



---

# $\LaTeX$ 模板说明

作者: Eureka

# 目录

---

<b>Chapter I: 前言</b>	<b>1</b>
<b>1 我为什么开发<math>\LaTeX</math>?</b>	<b>2</b>
1.1 关于本模板名字 . . . . .	2
1.2 关于模板开发 . . . . .	2
<hr/>	
<b>Chapter II: Introduction to <math>\LaTeX</math></b>	<b>3</b>
<b>1 模板选项</b>	<b>4</b>
1.1 页面布局 . . . . .	4
1.2 数学字体 . . . . .	4
1.3 模板语言 . . . . .	4
<b>2 封面设置</b>	<b>4</b>
2.1 封面信息 . . . . .	4
2.2 封面样式 . . . . .	4
<b>3 计数器</b>	<b>5</b>
3.1 基本计数器 . . . . .	5
3.2 计数器拓展 . . . . .	5
<b>4 页面布局</b>	<b>5</b>
4.1 geometry . . . . .	5
4.2 margin . . . . .	6
4.3 Margin 测试 . . . . .	7
4.4 Margin With FootNote . . . . .	7
<b>5 额外定义命令</b>	<b>8</b>
5.1 数学字体命令 . . . . .	8
5.2 Arrow commads . . . . .	8
5.3 缩写命令 . . . . .	8
<b>6 环境</b>	<b>8</b>
6.1 正文环境 . . . . .	8
6.1.1 Emphasis env . . . . .	9
6.1.2 List Env . . . . .	9
6.1.3 Recall Env . . . . .	9
6.2 数学环境 . . . . .	9
6.2.1 definition Env . . . . .	9
6.2.2 theorem Env . . . . .	9
6.2.3 proposition Env . . . . .	10

6.2.4 corollary Env . . . . .	10
6.2.5 lemma Env . . . . .	10
6.3 抄录环境 . . . . .	10
6.3.1 Specific Language . . . . .	11
6.3.2 Just listing . . . . .	11
<b>7 图形绘制</b>	<b>12</b>
7.1 基本图形绘制 . . . . .	12
7.2 函数绘制 . . . . .	12
7.3 参数方程绘图 . . . . .	13
<b>8 计算</b>	<b>14</b>
<b>9 Wolfram Mathematica</b>	<b>15</b>
9.1 符号计算 . . . . .	15
9.2 生成图片 . . . . .	15
9.2.1 二维图片 . . . . .	15
9.2.2 三维图片 . . . . .	16
9.3 微分方程 . . . . .	16
9.4 生成表格 . . . . .	16
<b>10 gnuplot</b>	<b>17</b>
10.1 gnuplot 的安装 . . . . .	17
10.2 gnuplot 调用 . . . . .	17
10.3 脚本介绍 . . . . .	17
10.4 使用方法 . . . . .	17
<hr/>	
<b>Chapter III: <a href="#">L<sup>A</sup>T<sub>E</sub>X</a> Insight</b>	<b>19</b>
<b>1 设计流程</b>	<b>20</b>
<b>2 设计原则</b>	<b>20</b>
2.1 简洁 . . . . .	20
2.2 编译速度 . . . . .	20
2.3 自定义 . . . . .	20
<hr/>	
<b>Chapter IV: 我与 T<sub>E</sub>X</b>	<b>22</b>
<b>1 回忆</b>	<b>23</b>
1.1 为什么是 T <sub>E</sub> X? . . . . .	23
1.2 为什么要用 T <sub>E</sub> X? . . . . .	23

Chapter V: 致谢	24
1 致谢	25
1.1 模板参考 . . . . .	25
1.2 文档参考 . . . . .	25
1.3 结语 . . . . .	25

# Chapter I

## 前言

## 1 我为什么开发 $\LaTeX$ ?

模板开发前言，我为什么还要开发这个文档类，有如下的几个原因：

- 为了给折腾了这么久  $\TeX$  的一个交代，在开发这个文档类的过程中也学到了很多的东西，不想我的经历随着时间而消逝
- 其次是为了进一步降低  $\LaTeX$  新手的入门门槛，让  $\LaTeX$  的入门更加简单，本模板力求给使用者一个完整而美妙的写作体验
- 最后，也是最重要的，为了给  $\TeX$  社区带来一点贡献

### 1.1 关于本模板名字

本模板的名字是  $\LaTeX$ ， $\LaTeX$  的意思是，其实可以从两个方面来看：

- $\TeX$  的版本从 3.14 一直稳定到如今的 3.14159，于是我也希望我这个模板尽量完美
- 同时，本文档类  $\LaTeX$  中的  $\pi$  是倒装的，说明本模板距离完美还有很大一段距离

### 1.2 关于模板开发

本模板目前全部由我本人开发，没有其他人员的参与

# Chapter II

Introduction to [\LaTeX](#)



## 1 模板选项

本模板有目前已经定义如下的选项:

- 页面布局, 其中包含两种布局, 含有 `margin` 和不含 `margin`
- 数学字体, 本模板在 Linux 系统上进行制作, 提供了 `mtpro2`, `eulervm` 字体选项
- 模板语言, 本模板提供了 `chinese`, `english` 两种语言选项

具体的使用方法见下:

### 1.1 页面布局

同样的, 本模板内置两种布局:

- `margin`: 带有 `margin` 的默认模式
- `plain`: 使用 `nomargin` 命令开启

### 1.2 数学字体

本套模板提供了三套数学字体, 分别是 `mtpro2`, `eulervm`, `newpx`, 用户可以根据自己本机安装的数学字体和数学字体偏好进行对应的数学字体选择。数学字体选择方法:

```
1 \documentclass[lang=cn, math=mathptmx]{ZLaTeX}
2 \documentclass[lang=cn, math=mtpro]{ZLaTeX}
3 \documentclass[lang=cn, math=eulervm]{ZLaTeX}
```

三个字体有各自的特色, 模板默认数学字体为:  $\TeX$  的 Computer Modern 字体。

### 1.3 模板语言

本模板提供了两套语言, `en`, `cn`, 分别表示中文和英文, 当进行了相关的设置之后, 模板中相关数学环境名和章节名会进行对应的更改

## 2 封面设置

### 2.1 封面信息

本模板支持自定义封面<sup>1</sup>信息, 模板定义了如下命令用于设置封面信息

```
1 \cover[./your pic name.<ext>]{Your Note Name}{Author}
```

### 2.2 封面样式

其实本命令就是对原始的 `\titlepage` 命令的封装, 其参数含义见上面的调用格式, 使用起来还是比较简单的。有关详细的自定义内容可以参考后文的模板源码说明。

本模板的数学字体均在 Linux 环境下进行过测试, 没有任何的字体警告, 若使用者想要引入其他的自定义数学字体, 请自行查阅相关文档

当然你也可以自定义封面的格式, 模板这里采用的是一个比较简单的样式

<sup>1</sup>封面图片来源:<https://www.pexels.com/zh-cn/photo/7301803/>, 可以免费商用



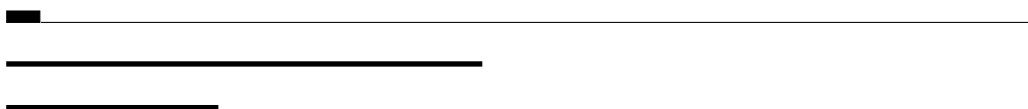
## 3 计数器

### 3.1 基本计数器

本模板继承自基本的`article`文档类, 于是能够在 `article` 中使用的计数器在本文均能使用。另外为弥补`article`文档类的不足, 本模板在此基本上重新定义了如下的一些计数器:

- `chapter`: 类似 `book` 模板中的 `chapter`
- 新增数学环境计`definition`, `proposition`, `theorem`, `lemma`, `corallary`计数器
- 除此之外, 本模板继承了 `article` 的所有基本计数器

### 3.2 计数器拓展



## 4 页面布局

### 4.1 `geomtry`

本模板采用 `geomtry` 宏包进行页面的设置, 其中对如下的原始命令进行了重定义:

```

1 \renewcommand{\newgeometry}[1]{
2   \Gm@restore@org
3   \Gm@initnewgm
4   \Gm@newgmtrue
5   \setkeys{Gm}{#1}
6   \Gm@newgmfalse
7   \Gm@process
8   \ifnum\mag=\@m\else\Gm@magtooffset\fi
9   \Gm@changelayout
10  \Gm@showparams{newgeometry}
11 }
12 \renewcommand{\restoregeometry}{
13   \Gm@restore@pkg
14   \Gm@changelayout
15 }
```

## 4.2 margin

由于`\marginpar`命令采用  $\LaTeX$  的浮动机制实现，所以可能不紧邻对应的内容，于是我们采用更加稳定的 `marginnote` 宏包提供的`\marginnote`命令，基础命令格式说明：

```
\marginnote[<left content>]{<right content>}[<voffset>]
```

—————→  
The First Margin(mandatory):

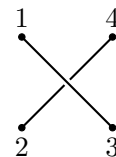
可以在 margin 中绘制其他的 tikz 图形，但如果两个 margin 的距离等参数调整的不太合理时， $\TeX$  就会丢出警告：Marginpar on page 1 moved.

本模板支持在特定页面取消 margin，重新设置当前页面的布局，只需使用本模板提供的 `\nomargin`命令即可。

同时本模板支持两种布局格式：

- 带有 margin 的默认模式，margin 的宽为 1.5cm，位于页面的右侧，当然本模板仍提供了没有 margin 的`nomarg`环境
- 取消 margin 的普通布局，需使用`\nomargin`开启，使用`\margin`结束，两者之间的内容为取消 margin 之后的页面布局，并且自动换页。如果忘记使用`\margin`命令结尾的话，后续的页面布局中布局会乱排。

所以一定要养成良好习惯，前后命令成对书写<sup>1</sup>



一个 tikz 示意图

<sup>1</sup>注意：没有 `nomargin` 对应的局部环境

### 4.3 Margin 测试

使用`\nomargin`后会自动换页，比如从这里的第 6 页自动换到 7 页：

```

1 % 第一页内容 ...
2 \nomargin
3 \section{Margin测试}
4 使用\verb|\nomargin|后会自动换页，...
5
6 \section{脚注测试}
7 本模板还对footnote进行了相关的定制，...
8 \begin{itemize}
9   \item footnote定制\footnote[1]{这是第一个脚注}
10  \item footnote定制\footnote[2]{这是第二个脚注}
11 \end{itemize}
12 \margin

```

还有就是可以从下面的公式布局看出当前是否有 margin，

$$\sum_{i=1}^{+\infty} \int_0^i -\frac{1}{t} dt = \frac{\pi^2}{6} \quad (1)$$

### 4.4 Margin With FootNote

本模板还对 footnote 进行了相关的定制，使用`\footnote`命令，会自动将脚注编号添加到脚注内容上。设置了 `footskip` 等值，使用效果如下：

- footnote 定制<sup>1</sup>
- footnote 定制<sup>2</sup>

<sup>1</sup>这是第一个脚注

<sup>2</sup>这是第二个脚注

## 5 额外定义命令

为了方便模板的使用，本模板附带了部分的自定义命令，位于`cmds.sty`宏包中，可以根据自己的喜好选择是否在模板中加载此宏包，定义的命令如下：

### 5.1 数学字体命令

```
1 % 4.1 Math Font Alias
2 \newcommand{\dd}{\mathrm{d}}
3 \newcommand{\C}[1]{\ensuremath{\mathcal{#1}}}
4 \renewcommand{\S}[1]{\ensuremath{\mathscr{#1}}}
5 \newcommand{\B}[1]{\ensuremath{\mathbb{#1}}}
6 \newcommand{\F}[1]{\ensuremath{\bm{#1}}}
7 \newcommand{\R}[1]{\ensuremath{\mathrm{#1}}}
```

### 5.2 Arrow commads

```
1 % 4.2 Math arrow
2 \newcommand{\lr}{\ensuremath{\longrightarrow}}
3 \newcommand{\equ}{\ensuremath{\Longleftarrow}},
4 \newcommand{\sr}{\ensuremath{\longmapsto}}
5 \newcommand{\lrr}[2][\ensurmath{\xrightarrow{#1}{#2}}}
6 \newcommand{\ns}{\ensuremath{\varnothing}}
7 \newcommand{\A}{\ensuremath{\forall}}
```

### 5.3 缩写命令

```
1 % 4.3 other cmd
2 \newcommand{\im}[1]{\ensuremath{\displaystyle #1}}
3 \newcommand{\dis}{\displaystyle}
4 \newcommand{\scale}[2]{\scalebox{#1}[#1]{#2}}
5 \newcommand{\sep}[1]{\hspace{#1}}
6 % \usepackage{pifont}
7 \newcommand{\lis}{\ensuremath{\text{\ding{227}\sep{.5em}}}}
```

## 6 环境

### 6.1 正文环境

由于本模板的简洁特性，于是相应的正文环境也进行了相应的简化，没有什么光彩夺目的文本环境。但是你可以把 `marginnote` 和 `footnote` 看作一种环境

### 6.1.1 Emphasis env

### 6.1.2 List Env

### 6.1.3 Recall Env

## 6.2 数学环境

本模板定义了如下的数学环境，都是相较于简洁的，符合整个模板的灰色主题。具体每一个环境的使用方法和效果，请参见如下的示例：

### 6.2.1 definition Env

```
1 \begin{definition}[Euler Formula]
2   测试如下著名的Euler公式，Take the famous Euler's Formula As Example:
3   \begin{align}
4     \sum_{i=1}^{+\infty} \int_0^i -\frac{1}{t} \mathrm{d}t =
5     \hookrightarrow \frac{\pi^2}{6}
6   \end{align}
7 \end{definition}
```

定义 1:Euler Formula

试如下著名的 Euler 公式，Take the famous Euler's Formula As Example:

$$\sum_{i=1}^{+\infty} \int_0^i -\frac{1}{t} dt = \frac{\pi^2}{6} \quad (2)$$

定义与定理环境  
可能在后期会进  
行合并，或单独  
的更改

### 6.2.2 theorem Env

```
1 \begin{theorem}
2   含有一阶导函数的泛函定义如下:
3   \begin{align}
4     \begin{aligned}
5       \delta F
6       &= F[x, U+\delta U, U'+\delta U'] - F[x, U, U'] \quad \backslash\backslash
7       &= \left[ \frac{\partial F}{\partial U} \delta U + \frac{\partial F}{\partial U'} \delta U' \right] + \frac{\partial^2 F}{\partial U^2} \delta U^2
9       &\hookrightarrow U' + \frac{1}{2!} \left[ \frac{\partial^2 F}{\partial U^2} \delta U^2 + \frac{\partial^2 F}{\partial U \partial U'} \delta U \delta U' + \frac{\partial^2 F}{\partial U'^2} \delta U'^2 \right] + \cdots \quad \backslash\backslash
10      &= \dots
11    \end{aligned}
12   \end{align}
13 \end{theorem}
```

## 定理 1

有一阶导函数的泛函定义如下:

$$\begin{aligned}\delta F &= F[x, U + \delta U, U' + \delta U'] - F[x, U, U'] \\ &= \left[ \frac{\partial F}{\partial U} \delta U + \frac{\partial F}{\partial U'} \delta U' \right] + \frac{\partial^2 F}{\partial U \partial U'} \delta U \delta U' + \frac{1}{2!} \left[ \frac{\partial^2 F}{\partial U^2} \delta U^2 + \frac{\partial^2 F}{\partial U'^2} \delta U'^2 \right] + \cdots \quad (3) \\ &= \varepsilon \left[ \frac{\partial F}{\partial U} \eta + \frac{\partial F}{\partial U'} \eta' \right] + \frac{\varepsilon^2}{2} [\cdots\end{aligned}$$

## 6.2.3 proposition Env

```
1 \begin{proposition}[某命题]
2   这是一个命题环境, this is a proposition env
3 \end{proposition}
```

这部分的内容可能也会在后期有着较大的变动

**命题 1(某命题)** 这是一个命题环境, this is a proposition env

## 6.2.4 corollary Env

```
1 \begin{corollary}[某推论]
2   这是一个推论环境, this is a corollary env
3 \end{corollary}
```

**推论 1(某推论)** 这是一个推论环境, this is a corollary env

## 6.2.5 lemma Env

```
1 \begin{lemma}[某引理]
2   这是一个引理环境, this is a lemma env
3 \end{lemma}
```

**引理 1(某引理)** 这是一个引理环境, this is a lemma env

## 6.3 抄录环境

本模板提供了部分代码抄录环境 `bytes`, `code`, 抄录环境 `code` 使用 `\lstset` 命令进行设置, 比如下面指定编程语言的高亮选择, 这里以 Python 举例.<sup>2</sup>

<sup>2</sup>lislising 已经定义了几十种语言的高亮, 详情请参见 lislising 的 2.4.1 小节

### 6.3.1 Specific Language

```

1 for circle_index in range(CIRCLE_NUM):
2     coors_set_pure = np.array([item for item in coors_set[circle_index, :] if item is
        ↪ not None and point_is_in(item)])
3     point_amount += coors_set_pure.size/2
4     if coors_set_pure.size == 0:
5         continue
6     # 把有效坐标保存
7     if circle_index == 0:
8         all_coors = pd.DataFrame(coors_set_pure)

```

```

1 int main() {
2     int data[] = {8, 7, 2, 1, 0, 9, 6};
3     int n = sizeof(data) / sizeof(data[0]);
4     printf("Original array: \n");
5     for (int i = 0; i < n; i++) {
6         printf("%d ", data[i]);
7     }
8     printf("\n");
9
10    quickSort(data, 0, n - 1);
11
12    printf("Sorted array: \n");
13    for (int i = 0; i < n; i++) {
14        printf("%d ", data[i]);
15    }
16    return 0;
17 }

```

注意：这里的语言类型 C/C++ 需要大写，否则会报错

### 6.3.2 Just listing

当然我们也可以不指定编程语言的类别，直接原样输出：

```

1 \newcounter{definition}
2 \newenvironment{definition}[1][]{%
3     \stepcounter{definition}%
4     \begin{tcolorbox}[%
5         enhanced,
6         arc=3pt,
7         frame hidden,
8         % suppressfootnotes=true,
9     ]{\end{tcolorbox}}

```

关于 footnote 还有一个 bug: 如果 height 没有填满，那么 footnote 就不会到最底部



## 7 图形绘制

在开发本文档类  $\LaTeX$  时，我也同时开发了对应的绘图宏包。往来说，新手认为在  $\TeX$  中绘制图形是一件很麻烦的事情，即便是经常用  $\LaTeX$  的朋友，在绘制一些简单的图形时，也会觉得比较麻烦。但是在  $\TeX$  中绘图，麻烦的还不知语法的本身，更重要的是引入 `tikz` 这中大型宏包之后，本身的编译速度就会变得十分的缓慢，于是本模板开率开发其他的宏包用于调用外部函数进行绘图，减少文档的编译时长。

$\LaTeX$  宏包中内置了 `FunctionPlot` 宏包，其作用就是用来绘制各种图像，包括普通形状绘制，2 维函数绘制，3 维函数绘制。自己在之前的命令基础上又重新定义了两个使用 `gnuplot` 的绘图命令，这样可以使得编译速度进一步加快。同时也对图形设置了默认的选项，免去了读者去翻阅官方文档的麻烦。

### 7.1 基本图形绘制

其中就包含基础的圆，三角形，直角三角形，平行四边形，梯形，平行线，以及各种基本图形。对应的命令调用格式如下：

```
1 % 1. 圆
2 \circle[center]{radius}
3 % 2. 三角形
4 \triangle[center]{side}
5 % 3. 矩形
6 \rectangle[center][width][angle]{height}
```

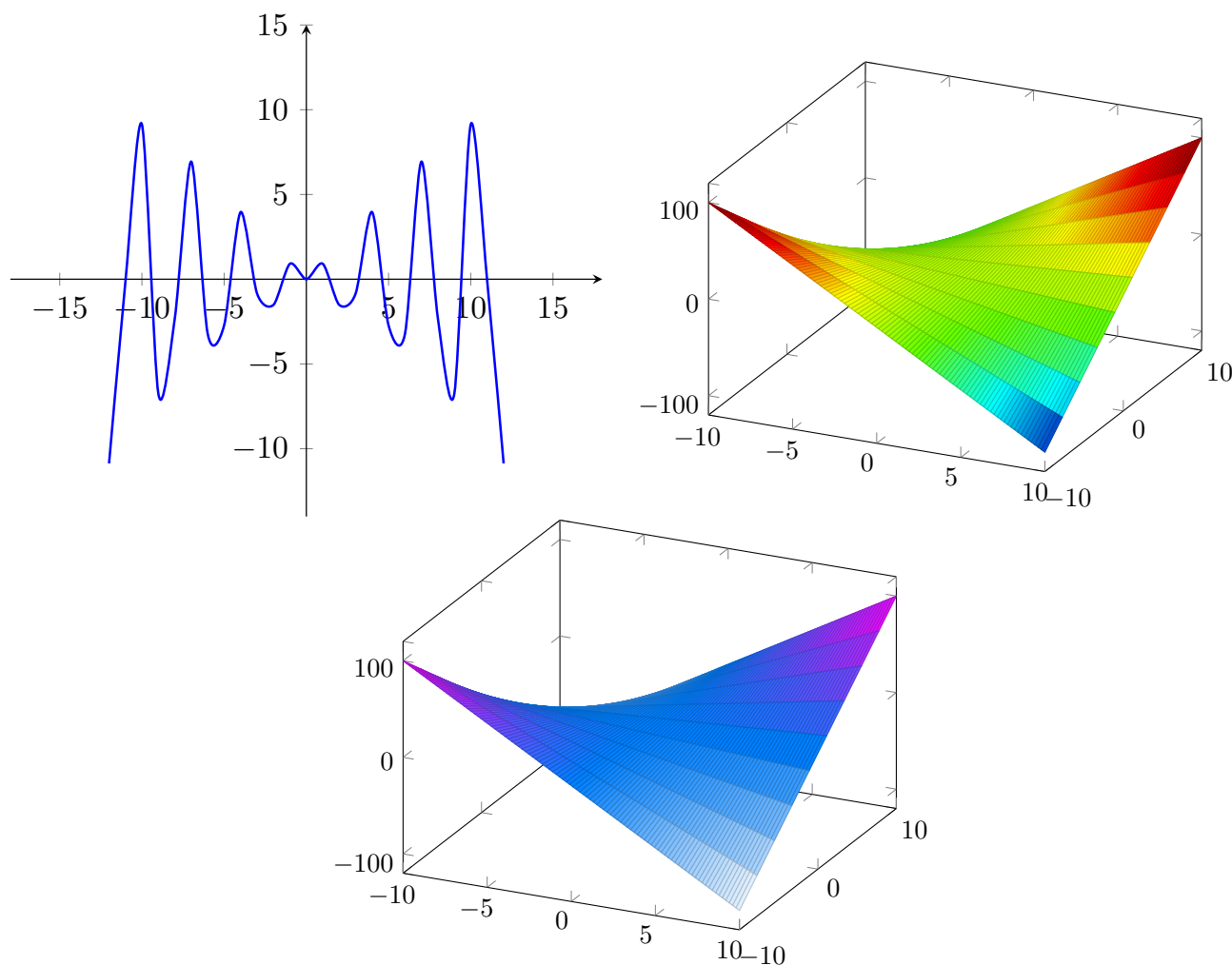
注意：这里只列出了部分图形绘制类型，完整列表请参考官方文档。对应的 `tikz` 指令也可以用户自定义

### 7.2 函数绘制

函数的绘制类型是十分的丰富的，包括：

- |         |         |               |
|---------|---------|---------------|
| • 显式函数  | • 隐式函数  | • 外部程序        |
| – 二维函数  | – 2 维函数 | – gnuplot     |
| – 3 维函数 | – 3 维函数 | – mathematica |
| – 参数方程  | – 参数方程  | – matlab      |
| – 数据绘图  | – 数据绘图  | – python      |

至于普通的二三维的函数绘图十分简单的，你可以指定一下绘制的颜色，或者说是 `colormap`，定义域之类的 `plot parameters`，尽情发挥。下面我们主要绘制了  $y = x \sin(2x)$ ， $z = x * y$  两个函数，效果还是可以的。



### 7.3 参数方程绘图

主要是在 vscode 中定义了下面 5 个绘制命令的 trigger

```

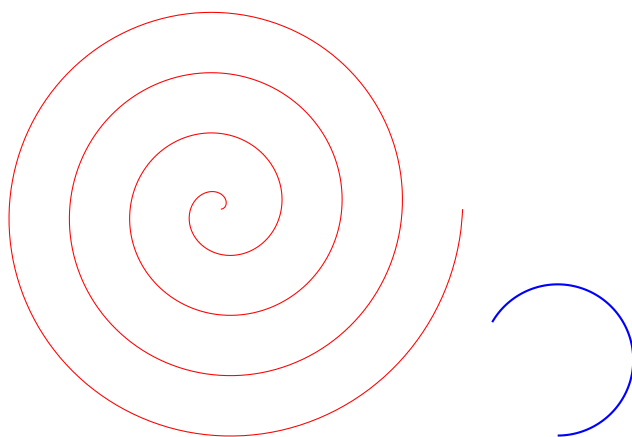
1 trigger --> 展开式
2 plot2d --> \Gplot[scale]{color}{f(x)}
3 plot3d --> \Gplotz[scale]{colormap style}{f(x, y)}
4 polarplot --> \polarplot[scale][plot parameters]{f(\t)}
5 paraplot2d --> \paraplot[scale][plot parameters]{{x(t)},{y(t)}}
6 paraplot3d --> \paraplotz[scale][plot parameters]{{x(t)},{y(t)}}

```

其实参数方程作图主要就是极坐标，二维参数方程，三维参数方程这三种常见的情形。那么我们就首先绘制极坐标的图形，下面我们绘制图形对应的方程分别为：

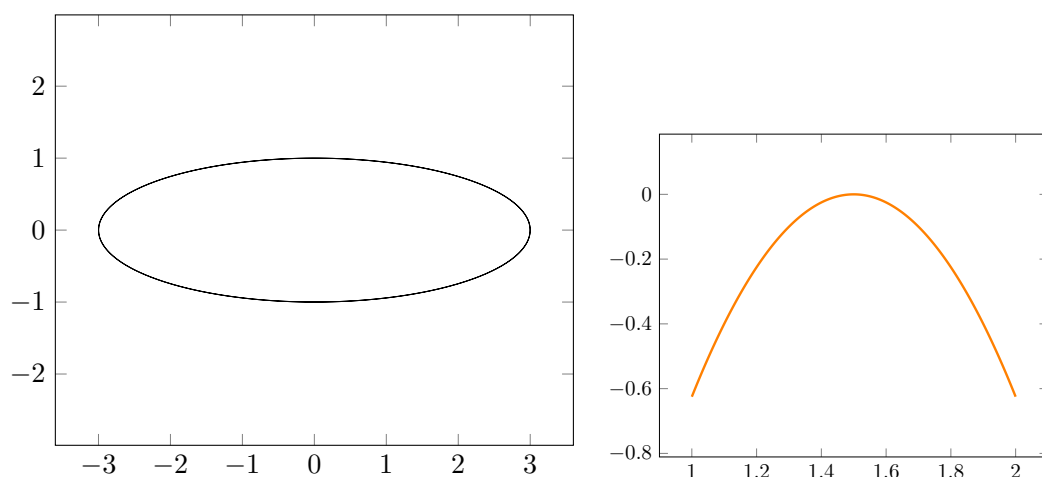
$$\rho = \frac{0.01}{\pi\theta} \quad (4)$$

$$\rho = \sin(\theta) \quad (5)$$

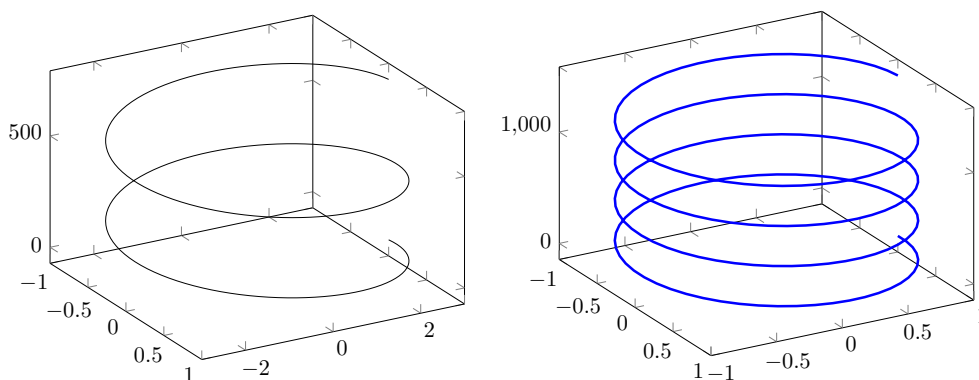


只需要注意的一个点:paraplot 默认使用的是角度,而非弧度

既然极坐标我们能够画出来了,那么接下来就是参数方程了,



最后就是我们的三维参数方程了,本质也是和上面的二维方程一样的,因为我们默认  $z(t) = t$ , 演示效果如下: 下面我们绘制了螺旋线的方程  $x = \sin(t), y = 3 * \cos(t), z = t$



同理,对应的 MMA 模块我也归纳到了 FunctionPlot 宏包中,用于调用 MMA 生成对应的 pdf 矢量图片

## 8 计算

本模板调用  $\LaTeX$  中的 `xfp` 和外部应用程序 `mathematica`, `gnuplot` 等进行计算,具体计算示例见下:

## 9 Wolfram Mathematica

### 9.1 符号计算

首先我们可以用 Mathematica 用于计算，众所周知 MMA 的符号计算是很强大的.

$$1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120} + O(x^6) \quad (6)$$

### 9.2 生成图片

#### 9.2.1 二维图片

MMA 图片测试，下面的这个绘图还是比较复杂的，于是我们使用 MMA 绘制，可以得到 1, 2 这两个绘图结果

```

1 \wolframgraphics{
2   ContourPlot[
3     x^2/4 + y^2/3 == 5, {x, -5, 5}, {y, -5, 5},
4     ContourStyle->{
5       % 同样的，被latex注释的部分不会传入MMA
6       % RGBColor["\#00C0A3"],
7       % 在传参的过程中，不要使用#，即使是\#,MMA也是会报错的。
8       RGBColor[0.,0.7529411764705882,0.6392156862745098],
9       Thickness[0.004]
10    },
11    AspectRatio->1,
12    AxesOrigin->{0,0},
13    Axes->True,
14    Frame->False,
15    AxesStyle->Arrowheads[{0, 0.03}],
16    AxesLabel->{"x", "y"}
17  ]}{MMA2d1}
18 \includegraphics[scale=.7]{MMA2d1.pdf}

```

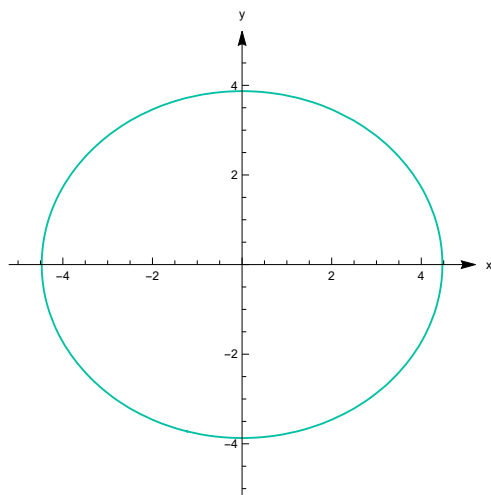


图 1: MMA 二维图形 1



图 2: MMA 二维图形 2

### 9.2.2 三维图片

二维的一些图形绘制了之后，自然要去绘制一些 MMA 中的 3 维对象了。下面就是一些例子：

```

1 \wolframgraphics{
2   Arrow[Tube[
3     BSplineCurve[{{0,0,0}, {.2,1,0.5},{2,1,1}}]
4   ]]/Graphics3D]{MMA3d1}
5 \wolframgraphics{
6   VectorPlot3D[
7     {x, y, z}, {x, -1, 1},
8     {y, -1, 1}, {z, -1, 1},
9     PlotTheme->"Classic"
10  ]]{MMA3d2}
11 \includegraphics[scale=.7]{MMA3d1.pdf}
12 \includegraphics[scale=.7]{MMA3d2.pdf}

```

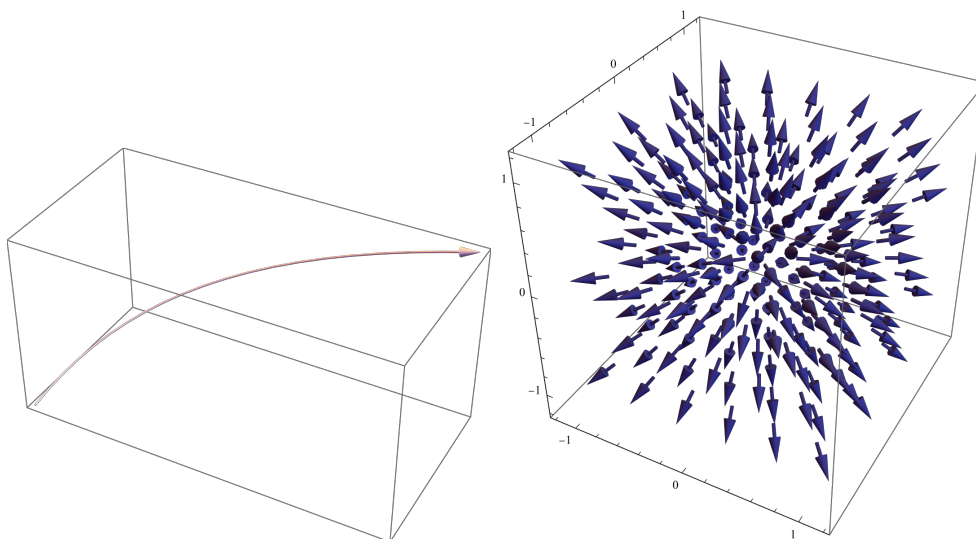


图 3: MMA 三维图形

## 9.3 微分方程

这里肯定少不了我们的微分方程求解

## 9.4 生成表格

MMA 还能够解方程，求根公式，输出表格等等.

## 10 gnuplot

鉴于 TiKZ 本身的运算能力，使用tikz来绘制数据量比较大的图形时显然是不明智的，于是我们在这里引入了gnuplot，让 TeX 在外边调用gnuplot来绘制图形，以此加快 tex 文档的编译速度。

### 10.1 gnuplot 的安装

安装 gnuplot 是很简单的

### 10.2 gnuplot 调用

想要使用外部程序gnuplot，必须要开启本模板的--shell-escape参数，在命令行手动编译时添加如下：

```
1 xelatex --shell-escape <file>.tex
```

如果你是使用的 vscode，那么可以仿照如下的配置对你的原始 json 文件进行设置：

```
1 {
2   "name": "xelatex",
3   "command": "xelatex",
4   "args": [
5     "-synctex=1",
6     "--shell-escape",
7     "-interaction=nonstopmode",
8     "-file-line-error",
9     "%DOC%"
10  ],
11 },
```

### 10.3 脚本介绍

本模板中的使用方法是十分的简单的，我已经把gnuplot的基础导出设置进行了设置，并把这些文件放在模板文件夹下的Scripts文件夹。首先说明脚本内容：

```
1 ----- 879 21 Sep 16:13 Gplot_2d.gp
2 ----- 774 22 Sep 12:33 Gplot_3d.gp
3 ----- 742 21 Sep 15:59 Gplot_3d.ps1
4 ----- 997 22 Sep 12:33 GPplot.gp
```

### 10.4 使用方法

本模板在此基础上，定义了如下几个命令：

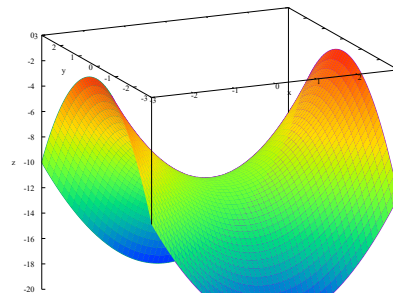
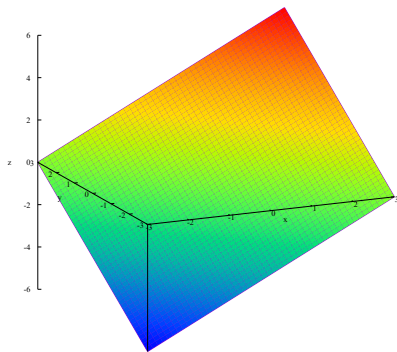
- `\GplotzNew{fun}`: 绘制三维显示函数
- `\GPplot{fun1, fun2, fun3}`: 绘制三维隐函数，也即三维参数方程绘制

使用语法是十分的简单的，如下为示例：

```

1 \GplotzNew{x+y}
2 \GplotzNew{x**2-y**2-10}

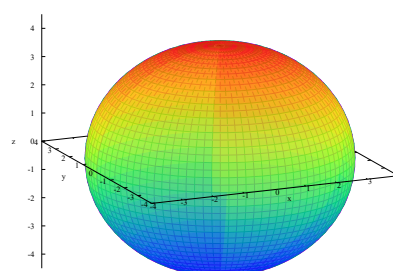
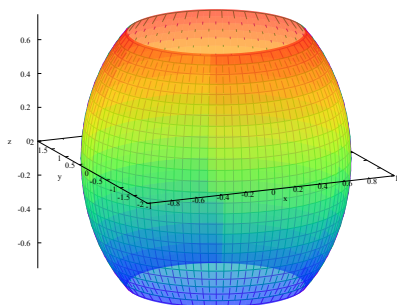
```



```

1 \GPplotz[-.75:.75]{1*cos(u)*cos(v), 2*cos(u)*sin(v), sin(u)}
2 \GPplotz[-4.5:4.5]{4*cos(u)*cos(v), 4*cos(u)*sin(v), 4*sin(u)}

```





# Chapter III

[L](#)<sup>A</sup>T<sub>E</sub>X Insight

## 1 设计流程

本模板的设计流程大致如下：

- 首先确定自己的页面布局，因为我这个模板是希望能够用于记笔记的，并且希望有 margin，于是就选择了一个较宽的页面布局，这就是为什么和其他的模板不同
- 在确定自己的基本页面尺寸后便可以考虑另一个比较大的方面：页脚，页眉，本模板采用的是fancyhdr宏包
- 然后是这个模板的语言，一般来说大部分人都是自用，也就没有去考虑英文，在这里我想说的是，你得有这种意识让你的模板能够有更多得适应面，也可以提高你使用 T<sub>E</sub>X 的能力
- 接下来是正文部分，在正文部分你首先应该考虑你需要使用的计数器（文章结构），需要 chapter 吗？需要 part 吗？确定层级顺序后便可以着手设计你的目录，本模板使用的是 tocloft宏包
- 接下来应该才是大部分人关心的部分，什么数学环境，代码环境，表格，图片等

## 2 设计原则

### 2.1 简洁

本模板没有花里胡哨的各种设计元素，主题颜色偏向灰色，没有很多的色彩，如果使用者想要更加多彩设计元素可以使用ElegantLaTeX系列模板。

简洁也就意味着本模板没有使用过多宏包，大部分命令均是通过 L<sup>A</sup>T<sub>E</sub>X2<sub>ε</sub> 的一些 primitive 定义，甚至部分结构的达成是通过 T<sub>E</sub>X 的 primitive 完成。这样的—个好处就是，本模板的使用者可以在此模板基础上自由的接入其他宏包，而不必考虑宏包的冲突问题。

### 2.2 编译速度

由于大部分人均是使用的 windows，只要你的文章稍微多一些，速度就下去了。于是本模板充分考虑到 L<sup>A</sup>T<sub>E</sub>X 中各个耗时的宏包，代码块，编译引擎差异。在 tikz 绘图和数值计算方面均采用外部程序完成，这样可以一定程度上增加编译速度。

### 2.3 自定义

在这一章节我会深入的介绍本模板的工作方式，以及其中定义的命令。比如我们这里对 content 进行了重写： [-12em]

这里自定义命令  
大部分基于  
L<sup>A</sup>T<sub>E</sub>X2<sub>ε</sub>

```
1 %% 1.原始自定义命令
2 \raise\the\dimexpr-\ch@pterskip-\title@skip%
3   \hbox{\rule{.03\textwidth}{.006\textheight}}\kern-1em%
4 \raise\the\dimexpr-\ch@pterskip-\title@skip%
5   \hbox{\rule{\dimexpr.95\textwidth-1em}{.5pt}}
6 \raise\the\dimexpr\ch@pterskip-2em\hbox{\vspace*{\the\ch@pterskip}}%
7 \kern-1em\large Chapter \Roman{chapter}: #1
```

同时考虑到 overfull hbox, 改进了原始的命令, 改进后的命令如下, 此时我们便已经修复了原始 contents 中的 overfull hbox 问题:

```
1 % 使用新的chapter命令
2 \let\old@chapter\chapter
3 \renewcommand{\chapter}[1]{%
4   \ifnum\value{chapter}=1\relax
5     \setlength{\title@skip}{-4em}
6   \else%
7     \setlength{\title@skip}{0em}
8   \fi
9   \addtocontents{toc}{%
10    \protect\mbox{\}\protect\hspace*{1.7em}%
11    \rule[\the\title@skip]{.03\textwidth}{.006\textheight}%
12    \rule[\the\title@skip]{\dimexpr.96\textwidth-2em}{.5pt}\par%
13  }
14  \old@chapter{#1}
15 }
```

# Chapter IV

我与 T<sub>E</sub>X

## 1 回忆

我为什么用 T<sub>E</sub>X? T<sub>E</sub>X 带给了我什么?

### 1.1 为什么是 T<sub>E</sub>X?

究竟什么是 T<sub>E</sub>X??

### 1.2 为什么要用 T<sub>E</sub>X?

# Chapter V

致谢

# 1 致谢

## 1.1 模板参考

本模板参考了如下的优秀模板:

- BeautifulL<sup>A</sup>T<sub>E</sub>X
- Memoir
- ElegantL<sup>A</sup>T<sub>E</sub>X
- NotesT<sub>E</sub>X V3
- CUMCMThesis

## 1.2 文档参考

## 1.3 结语

望诸君皆能找到自己之所爱.