还有呢？去读你使用的每一个宏包的说明文档

(e) 还有许多专题文档，如讲数学公式、图形、表格、字体等

## 2. 我更愿意阅读中文

(a) 我要迅速入门: lnotes.pdf (LaTeX Notes, 1.20, Alpha Huang)

(b) 我要系统学习: 《LaTeX2 $\epsilon$  科技排版指南》，邓建松（电子版）如果不好找，看《LaTeX 入门与提高》第二版，陈志杰等

(c) 我要深入学习: TeXbook0.pdf (特可爱原本, TeXbook 的中译, xianxian)

(d) 还有呢？英语，绝大多数 TeX 资料还是英文的

### 1.10.5 什么知识会过时？什么不会？

T<sub>E</sub>X 是排版语言，也是广泛使用的软件，并且不断在发展中；因此，总有一些东西会很快过时。作为学习 T<sub>E</sub>X 的人，免不了要看各种各样的书籍、电子文档和网络论坛上的只言片语，因此了解什么知识会迅速过时，什么知识不会是十分重要的。

最稳定的是关于 Primitive T<sub>E</sub>X 和 Plain T<sub>E</sub>X 的知识，也就是 Knuth 在他的《The TeXbook》中介绍的内容。因为 T<sub>E</sub>X 系统开发的初衷就是稳定性，要求今天的文档到很久以后仍可以得到完全相同的结果，因此 Knuth 限定了他的 T<sub>E</sub>X 语言和相关实现的命令、语法。这些内容许多年来就没有多少变化，在未来的一些年里也不会有什么变化。Primitive T<sub>E</sub>X 和 Plain T<sub>E</sub>X 的知识主要包括 T<sub>E</sub>X 排版的基本算法和原理，盒子的原理，底层的 T<sub>E</sub>X 命令等。其中技巧性的东西大多在宏包设计中，初学者一般不会接触到很多；而基本原理则是常常被提到的，譬如，T<sub>E</sub>X 把一切排版内容作为盒子（box）处理。

相对稳定的是关于基本 L<sup>A</sup>T<sub>E</sub>X 2 $\epsilon$  的知识，也包括围绕 L<sup>A</sup>T<sub>E</sub>X 2 $\epsilon$  的一些核心宏包的知识。L<sup>A</sup>T<sub>E</sub>X 2 $\epsilon$  是自 1993 年以来的一个稳定的 L<sup>A</sup>T<sub>E</sub>X 版本，直到最近的一次修订（2005 年）都没有大的变动。L<sup>A</sup>T<sub>E</sub>X 的下一个计划中的版本 L<sup>A</sup>T<sub>E</sub>X 3 遥遥无期，在可预见的将来，L<sup>A</sup>T<sub>E</sub>X 2 $\epsilon$  不会过时。L<sup>A</sup>T<sub>E</sub>X 2 $\epsilon$  的知识是目前大部分 L<sup>A</sup>T<sub>E</sub>X 书籍的主体内容。关于 L<sup>A</sup>T<sub>E</sub>X 的标准文档类（article、report、book、letter、slide 等），关于基本数学公式的输入，文档的章节层次，表格和矩阵，图表浮动体，LR 盒子与段落盒子……这些 L<sup>A</sup>T<sub>E</sub>X 的核心内容都是最常用的，相对稳定的。与 L<sup>A</sup>T<sub>E</sub>X 2 $\epsilon$  相匹配的核心宏包，如 graphics(x)、ifthen、fontenc、doc 等，也同样是

相对稳定的。还有一些被非常广泛应用的宏包，如 `amsmath` 系列，也可以看作是相对稳定的。

简单地说，关于基本  $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  的语言，都是比较稳定的。与之对应，实现或者支持  $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  语言的软件，包括在  $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  基础上建立的新的宏，都不大稳定。

容易过时的是关于第三方  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  宏包的知识、第三方  $\text{T}_{\text{E}}\text{X}$  工具的知识，以及新兴  $\text{T}_{\text{E}}\text{X}$  相关软件的知识等。 $\text{T}_{\text{E}}\text{X}$  和  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  语言是追求稳定的；但无论是宏包还是工具，作为不断更新软件，它们是不稳定的。容易过时的技术很多，而且现在广泛地出现在几乎所有  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  文档之中，因此需要特别引起注意：宏包的过时的原因可能是宏包本身的升级换代带来了新功能或不兼容，也可能是同一功能的更新更好的宏包代替了旧的宏包。前者的典型例子比如绘图宏包 `PGF/TikZ`，现在的 2.00 版功能十分强大，和旧的 1.1x 版相差很大，和更旧的 0.x 版本则几乎完全不同；后者的典型例子比如 `caption` 宏包先是被更新的 `caption2` 宏包代替，后来 `caption` 宏包更新又使得 `caption2` 宏包完全过时。——安装更新的发行版可以避免使用过旧的宏包；认真阅读宏包自带的文档而不是搜索得到的陈旧片断可以避免采用过时的代码。

工具过时的主要原因也是升级换代和被其他工具替换。前者的典型例子是编辑器 `WinEdt` 在 5.5 以后的版本支持 UTF-8 编码，而旧版本不支持；后者的典型例子是中文字体安装工具从 `GBKFonts` 到 `xGBKFonts` 到 `FontsGen` 不断被取代。图形插入是一个在  $\text{T}_{\text{E}}\text{X}$  实现、宏包与外围工具方面都更新很快的东西。在过去，最常用的输出格式是 PS (PostScript) 格式，因此插入的图像以 EPS 为主流。使用 `Dvips` 为主要输出工具，外围工具有 `GhostScript`、`bmeps` 等等，相关宏包有 `graphics` 等，相关文档如《 $\text{LaTeX}2\text{e}$  插图指南》。

但凡提及“ $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  只支持 EPS 图形”的，就是这个过时的时代的产物。事实上  $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  并不限定任何图形格式，只不过是当时的输出格式 (PS) 和工具 (`Dvips`) 对 EPS 情有独钟而已。后来 PDF 格式成为主流，`pdfTEX`、`DVIPDFM`、`DVIPDFMx`、`XYTEX` 等工具则主要支持 PDF、PNG、JPG 格式的图形，涉及一系列工具如 `ImageMagick`、`ebb` 等。

值得特别提出注意的就是，中文处理也一起是更新迅速、容易过时的部分。



而且因为中文处理一直没有一个“官方”的“标准”做法，软件、工具、文档以及网上纷繁的笔记也就显得相当混乱。从八十年代开始的 CCT 系统、天元系统，到后来的 CJK 方式，到近来的 Xe<sub>La</sub>TeX、LuaTeX 方式，中文处理的原理、软件、宏包、配置方式等都在不断变化中。

### 1.10.6 插图格式

前面提到， $\text{LaTeX}$  主要支持 EPS 格式的插图文件，而  $\text{PDFLaTeX}$  则更喜欢 PDF、PNG、JPG 格式的图形。为解决兼容性，最新版的  $\text{PDFLaTeX}$  会自动把 EPS 文件转换为 PDF 文件。因此，使用 EPS 格式的插图可能具有最广泛的通用性。

### 1.10.7 $\text{LaTeX}$ 作图

目前已经有很多优秀的  $\text{LaTeX}$  作图宏包，如 pgf/Tikz 和 pstricks，两者都具有强大的作图能力。

## 第二章 说明

本模板参照南开大学学位论文写作规范编写，仅仅提供了论文的基本格式，包括章节标题和正文字体、字号等等的设置。

您自愿使用这个模板。提供本模板的目的是为了给您的论文写作带来方便，然而，作者不保证这个模板完全符合学校的要求，也不对由此产生的任何后果负责。如果您不同意这些条款，请不要使用这个模板。

参考文献的录入请参考第八节。

## 参考文献

- [1] 陈建军, 车建文, 陈勇·强. 具有频率和振型概率约束的工程结构动力优化设计 [J]. 计算力学学报, 2001, 18(1): 74 ~ 80.
- [2] M. A. Nadkarni, C. K. K. Nair, V. N. Pandey, et al. Characterization of  $\alpha$ -galactosidase from *Corynebacterium murisepticum* and mechanism of its induction [J]. J Gen App Microbiol, 1992, 38: 223 ~ 234.
- [3] 华罗庚, 王元. 论一致分布与近似分析: 数论方法 (I)[J]. 中国科学, 1973, (4): 339 ~ 357.
- [4] 竺可桢. 物候学 [M]. 北京: 科学出版社, 1973.
- [5] 霍夫斯塔主编; 胡祥璧译. 禽病学: 下册. 第 7 版 [M]. 北京: 农业出版社, 1981: 798 ~ 799.
- [6] 姜锡洲. 一种温热外敷药制备方案 [P]. 中国: 88105607.3. July 26, 1989.
- [7] S. P. Timoshenko. Theory of plate and shells. 2nd ed. [M]. New York: McGraw-Hill, 1959: 17 ~ 36.
- [8] 张全福, 王里青. “百家争鸣”与理工科学报编辑工作 [M]. 见: 郑福寿. 学报编辑论丛: 第 2 集. 南京: 河海大学出版社, 1991: 1 ~ 4.
- [9] 张全福, 王里青. “百家争鸣”与理工科学报编辑工作 [M]. 见: 郑福寿. 学报编辑论丛: 第 2 集. 南京: 河海大学出版社, 1991: 1 ~ 4.
- [10] Kai Li. IVY: A Shared Virtual Memory System for Parallel Computing. ICPP (2), 1988, 88: 94.
- [11] Ding-Yong Hong, Chun-Chen Hsu, Pen-Chung Yew, et al. HQEMU: a multi-threaded and retargetable dynamic binary translator on multicores. In: Proceedings of the Tenth International Symposium on Code Generation and Optimization, 2012: 104 ~ 113.
- [12] Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin, et al. Spark: Cluster computing with working sets. HotCloud, 2010, 10(10-10): 95.

- [13] Wenwen Wang, Stephen McCamant, Antonia Zhai, et al. Enhancing Cross-ISA DBT Through Automatically Learned Translation Rules. In: Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems. Williamsburg, VA, USA: Association for Computing Machinery, 2018: 84 ~ 97. <https://doi.org/10.1145/3173162.3177160>.

## 致 谢

致谢要带着感情来写，要有感而发。

在本论文完成之际...， ... 感谢...

再者，在系统设计和开发过程中，我也得到了家人、同学、同事的大力帮助和默默的支持使我顺利地完成学业和论文的写作...

最后衷心地感谢在百忙之中评阅论文和参加答辩的各位专家、教授!