

中图分类号:

学校代码: **10856**

学 号: **M987654321**



上海工程技术大学硕士学位论文

中文题目

作者姓名: 作者 **Mr.Z**

指导教师: 导师 **Mr.H**

专 业: 控制科学与工程

学 院: 电子电气工程学院

申请学位: 工学硕士

完成时间: **11 月 13 日**

评阅人: A 评阅

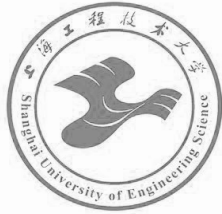
答辩委员会: 主席:A 主席

成员: 成员一、成员二

成员三、成员四

University Code: 10856

Student ID:



English Thesis Title

Candidate: Author Mr.Z

Supervisor: Supervisor Mr.H

**Major: Control Science and
Engineering**

English Thesis Title

Shanghai University of Engineering Science

Shanghai, P.R. China

11, 2022

上海工程技术大学

学位论文原创性声明

本人郑重声明：所递交的学位论文，是本人在导师的指导下，独立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，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期： 年 月 日

上海工程技术大学

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权上海工程技术大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密 ☐，在____年解密后适用本授权书。

本学位论文属于

不保密 ☐。

（请在以上方框内打“√”）

学位论文作者签名：

指导老师签名：

日期： 年 月 日

日期： 年 月 日

中文题目

摘 要

本模板是为上海工程技术大学硕士研究生毕业设计论文编写的 \LaTeX 模板, 旨在让大家专注于论文的内容写作, 而不用花费过多精力在格式的定制和调整上. 本手册是相应的参考, 其中提供了一些环境和命令可以让模板的使用更为方便. 同时需要注意, 使用者需要有一定的 \LaTeX 的使用经验, 至少要会使用 `ctex` 宏包的一些功能, 比如调节字距或修改字体大小等等.

上海工程技术大学硕士毕业设计论文 latex 模板更新地址详细参见[GitHub](#)。

关键字： 关键词 1，关键词 2，关键词 3

English Thesis Title

ABSTRACT

This template is a \LaTeX template written for the graduate design thesis of Shanghai University of Engineering Technology. It aims to let everyone focus on the content of the thesis without spending too much energy on the customization and adjustment of the format. This manual is the corresponding Reference, which provides some environments and commands to make the use of templates more convenient. At the same time, it should be noted that users need to have some experience in using \LaTeX , and at least use some functions of the `ctex` package, such as adjusting Kerning or modifying font size, etc.

For details on the updated address of the latex template of Shanghai University of Engineering and Technology Master's Graduation Design Thesis, please refer to [GitHub](#).

KEY WORDS: keywords1,keywords2,keywords3

目 录

第一章 绪论 3

1.1 关于 L^AT_EX 模板 3

1.2 关于 T_EX 和 L^AT_EX 3

1.3 关于使用的平台和发行版本 3

1.4 使用哪个 T_EX 编辑器 4

第二章 公式、图表 5

2.1 公式 5

第三章 算法、数学定义等 7

3.1 算法 7

3.1.1 插入代码 7

3.1.2 插入伪代码 9

3.2 数学定义等 9

第四章 交叉引用、链接 13

4.1 交叉引用 13

第五章 其他 15

第六章 总结与展望 17

附 录 A 参考文献 19

附 录 B 附录 21

附 录 C 攻读硕士学位期间发表的学术论文及取得的相关科研成果 23

附 录 D 致谢 25

符号和缩略词说明

对文中所用符号缩略词所表示的意义及单位（或量纲）的说明。在目录中不出现。若不需要说明，则删除此页。

这里使用三线表举个例子。这里的符号描述可以使用三线表描述一些符号和缩略词，在学术文章当中经常使用到的一种表格。插入表格，一般最上面线和最下面线宽度为 1.5pt，中间线条宽度为 0.75pt。表格线宽度设置需要用到 booktabs 包，这个包当中包含有 toprule、midrule 和 bottomrule 三类线。下面表 1 是一个三线表简单的例子

表 1: 一个三线表的示例

符号表示	符号意义
e	数学自然对数
π	数学圆周率
ϵ	介电值常数
G	万有引力常数
k	玻尔兹曼常数

第一章 绪论

1.1 关于 L^AT_EX 模板

本模板是本人为写硕士学位论文而写的，本项目地址代码均在[GitHub](#)上。论文模板写得稍微有一些简陋，但是足够用于完成上海工程技术大学的硕士学位论文。

1.2 关于 T_EX 和 L^AT_EX

T_EX 是由图灵奖得主，程序 (program) 和算法 (algorithm) 这两个概念的提出者，《计算机程序设计的艺术》(The Art of Computer Programming) 的作者，著名计算机科学家 Donald E. Knuth（高德纳）发明的排版系统。T_EX 是特别优秀的排版工具，尤其善于处理复杂的图表和公式。L^AT_EX（拉泰赫）是一种基于 T_EX 的排版系统，由由美国计算机学家 Leslie Lamport（莱斯利·兰伯特）在 20 世纪 80 年代初期开发，因此被称为 Lamport Tex，简称 LaTeX。

1.3 关于使用的平台和发行版本

L^AT_EX 拥有众多的发行版，主要有一下几个：

支持平台 发行版	Windows	Linux	OSX
TexLive	✓	✓	✓
MikTex	✓	✗	✗
MacTex	✗	✗	✓

表 1.1: 主要的 L^AT_EX 发行版。

我比较推荐 TexLive，因为它支持主流的平台，而且更新频率也比较高。

1.4 使用哪个 $\text{T}_{\text{E}}\text{X}$ 编辑器

目前 $\text{T}_{\text{E}}\text{X}$ 编辑器是多种多样的,使用一个较好的编辑器对于编写 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 文档是有着非常好的效果。 TexLive 默认的编辑器是 texstudio 编辑器,但是笔者更喜欢 VSCode 作为编辑器,它具有非常好的联想功能和代码主题,看起来也非常省眼。

在线编辑器 现在有很多在线的 latex 编辑器,最为常见和著名的编辑器要属 OverLeaf 。这种在线 latex 平台编辑器能够非常好的支持各种排版的功能,类似于腾讯文档,它支持多人协作编辑文档,具有较好的使用效果。

第二章 公式、图表

2.1 公式

方便快捷写入公式是 \LaTeX 相对于 Word 编辑器最为主要的优势之一，特别是熟练掌握之后，在输入公式的时候具有非常大的提升效果。

\LaTeX 中的公式分为两类，包括有行内公式和行间公式，例如这是一个行间公式 $f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ ，下面举例几个行间公式

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (2.1)$$

例如，定义一个分段函数

$$f(x) = \begin{cases} -x^3 + x + 8 & , x \leq 2 \\ \frac{1}{2}x^2 & , 2 < x \leq 10 \\ x + 10 & , x > 10 \end{cases} \quad (2.2)$$

也可以定义一个多行的连等的等式，定义如下所示

$$\cos 2x = \cos^2 x - \sin^2 x \quad (2.3)$$

$$= 2 \cos^2 x - 1 \quad (2.4)$$

$$= 1 - 2 \sin^2 \quad (2.5)$$

可以将多个等式对齐写在同一个语句块当中，例如麦克斯韦方程组积分形式：

$$\left\{ \begin{array}{l} \oint_l \mathbf{H} \cdot d\mathbf{l} = \iint_S J \cdot d\mathbf{S} + \iint_S \frac{\partial \mathbf{D}}{\partial t} \cdot d\mathbf{S} \\ \oint_l \mathbf{E} \cdot d\mathbf{l} = - \iint_S \frac{\partial \mathbf{B}}{\partial t} \cdot d\mathbf{S} \\ \oint_S \mathbf{B} \cdot d\mathbf{S} = 0 \\ \oint_S \mathbf{D} \cdot d\mathbf{S} = \iiint_V \rho dV \end{array} \right. \quad (2.6)$$

微分形式：

$$\begin{cases} \nabla \times \mathbf{H} = J + \frac{\partial \mathbf{D}}{\partial t} \\ \nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \\ \nabla \cdot \mathbf{B} = 0 \\ \nabla \cdot \mathbf{H} = \rho \end{cases} \quad (2.7)$$

带有矩阵定义的公式：

$$\mathbf{H} = -\boldsymbol{\mu} \cdot \mathbf{B} = -\gamma B_o \mathbf{S}_z = -\frac{\gamma B_o \hbar}{2} \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}. \quad (2.8)$$

在求解凸优化问题的时候，问题研究最后求解归结为以下的方程形式：

$$\underset{x_j, j=1, \dots, N}{\operatorname{argmin}} \sum_{j=1}^N c_j x_j \quad (2.9)$$

$$\text{s.t.} \begin{cases} \sum_{j=1}^N a_{ij} x_j = b_i, & i = 1, \dots, m \\ x_j \geq 0, \end{cases} \quad (2.10)$$

在文章当中每一个公式的后面均可以添加一个 label 的标签，这样就可以应用公式了，例如式 (2.8) 就是刚刚我们表达的矩阵表达式。

第三章 算法、数学定义等

3.1 算法

有时候我们在论文中提到一种算法需要插入代码和一些流程代码来详细说明我们所提出的算法内容，或者是说需要插入一个伪代码片段来说明算法的流程，就此下面详细说明这一个问题。

3.1.1 插入代码

这里我们举例四种语言，即 C/C++ 语言，Java 语言，Python 语言和 MATLAB 语言。

- C 语言

```
# include<sdtio.h>
int main(){
    printf("This is C language block.");
    return 0;
}
```

- C++ 语言

```
# include<iostream>
int main(){
    std::cout<<"This is C++ programming language."<<std::endl;
    return 0;
}
```

- Java 语言

```
public class Main{
```

```
public static void main(String [] args ){
    System.out.printf("This is a Java programing.");
}
}
```

- Python 语言

```
def partition(arr,low,high):
    i = ( low-1 )          # 最小元素索引
    pivot = arr[high]
    for j in range(low , high):
        # 当前元素小于或等于 pivot
        if arr[j] <= pivot:
            i = i+1
            arr[i],arr[j] = arr[j],arr[i]
    arr[i+1],arr[high] = arr[high],arr[i+1]
    return ( i+1 )
# arr[] --> 排序数组
# low --> 起始索引
# high --> 结束索引
# 快速排序函数
def quickSort(arr,low,high):
    if low < high:
        pi = partition(arr,low,high)
        quickSort(arr, low, pi-1)
        quickSort(arr, pi+1, high)
arr = [10, 7, 8, 9, 1, 5]
n = len(arr)
quickSort(arr,0,n-1)
print ("排序后的数组:")
for i in range(n):
    print ("%d" %arr[i]),
```

- Matlab 语言

```
kk=2;[mdd,ndd]=size(dd);
while ~isempty(V)
    [tmpd,j]=min(W(i,V));tmpj=V(j);
    for k=2:ndd
        [tmp1,jj]=min(dd(1,k)+W(dd(2,k),V));
        tmp2=V(jj);tt(k-1,:)= [tmp1,tmp2,jj];
    end
    tmp=[tmpd,tmpj,j;tt];[tmp3,tmp4]=min(tmp(:,1));
```

```

if tmp3==tmpd
    ss(1:2, kk)=[i;tmp(tmp4,2)];
else
    tmp5=find(ss(:,tmp4)~=0);tmp6=length(tmp5);
    if dd(2,tmp4)==ss(tmp6,tmp4)
        ss(1:tmp6+1, kk)=[ss(tmp5,tmp4);tmp(tmp4,2)];
    else
        ss(1:3, kk)=[i;dd(2,tmp4);tmp(tmp4,2)];
    end
end
dd=[dd,[tmp3;tmp(tmp4,2)]];V(tmp(tmp4,3))=[];
[mdd, ndd]=size(dd);kk=kk+1;
end
S=ss;D=dd(1,:);

```

3.1.2 插入伪代码

下面是一个粒子群算法的伪代码

算法 1 粒子群算法

输入: 群体规模 N , 每个粒子的位置 x_i 和速度 v_i

输出: output result

- 1: 初始化粒子群
 - 2: **while** 不满足结束条件 **do**
 - 3: 计算每个粒子的适应度 $F_{it}(i)$
 - 4: 对每个粒子, 用它的适应度值 $F_{it}(i)$ 和个体极值 $P_{best}(i)$ 比较,
 如果 $F_{it}(i) > P_{best}(i)$, 则用 $F_{it}(i)$ 替换掉 $P_{best}(i)$;
 - 5: 对每个粒子, 用它的适应度值 $F_{it}(i)$ 和全局极值 g_{best} 比较, 如
 果 $F_{it}(i) > g_{best}$, 则使用 $F_{it}(i)$ 替换 g_{best} ;
 - 6: 更新例子的速度 v_i 和位置 x_i 。
 - 7: **return** result
-

3.2 数学定义等

在文章当中, 有些地方需要一些数学的定义、解释、证明等等, 这需要一定的格式, 这里提供了以下若干个数学当中需要的命令:

下面简单介绍一下定理、证明等环境的使用

定义 1 ($\varepsilon - \delta$ 极限定义). 如果对于 $\forall \varepsilon > 0$ (不论它多么小), $\exists \delta > 0$, 是的对于适合不等式

$$0 < |x - x_0| < \delta \quad (3.1)$$

的一切 x , 对应的函数值 $f(x)$ 均满足不等式

$$|f(x) - A| < \varepsilon \quad (3.2)$$

那么常数 A 就叫做函数 $f(x)$ 当 $x \rightarrow x_0$ 时的极限, 记作

$$\lim_{x \rightarrow x_0} f(x) = A \text{ Or } f(x) \rightarrow A \text{ When } x \rightarrow x_0 \quad (3.3)$$

定义 1 从根本上定义了极限。

除了 definition 环境, 还可以使用 theorem、lemma、corollary、assumption、conjecture、axiom、principle、problem、example、proof、solution 这些环境, 根据论文的实际需求合理使用。

定理 1 (勾股定理). 在平面上的一个直角三角形中, 两个直角边边长 (a, b) 的平方加起来等于斜边长 (c) 的平方。

由定理 1 我们知道了勾股定理的使用。

引理 1. 设素数 $p > 3$, 那么不定方程 $x^2 + 3y^2 = p$ 有解的充要条件是 $\left(\frac{-3}{p}\right)$, 即 p 是形如 $6k + 1$ 的素数。

由引理 1 我们知道了引理。所以证明如下所示

证明 1. 这是一个证明环境。

由证明 1 我们知道了证明环境的使用。

推论 1. 这是一个推论。

由推论 1 我们知道了推论环境的使用。

假设 1. 这是一个假设。

由假设 1 我们知道了假设环境的使用。

猜想 1. 这是一个猜想。

由猜想 1 我们知道了猜想环境的使用。

公理 1. 这是一个公理。

由公理 1 我们知道了公理环境的使用。

定律 1. 这是一个定律。

由定律 1 我们知道了定律环境的使用。

问题 1. 这是一个问题。

由问题 1 我们知道了问题环境的使用。

例 1. 这是一个例子。

由例 1 我们知道了例子环境的使用。

解 1. 这是一个解。

由解 1 我们知道了解环境的使用。

第四章 交叉引用、链接

4.1 交叉引用

这里的交叉引用指的是，需要再 `response.tex` 中引用 `revision.tex` 的某一段文本，可以通过以下步骤实现：

- 1 在 `revision.tex` 中，将需要引用的文本放在 `%<*tag>` 和 `%</tag>` 之间，即：

```
%<*tag>
Here is the text to be cited.
%</tag>
```

- 2 在 `response.tex` 中，利用 `\ExecuteMetaData` 命令引用文本，

即：`\ExecuteMetaData[revision]{tag}`

第五章 其他

第六章 总结与展望

附录 A 参考文献

附 录 B 附录

附录 C 攻读硕士学位期间发表的学术论文及取得的相关科研成果

附 录 D 致 谢