

分类号: \_\_\_\_\_



陕西师范大学  
SHAANXI NORMAL UNIVERSITY

## 本科生毕业论文(设计)

题目: \_\_\_\_\_ 你的题目

作者单位 \_\_\_\_\_ 陕西师范大学

作者姓名 \_\_\_\_\_ 你的名字

专业班级 \_\_\_\_\_ 物(新) 1902

作者学号 \_\_\_\_\_ 你的学号

指导教师(职称) \_\_\_\_\_ 教师名字(职称)

2023年1月

## 学位论文原创性声明

本人声明所呈交的学位论文是我在导师指导下进行的研究工作所取得的研究成果。尽我所知，除本文中已经注明引用的内容和致谢的地方外，本论文不包含其他个人或集体已经发表或撰写过的研究成果，也不包含本人或他人已申请学位或其他用途使用过的成果。对本文的研究做出重要贡献的个人和集体，均已在文中做了明确说明并表示谢意。

本学位论文若有不实或者侵犯他人权利的，本人愿意承担一切相关的法律责任。

作者签名：                    日期：          年    月    日

## 学位论文知识产权及使用授权声明书

本人在导师指导下所完成的学位论文及相关成果，知识产权归属陕西师范大学。本人完全了解陕西师范大学有关保存、使用学位论文的规定，允许本论文被查阅和借阅，学校有权保留学位论文并向国家有关部门或机构送交论文的纸质版和电子版，有权将本论文的全部或部分内内容编入有关数据库进行检索，可以采用任何复制手段保存和汇编本论文。本人保证毕业离校后，发表本论文或使用本论文成果时署名单位仍为陕西师范大学。

保密论文解密后适用本声明。

作者签名：                    日期：          年    月    日

## 摘要

主要工作是基于已有 Overleaf 模板——陕师大学位论文模板，进行小修小补，提供了一个基于 Overleaf 的陕师大理工科本科生毕业学位论文模板。为了让大家使用模板更加方便，本模板基本上对于每一条代码都作了注释，并且优化了整体的结构，上手更加容易。对于使用的宏包，也在注释里提供了一个或者多个参考的网页，供大家快速了解使用。祝大家使用愉快！如有疑问请联系我：2829259839@qq.com

**关键词：** L<sup>A</sup>T<sub>E</sub>X 模板，学位论文，陕师大

## Abstract

The main work of this article is based on the existing Overleaf template - the Shaanxi Normal University dissertation template, with some repairs, and provides an Overleaf-based dissertation template for Shaanxi Normal University's undergraduates of science and engineering. In order to make it more convenient for everyone to use the template, this template basically has annotated every piece of code, and optimizes the overall structure, making it easier to get started. For the macro package used, one or more reference web pages are also provided in the comments for everyone to quickly understand and use. I wish you a happy use! If you have any question, please contact me: 2829259839@qq.com

**Key words:** L<sup>A</sup>T<sub>E</sub>XTemplate, Thesis, Shaanxi Normal University

# 目 录

摘 要 .....	I
Abstract .....	II
<b>1 这是什么 .....</b>	<b>1</b>
1.1 文档说明 .....	1
1.1.1 准备工作 .....	1
1.1.2 怎么编译 .....	1
1.2 模板文件结构 .....	2
1.2.1 配置文件 <code>main.tex</code> .....	2
1.2.2 内容目录 <code>tex/</code> .....	2
1.2.3 参考文献目录 <code>bib/</code> .....	2
1.2.4 图片目录 <code>figure/</code> .....	2
<b>2 写作示例 .....</b>	<b>3</b>
2.1 列表环境 .....	3
2.1.1 无序列表 .....	3
2.1.2 有序列表 .....	3
2.1.3 描述列表 .....	3
2.2 数学排版 .....	4
2.2.1 公式排版 .....	4
2.2.2 定理环境 .....	4
2.3 表格 .....	5
2.4 插入图片 .....	5
2.5 插入代码 .....	6
2.6 标签和引用 .....	7
2.6.1 标签设计原则 .....	7
2.6.2 引用 .....	7
2.7 文献的添加和引用 .....	7

<b>3</b>	<b>绪论</b> .....	<b>9</b>
3.1	研究背景 .....	9
3.2	所做工作 .....	9
<b>4</b>	<b>预备知识</b> .....	<b>10</b>
	<b>总结</b> .....	<b>11</b>
	<b>参考文献</b> .....	<b>12</b>
	<b>附录</b> .....	<b>13</b>
	<b>致谢</b> .....	<b>15</b>

# 1 这是什么

这是一个陕西师范大学学位论文模板。

这份模板主要基于同学之间流传的模板以及网上其他高校的学位论文模板，结合少量个人审美。由于学校没有相应的论文规范文件，因此模板里的很多参数都是参考其他模板以及个人审美得来的。大家可以根据自己需要适当修改。

## 1.1 文档说明

### 1.1.1 准备工作

如果在 Overleaf 上编译，只需在菜单栏里设置编译器为 XeLatex 即可正常运行。

如果下载到本地使用，首先必须安装 Texlive2017 或更新的版本，编译器选择为 Xelatex。然后安装编辑器，推荐安装 TeXstudio，在使用 TeXstudio 之前，需要进行一些设置。进入到 TeXstudio 设置界面，在命令子界面将 Latexmk 设置为（注意 Latexmk 在比较下面的位置，鼠标往下滑）

```
latexmk.exe -xelatex -silent -synctex=1 %
```

然后在构建子界面将默认编译器修改为 latexmk，最后打开模板的 main.tex 然后按 F5 来构建查看。

### 1.1.2 怎么编译

Overleaf 上只需要点击编译即可。

在 TeXstudio 上，首先打开 main.tex 文件，然后左侧的文件结构会显示与之关联的 tex 文件以及参考文献。点击任何一个关联的文件都可以直接打开对应的编辑界面。最后在任意一个 tex 文件的编辑界面都可以直接按 F5 来构建查看。

一个小技巧，在第一次编译之后，新增内容后只需要按 ctrl+s 键保存，等待两秒，TeXstudio 就会自动编译，内嵌的 PDF 阅读器也会同步显示新增的内容。

## 1.2 模板文件结构

本节介绍模板的文件结构，该模板采用配置与内容分离的设计，主要包含根目录下的配置文件 `main.tex` 以及三个子目录 `bib`，`figure`，`tex`。

如果想要修改全局的配置，就去 `main.tex`；想要编辑论文的内容，就去 `tex/`目录；图片都放到 `figure/`目录；参考文献数据放在 `bib/`目录。

### 1.2.1 配置文件 `main.tex`

配置文件 `main.tex` 的作用在于定义全局配置，例如文档类型，引入宏包，页面布局等，可以理解为 `tex` 文件的导言区。

### 1.2.2 内容目录 `tex/`

`tex` 目录下共有九个文件 `tex` 文件，分别对应于封面，原创性声明页，中文摘要，英文摘要，正文页，总结页，参考文献，致谢，研究成果页。此外还有一个子目录 `chapters/`用来存放正文页导入的内容。

封面页只需要修改最开始的几个参数即可。

原创性声明页一般无需改动。

中文摘要，英文摘要，正文页，总结页，致谢需要自行编辑。

研究成果页已经给出了示例，仿造一下即可。

### 1.2.3 参考文献目录 `bib/`

目录下有一个参考文献数据库文件 `ref.bib`来存放学位论文参考的文献信息。

### 1.2.4 图片目录 `figure/`

存放需要用到的图片，本模板可以使用的格式包括 `pdf`，`jpg`，`png`，`eps`。配置文件中已经定义了图片的存放路径，所以插入图片的时候，顶层目录 `figure/`可以省略。作者建议不要省略，可读性更强一些。



## 2 写作示例

### 2.1 列表环境

#### 2.1.1 无序列表

使用环境`itemize`是一个无序列表的例子，列表的每个条目单独分段。

- 这是一个无序列表。
- 这是一个无序列表。
- 这是一个无序列表。

#### 2.1.2 有序列表

使用环境`enumerate`创建有序列表，使用方法与无序列表类似。

- 1 这是一个有序列表。
- 2 这是一个有序列表。
- 3 这是一个有序列表。

通过修改`enumerate`环境后的参数`[label={\arabic*}]`可以实现编号数字类型的切换。`\arabic`可以替换为`\roman`、`\Roman`、`\alph`、`\Alph`来表示小写罗马数字，大写罗马数字，小写字母编号，大写字母编号。

可以加入一些括号让列表更加好看，例如：`[label={\roman*}]`显示效果如下

- i) 这是一个有序列表。
- ii) 这是一个有序列表。
- iii) 这是一个有序列表。

#### 2.1.3 描述列表

使用环境`description`可创建带有主题词的列表，条目语法是`\item[主题] 内容`。示例如下：

- 主题一 详细内容
- 主题二 详细内容
- 主题三 详细内容 ...

## 2.2 数学排版

### 2.2.1 公式排版

这里有举一个长公式排版的例子，来自《Math mode》：

$$\frac{1}{2}\Delta(f_{ij}f^{ij}) = 2 \left( \sum_{i < j} \chi_{ij}(\sigma_i - \sigma_j)^2 + f^{ij}\nabla_j\nabla_i(\Delta f) + \nabla_k f_{ij}\nabla^k f^{ij} + f^{ij}f^k [2\nabla_i R_{jk} - \nabla_k R_{ij}] \right) \quad (2.1)$$

### 2.2.2 定理环境

在这个模板中，我们有如下几个环境：definition(定义)，theorem(定理)，lemma(引理)，corollary(推论)，remark(注)，example(例)。amsmath 还提供了一个 proof(证明) 的环境。

其中，定义，定理，引理按 section 连续编号，推论，注，例按 chapter 单独编号。我们举例说明它们的用法。

定义环境

**定义 2.2.1 (域).** 设  $S$  为一个非空集合，其上有“加法”（记作  $+$ ）与“乘法”（记作  $\cdot$ ）两种代数运算。若满足以下条件，则称  $(S, +, \cdot)$  构成一个域 (*field*)。

- i)  $(S, +)$  构成一个交换群。
- ii) 若记  $S^* = S - \{0\}$ ，其中  $0$  为群  $(S, +)$  中的单位元，则  $(S^*, \cdot)$  也构成一个交换群。
- iii) 乘法对加法有分配律： $a(b + c) = ab + ac$ 。

引理环境

**引理 2.2.2.** Azizov2003On 实轴上任一有界无限点集  $S$  至少有一个聚点。

推论环境

**推论 2.2.3.** 根据引理2.2.2，我们可以得到柯西收敛准则。

定理环境

**定理 2.2.4 (望远镜公式).**  $[\mathbb{Q}(a, b) : \mathbb{Q}] = [\mathbb{Q}(a, b) : \mathbb{Q}(a)] [\mathbb{Q}(a) : \mathbb{Q}]$

注环境

**注 2.1.** 每个操作都可逆。

证明环境

**证明.** Theorem 2.2.4 告诉我们, 对任意  $s \in S$ , 均有  $|Orb(s)| \cdot |Stab(s)| = |G| = p$ 。于是  $|Orb(s)|$  整除  $p$ , 这里  $p$  是一个素数。从而  $|Orb(s)|$  等于 1 或  $p$ , 也就是说, **所有轨道的大小要么为 1, 要么为  $p$** 。于是整个集合  $S$  就被划分为两部分, 一部分是大小为 1 的轨道, 另一部分是大小为  $p$  的轨道, 如图 9.4 所示。

假设大小为 1 的轨道有  $m$  个, 大小为  $p$  的轨道有  $n$  个, 则有

$$m + p \cdot n = |S| \tag{2.2}$$

注意到定义 2.2.1, **那些  $|Orb(s)| = 1$  的元素  $s$  即为稳定元**, 这就表明有  $m$  个稳定元。从上式立刻看出  $|S| \equiv m \pmod{p}$ 。 □

例子环境

**例 2.1.** 用数列的柯西准则证明确界有界。

## 2.3 表格

这一节给出的是一些表格的例子。先给出一个简单的表格样式 Table 2.1, 具体可看 tex 文件源代码。

一个更为简便的方法是利用 Texstudio 自带的表格助手, 点击菜单栏的向导, 有两种表格类型的图形化界面, 不再赘述。

表 2.1: 表格名

列 1	列 2	列 3	列 4
条目 1	条目 2	条目 3	条目 4
条目 1	条目 2	条目 3	条目 4

## 2.4 插入图片

XeLatex 可以很方便地插入 PDF、PNG、JPG 格式的图片。插入 PNG/JPG 的例子如 Figure 2.1 所示。这两个水平并列放置的图共享一个“图标题” (table caption), 没有各自的小标题。

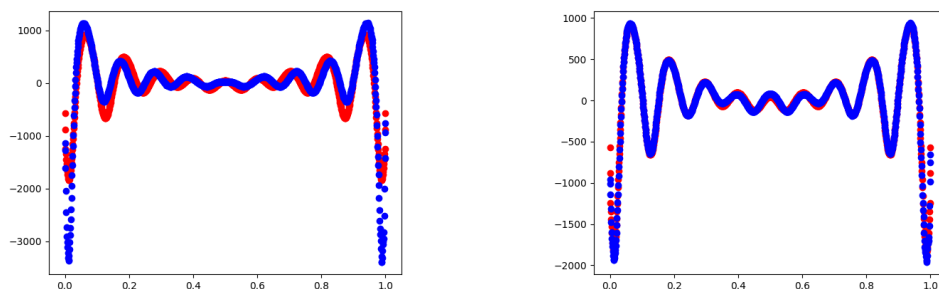
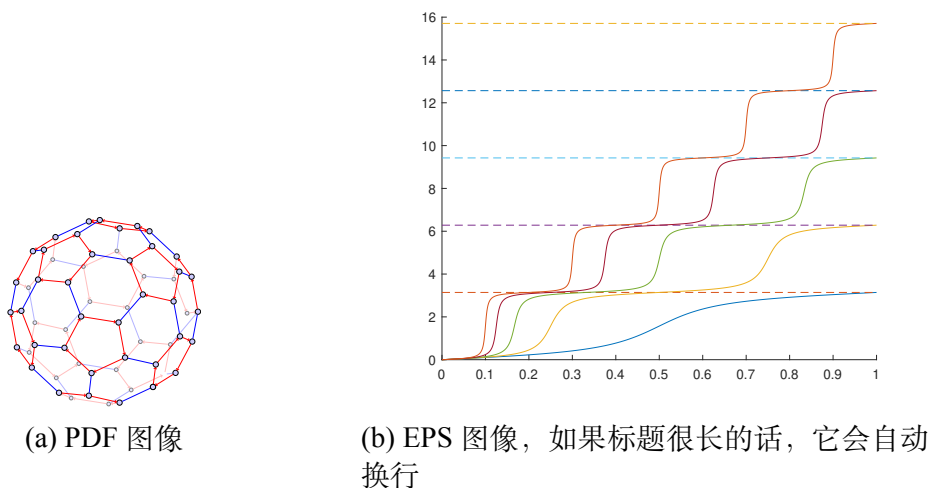


图 2.1: 中文题图  
图 2.1: English caption

这里还有插入 EPS 图像和 PDF 图像的例子，如Figure 2.2a 和Figure 2.2b。这里将 EPS 和 PDF 图片作为子图插入，每个子图有自己的小标题。子图标题使用 subcaption 宏包添加。



(a) PDF 图像

(b) EPS 图像，如果标题很长的话，它会自动换行

图 2.2: 插入 eps 和 pdf 的例子（使用 subcaptionbox 方式）  
图 2.2: An EPS and PDF demo with subcaptionbox

## 2.5 插入代码

这里给一个使用 listings 宏包插入源代码的例子:

算法 2.1: 一段 C 源代码

```
#include <stdio.h>
```

```
int main() {
    printf("Hello World!\n");
    return 0;
}
```

在算法 2.1 中，我们插入了一个 C 语言的 Hello World 代码。

## 2.6 标签和引用

这一节给出的是一些标签和引用的例子。

### 2.6.1 标签设计原则

为了更好的方便作者引用已经存在的标签，标签名必须见名知意。一般的设计原则是类型名:内容描述。在 Table 2.2 中给出通常使用的类型的简称。

例如: thm:limit, tab:typename 等等。

表 2.2: 常见类型缩写

缩写	全称	缩写	全称
part	part	fig	figure
chap	chapter	tab	table
sec	section	eq	equation
subsec	subsection	algo	algorithm
thm	theorem	def	definition
lem	lemma	rem	remark

### 2.6.2 引用

最简单的引用标签的方法就是使用 `\ref` 命令，会显示被引用的定理（或者章节）对应的编号。为了让文章可读性更强，我们通常会在 `\ref` 命令前加入类型名，如 `<定理\ref{thm:telescope}>`。

## 2.7 文献的添加和引用

本模板的参考文献数据统一存放在 bib/ 目录下的 `ref.bib` 文件中，如需添加文献，可以在 INSPIRE HEP，百度学术，Google Scholar 等界面查找到对应文献的 `bibtex` 格

式引用信息，然后附在ref.bib文件末端即可。一个比较规范的bibtex文献信息如下：

```
@article{Sjostrand:2006za,
  author = "Sjostrand, Torbjorn and Mrenna, Stephen and
    Skands, Peter Z.",
  title = "{PYTHIA 6.4 Physics and Manual}",
  eprint = "hep-ph/0603175",
  archivePrefix = "arXiv",
  reportNumber = "FERMILAB-PUB-06-052-CD-T, LU-TP-06-13",
  doi = "10.1088/1126-6708/2006/05/026",
  journal = "JHEP",
  volume = "05",
  pages = "026",
  year = "2006"
}
```

第一行表明这个文献的类型是article以及引用关键字为Sjostrand:2006za。

文献添加后，如果需要引用，使用\cite{label}命令即可，支持引用单个或多个文献。演示如下：

命令\cite{Sjostrand:2006za}的使用效果：在<sup>[1]</sup>中，作者已经阐述了有关结论。

命令\cite{bierlich2022comprehensive,王亚平2006}的使用效果：在<sup>[2-3]</sup>中，作者有...

**重要提示：**如果在引用时发现不能正常显示引用标志的话，进入模板根目录下的cmd环境，输入biber main命令，然后再次编译即可。

## 3 绪论

绪论 我是绪论

### 3.1 研究背景

这样引用<sup>[2]</sup>。如果 ref.bib 中的哪条参考文献没有在正文被引用，那么该条参考文献将不会在参考文献那一页显示出来。

### 3.2 所做工作

## 4 预备知识

我是预备知识



## 总结

我是总结

## 参考文献

- [1] SJOSTRAND T, MRENNA S, SKANDS P Z. PYTHIA 6.4 Physics and Manual[J]. JHEP, 2006, 05: 026. arXiv: hep-ph/0603175. DOI: 10.1088/1126-6708/2006/05/026.
- [2] BIERLICH C, CHAKRABORTY S, DESAI N, et al. A comprehensive guide to the physics and usage of PYTHIA 8.3[J]. SciPost Physics Codebases, 2022: 008.
- [3] 王亚平. LHC 能区 ALICE 实验及其 PHOS 触发选判[C]//中国物理学会高能物理分会第七届学术年会实验分会场. 2006.

## 附录

### 一、处理碰撞事例的 Python 代码

算法 4.1: Python 源代码

```
## 2023/1/19 version.
import IPython.core.pylabtools as pyt
import numpy as np
import matplotlib.pyplot as plt
import math
import time
#Because this module structure is too long, use "as" to
    rename it for conveniently invoking.

T1 = time.time()
#read data from 'main_out.txt' into arrays.
data =np.loadtxt("Jpsi_13TeV_1000.txt",dtype='str',delimiter=
    '\t')
#or delimiter='\s+'

#Clear out the useless information and split the strings.
data=np.delete(data,len(data)-1,0)
data=[item.split() for item in data]
JpsiData=[item for item in data if (item[0] == '443' or item
    [0] == '-443')]
#extract physical quantities from data(its type is "list")
KF=[JpsiData[i][0] for i in range(len(JpsiData))]
KF = np.array(KF,dtype=float)
px=[JpsiData[i][1] for i in range(len(JpsiData))]
px = np.array(px,dtype=float)
```

```

py=[JpsiData[i][2] for i in range(len(JpsiData))]
py = np.array(py,dtype=float)
pz=[JpsiData[i][3] for i in range(len(JpsiData))]
pz = np.array(pz,dtype=float)
E=[JpsiData[i][4] for i in range(len(JpsiData))]
E = np.array(E,dtype=float)
m=[JpsiData[i][5] for i in range(len(JpsiData))]
m = np.array(m,dtype=float)
floatdata=[KF ,px ,py ,pz ,E ,m]
JpsiData=np.transpose(floatdata)
print(JpsiData)
f=open('JpsiData_13TeV.txt','a+')
for index in range(len(JpsiData)):
    print(*JpsiData[index,:],file=f)
f.close()

T11 = time.time()
print('数据处理时间:%s mins' % ((T11 - T1)*1.0/60))

```

## 致谢

我是致谢