

届 别 2024 届  
学 号 20201405xxxx



湘南学院

# 毕业论文（设计）

基于  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  的两月半研究  
—以 xxx 为例

姓 名 xxx

学 院、专 业 数学与信息科学学院

计算机专业

导师姓名、职称 指导老师 副导师

完 成 时 间 2024 年 5 月

# 目 录

摘要 .....	I
Abstract .....	II
1 引言 .....	1
1.1 为什么学 $\text{\LaTeX}$ 2 $\epsilon$ .....	1
1.2 $\text{\LaTeX}$ 如何安装 .....	1
1.3 $\text{\LaTeX}$ 模板使用 .....	1
2 $\text{\LaTeX}$ 快速入门 .....	1
3 宏包与字体 .....	2
3.1 宏包的使用 .....	2
3.2 浅谈字体 .....	2
4 浮动体 .....	3
4.1 图片 .....	3
4.2 表格 .....	3
4.3 算法 .....	3
4.4 代码 .....	4
4.5 其他浮动体 .....	4
5 数学 .....	5
5.1 数学符号 .....	5
5.1.1 分数 .....	5
5.1.2 积分符号 .....	5
5.1.3 对齐点 .....	5
5.2 证明、定理和公理 .....	6
6 TikZ .....	7
7 文献引用与 $\text{\LaTeX}$ 模板 .....	8
7.1 文献 .....	8
7.2 模板 .....	8
8 总结 .....	10
致谢 .....	12

附录 A 球体 ..... 13

附录 B 说明书..... 14

附录 C 流程图..... 15

## 摘 要

本文是仿湘南学院理科论文设计（2018 修订版）要求制作的简易版  $\text{\LaTeX}$  模板生成的。此模板属于个人作品，非官方模板，大多数格式遵循撰写规范，部分小格式手册未曾提及，模板参考了其他高校的格式，常见的如：证明，定理，推论等环境；代码片段的格式；附录等等。作者学  $\text{\LaTeX}$  时间不长，第一次用  $\text{\LaTeX}$  写模板，因此本模板在使用上仍然会存在问题，还请见谅，也希望有更多热爱  $\text{\LaTeX}$  的同学一起构建更好的版本。

**关键词：** $\text{\LaTeX}$ ；明德；博学；创新；笃行；毕业论文；第七个关键词

## ABSTRACT

Xiangnan University ('Xiangnan College'; XNU) is a provincial public college in Suxian, Chenzhou, Hunan, China. Despite its English name, the institute has not been granted university status. The college is under the Hunan Provincial Department of Education.

**Keywords:** XNU; master thesis; XeTeX/LaTeX template

# 1 引言

## 1.1 为什么学 $\text{\LaTeX}$ 2 $\epsilon$

虽然 word 处理文字方便，但想要获得一个准确的样式却很难，特别是在公式与图片较多的时候会存在很多问题，而  $\text{\LaTeX}$  排版数学公式非常高效，你一定没见过

$$\int_a^b \frac{d}{dx} \left( \frac{\sqrt{x^4 - 3x^2 + 2x + 1}}{\ln(x) - \sin(x)} \right) dx = \sum_{n=1}^{\infty} \frac{1}{n^2 + nx} \Bigg|_a^b + \prod_{i=1}^n i^2 - \left( \frac{\cos(x) + \sin(x)}{e^{x^2} - 1} \right)^{\frac{3}{2}} \quad (*)$$

这行公式仅需 3 行代码就搞定了。现在就带你一起来看看吧！

## 1.2 $\text{\LaTeX}$ 如何安装

MiKTeX 和 Texlive 都是主流软件，网上都有教程这里就不细说了。编辑体验比较好的有 **OverLeaf** 和 VSCode。OverLeaf 浏览器打开即用，本模板就由该网站提供的工具所作，VSCode 需要配置环境，它的主要在输入提示上非常友好，不过这一点似乎 TexLive 支持的也挺好，VSCode 适合有编程基础的同学。

## 1.3 $\text{\LaTeX}$ 模板使用

想要写出一篇完整论文，除了 .cls 文件配置的格式文件，写作的过程的其他宏包仍然需要手动添加。本文就是基于 xnuthesis.cls 编写的，不一定适合所有人，模板提供了一些基础格式，如果需要更精细的格式还需自行重构。

# 2 $\text{\LaTeX}$ 快速入门

作者也不是什么 LaTeX 专家，只能说是入门比较快的新手，浅浅分享一波经验！新手当然是推荐看视频了，如果你是码农，那么直接上文档！

1. 《一份不太简短的  $\text{\LaTeX}$  2 $\epsilon$  介绍》这本书做的不错，翻阅了无数遍，甚至写这篇文章还在用，入门前觉得像词典，入门后感觉是比较简洁的，放下它的 github 仓库地址: <https://github.com/CTeX-org/lshort-zh-cn>。
2. AI: 国内的通义千问，国外有许多免费 ChatGPT-4 站点可用，个人一直在用的 <https://www.coze.com> 挺不错的，作为新手几乎有一半的问题是靠它解决的。唯一不足之出就是大模型写代码还是一些存在 bug 的。
3. 各类开源网站论坛

- **LaTeX 工作室**: 该工作室在 b 站上的视频也值得去看。
  - **Tex.StackexChange**: 一个专注解决 Tex 问题的“stackoverflow”。
  - **CTAN**: 开源宏包海洋，入海，做一个 L<sup>A</sup>T<sub>E</sub>X 极客。
4. 高校毕业论文模板，如<https://github.com/sjtug/SJTUThesis>，是上海交通大学的论文模板，这个项目用了 L<sup>A</sup>T<sub>E</sub>X3 语法，是一个非常新的项目。入门的话更推荐北航，天津大学等的模板，它们的类文件语法偏 Tex。

## 3 宏包与字体

### 3.1 宏包的使用

L<sup>A</sup>T<sub>E</sub>X 宏包众多，能尽量用新宏包就用新的，优先用 L<sup>A</sup>T<sub>E</sub>X3 语法重构的，简要谈谈这几个宏包吧。

1. **newtxmath** 这个宏包用于修改数学字体，与宏包 **amsthm** 发生冲突\openbox重定义。
2. **tocloft**与**titletoc**如果同时使用在定义目录样式的时候会出现修改无效的问题。同样**titlesec**与\ctexset修改标题也会冲突，因为他们本身属于同一类宏包，修改相同的东西。

### 3.2 浅谈字体

L<sup>A</sup>T<sub>E</sub>X 字体与 word 不同，默认使用的是 Fandol 系列字体，大多时候不建议加载一些奇怪的字体，比较麻烦，用 ctex 宏集默认字体即可，ctex 基本的中文字体都比较全面。

英文	<b>font</b>	<i>font</i>	font
中文	<b>粗体</b>	斜体	<b>无衬</b>
宋体	<b>黑体</b>	仿宋	<b>楷书</b>

可以看到 Fandol 系列中文的黑体，就是无衬线字体 sffamily，本文使用的 \heiti 与 \sffamily 效果一样，不同的是 \heiti 只对中文有效，而 \sffamily 是中英文都可使用。

## 4 浮动体

### 4.1 图片

注意：分辨率较高的图片会增加编译时长。图片格式一般最好用 pdf,eps 等矢量图，放大不会失真,pdf 图片兼容性好，图1。



图 1 网络素材

### 4.2 表格

长表格一般比较比较难输入，推荐使用工具，如<https://tablesgenerator.com/>, 实在不行用 overleaf 自带的也行，长表格如果实在需要断页，一般来说需要标注续表。

### 4.3 算法

算法目前推荐使用 `algorithm2e` 宏包，代码环境一般用 `listings` 宏包并使用 `\lstset` 设置风格。



表 1 三线表格

左对齐	居中	右对齐
$a_1$	$\int$	$\aleph$
$a_1, b_2$	$\Sigma \Sigma$	$\blacktriangleright \blacktriangleleft$
$a_1, b_2, c_3$	$\Pi \Pi \Pi$	$\star \textcircled{\text{S}} \circ$

Algorithm 1: 示例算法

**Data:** 输入数据  
**Result:** 输出结果

1 **Function** Main():  
    **Data:** 这里可以描述函数的输入  
    **Result:** 这里可以描述函数的输出  
2     Sum( $a, b$ )  
3     **return**  $a + b$

4 **Function** Sum( $a, b$ ):  
    **Data:** 这里可以描述函数的输入  
    **Result:** 这里可以描述函数的输出  
5      $sum \leftarrow a + b$   
6     **return**  $sum$

4.4 代码

```
1 %双曲抛物面
2 clear;
3 x = linspace(-10, 10, 100);
4 y = linspace(-10, 10, 100);
5 [X, Y] = meshgrid(x, y);
6 Z = (X.^2)/3 - (Y.^2)/5;
7 surf(X, Y, Z);
```

代码 1 MATALB

4.5 其他浮动体

据说支持 JavaScript 的 PDF 阅读器，通过media9宏包，能实现视频或动画的播放。当然利用 TikZ(在本文第 5 节介绍) 一帧一帧的放也能实现动画效果，与 beamer 播放类似。

## 5 数学

L<sup>A</sup>T<sub>E</sub>X 公式符号系统比较完整，不会基本可查，作者虽是数学系但所选数学论文中公式使用的也较少，但还是总结了一些值得注意的坑。

### 5.1 数学符号

#### 5.1.1 分数

使用行内公式会显得小，而使用`\dfrac`会感觉太挤了，由于行间距一般不能改变，所以要么调行间距，要么使用行间公式，看下面示例

有志登山  
顶  $\frac{3}{9}$ ，无  
志站  $\frac{0}{9}$ 。

一般情况分数上下太宽建议直接放在行间。

#### 5.1.2 积分符号

一些情况下我们可能需要直立体，而不是斜体。

牛顿-莱布尼茨公式：

$$\int_a^b f(x) dx = F(b) - F(a), \forall x \in [a, b], F'(x) = f(x)$$

高斯定理（散度定理）可以表示为：

$$\oiint_{\partial V} \mathbf{F} \cdot d\mathbf{S} = \iiint_V \nabla \cdot \mathbf{F} dV$$

其中，

- $\oiint_{\partial V} \mathbf{F} \cdot d\mathbf{S}$  表示向量场  $\mathbf{F}$  通过闭合表面  $\partial V$  的向外通量，
- $\iiint_V \nabla \cdot \mathbf{F} dV$  表示向量场  $\mathbf{F}$  在体积  $V$  内的散度的积分。

#### 5.1.3 对齐点

这里 `align` 与 `aligned` 完全是两个环境，`aligned` 不是一个公式环境，

`align`:

$$a = b + c \tag{5.1}$$

$$= d + e \tag{5.2}$$

aligned:

$$\begin{aligned}a &= b + c \\d &= e + f + g \\h + i &= j + k \\l + m &= n\end{aligned}\tag{5.3}$$

对齐需要对齐点，一般在 `&` 处对齐。

## 5.2 证明、定理和公理

本模板序号都用 `section` 编号，如果想使用单个序号 1, 2, 3 等，当然也可以通过类文件自定义。

**推论 5.1:** 生活可能不像你想象的那么好，但是也不会像你想象的那么糟，人的脆弱和坚强都超乎了自己的想象，有时候脆弱的一句话会让你泪流满面，有时候你发现自己咬着牙已经走过了很长的路。

**定理 5.1 (三角形的内积和):** 两直角的平方差一定小于正弦的  $30^\circ$  的一半。

**练习 5.1:** 子曰：打架用砖乎！不亦乱乎！ — 《论语》

**引理 5.1:** 可导函数的每一个可导的极值点都是驻点（函数的导数在该点为零）。

**证明:** 利用

$$i + j = m \vec{a}$$

，可以得到

$$\lim_{x \rightarrow y} = C + + \quad \square$$

**例 5.1 (新高考 II):** 想象有  $n$  个有序排列的箱子，其中每个箱子可以放一个球或者不放球。令二项式系数 (或称为组合数)  $m$  个球的情况种数。

**解:** 利用组合数，

$$C_n^m = \binom{n}{m} = \frac{n!}{m!(n-m)!}$$

我们可以归纳得出...

## 6 TikZ

与 TikZ 有关的 L<sup>A</sup>T<sub>E</sub>X 中宏包非常丰富，常规的流程图（附录 C 图6），函数图，电路图等等都有对应的宏包支持，它的语法也不同于 Tex。入门 TikZ 建议看视频，这一块不建议上来就磕文档，官方文档有上千页。当然其绘图功能也是十分强大！

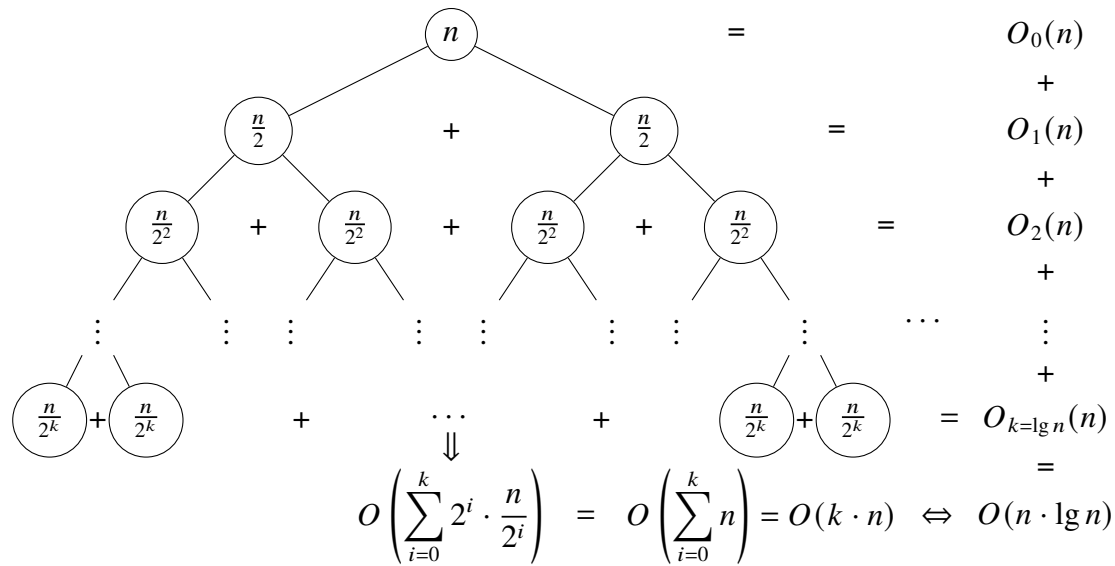


图 2 二叉树

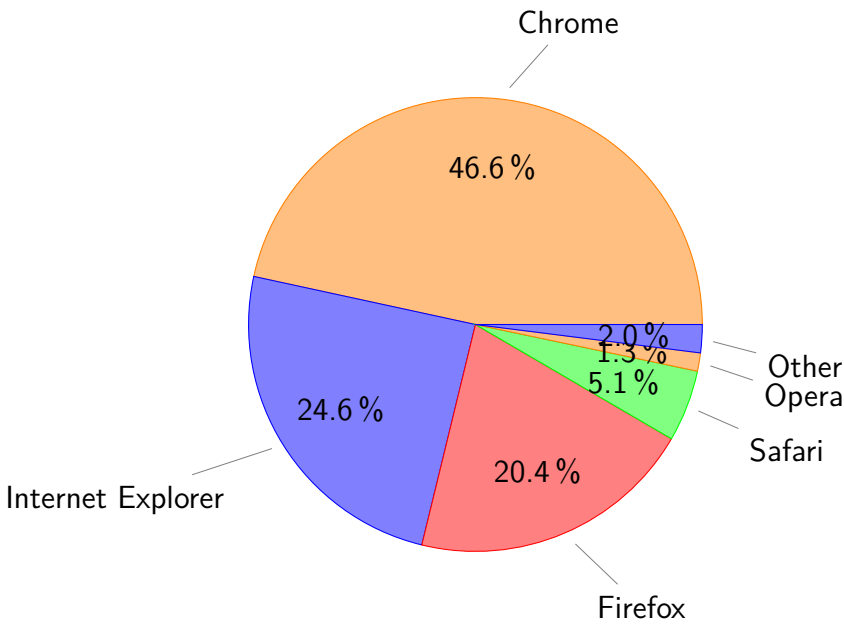


图 3 扇形图

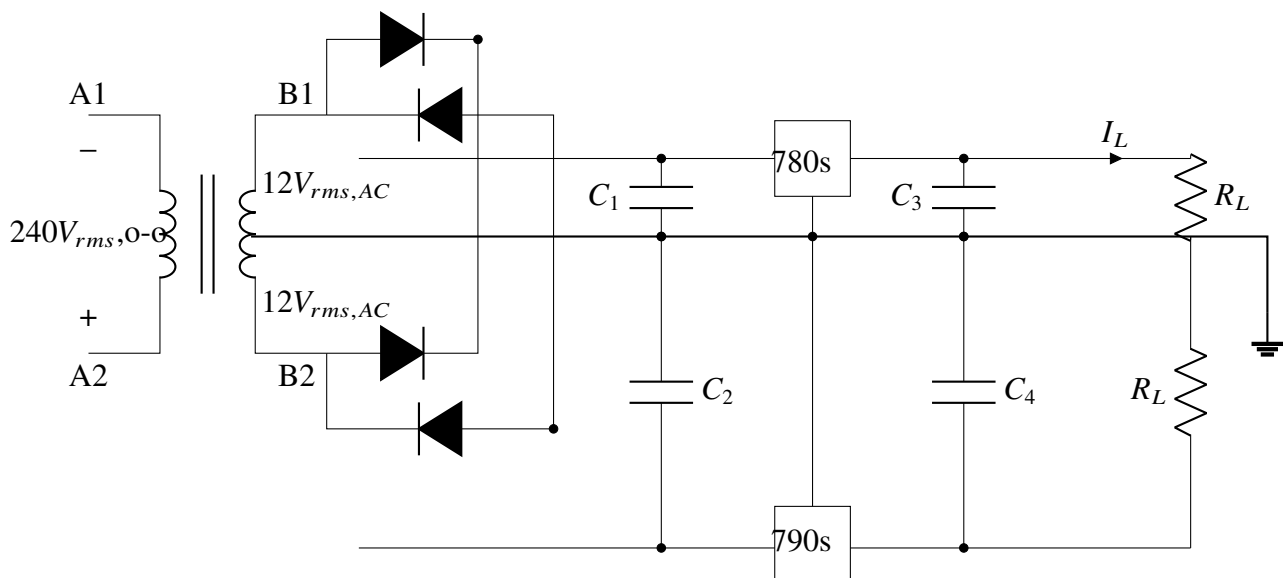


图 4 电路图

上面的图形都是 TikZ 代码绘制的，TikZ 功能远不止此，由于国内视频缺乏完整教程，所以学习起来会有些吃力，作者也只是触碰到它的冰山一角，图2 和图3来自 <https://texample.net/tikz>，图4转载自 [latexstudio](https://www.latexstudio.net)。

## 7 文献引用与 L<sup>A</sup>T<sub>E</sub>X 模板

### 7.1 文献

本文参考文献的格式为 GB/T7714-2015, 使用 bibtex 进行管理, `\cite{}`对文献引用:

单文件引用<sup>[1]</sup>, 多个文献引用<sup>[2-13]</sup>。

### 7.2 模板

xnuthesis 论文结构大致如下, 更多使用方法可以参考本文的源码, 如需学习撰写类文件可以看<https://github.com/rockyzzh/latexdoc-chinese-translation>提供的文档, 有许多翻译后的中文文档可以参考。

L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>对类文件开发维护困难, 主要体现在许多命名规范上的问题, 于是就产生了 L<sup>A</sup>T<sub>E</sub>X 3, L<sup>A</sup>T<sub>E</sub>X 3 语法就是由 expl3 宏包提供的, <https://texdoc.org/serve/interface3/0>是 L<sup>A</sup>T<sub>E</sub>X 3 语法的完整手册。L<sup>A</sup>T<sub>E</sub>X 3 更像是一门面向开发者的编程语言, 有类型并且支持函数式编程, 命名规范, 缺点是目前还未完全普及, 许多宏包还没支持新的 l3 语法, 但这是未来的发展趋势。为了以后兼容性, 本模板在类文件中部分也使用了 L<sup>A</sup>T<sub>E</sub>X 3 语法。

```

\documentclass{xnuthesis}
%=====引言区
\input{setup} %配置文件，自由加载宏包与论文的信息

\begin{document}
%=====封面
\input{cover/cover} %如果有问题，封面可以用word转PDF再进行插入

%=====目录
\tableContents
\frontmatter

%=====摘要
\input{contents/abstract}
\mainmatter

%=====正文
\input{contents/intro}
\input{contents/sect_pkg_font}
\input{contents/sect_float}
\input{contents/sect_math}
\input{contents/sect_tikz}
\input{contents/sect_cite_template}

%=====参考文献
\printreferences

%=====致谢
\input{contents/acknowledgements}

%=====附录
\appendix
\input{contents/app_tikzpic}
\input{contents/app_interpreter}
\input{contents/app_process}
\end{document}

```

代码 2 LaTeX 论文结构

## 8 总结

本模板的许多结构方面参考了北航和上交的模板, 作者全程依赖 GPT 解决了许多疑问, 不得不说的是人工智能时代, 信息获取方式越来越快, AI 已经能实现模块化的基本需求, 学习和写出  $\text{\LaTeX}$  程序变得更加轻松了, Typst 与 MathML 发展日益强大, 也希望某一天能出现更优秀简洁排版方式吧!

## 参考文献

- [1] 程根伟. 1998 年长江洪水的成因与减灾对策[M]//许厚泽, 赵其国. 长江流域洪涝灾害与科技对策. 北京: 科学出版社, 1999: 32-36.
- [2] 中国力学学会. 第 3 届全国实验流体力学学术会议论文集[C]. 天津: 天津大学出版社, 1990: 20-24.
- [3] 国家标准局信息分类编码研究所. GB/T 2659-1986 世界各国和地区名称代码[S]//全国文献工作标准化技术委员会. 文献工作国家标准汇编: 3. 北京: 中国标准出版社, 1988: 59-92.
- [4] 河北绿洲生态环境科技有限公司. 一种荒漠化地区生态植被综合培育种植方法: 中国, 01129210.5[P/OL]. 2001-10-24 [2002-05-28]. <http://211.152.9.47/sipoasp/zlijs/hyjs-yx-new.asp?recid=01129210.5&leixin>.
- [5] Hopkinson A. UNIMARC and metadata: Dublin core[EB/OL]. [1999-12-08]. <http://www.ifla.org/IV/ifla64/138-161e.htm>.
- [6] 姜锡洲. 一种温热外敷药制备方案: 中国, 88105607.3[P]. 1989-07-26.
- [7] 蒋有绪, 郭泉水, 马娟, 等. 中国森林群落分类及其群落学特征[M]. 北京: 科学出版社, 1998: 11-12.
- [8] 李炳穆. 理想的图书管理员和信息专家的素养与形象[J]. 图书情报工作, 2000(2): 5-8.
- [9] 李晓东, 张庆红, 叶瑾琳. 气候学研究的若干理论问题[J]. 北京大学学报: 自然科学版, 1999, 35(1): 101-106.
- [10] 中国图书馆学会. 图书馆学通讯[J]. 1957(1)-1990(4). 北京: 北京图书馆, 1957-1990.
- [11] World Health Organization. Factors regulating the immune response: report of WHO Scientific Group[R]. Geneva: WHO, 1970.
- [12] 余敏. 出版集团研究[M]. 北京: 中国书籍出版社, 2001: 179-193.
- [13] 张志祥. 间断动力系统的随机扰动及其在守恒律方程中的应用[D]. 北京: 北京大学数学学院, 1998: 50-55.



## 致 谢

感谢湘南学院提供论文格式！感谢室友及指导老师给出的意见！

## 附录 A 球体

这是一个 TikZ 绘制的 3D 图形，也是 [example.net/tikz](https://example.net/tikz) 上面的例子。

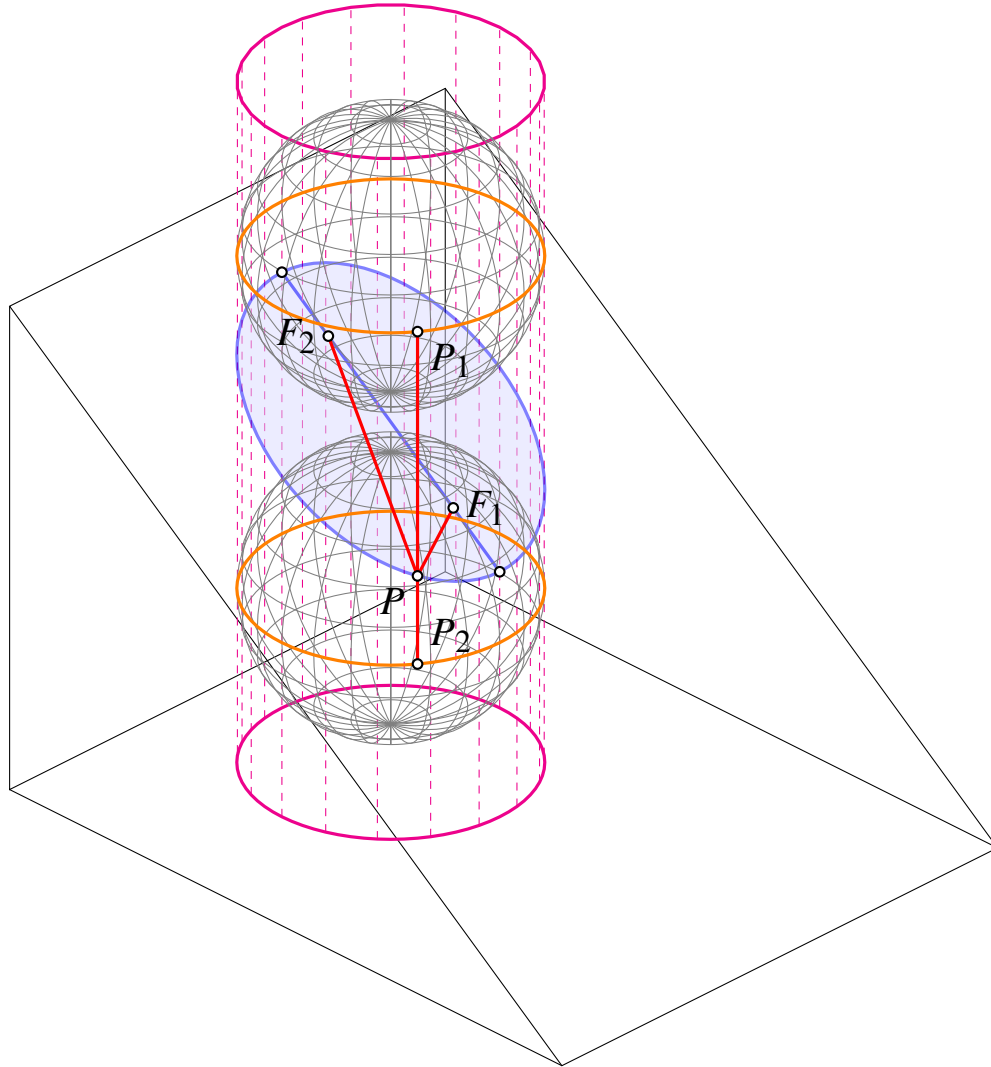


图 5 dandelin-spheres

## 附录 B 说明书

C++ 是一种被广泛使用的计算机程序设计语言。它是一种通用程序设计语言，支持多重编程范式，例如过程化程序设计、面向对象程序设计、泛型程序设计和函数式程序设计等。

比雅尼·斯特劳斯特鲁普博士在贝尔实验室工作期间在 20 世纪 80 年代发明并实现了 C++。起初，这种语言被称作“C with Classes”（“包含‘类’的 C 语言”），作为 C 语言的增强版出现。随后，C++ 不断增加新特性。虚函数、运算符重载、多继承、标准模板库、异常处理、运行时类型信息、命名空间等概念逐渐纳入标准草案。1998 年，国际标准组织颁布了 C++ 程序设计语言的第一个国际标准 ISO/IEC 14882:1998，目前最新标准为 ISO/IEC 14882:2020。ISO/IEC 14882 通称 ISO C++。ISO C++ 包含了主要包含了核心语言和标准库的规则。尽管从核心语言到标准库都有显著不同，ISO C++ 直接正式 (normative) 引用了 ISO/IEC 9899 (通称 ISO C)，且 ISO C++ 标准库的一部分和 ISO C 的标准库的 API 完全相同，另有很小一部分和 C 标准库略有差异（例如，`strcat` 等函数提供对 `const` 类型的重载）。这使得 C 和 C++ 的标准库实现常常被一并提供，在核心语言规则很大一部分兼容的情况下，进一步确保用户通常较容易把符合 ISO C 的源程序不经修改或经极少修改直接作为 C++ 源程序使用，也是 C++ 语言继 C 语言之后流行的一个重要原因。

作为广泛被使用的工业语言，C++ 存在多个流行的成熟实现：GCC、基于 LLVM 的 Clang 以及 Visual C++ 等。这些实现同时也是成熟的 C 语言实现，但对 C 语言的支持程度不一（例如，VC++ 对 ANSI C89 之后的标准支持较不完善）。大多数流行的实现包含了编译器和 C++ 部分标准库的实现。编译器直接提供核心语言规则的实现，而库提供 ISO C++ 标准库的实现。这些实现中，库可能同时包含和 ISO C 标准库的共享实现（如 VC++ 的 `msvcrt`）；而另一些实现的 ISO C 标准库则是单独于编译器项目之外提供的，如 `glibc` 和 `musl`。C++ 标准库的实现也可能支持多种编译器，如 GCC 的 `libstdc++` 库支持 GCC 的 `g++` 和 LLVM Clang 的 `clang++`。这些不同的丰富组合使市面上的 C++ 环境具有许多细节上的实现差异，因而遵循 ISO C++ 这样的权威标准对维持可移植性显得更加重要。现今讨论的 C++ 语言，除非另行指明，通常均指 ISO C++ 规则定义的 C++ 语言（虽然因为实现的差异，可能不一定是最新的正式版本）。

## 附录 C 流程图

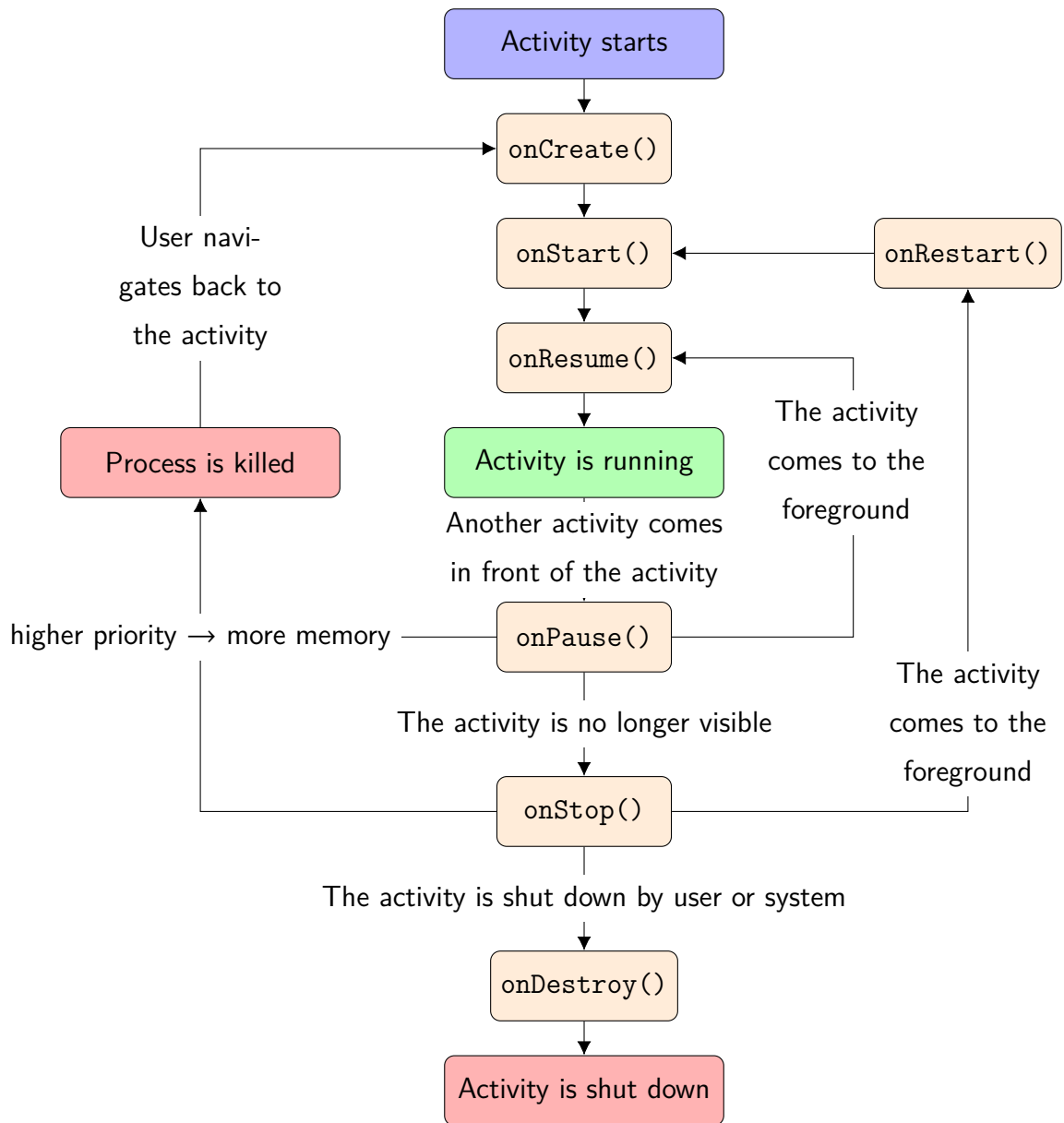


图 6 android