



南京大学

本科毕业论文

院 系 _____ 物理学院

专 业 _____ 物理基地

题 目 _____ 南京大学本科毕业设计 L^AT_EX 模板

年 级 _____ 09 学 号 _____ 091120000

学生姓名 _____ 曹增乐

指导老师 _____ 老板 职 称 _____ 教授

提交日期 _____ 2013 年 2 月 6 日

南京大学本科生毕业论文（设计）中文摘要

毕业论文题目： 南京大学本科毕业设计 L^AT_EX 模板

物理学院 院系 物理基地 专业 09 级本科生姓名： 曹增乐

指导教师（姓名、职称）： 老板 教授

摘要：

这是中文摘要。

关键词： 南京大学；本科毕设；L^AT_EX 模板

南京大学本科生毕业论文（设计）英文摘要

THESIS: L^AT_EX Template for Bachelor's Thesis of Nanjing University

DEPARTMENT: School of Physics

SPECIALIZATION: Physics

UNDERGRADUATE: Zengle Cao

MENTOR: Prof. Boss

ABSTRACT:

This is English abstract.

KEY WORDS: Nanjing University (NJU); Bachelor's Thesis; L^AT_EX Template

目 录

第 1 章	\LaTeX 之我见	1
1.1	工具与艺术	1
1.2	软件与语言	2
1.3	\LaTeX 与 Word	3
第 2 章	\LaTeX 基础	6
2.1	字符	6
2.2	命令	6
2.3	文档的基本结构	7
2.4	标题与文档逻辑结构	8
2.5	公式	10

第 1 章 L^AT_EX 之我见

1.1 工具与艺术

之前在人人上看到一篇介绍 L^AT_EX 的文章，第一句话说的便是

L^AT_EX 并不难，L^AT_EX 也不是艺术。

对此，我并不完全赞同。诚然，对于一部分，甚至是很大一部分人来说 L^AT_EX 的确只是一个写文章的工具，但其底层 T_EX 设计的初衷绝非仅限于此。

T_EX 之父，正是有编程圣经之称的 *The Art of Computer Programming* 的作者，大名鼎鼎的 *Donald E. Knuth* 教授。他在为 T_EX 系统撰写的手册 *The TeXBook* 第一章便写道

English words like ‘technology’ stem from a Greek root beginning with the letters $\tau\epsilon\chi$...; and this same Greek word means *art* as well as technology. Hence the name T_EX, which is an uppercase form of $\tau\epsilon\chi$.

.....

The purpose of this pronunciation exercise is to remind you that T_EX is primarily concerned with high-quality technical manuscripts: Its emphasis is on art and technology, as in the underlying Greek word. If you merely want to produce a passably good document—something acceptable and basically readable but not really beautiful—a simpler system will usually suffice. With T_EX the goal is to produce the *finest* quality; this requires more attention to detail, but you will not find it much harder to go the extra distance, and you’ll be able to take special pride in the finished product.

足见 T_EX/L^AT_EX 并不仅仅是工具而已，它的诞生就是为了满足高质量排版的各种苛刻要求。

又或者，L^AT_EX 本身是一种工具，而排版是艺术。就好比一支笔。很多人用笔写字，只是为了记录和表达，于他们而言，笔仅仅是工具。但也有人，成了书法家，书法是一种艺术。要写得好字，功夫自是必不可少的，一支好笔也同样很重要。L^AT_EX 对于排版艺术，就是一支好笔。

1.2 软件与语言

就好比 C++ 是一种编程语言，而微软的 VC++ 是开发环境，它只是为了提高效率而已。没有 VC++，还可以选择 devC++，或者使用任何文本编辑器配合编译器。语言的核心，只是代码和编译器而已。

同样的，虽然很多人把 L^AT_EX 视为软件，私以为以排版语言理解之更为合适。其实只需一个编辑器编辑源文件，在命令行里编译，就可以得到高质量的排版结果。而所说的软件，如 C_T_EX 套装里自带的 WinEdt，以及开源的 TeXworks，TeXstudio 等，就好比 VC++ 或者 devC++。所以说，L^AT_EX 其实是一种编译器，或者语言。正如 C++ 编译器用来编译 C++ 语言，当谈论 L^AT_EX，我们既指编译器，也指语言。

当然，令人惊奇的是，当我们谈论 L^AT_EX，很多时候我们用的却并非在用 L^AT_EX 进行编译。首先，我们应该先搞清楚 TeX，pdfTeX，XeTeX 以及 LaTeX，pdfLaTeX，XeLaTeX。当然，继续了解，还有 ConTeXt，LuaTeX 等。比如 TeX，pdfTeX，XeTeX，是 TeX 的三种编译器，它们共同编译满足 TeX 语法的源文件。不同的是 TeX 输出的是 .dvi 格式的文件，需要经过额外的转换才能得到 pdf 文件。要知道，TeX 的诞生可比 pdf 的流行要早得多。而 pdfTeX，顾名思义，可以直接输出 pdf 文件。TeX 系统原本并不支持调用操作系统字体，它调用的字体都是 TeX 系统自带的文件定义的，而标准的 LaTeX 发行版，因为版权原因并没有包含 windows 上常用中易字体的定义文件。另一方面，原先的 TeX 系统只具备处理标准 ASCII 码的能力，对于中文这样字符数量庞大的语言，就更加无法提供了。虽然 TeX 系统诞生之初并不支持非英文语言，但在世界各地用户的努力下，TeX 系统也开始支持越来越多的语言。至于 XeTeX，则可以调用系统字体，并且使用通用性更好的 UTF-8 编码，让非英文排版变得简单。再说说我们的主角 LaTeX，原本只是基于 TeX 的一个拓展。它的语法与 TeX 并不兼容。虽然 TeX 文档和 LaTeX 文档的后缀都是 .tex，但既不能用 TeX 编译器编译 LaTeX 源文件，也不能用 LaTeX 编译器编译 TeX 源文件。LaTeX 和 ConTeXt，LuaTeX，都是 TeX 的拓展。LaTeX 是其中最流行的拓展，另外两个不甚了解，就不作介绍了。这里已经不难理解 pdfLaTeX 和 XeLaTeX，分别是 pdfTeX 和 XeTeX 向上兼容 LaTeX 拓展的产物了。

当然 L^AT_EX 现在已经是一个很笼统的称呼了。一个完整的 L^AT_EX 发行套装，包含上述提到的全部编译器，甚至还提供了 `metapost` 和 `Asymptote` 这样强大的绘图语言。另外 windows 版本还会自带编辑器。而我们最常用的还是 L^AT_EX 语言和他相应编译器。由于以前用 Linux 时 LaTeX 和 pdfLaTeX 还不能处理中文，我自己对 XeLaTeX 情有独钟，以至于这份模板仅支持 XeLaTeX 一种编译途径。

1.3 L^AT_EX 与 Word

L^AT_EX 与 Word 的差别，其实就是一个开源的可扩展的排版语言，和一个封闭的商业软件的差别。

首先，作为一种排版语言，L^AT_EX 体现出了惊人的可扩展性。只要有人为它开发宏包，它就能拥有很多令人吃惊的功能。除了自带的优秀的公式排版，它还可以借助相应的宏包排版化学结构式、流程图、交换图，甚至各种棋谱和乐谱。使用 Word，你需要另外购买的功能，L^AT_EX 很多都可以通过宏包实现拓展。

同时，作为排版语言，L^AT_EX 让你可以通过命令对排版效果进行细致的控制。就比如上边这段话，我们可以试着在“甚至各种棋谱和乐谱。”后加上 `\linebreak` 命令来建议 L^AT_EX 断行。得到的效果如下边这段话：

首先，作为一种排版语言，L^AT_EX 体现出了惊人的可扩展性。只要有人为它开发宏包，它就能拥有很多令人吃惊的功能。除了自带的优秀的公式排版，它还可以借助相应的宏包排版化学结构式、流程图、交换图，甚至各种棋谱和乐谱。使用 Word，你需要另外购买的功能，L^AT_EX 很多都可以通过宏包实现拓展。

在 Word 里没有这样的自由吧，是不是很赞！再看看这个，如果我要输入公式

$$-x < y < z$$

$$p < y < z$$

这样的效果似乎还不够，在 p 前加入 `` 占据一个负号的位置呢？

$$-x < y < z$$

$$p < y < z$$

我想这就是工具与艺术的区别！

L^AT_EX 另外个巨大的优势就在于用 `\chapter{}`, `\section{}` 这样的命令来整理文章的逻辑结构, 通过预先定义章节标题的样式来实现各种效果。试想一下 Word 里, 你打了“1. 标题”, 然后回车准备输入正文, 这时候出现了自动编号。可是下一节写什么你还没有想好啊, 于是退格删除, 把格式改回正文的格式。打到下一节, 输入“2. 标题”, 再用格式刷把第一节的标题刷到第二节。为了让标题与上一段分开一点, 你在上一段末尾多敲了一个回车。有时候还会遇到更尴尬的情况, 这个标题正好在一页的末尾, 这一节就这样被“斩首”了, 不太好看, 于是你回到上一节, 啪啪啪一路回车愣是把这一节的标题换到下一页。接着写, 你突然想起来前面要再插入一节。如果你不太清楚大纲视图和自动编号, 那惨了, 你得把后面的节前面的数字全部改了。别忘了还有你之前换页敲的一溜回车, 重新删了吧……

好吧, 我又在意淫 Word 用户的困扰了。其实我并不是要黑 Word, 只是拿全靠回车、空格和格式刷排版的 Word 用户举个例子。习惯了 L^AT_EX 的 `\chapter{}`, `\section{}` 这一套, 回到 Word, 便会自觉地开始学习用样式来规定标题的格式和编号, 标题设置成与下段同页就避免了上述的尴尬, 孤行控制还可以防止某段的最后一行落到下一页, 诸如此类。当习惯了 L^AT_EX 通过 `\label{}` 和 `\ref{}` 统一地实现交叉引用, 回到 Word, 就会学习使用题注来添加图表标题。当然, 你最终会发现, 这些没有 L^AT_EX 里的命令来得好用。

这便是 L^AT_EX 所谓“所想既所得”与 Word 之“所见即所得”的差别。L^AT_EX 让你专注于写文章, 而非文章的排版。我想很多人之所以把 L^AT_EX 视为工具, 原因就在于此。既然不需要写文章的人关注排版, 那么事先定义好详细的排版规则, 让排版出来的文章美观、高质量, 这就是 L^AT_EX 和其它宏包设计者该做的事了。对于这些人, L^AT_EX 必须是艺术。当然, 前面给出的两个关于细节控制的例子, 对于将 L^AT_EX 完全视为工具的使用者而言, 是不必要了解的, 对于重视排版效果的人, 则会为这个系统贴心的设计感到欣喜。

写了那么多, 并不是为了显示 L^AT_EX 有多么优越。Word 也很强大, 只是它太容易上手了, 容易到用户会很容易无视它强大的功能而选择最“暴力”的办法。当然, 说到论文模板这种东西, 真不得不说还是 L^AT_EX 好用。至少每次在 Word 的某条下划线上打字, 字的位置还得尽量居中的时候, 我都感到十分头大。况且 Word 整理各种编号和文档结构的效率, 还真是远不如 L^AT_EX。有人说 L^AT_EX 是理

科生写论文用的，甚至理科生里搞实验的会说那是搞理论的人用的。我看其实不然，诚然 \LaTeX 高效的公式编辑和美观的公式排版确实对搞理论的人很有用，但如果一个的文科生，一个脱离了公式和图表的纯粹的文科生，用 \LaTeX 应该感到同样的舒心痛快，乃至更甚。试想一下，一篇论文，只需要各级标题和引用，全文几乎只用到 `\chapter{}` 和 `\section{}` 开启新的章节，`\cite{}` 引用参考文献，`\footnote{}` 添加脚注，岂不是比用 Word 排版还要容易得多！

第 2 章 L^AT_EX 基础

2.1 字符

字符是文档最基本元素。最基础的 T_EX/L^AT_EX 系统仅支持英文，我们就先从英文排版说起。不同于 Word 中任何一个用键盘输入的字符都可以直接显示出来，L^AT_EX 作为一种排版语言，不可能把任何一个字符都仅当作输出字符处理，而是需要占用一些字符，来表达指令，帮助系统完成排版的任务。这些字符是不能直接输出的。此外，如果这些字符不合时宜地出现在正文里，还会引起错误。这些字符有

%	注释符	\	命令前导符
^	上标符	\$	数学模式符
_	下标符	#	参数表达符
~	空格符	{	参数起始符
&	分列符	}	参数结束符

如果你希望这些字符可以在正文中显示，“\”符号可以用 `\textbackslash` 显示，其他符号都可以用 `\符号{}` 的方式来显示，如“{”可以通过输入 `\{ }` 来显示。但 `\\{ }` 不能用来输入“\”，它是换行命令。

2.2 命令

L^AT_EX 中，命令最基本的形式是

`\命令[可选参数]{参数}`

其中 `\` 起到引导命令的作用，方括号内的参数是可选的，没有方括号可以使用默认参数，花括号内的参数是必选的，对于定义了必选参数的命令，必须给出参数。

下面从最简单的看起。就像上一节介绍怎样在正文中打印出字符，其实就是用简单的命令实现的。其中的“{ }”并没有给出参数，因此它不是完全必要的，但有些字符的打印，没有它无法通过编译，为了便于记忆，打印保留字符时不妨都加上。

再说说前面提到的换行命令 `\\`，它还可写成带可选参数的形式。`\\[1em]`

方括号里的参数代表距离。L^AT_EX 中的距离单位有 cm, mm, pt, em 等，其中 em 表示一个字符的宽度。效果嘛，你已经看到了。

L^AT_EX 中还有一种特殊的成对出现的命令，它以下面的形式出现

```
\begin{环境名}[可选参数]{必选参数}
    文字或命令
\end{环境名}
```

利用环境可以实现各种更加复杂的排版效果，例如列表和浮动的图表等。

最后，试试命令 `\LaTeX{}` 吧。

2.3 文档的基本结构

最基本的 L^AT_EX 文档实现形式如下

```
\documentclass[参数]{article}
    导言区
\begin{document}
    正文
\end{document}
```

其中 `article` 可以用其他文类代替，输入特定参数则可以实现特殊的排版效果。

导言区内可以利用命令 `\usepackage[参数]{宏包}` 来实现各种复杂的功能，或者自定义命令，满足自己写文章时特殊的需求。如输入一个公式

$$\int \sin x \, dx$$

其中 d 是微分算符，应该写成正体，可以用命令 `\text{trm}{d}` 实现。于是我们输入 `\int \sin x \, \text{trm}{d} x` 得到

$$\int \sin x \, dx$$

但每次都这样输入是在麻烦，我们只需要在导言区做这样的定义

```
\newcommand{\diff}{\text{trm}{d}}
```

这样，我们输入 `$$ \int \sin x \, dx \, \frac{d}{dx} x \, dx \, dx` 也能达到一样的效果。对于某些高频率的比较复杂的输入，这样的自定义命令是不是会给你带来很大的便利呢。

下面说说正文。正文部分可以输入文字、公式，可以插入图表，还可以利用各种命令和环境来实现特定的排版效果。首先，要了解的是 L^AT_EX 正文排版规则。不同于 Word 利用软回车换行，用硬回车分段，连续的空格可以增加横向的间距。L^AT_EX 中回车和空格都自动产生一个空格以分隔单词。连续的空格并不能在输出上产生更多的横向间距。例如这段话，它的源代码如下。

```
下面说说正文。正文部分可以输入文字、公式，可以插入图表，
还可以利用各种命令和环境来实现特定的排版效果。
首先，要了解的是 \LaTeX{} 正文排版规则。
不同于 Word 利用软回车换行，用硬回车分段，
连续的空格可以增加横向的间距。
\LaTeX{} 中回车和空格都自动产生一个空格以分隔单词。
连续的空格并不能在输出上产生更多的横向间距。
例如这段话，它的源代码如下。
```

前面我们已经知道了换行命令，那么 L^AT_EX 中如何产生新的一段呢？只要本段内容与上一段内容间有空白行（仅有空格或制表符，无其他字符），L^AT_EX 就会开起新的段落。与空格同样，更多的回车也不能增加段落的间距。

2.4 标题与文档逻辑结构

L^AT_EX 最大的特点就是你主要注意文章的逻辑结构。即告诉 L^AT_EX 什么地方新起一章，什么时候新起一节，L^AT_EX 会自动给各个章节标题编号并套用相应的格式。一般而言，作者无需关心章节标题格式的问题，但有时为了满足特定的需求，通过一些宏包，也可以修改章节标题的格式。大多科学杂志出版社要求使用 L^AT_EX 格式的文档投稿，最大的优势就在于可以方便套用预设的模板达到出版社所需的排版效果，而无需对正文的内容作其他的调整。

首先介绍文章标题。L^AT_EX 可自动生成标题。在导言区或正文开头用命令

```
\title{{标题}}, \author{作者} 和 \date{日期}
```

可以对标题的要素进行申明，而后用命令

```
\maketitle
```

就可以自动生成格式优美的标题了。需要说明的是，如果不需要打印作者，可以略去对作者的声明，但如果略去对时间的声明，则会打印出当前日期。如果不需要打印日期，则要这样声明：`\date{}`。

正文中，开起新的章节的命令分别是

`\chapter{章 标题}` 和 `\section{节 标题}`

同时，还可以用 `\subsection{标题}` 和 `\subsubsection{标题}` 来开起更细的标题结构。需要注意的是，`article` 文类是没有章标题的，它一般比较适合排版短篇论文。对于长篇报告和书籍，需要使用文类 `report` 和 `book`。

同时，文章中有时候还需要用到罗列环境。最基本的罗列环境有常用和排序两种。常用罗列环境的形式和排版效果如下

<code>\begin{itemize}</code>	• 条目 1
<code>\item 条目1</code>	
<code>\item 条目2</code>	• 条目 2
<code>\item[**] 条目3</code>	
<code>\end{itemize}</code>	** 条目 3

排序罗列环境的形式和排版效果如下

<code>\begin{enumerate}</code>	1. 条目 1
<code>\item 条目1</code>	
<code>\item 条目2</code>	2. 条目 2
<code>\item[**] 条目3</code>	
<code>\end{enumerate}</code>	** 条目 3

罗列环境还可以嵌套使用。

最后说一下中文排版。中文排版只需要将 L^AT_EX 预置的三种文类 `article`, `report` 和 `book` 替换为 `ctexart`, `ctexrep` 和 `ctexbook` 即可。中英文混排中，需要注意的一个细节是中英之间的空格。英文中把回车和空格都当作一个空格处理。这样处理，显然不符合中文的排版习惯，因为汉字不像单词，之间是不会出现空格的。而这种情况下，源文件换行必定会在输出上出现不必要的空格。用 L^AT_EX 或 pdfL^AT_EX 编译时，一般有两种解决方案。一种是依旧把回车和空格当作空格处理，在每一行末尾加上 `%` 以注释掉回车。这显然过于繁琐了。另一种是让编译器忽略掉中文字符间的空格，这时系统也会自动把中文和英文之间的空格忽略掉。（英文单词之间的空格还是会保留）一般我们在中文和英文之间需要插入

一个 ~ 符号产生一个小空格，排版效果才比较好。这又是我推荐使用 XeLaTeX 进行编译的另一个重要原因。编译器会自动处理这些空格，而无需对行末的回车进行注释或是在中文和英文单词间加入额外的空格，就能得到理想的排版效果。但我仍习惯在中文和英文之间键入一个空格，不过键入空格可比键入 ~ 方便得多。

至此，纯文字的排版基本上就介绍完了。

2.5 公式

前面多次提到 L^AT_EX 比起 Word 很大的一个优势，就是它的公式输入和排版。先说说它的排版效果吧。L^AT_EX 非常智能，对于出现在文字中的公式（行内公式）和单独占用一行的公式（行间公式）可以自动作出调整。用 Word 写文档一般要求用 Mathtype 插入公式，插入行内公式，哪怕只是 x , y , z 这样的变量，也会引起行间距增加，使排版效果大打折扣。如果插入 $\frac{1}{x}$ 这样的分式，那排版的效果简直就 undirectlookable 了。下面给大家看一下，同样一段话用 Word 排版的效果。

前面多次提到 LaTeX 比起 Word 很大的一个优势，就是它的公式输入和排版。先说说它的排版效果吧。LaTeX 非常智能，对于出现在文字中的公式（行内公式）和单独占用一行的公式（行间公式）可以自动作出调整。用 Word 写文档一般要求用 Mathtype 插入公式，插入行内公式，哪怕只是 x , y , z 这样的变量，也会引起行间距增加，使排版效果大打折扣。如果插入 $\frac{1}{x}$ 这样的分式，那排版的效果简直就 undirectlookable 了。

下面说说在 L^AT_EX 如何输入公式。虽然 L^AT_EX 的公式输入都是靠代码实现，但 L^AT_EX 输入公式的语法非常简单，甚至可以说，以英文为母语的人只要会读，就可以把公式写出来。先看看下面这个公式

$$\int_{-1}^1 \frac{f(x)}{\sqrt{1-x^2}} dx = \frac{\pi}{n} \sum_{k=1}^n f\left(\cos \frac{2k-1}{2n} \pi\right) + \frac{\pi}{2^{2n-1}(2n)!} f^{2n}(\theta)$$

看起来有点小复杂吧。别急，只要稍加说明，打出比这更复杂的公式都是很简单的的事情。

首先，需要学会开始一个公式。最基础的 L^AT_EX 中，要输入行内公式，应该这样做 $\$ 行内公式 \$$ 。如果公式比较重要，或者形式比较复杂，则一般写成行间公式的形式

$\$ \$ 行间公式 \$ \$$

前面出现的公式就是一个行间公式，我们也可以把它改写成行内公式的形式，效果是这样的 $\int_{-1}^1 \frac{f(x)}{\sqrt{1-x^2}} dx = \frac{\pi}{n} \sum_{k=1}^n f\left(\cos \frac{2k-1}{2n}\pi\right) + \frac{\pi}{2^{2n-1}(2n)!} f^{2n}(\theta)$ 。与其行间公式形式比较，L^AT_EX 自动调整了字体大小，并将求和上下限的位置改写到了求和符号的右边以减小竖直方向的空间占用，保持行距。虽然这样复杂的公式很少有人将它写成行内形式，但也可以想象一下在 Word 里插入这样复杂的行内公式，排版效果会变得多么糟糕。

学会了开始公式，我们学习最简单的表达式。数字和变量直接输入就行，例如 `$2x$` 就可以输出 $2x$ 了。希腊字母怎么读就怎么输，`$\alpha\beta\gamma$` 就可以输出 $\alpha\beta\gamma$ 了。大写字母只要开头大写就行，输入 `$\Delta\Phi\Psi$` 就能输出 $\Delta\Phi\Psi$ 。等等，还有个小问题，有些字母是有变体的，需要在命令前加 `var`。如输入 `$\varepsilon\varphi\Phi\Psi$` 就得到了 $\varepsilon\varphi\Phi\Psi$ 。

还有各种算符，算符需要用数学正体表示，以区别于变量。L^AT_EX 已经定义了很多算符命令，比如 `\lim, \ln, \sin, \cos` 得到输出 \lim, \ln, \sin, \cos 。还有积分符号 \int ：`\int`，求和符号 \sum ：`\sum` 等，别忘还有了我们前面自定义的微分算子 d ：`\diff`。现在你应该能理解前面输入的命令 `$\int \sin x \, \diff x$` 是怎样被翻译成公式 $\int \sin x dx$ 的了吧？等等 `\,` 是做什么的？忘了说了，数学模式下空格是被自动忽略的，输入 `\,` 让 $\sin x$ 和 dx 之间隔开一点（具体说是六分之一空格），会比较好看哦，如果没有它就变成了 $\int \sin x dx$ ，是不是有点别扭呢。另外，加号减号键盘上就有，乘号就是 `\times`，除号嘛，自己百度吧，反正我从没用过。还有各种等号不等号，等于，大于小于，键盘上就有。至于不等号，只要记住，等于是 `eq`，大于是 `g`，小于是 `l`，那不等号就是 `\neq`，大于等于和小于等于分别是 `\geq/\leq`，远大于远小于就是 `\gg/\ll`，是不是很简单。当然，L^AT_EX 还支持更多你见过或者没见过的字符，可以参考 *symbols* 文档。

至于上下标，分别用 `^{上标}` 和 `_ {下标}` 来输入。上下标的运用是非常灵活的。除了各种变量，数字，希腊字母的右侧可以输入上下标，左侧也可以输入，即使没有字母也可以输入，甚至算符也可以有上下标，积分和求和的上下标分别是他们的上下限。注意，上下标必须在公式模式下使用。公式里还可以输入文字

`公式中的文字`

就可以插入文字了。如果你要将一段文字上标，只要 `$^{\text{上标文字}}$` 就可以了。

还有一些多元运算符。比如分式 `\frac{分子}{分母}` 就可以输入分式。再比如开 n 次根号的表达式 `\sqrt[n]{数字, 变量或表达式}`，其中 $[n]$ 是可选参数，没有时显然就是开平方根啦。

现在，回头看看那个复杂的公式，它的代码是这样的

```
$$ \int_{-1}^1 \frac{f(x)}{\sqrt{1-x^2}} dx, \frac{d}{dx} \left( \frac{f(x)}{\sqrt{1-x^2}} \right) \\ = \frac{\pi}{n} \sum_{k=1}^n f\left(\cos\frac{2k-1}{2n}\pi\right) \\ + \frac{\pi}{2^{2n-1}(2n)!} f^{(2n)}(\theta) $$
```

应该不难理解吧。