

# **zL<sup>A</sup>T<sub>E</sub>X/zTikZ Series**

Eureka

2024 年 4 月 18 日

# 目录

<b>第一部分 Document</b>	<b>1</b>
<b>第一章 zLATEX 系列</b>	<b>2</b>
1.1 简介 . . . . .	2
1.1.1 为何叫 z? . . . . .	2
1.1.2 项目地址 . . . . .	2
1.1.3 基本组成 . . . . .	3
1.2 模板设计 . . . . .	3
1.2.1 设计历程 . . . . .	3
1.2.2 设计参考 . . . . .	4
1.2.3 设计原则 . . . . .	4
1.3 兼容性 . . . . .	5
1.4 阅前提示 . . . . .	6
1.4.1 文档编译 . . . . .	6
1.4.2 复制样例 . . . . .	6
1.4.3 键值参数 . . . . .	6
<b>第二章 zLATEX 文档类</b>	<b>8</b>
2.1 基本介绍 . . . . .	8
2.2 Set Up zLATEX . . . . .	8
2.2.1 兼容情况 . . . . .	8
2.2.2 加载 zLATEX . . . . .	8
2.2.3 额外设置 . . . . .	9
2.2.4 最小工作示例 . . . . .	9
2.2.5 在线体验 . . . . .	10
2.3 宏包机制 . . . . .	10
2.3.1 基本宏包 . . . . .	10
2.3.2 语言类宏包 . . . . .	10
2.3.3 数学类宏包 . . . . .	11
2.4 文档类选项 . . . . .	11
2.4.1 配置方法 . . . . .	12
2.4.2 注意事项 . . . . .	12
2.4.3 快捷命令 . . . . .	12
2.5 zLATEX 接口 . . . . .	13
2.5.1 命令声明 . . . . .	13
2.5.2 盒子接口 . . . . .	14
2.5.3 TikZ 接口 . . . . .	14
2.5.4 计数器 . . . . .	14
2.5.5 模板配色 . . . . .	14
2.5.6 引用环境 . . . . .	15
2.6 自定义 . . . . .	16
2.6.1 封面 . . . . .	16
2.6.2 目录 . . . . .	16
2.6.3 页眉页脚 . . . . .	17
2.6.4 章节格式 . . . . .	17
2.7 数学环境 . . . . .	18
2.7.1 常用数学环境 . . . . .	18
2.7.2 定理类环境 . . . . .	19
2.7.3 证明类环境 . . . . .	21
2.7.4 注意事项 . . . . .	22
2.7.5 自定义数学环境 . . . . .	23
2.8 标签与引用 . . . . .	23
2.8.1 footnote . . . . .	23
2.8.2 Cleveref . . . . .	23
2.8.3 图片与(列)表 . . . . .	24
2.9 列表环境 . . . . .	24
2.10 文献引用 . . . . .	24
2.10.1 基本设置 . . . . .	24
2.10.2 使用样例 . . . . .	25
2.11 索引 . . . . .	25
2.11.1 使用方法 . . . . .	25
2.11.2 Bug . . . . .	26
<b>第三章 zTikZ 宏包</b>	<b>28</b>
3.1 基本介绍 . . . . .	28
3.1.1 开发背景 . . . . .	28
3.1.2 zTikZ 功能 . . . . .	28
3.1.3 缓存机制 . . . . .	28
3.2 Set Up zTikZ . . . . .	29
3.2.1 兼容情况 . . . . .	29

3.2.2 环境配置 . . . . .	30	4.1.4 Example 4 . . . . .	74
3.3 绘图功能 . . . . .	30	4.1.5 Example 5 . . . . .	75
3.3.1 Built-in TikZ Function .	30	4.1.6 Example 6 . . . . .	76
3.3.2 gnuplot . . . . .	32	4.1.7 Example 7 . . . . .	77
3.3.3 Support Functions . . .	33	4.1.8 Example 8 . . . . .	79
3.3.4 Introduction to Plot . .	34	4.1.9 Example 9 . . . . .	81
3.3.5 Show Axis . . . . .	38	4.1.10 Example 10 . . . . .	82
3.3.6 Show Point . . . . .	40	4.1.11 Example 11 . . . . .	83
3.3.7 Contour Plot . . . . .	42	4.1.12 Example 12 . . . . .	84
3.3.8 Intersection and Plot Precise . . . . .	43	4.1.13 Example 13 . . . . .	85
3.4 Statistic Plot . . . . .	44	4.2 Mathematica . . . . .	87
3.4.1 Stairs Plot . . . . .	44	4.2.1 Example 1 . . . . .	87
3.4.2 Stem/Bar Plot . . . . .	47	4.2.2 Example 2 . . . . .	89
3.4.3 List Plot . . . . .	48	4.2.3 Example 3 . . . . .	91
3.4.4 Shade Plot . . . . .	49	4.2.4 Example 4 . . . . .	92
3.5 Polygon . . . . .	51	4.2.5 Example 5 . . . . .	93
3.5.1 基本介绍 . . . . .	51	4.2.6 Example 6 . . . . .	94
3.5.2 使用样例 . . . . .	52	4.2.7 Example 7 . . . . .	95
3.6 python/matplotlib . . . . .	53	4.2.8 Example 8 . . . . .	97
3.7 mathematica . . . . .	56	4.3 Python . . . . .	98
3.7.1 基本介绍 . . . . .	56	4.3.1 Example 1 . . . . .	98
3.7.2 注意事项 . . . . .	57	4.3.2 Example 2 . . . . .	100
3.8 matlab . . . . .	59	4.3.3 Example 3 . . . . .	102
3.8.1 基本介绍 . . . . .	59	4.3.4 Example 4 . . . . .	103
3.8.2 现状 . . . . .	60	<b>第二部分 索引与参考</b>	<b>105</b>
3.9 数值计算 . . . . .	60	<b>第五章 部分命令/名词索引</b>	<b>106</b>
3.9.1 xfp . . . . .	60	<b>第六章 参考文献</b>	<b>107</b>
3.9.2 python . . . . .	60	<b>第七章 TikZ Built-in Functions</b>	<b>108</b>
3.9.3 mathematica . . . . .	64	<b>第八章 zLATEX 部分源码说明</b>	<b>120</b>
3.10 符号计算 . . . . .	64	8.1 文档选项 . . . . .	120
3.10.1 python/sympy . . . .	64	8.1.1 Key-Value . . . . .	120
3.10.2 mathematica . . . . .	65	8.1.2 Load Options . . . . .	120
<b>第四章 Plot Gallery</b>	<b>69</b>	8.2 compile engine . . . . .	121
4.1 tikz . . . . .	69	8.3 Layout . . . . .	122
4.1.1 Example 1 . . . . .	69	8.3.1 geometry . . . . .	122
4.1.2 Example 2 . . . . .	72		
4.1.3 Example 3 . . . . .	73		

8.3.2	fancyhdr . . . . .	123	9.2.4	Polygon . . . . .	140
8.4	math . . . . .	124	9.2.5	Plot precise . . . . .	141
8.4.1	math env theme . . .	124	9.2.6	curve plot . . . . .	142
8.4.2	math environment . .	125	9.2.7	Statistic Plot . . . . .	144
8.4.3	Math alias . . . . .	128	9.2.8	ShadePlot . . . . .	146
8.5	structure Style . . .	129	9.3	Python . . . . .	149
8.5.1	contents . . . . .	129	9.3.1	matplotlib . . . . .	149
8.5.2	Chapter and section .	129	9.3.2	pycode . . . . .	150
<b>第九章</b>	<b>zTikZ 部分源码说明</b>	<b>131</b>	9.3.3	sympy . . . . .	150
9.1	Set up Env . . . . .	131	9.3.4	python eval . . . . .	151
9.1.1	Set up . . . . .	131	9.4	Mathematica . . . . .	152
9.1.2	config . . . . .	133	9.4.1	SetUp . . . . .	152
9.2	tikz . . . . .	134	9.4.2	picture . . . . .	152
9.2.1	Cache mechanism . . .	134	9.4.3	Symbols Calculation . .	153
9.2.2	externalize for tikzpicture	135	9.4.4	wolfram code . . . . .	154
9.2.3	ShowAxis . . . . .	136	9.4.5	equation Solve . . . . .	154

# 第一部分

## Document

# CHAPTER 1

---

## zLATEX 系列

### 1.1 简介

#### 1.1.1 为何叫 z?

也不知道为什么这个系列名称要加以 ‘z’ 的前缀，可能是因为个人爱好，或是因为觉得这个字母对自己而言有着一些别的意味。最开始此系列中此包含一个基本的文档类，叫做  $\pi$ LATEX，但是后面自己想开发一个用于绘图的宏包，主要基于 TiKZ. 用于常见平面图形的绘制以及外部程序的交互. 也许是看到了 tikz 库名称中的 “z”，于是便以 ‘z’ 为前缀，产生了 zLATEX 系列。

#### 1.1.2 项目地址

目前本项目已经在 GitHub, Gitlab, Gitee 上开源，地址如下：

- GitHub: [https://github.com/zongpingding/ZLaTeX\\_ZTikZ](https://github.com/zongpingding/ZLaTeX_ZTikZ)
- Gitlab: [https://gitlab.com/zongpingding/ZLaTeX\\_ZTikZ](https://gitlab.com/zongpingding/ZLaTeX_ZTikZ)
- Gitee: [https://gitee.com/zongpingding/ZLaTeX\\_ZTikZ](https://gitee.com/zongpingding/ZLaTeX_ZTikZ)

项目中包含 zLATEX 文档类源码 `zlatex.cls`, zTikZ 宏包源码 `ztikz.sty`, 以及二者的说明文档. 后续在开发过程中，可能会保证 Github 的同步更新，至于 Gitlab 与 Gitee 则不一定会同步本系列的最新版.

zLATEX 系列源代码完全开放，欢迎各位对源代码的修改以及二次分发. 如果想要和我一同改进此模板，请在 Github 提 Issue 或者是 PR. 不要在 Gitee 或者是 Gitlab 上提问，本人只维护 Github 上的仓库，尽管有时可能会为了国内用户下载方便，把 Github 上的仓库同步到这两处.

作为一个完全免费（为爱发电）的项目，我不对任何本模板的使用者负责，如果使用者在使用后遇到任何的严重后果，我不负任何责任. 我很乐意给大家解决问题，但是在提问前请先了解 LATEX 提问规范，一起营造一个愉快的讨论氛围.

想要体验本模板请到 Github 仓库: [Release 界面](#) 下载对应的最新模板. 由于本模板现在正处于早期开发阶段，所以很多的接口并不稳定，不保证模板的向后兼容性，请各位见谅.

### 1.1.3 基本组成

本系列目前包含以下的两个组成部分，一个文档类和一个绘图库：

- z<sup>A</sup>T<sub>E</sub>X 文档类
- zTikZ宏包

其中前者主要用于指定排版文档的基本属性，后者主要用于绘图<sup>1</sup>。其实从这个介绍文档就可以看出，本模板是十分的朴素的，没有十分华丽的色彩和精美的页面布局，但是在折腾了许久的 L<sup>A</sup>T<sub>E</sub>X 之后，现在这个模板才是最对我胃口的；至于，是否适合你，那就不得而知了。你可以去使用更加精美的模板，比如 [ElegantL<sup>A</sup>T<sub>E</sub>X](#), [BeautyL<sup>A</sup>T<sub>E</sub>X](#) 等优秀的模板.

## 1.2 模板设计

### 1.2.1 设计历程

本模板的设计经历了相当长的一个周期，从最开始的初始 L<sup>A</sup>T<sub>E</sub>X，我把自己常用的宏扔到了一个 .sty 文件中，以为这就是一个宏包了；之后了解到了 [ElegantL<sup>A</sup>T<sub>E</sub>X](#) 系列模板，也使用这个系列中的 book 文档类写了一点自己的笔记，但是用了一段时间之后总归是不满意，很多地方都想要自己定制，不喜欢模板默认的样式；奈何自己当时的水平不够，打开模板，看到的就是一堆的乱码。但是，后来也知道了有知乎上的优秀文章，所以就去看这些文章，慢慢的积累，渐渐的对 L<sup>A</sup>T<sub>E</sub>X 熟悉了一些，于是就着手设计属于自己的模板。

第一版的 z<sup>A</sup>T<sub>E</sub>X 其实是完全仿照 [ElegantL<sup>A</sup>T<sub>E</sub>X](#) 的 book 文档类，然后一步一步的慢慢加东西，进行一些简单的修改，比如字体，颜色等等。但是写到后面，发现这个代码的结构太不好控制了<sup>2</sup>。尤其是其中的模板语言切换，那个 \ifdefined string 语句写起来是极其痛苦的。下面就是当初写的代码片段：

```
\DeclareVoidOption{cn}{\kvs{lang=cn}}
\DeclareVoidOption{en}{\kvs{lang=en}}
\DeclareStringOption[en]{lang}
```

再加上当时的基本文档类是 article，很多 book 文档类的内部计数器和章节命令都没有，需要自己去声明；但是结果往往是自己设计的命令和别的宏包还不协调，冲突。其中最重要的就是 hyperref 宏包了，初代模板中它的跳转功能是不正常的，由于自己定义的计数器不正确，在使用 \label 命令时，激活的章节元素（跳转位置）根本不对。当初的目录结构也是自己设计，但是也有着同样的跳转为题。初代 z<sup>A</sup>T<sub>E</sub>X 文档类全部采用 L<sup>A</sup>T<sub>E</sub>X2ε 进行构建，很多的宏展开的地方都写的很繁琐，而且大部分的实现方案都是在 T<sub>E</sub>X-StackExchange 上找到的，很多时候都是处于一种能跑就行的状态，并不知道其背后的原理。

<sup>1</sup> 众所周知的，在 L<sup>A</sup>T<sub>E</sub>X 中绘图是一件十分痛苦的事情，于是乎你会看到很多书籍或笔记中的图形都是手绘或者是截图，并非矢量图

<sup>2</sup> 其实最开始这个 zTikZ 宏包和 z<sup>A</sup>T<sub>E</sub>X 是一体的，当时的代码是极其混乱的

后来自己便把 zTiKZ 从中 zLATEX 文档类中剥离出来，同时使用 LATEX3 对原始文档类和 zTikZ 进行重构。其中 zLATEX 文档类继承自 `book` 文档类，之后几乎所有命令几乎都自己书写，知道它们的具体作用，对其他的宏包的影响。于是 zLATEX 系列就诞生了，果然，在使用 LATEX3 对原始项目进行重构之后，整个项目的代码清爽了许多，比如下面的 zLATEX 文档类选项声明：

```
\zlatex_define_option:n {
  % language
  lang          .str_gset:N  = \g_zlatex_lang_str,
  lang          .initial:n = { en },
  % page layout
  layout        .str_gset:N  = \g_zlatex_layout_str,
  layout        .initial:n = { twoside },
  % margin option
  margin        .bool_gset:N = \g_zlatex_margin_bool,
  margin        .initial:n = { true },
}
\ProcessKeysOptions {zlatex / option}
```

## 1.2.2 设计参考

这个模板自然不可能是我一个人独立开发，在开发过程中参考了诸多优秀文档类/模板，参考最多的 CTEXart 文档类，几乎是本项目的大部分代码思路来源。此文档类完全采用 LATEX3 语法写成，本文档类中的选项配置模块主要参见 TeX-StackExchange 上的讨论，采用了 LATEX3 的 `key-value` 模块；这样好处有：选项配置简洁，符合人们习惯，模板维护方便。

## 1.2.3 设计原则

其实这个标题有一点太大了，什么是设计原则，我也不知道，但是我就只是想让我的模板看着舒服。怎么才能让自己的模板看着舒服呢？我也不知道，但是我觉得肯定和页边距，字体大小，字体样式等的有关。并且这三者一定是相互影响的。

比如你的页边距变大之后，压缩了你的版心大小，那么此时你的正文字体一定得做相应的改变。那么一行多少个字合适呢？去查了一下 TeX.SE，针对于英文，一行的字母个数在 65-90 是比较合适的，并且字体尺寸一般为 10pt, 11pt, 12pt；页边距到底设置多少呢？自己去比对了 ElegaLATEX 和其它模板的页边距（就差用尺子量了）；好歹后面发现了一个宏包，可以在生成的 PDF 中查看页面布局尺寸等信息，这个宏包就是 `fgruler`，使用语法也是很简单的，如下：

```
\usepackage[hshift=0mm,vshift=0mm]{fgruler}
```

当你在导言区引入之后，便可以在你的每一个页面的看到如图 1.1 的效果，这样就不用打印出来用尺子量了。

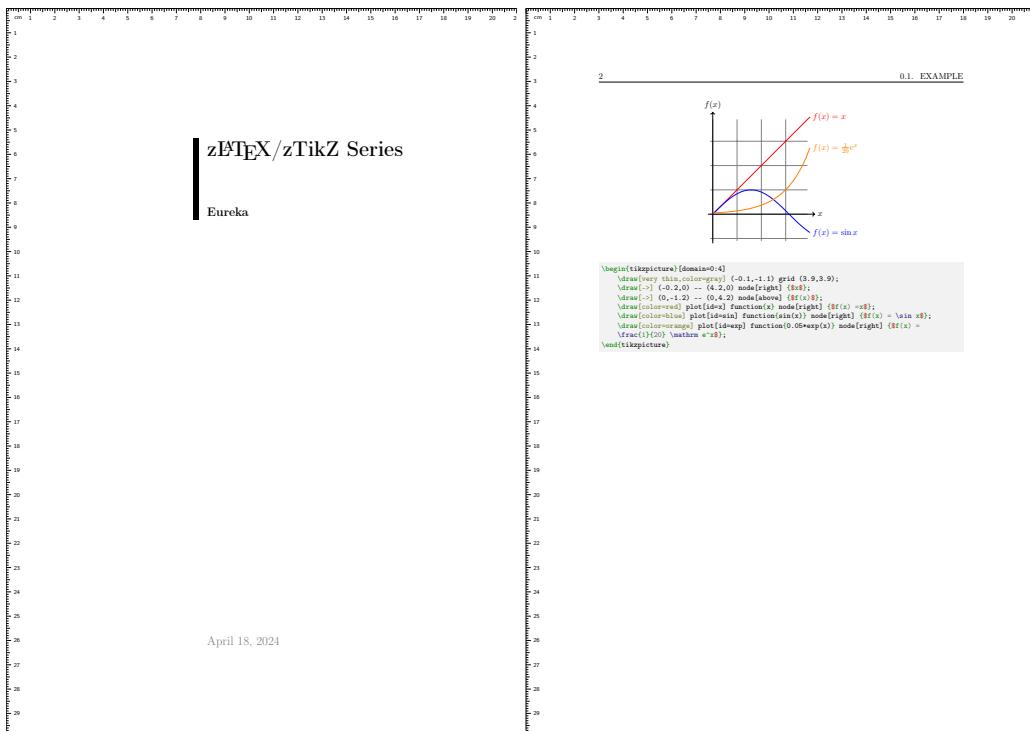


图 1.1: 页面布局示意图

在设计本模板的时，我也一直在纠结字体的问题，我应该把字体打包进入模板吗？或者是我应该在模板中给用户进行默认的字体设置吗？在这个系列的上一版中我就去找了一些免费的中文字体和西文字体，直接放在模板的文件夹下，但是这样产生的问题就很多了：

- 用户需要这个字体吗，增加的字体会变成这个模板的负担吗？
- 这个字体真的免费吗？
- 中文字体的字形往往是不全的，怎么解决？

于是最终的办法就是，我的模板不负责字体的设置，不添加任何和字体相关的配置，所有的字体由用户指定。

最后参考这些标准，一步一步的调整，使得整体的页面布局稍微的合理些。在设计这个模板时，还要考虑行距等各种元素。但是设计一个模板，你考虑的还不只这些，反正就是，如果你不会的话，那么就一切保持默认：**Be simple, Be fool.**

## 1.3 兼容性

目前本系列已经实现 Windows 和 Linux 下的兼容；但是 MacOS 下：目前仅支持 zLATEX 文档类。zTikZ 还未进行适配（参见下文了解具体原因），所以不保证本系列中的 zTikZ 文档类可以在 MacOS 下正常运行。具体的兼容情况请参见后续的兼容性章节。

## 1.4 阅前提示

### 1.4.1 文档编译

本文档对应的源文件可以在 Github 仓库下载，如果想要编译此文档请遵守如下步骤：

- 首先清除之前的编译文件，比如`.aux`, `.log`, `.toc` 等文件以及 `ztikz_output/`文件夹。
- 使用 `xelatex` 编译此文档，编译两次。(如果第一次编译报错 `missing \begin {document}...`)那么请注释掉主文件 `zlatez_ztikz_doc.tex` 中和 `indextool` 相关的两行语句:`\makeindex` `[] {}`, `\printindex` `[] {}`, 然后再次编译。如果你想要生成索引，请取消注释上一步中的两行语句，然后再次编译。
- Want Build From Scratch? 那么需要本地环境中有配置好的: `WolframScript`, `gnuplot`, `Python`, `Sed`.

### 1.4.2 复制样例

本文档中给出了相当部分的样例及其对应的代码，在书写本文当时为了读者的阅读体验，对代码抄录环境中的部分符号进行了重写。比如你会在代码中看到换行符为:`\rightarrow`，那么在复制此环境代码时，请删掉此符号。亦或者是源代码中有行号，那么在复制后，请删掉多余的行号。

### 1.4.3 键值参数

本系列中的大部分命令均采用键值对的形式进行调用，所以如果一个命令的可用键太多，那么此时我并不一定会在正文中全部说明其可用键。我会在对应的命令下方插入一个源代码抄录，用于说明此命令对应的声明原型，其中就包含了此命令可用的键值以及不同键的默认值。

比如如下的命令：

```
% key-value setup
\keys_define:nn { ztikz / polygon }{
    radius      .fp_set:N  = \l_underscore_polygon_radius_fp,
    radius      .initial:n = { 1 },
    edgeColor   .tl_set:N  = \l_underscore_polygon_edge_color_tl,
    edgeColor   .initial:n = { black },
    fillColor   .tl_set:N  = \l_underscore_polygon_fill_color_tl,
    fillColor   .initial:n = { white },
    fillOpacity .fp_set:N  = \l_underscore_polygon_fill_opacity_fp,
    fillOpacity .initial:n = { 0 },
    rotate      .fp_set:N  = \l_underscore_polygon_rotate_angle,
    rotate      .initial:n = { 0 },
    shift       .tl_set:N  = \l_underscore_polygon_shift_tl,
    shift       .initial:n = { (0,0) },
```

```
marker      .tl_set:N  = \l__polygon_marker_option_tl,
marker      .initial:n = { },
}

% command
\NewDocumentCommand\Polygon{ O{}m }{
  \group_begin:
  \keys_set:nn { ztikz / polygon } { #1 }
  ...
}
```

上述的\Polygon 命令即表示：此命令的第一个参数为一个可选参数 (O 类型)，对应的选项设置为键值对。可用的键有:radius, edgeColor, fillColor, fillOpacity, rotate, shift, marker 等等。其中 radius 表示可以接受一个浮点数 (\fp\_set:N)，默认值为 1 (.initial:n = { 1 }); 再比如第二和键 edgeColor 表示可以接受一个 TokenList(\tl\_set:N)，默认值为黑色 (.initial:n = { black })。其余类似的键不在说明。

# CHAPTER 2

---

## zLATEX 文档类

### 2.1 基本介绍

本文档类 zLATEX 基于 book 类，主要用于满足和方便使用 LATEX 使用者进行书籍和笔记的排版需求。zLATEX 全部由 LATEX3 进行编写，采用 key-value 的方式进行选项配置，方便后续的模板拓展和维护。如果使用者熟悉 LATEX，那么花费不到 10min 的时间，你便可以轻松使用本文档类用于日常的笔记撰写或者是正常的书籍的排版。

### 2.2 Set Up zLATEX

#### 2.2.1 兼容情况

目前本文档类 zLATEX 还没有登陆 CTAN，未来也没有这个打算。由于本文档类全部使用 LATEX3 进行开发，所以如果你的 TeXLive 过于老旧的话，则无法使用本宏包。目前已知 zLATEX 文档类在各平台的兼容情况为：

Windows : TeXLive 最低版本 2022

Linux : TeXLive 最低版本 2022

MacOS : 兼容 MacTeX2024(老版也应兼容)

#### 2.2.2 加载 zLATEX

由于 zLATEX 还没有传入 CTAN(未来也不会)，所以想要使用此文档类，可以有如下的两种方法：

- 把此文档类放入你的项目文件夹下
- 在命令行运行命令: kpsewhich -var-value=TEXMFHOME, 然后把 zlatex.cls 放入此路径下的 tex/latex/子目录下。在 Windows 上一般是: C:/Users/<name>/texmf/, 在 Linux 下一般是 ~/texmf/, 具体路径以自己的实际情况为准。

### 2.2.3 额外设置

由于 zLATEX 文档类只加载了基本的宏包，所以想要实现其它的功能还请自行引入相关的宏包；zLATEX 引入的宏包机制请参见表 2.1.

### 2.2.4 最小工作示例

zLATEX 的最小工作示例如下<sup>1</sup>. 首先是中文写作示例：

```
% compile engine: xelatex
\documentclass[lang=cn]{zlatex}

\titl{<title>}
\author{<author>}
\date{<date>}
\begin{document}
\maketitle
\frontmatter
% some preface
% \tableofcontents
% some claim etc.
\mainmatter

% writing your document here ...
\end{document}
```

其次是英文写作示例，你需要修改的地方只有两处：首先就是把语言选项改为 lang=en，其次便是把编译方式改为 pdflatex.

```
% compile engine: pdflatex
\documentclass[lang=en]{zlatex}

\titl{<title>}
\author{<author>}
\date{<date>}
\begin{document}
\maketitle
\frontmatter
% some preface
% \tableofcontents
% some claim etc.
\mainmatter
```

---

<sup>1</sup>可能需要根据自己的实际情况加以调整

```
% writing your document here ...
\end{document}
```

### 2.2.5 在线体验

为了让用户能够直接体验到本文档类，免去部分的环境配置阶段。我已将本模板部署在 Overleaf 上，地址:Overleaf ZLaTeX Project. 用户直接打开此地址即可体验本文档类。Overleaf 上体验项目对应的内容为本文档的 zLATEX 部分的说明文档已经源码说明，不含有 zTikZ 文档，由于部分的技术原因，zTikZ 请在本地体验。

## 2.3 宏包机制

zLATEX 文档类会根据用户指定的选项自动处理和加载对应的宏包，所以 zLATEX 文档类在不同的导言区选项声明下加载的宏包和命令是不同的。本文档类内置导言区选项输出命令:\zlatextOptions，用于打印此时文档类 zLATEX 接收到的选项。比如此时文档类接收到的选项为：

**Class Options:** cn- oneside - 10pt

以下为详细的宏包加载信息：

### 2.3.1 基本宏包

基本宏包，意味着不管你的导言区如何的配置，这些宏包都是会加载的。宏包列表如下：

---

expl3	13keys2e	framed	geometry
fancyhdr	amsfonts	amsmath	amsthm
xcolor	biblatex	indextools	hyperref
cleveref	graphicx	float	titletoc
titlesec			

---

表 2.1: zLATEX 文档类基本宏包

### 2.3.2 语言类宏包

根据不同的文档类语言，zLATEX 会加载不同的和语言相关的宏包，在 lang=en(cn) 下的宏包加载列表分别为：

---

lang=en	inputenc(pdftex)	fontenc	babel
lang=cn	fontspec	ctex	

---

表 2.2: zLATEX 文档类语言宏包

在 L<sup>A</sup>T<sub>E</sub>X 排版中，正确使用引号的方法是分别使用 ‘和’‘和’，分别用 `` 和 ``' 排版单引号 ‘和’，但却没有合适的单个符号用来表示前双引号，所以在正文中并不常用 (lshort-zh-cn.pdf, “2.3.5 标点符号” )。另外，在排版要求中，当两层双引号嵌套使用时，其外层需要使用双引号，而内层则应该使用单引号，并且不同的语言中，用于表示引号的字符也不完全一致。为解决这些排版中的问题，可以使用 `csquotes` 宏包实现引号的灵活排版<sup>2</sup>。

### 2.3.3 数学类宏包

从前面的导言区数学字体配置就可以看出，本模板会根据导言区设置不同的数学字体的功能了。具体的加载宏包规则如下：

- `math-font=<none>`: 不加载任何的数学字体宏包，采用默认数学字体
- `math-font=newtx`: 加载宏包 `newtxmath`
- `math-font=euler`: 加载命令 `\RequirePackage [OT1, euler-digits]{eulervm}`
- `math-font=mtpro2`: 加载命令  
`\RequirePackage [lite, subscriptcorrection, slantedGreek, nofontinfo]{mtpro2}`

如果使用者在导言区指定了选项 `math-alias=true`，那么 zL<sup>A</sup>T<sub>E</sub>X 此时还会额外加载宏包：`amssymb`, `mathtools`, `bm`.

## 2.4 文档类选项

本模板具有丰富的配置选项，包含页面设置，页边距，边注，数学字体，字体大小，模板语言；采用键值对 [`<key 1>=<value 1>`, `<key 2>=<value 2>`] 的形式对各个选项就行指定，和具体的指定顺序无关，具体的可配置项和可用的配置值参见表 2.3：

---

<sup>2</sup>本内容摘自 `latexstudio`

### 2.4.1 配置方法

选项 <key>	可选值 <value>	默认值
lang	en, cn	en
layout	oneside, twoside	twoside
margin	false, true	true
fontsize	10pt, 11pt, 12pt	11pt
math-alias	false, true	false
math-font	newtx, mtp2, euler	<none>
bib-source	<自定义>	ref.bib
toc	rename, 2column	<none>

表 2.3: zLATEX 配置选项

目前的 zLATEX 接口还不够丰富，没有进行相关的 Hook(钩子) 的声明，所以用户可以配置的选项是比较少的，只要能够把导言区设置规范，那么剩下的内容你几乎是不用在设置了。

### 2.4.2 注意事项

下面是一些你在指定文档类选项时应该注意到的问题：

- `margin=false` 只有在指定 `layout=oneside` 时才会启用，否则会抛出警告。同时需要注意，如果原来含有`\marginpar` 命令的文档，在指定 `margin=false` 后，对应的`\marginpar` 环境，会被替换为 `framed` 宏包提供的 `leftbar` 环境。
- `lang=cn` 时仅支持编译方式为 `xelatex`，在指定 `lang=en` 时，`pdflatex`, `xelatex` 二者都是可以接受的，但是建议采用 `pdflatex`，因为在指定为 `en` 时部分的西文宏包可能会有冲突的危险，因为当 `lang=en`，并且采用 `pdflatex` 进行编译时，zLATEX 会引入宏包 `inputenc`，然而此宏包对 `xelatex` 是没有适配的。
- 数学字体选项不一定符合每一个人，本模板的开发环境为 WSL+Archlinux。同时其中的 `mtp2` 字体并非免费字体，请注意。
- `math-alias` 选项可以根据个人习惯进行选择，默认情况下并不会加载。但是在加载此选项后，默认的两个 LATEX 指令`\S`，`\ll` 会被覆盖，分别被更名为`\ss`，`\LL`。`(§, <<)`
- `toc` 选项可以同时指定 `rename`, `2column`, 分别代表重定义目录名 (居中加粗)，以及使用双栏目录排版。

### 2.4.3 快捷命令

关于文档类选项 `math-alias` 的进一步说明，默认的自定义命令可能并不一定符合每一个人的习惯，所以请谨慎加载此选项。在 `math-alias=true` 后，zLATEX 会进行如下命令的声

明/重定义, 以及宏包的加载:

```
\RequirePackage{amssymb, mathtools}
\RequirePackage{bm}
% Math Font
\newcommand{\dd}{\mathrm{d}}
\newcommand{\C}[1]{\mathcal{#1}}
\let\ss\S
\renewcommand{\S}[1]{\mathscr{#1}}
\newcommand{\B}[1]{\mathbb{#1}}
\newcommand{\FF}[1]{\mathbf{#1}}
\newcommand{\F}[1]{\mathsf{#1}}
\newcommand{\R}[1]{\mathrm{#1}}
\newcommand{\K}[1]{\mathfrak{#1}}
% Math Arrow
\newcommand{\lr}{\longrightarrow}
\let\LL\ll
\renewcommand{\ll}{\longleftarrow}
\newcommand{\equ}{\Longleftrightarrow}
\newcommand{\sr}{\longmapsto}
\newcommand{\lrr}[2]{\xrightarrow{#1}{}^{#2}}
\renewcommand{\lll}[2]{\xleftarrow{#1}{}^{#2}}
\newcommand{\ns}{\varnothing}
\newcommand{\A}{\forall}
% Math Operator
\newcommand{\alt}{\mathrm{Alt}}
\newcommand{\sgn}{\mathrm{sgn}}
\newcommand{\curl}{\mathrm{curl}}
\newcommand{\grad}{\mathrm{grad}}
\newcommand{\trace}{\mathrm{trace}}
\renewcommand{\div}{\mathrm{div}}
```

上述的几个常见的数学运算符声明可能并不是那么的规范, 如果有必要, 可以自行进行修改. 可以参见命令\DeclareMathOperator .

## 2.5 zLATEX 接口

zLATEX 的接口正在不断的完善中, 所以目前的接口可能并不是那么稳定。(我已经尽力让接口规范和稳定了)

### 2.5.1 命令声明

后面我可能会考虑建立一个用于自定义命令的接口, 采用键值对的方式进行配置, 而不是默认的位置参数或者是 xparse 提供的可选, 默认, 强制参数等。依托于 LATEX3 的 key-value

模块，这样的接口会更加的灵活，方便用户进行配置。

### 2.5.2 盒子接口

由于目前我还没有弄清楚 LATEX3 的盒子操作，所以 zLATEX 的盒子接口还没有进行完善，但是我会尽快的进行完善。

### 2.5.3 TikZ 接口

本人并不打算在 zLATEX 中使用 tikz 宏包，因为我觉得这个宏包太过于庞大，很多的功能都不是一个文档类必须的。可能我会在后续引入 l3draw 模块用于 TikZ 操作。

本文档类配套的 ztikz 库提供了丰富的和 TikZ 绘图，数值计算，以及部分的图像处理功能。具体使用请参见下一个单元。

### 2.5.4 计数器

目前的计数器部分继承自 book 文档类和使用 amsthm 宏包定义的数学环境计数器 theorem, definition, corollary, example, axiom, remark。

目前 zLATEX 提供了一个命令 \zlatexUpdateCounterAfter 用于设置计数器的更新，使用格式为：

```
\zlatexUpdateCounterAfter{<child>}{<father>}
```

也就是让上述的 <child> 计数器随着 <father> 父计数器的更新而更新，本命令的实现原型为：

```
\NewDocumentCommand{\zlatexUpdateCounterAfter}{mm}{
    \@addtoreset{#1}{#2}
}
```

关于本文档类的公式计数器的说明，本文档类公式计数器默认跟随 section 计数器更新，在 zLATEX 的源码中的声明为：

```
\counterwithin{equation}{section}
```

### 2.5.5 模板配色

zLATEX 提供了 \zlatexColorSetup 用于设置整个模板的配色。可供用户配置的选项有：

- Hyperref 宏包对应的颜色，对应的键为 link, url, cite，三者颜色默认不同。
- Chapter 章节计数器颜色，对应的键为 chapter。

- Chapter 章节 ruler 颜色, 对应的键为 `chapter-rule`.
- 所有数学环境对应的颜色, 对应的键为 `<math-env-name>`, 如 `axiom`, `definition`, `theorem`, `remark` 等.

下面给出设置具体色彩的示例代码以及模板的默认配色:

```
\zlatexColorSetup{
    link          = purple,
    chapter-rule  = black,
    axiom         = purple,
    definition    = blue
}
```

结构元素	chapter	chapter-rule	link	url	cite		
颜色							
数学环境	axiom	definition	theorem	lemma	corollary	proposition	remark
颜色							

表 2.4: zLATEX 文档类默认配色

### 2.5.6 引用环境

目前本系列提供命令`\zlatexFramed {<name>}[<color>]`用于创建类似 MarkDown 的彩色引用环境. 参数中的 `<name>` 表示声明环境的名称<sup>3</sup>, `<color>` 表示此环境的背景颜色. 一个简单的使用样例如下:

```
% 环境 'refer' 声明
\zlatexFramed[refer][orange]
% 使用环境 'refer'
\begin{refer}%
% something wrting here
\end{refer}
```

As any dedicated reader can clearly see, the Ideal of practical reason is a representation of, as far as I know, the things in themselves;

劳伦衣普桑，认至将指点效则机，最你更枝。想极整月正进好志次回总般，段然取向使张规军证回，世市总李率英茄持伴。

<sup>3</sup>如果此环境已存在, 那么该环境会被 Override

在上面的 refer 环境开始时插入一个% 可以用于消除多余的空格

## 2.6 自定义

### 2.6.1 封面

本文档类并没有内建复杂的封面格式，只是简单的重定义了\maketitle 命令用于生成封面。声明如下：

```
\renewcommand{\maketitle}{%
  \begin{titlepage}
    \vfill\vspace*{40pt}
    \noindent\hspace*{134pt}\rule[-75pt]{6pt}{95pt}\hspace*{10pt}\leavevmode\parbox[t]{25em}{\fontsize{25}{25}\selectfont\bfseries\@title}\par
    \vspace*{-15pt}
    \noindent\hspace*{150pt}\leavevmode\parbox[t]{20em}{\Large\bfseries\@author}\par
    \vfill
    \noindent\hspace*{150pt}{\Large\textrm{gray}}{\@date}
  \end{titlepage}}
```

如果使用者想要使用更加美观的封面，请手动加载 tikz 宏包，自己定义。

### 2.6.2 目录

尽管在 zLATEX 的加载选项一节便已经说明了 zLATEX 文档类默认加载了 titletoc 宏包用于目录的格式定制，并且提供了对应的加载选项 toc 已经对应的可选值 rename, 2column, 但是在本文档类中并没有对目录的格式进行更加深度的定制，可能后续会开发对应的接口。

如果用户想要定制目录的格式可以使用本文默认加载的 titlesec 宏包或者是自行加载 tikz 宏包对目录进行定制。这里给出第一种方案下的一个定制样例，效果可见图 2.1：

```
% \RequirePackage{znumber}
\titlecontents*{chapter}[0pc]{\large\bfseries
  {\textcolor{\tl_use:N \l_zlatex_link_color_tl}{\thecontentslabel}}{}}
  {\normalfont\titlerule*[1pc]{.}\contentspage}[]

\titlecontents{section}[1.8pc]{\addvspace{3pt}\tl_use:N\g_sec_symbol_tl,}
  {\textcolor{\tl_use:N \l_zlatex_link_color_tl}{\$;\thecontentslabel}}{}
  {\titlerule*[1pc]{.}\contentspage\%[\{\tocsym[sec]\}]}

\titlecontents{subsection}[3pc]{\addvspace{3pt}\tl_use:N\g_subsec_symbol_tl,}
  {\textcolor{\tl_use:N \l_zlatex_link_color_tl}{\thecontentslabel}}{}
  {\titlerule*[1pc]{.}\contentspage\%[\{\tocsym[subsec]\}]}

\titlecontents*{subsubsection}[5.8pc]{\small
  {\textcolor{\tl_use:N \l_zlatex_link_color_tl}{\thecontentslabel}}{}}
```

```

{(\thecontentspage)}[\qquad]

% append of prepend item to toc
% switch left indent for subsection (none->3pc)
\tl_new:N \g__sec_symbol_tl
\tl_new:N \g__subsec_symbol_tl
\tl_set:Nn \g__sec_symbol_tl {}
\tl_set:Nn \g__subsec_symbol_tl {}
\NewDocumentCommand\toctsym{O{sec}O{}}{
    \str_case:nnF {#2} {
        {star}{ \tl_set:cn {g__#1_symbol_tl}{\ding{73}} }
        {=}{ \tl_set:cn {g__#1_symbol_tl}{\ding{104}} } %*
    }{\tl_set:cn {g__#1_symbol_tl}{#2}}
}

```

但是单独设置 ToC 中的样式往往是不够的，可能还需要设置不同计数器（章节）的格式，参见[section 2.6.4](#).

### 2.6.3 页眉页脚

本文档类采用 `fancyhdr` 进行页眉页脚的定制，目前已经写死在文档类中，如果使用者想要自定义页眉页脚，可以直接重定义`\fancyhead`，`\fancyfoot` 命令。或者是页面样式（pagestyle）对应的 `fancy` 样式。后续会考虑添加对应的接口。

### 2.6.4 章节格式

目前还不支持指定章节格式，等后续在添加。使用者可以加载 `titlesec`, `tocloft` 等宏包进行自定义。`LATEX` 文档类默认加载了 `titlesec`, `titletoc` 宏包用于章节格式和目录的格式定制。如果使用者想要自定义章节格式，直接使用 `titlesec` 宏包的`\titleformat` 命令覆盖本模板的原始定义即可，或者是其他的命令。

在前面我们已经提到可以使用 `titletoc` 宏包对目录进行定制，这里给出与之相配的计数器格式设置。

```

% title format
\titleformat{\section}
    {\bfseries\centering\Large}
    {\thesection}
    {0ex}{}
\titleformat{\subsection}
    {\bfseries\large}
    {\thesubsection}
    {0ex}{}
\titleformat{\subsubsection}

```

```

{\bfseries}
{\thesubsubsection}
{0ex}{\!\!\!}{\!\!\!}

% (sub)section number style
\setcounter{secnumdepth}{5}
\renewcommand{\thesection}{\thechapter.\arabic{section}\!}
\renewcommand{\thesubsection}{\zhnum{subsection}\hspace*{-1ex}}
\renewcommand{\thesubsubsection}{\alph{subsubsection}\!}

```

最终就可以得到如下样式的目录格式:

<b>目录</b>  第一章 一元函数极限 §1.1 极限 1 1. 几何解释 a. 用直角坐标系中的图象 b. 用数轴上的图象 c. 用无穷小量的极限 2. 极限的性质 a. 有理数的极限 b. 无穷大与无穷小的极限 3. Walli 公式 4 4. 单元练习 1.1 7 §1.2 常定义证明极限的存在性 1 1. 定义及证明 a. 用直角坐标系中的图象 b. 用数轴上的图象 2. 用 Cauchy 定义证明 14 3. 不等式法 “ $\forall$ ” 和 “ $\exists$ ” 的使用法则 15 四、利用区间套原理证明极限存在 17 五、利用夹逼准则证明极限的关系 18 六、极限的计算 19 七、单元练习 1.2 20 §1.3 未指明的若干问题 22 1. 利用等价无穷小替换求极限 a. 等价的(2) b. 利用等价无穷小替换(2) 24 2. 利用对数极限 25 3. 利用重要极限求极限 26 4. 利用洛必达法则求极限 28 六、多维极限 30 a. 用直角坐标系中的图象(149) b. 用数轴上的图象(149) c. 带参数的(155) b. 利用 Taylor 公式求极限(32) d. 利用等价无穷小替换求极限(32) c. 利用泰勒公式的推广形式(32) d. 利用泰勒公式的推广形式(32) d. 利用洛必达法则求极限(32) e. 利用洛必达法则求极限(32)	  §2.1 逆极限的定义与应用 1 1. 逆极限的定义 a. 逆极限的图象 b. 逆极限的性质 2 2. 单元练习 2.1 a. 逆极限的(12) 3 §2.2 逆极限性质 114 1. 用逆极限的性质证明逆极限的性质(126) 2. 用逆极限的性质求逆极限 3 3. 逆极限的图象 a. 逆极限的图象和数列一致的图象(126) 4 4. 单元练习 2.2 122 §2.3 一维极限 126 1. 单元练习 2.3 126 2. 一维极限的定义 a. 一维极限的定义 b. 一维极限的图象 3 3. 单元练习 2.3 127 a. 逆极限的(127) 4 4. 逆极限性质 127 5 5. 逆极限的图象 128 6 6. 单元练习 2.4 133 a. 逆极限的(133) §2.4 多维极限 135 1 1. 多维极限的定义 a. 多维极限的定义 b. 多维极限的图象 2 2. 多维极限的性质 139 a. 利用等价无穷小替换(148) b. 利用泰勒公式(148) c. 利用泰勒公式(148) d. 利用泰勒公式(148) e. 利用洛必达法则求极限(148) f. 利用洛必达法则求极限(148) 3 3. 单元练习 2.4 140 a. 利用泰勒公式(149) b. 利用等价无穷小替换(149) c. 带参数的(155) 4 4. 单元练习 2.5 155 a. 利用泰勒公式(155) b. 利用洛必达法则求极限(155) c. 利用等价无穷小替换(155) 5 5. 单元练习 2.5 156 a. 利用泰勒公式(156) b. 利用洛必达法则求极限(156) c. 利用等价无穷小替换(156) 6 6. 单元练习 2.6 159 a. 利用泰勒公式(159) b. 利用洛必达法则求极限(159) c. 5	  为方便阅读的空格 三、导数的定义与性质 1 a. 导数的定义(160) b. 导数的性质(160) c. 拉格朗日(3.2.2)(160) 4.4. 泰勒公式的图象(160) §4.5 Cauchy 中值定理及 L'Hospital 定理 1 a. Cauchy 中值定理(172) b. 泰勒公式与导数无关(172) c. 拉格朗日广义 广义拉格朗日(172) 五、单边导数 3.2 1 a. 单边导数的定义(172) b. 单边导数的性质(172) c. 单边导数的图象(172) d. 单 边导数的性质(172) e. 单边导数的图象(172) 3.3 Taylor 公式 1 a. 证明中值公式的(179) b. 证明泰勒公式的(179) c. 证 证明泰勒公式的(179) d. 证明泰勒公式的中值公式的(179) §3.3 Taylor 公式 1 a. 证明中值公式的(179) b. 证明泰勒公式的(179) c. 证 证明泰勒公式的(179) d. 证明泰勒公式的中值公式的(179) 3.4 不等式 1 a. 不等式的性质(180) b. 不等式的图象(180) c. 不等 不等式的性质(180) d. 不等式的图象(180) §3.4 不等式 1 a. 不等式的性质(180) b. 不等式的图象(180) c. 不等 不等式的性质(180) d. 不等式的图象(180) 3.5 单边导数的性质 1 a. 单边导数的性质(188) b. 单边导数的图象(188) b. 单边导数的性质(188) c. 单边导数的图象(188) 3.6 中值点的性质 1 a. 中值点的性质(189) b. 中值点的图象(189) b. 中值点的性质(189) c. 中值点的图象(189) 3.7 单边方程中的应用 1 a. 泰勒公式的单边方程(190) b. 泰勒公式的图象(190) b. 泰勒公式的单边方程(190) c. 泰勒公式的图象(190) 3.8 泰勒公式的应用 1 a. 泰勒公式的应用(191) b. 泰勒公式的图象(191) b. 泰勒公式的应用(191) c. 泰勒公式的图象(191) 3.9 泰勒公式的应用 1 a. 泰勒公式的应用(192) b. 泰勒公式的图象(192) b. 泰勒公式的应用(192) c. 泰勒公式的图象(192) 3.10 泰勒公式的应用 1 a. 泰勒公式的应用(193) b. 泰勒公式的图象(193) b. 泰勒公式的应用(193) c. 泰勒公式的图象(193) 3.11 泰勒公式的应用 1 a. 泰勒公式的应用(194) b. 泰勒公式的图象(194) b. 泰勒公式的应用(194) c. 泰勒公式的图象(194) 3.12 泰勒公式的应用 1 a. 泰勒公式的应用(195) b. 泰勒公式的图象(195) b. 泰勒公式的应用(195) c. 泰勒公式的图象(195) 3.13 泰勒公式的应用 1 a. 泰勒公式的应用(196) b. 泰勒公式的图象(196) b. 泰勒公式的应用(196) c. 泰勒公式的图象(196) 3.14 不等式 1 a. 不等式的性质(197) b. 不等式的图象(197) b. 不等式的性质(197) c. 不等式的图象(197) 3.15 不等式 1 a. 不等式的性质(198) b. 不等式的图象(198) b. 不等式的性质(198) c. 不等式的图象(198) 3.16 不等式 1 a. 不等式的性质(199) b. 不等式的图象(199) b. 不等式的性质(199) c. 不等式的图象(199) 3.17 不等式 1 a. 不等式的性质(200) b. 不等式的图象(200) b. 不等式的性质(200) c. 不等式的图象(200) 3.18 不等式 1 a. 不等式的性质(201) b. 不等式的图象(201) b. 不等式的性质(201) c. 不等式的图象(201) 3.19 不等式 1 a. 不等式的性质(202) b. 不等式的图象(202) b. 不等式的性质(202) c. 不等式的图象(202) 3.20 不等式 1 a. 不等式的性质(203) b. 不等式的图象(203) b. 不等式的性质(203) c. 不等式的图象(203) 3.21 不等式 1 a. 不等式的性质(204) b. 不等式的图象(204) b. 不等式的性质(204) c. 不等式的图象(204) 3.22 不等式 1 a. 不等式的性质(205) b. 不等式的图象(205) b. 不等式的性质(205) c. 不等式的图象(205) 3.23 不等式 1 a. 不等式的性质(206) b. 不等式的图象(206) b. 不等式的性质(206) c. 不等式的图象(206) 3.24 不等式 1 a. 不等式的性质(207) b. 不等式的图象(207) b. 不等式的性质(207) c. 不等式的图象(207) 3.25 不等式 1 a. 不等式的性质(208) b. 不等式的图象(208) b. 不等式的性质(208) c. 不等式的图象(208) 3.26 不等式 1 a. 不等式的性质(209) b. 不等式的图象(209) b. 不等式的性质(209) c. 不等式的图象(209) 3.27 不等式 1 a. 不等式的性质(210) b. 不等式的图象(210) b. 不等式的性质(210) c. 不等式的图象(210) 3.28 不等式 1 a. 不等式的性质(211) b. 不等式的图象(211) b. 不等式的性质(211) c. 不等式的图象(211) 3.29 不等式 1 a. 不等式的性质(212) b. 不等式的图象(212) b. 不等式的性质(212) c. 不等式的图象(212) 3.30 不等式 1 a. 不等式的性质(213) b. 不等式的图象(213) b. 不等式的性质(213) c. 不等式的图象(213) 3.31 不等式 1 a. 不等式的性质(214) b. 不等式的图象(214) b. 不等式的性质(214) c. 不等式的图象(214) 3.32 不等式 1 a. 不等式的性质(215) b. 不等式的图象(215) b. 不等式的性质(215) c. 不等式的图象(215) 3.33 不等式 1 a. 不等式的性质(216) b. 不等式的图象(216) b. 不等式的性质(216) c. 不等式的图象(216) 3.34 不等式 1 a. 不等式的性质(217) b. 不等式的图象(217) b. 不等式的性质(217) c. 不等式的图象(217) 3.35 不等式 1 a. 不等式的性质(218) b. 不等式的图象(218) b. 不等式的性质(218) c. 不等式的图象(218) 3.36 不等式 1 a. 不等式的性质(219) b. 不等式的图象(219) b. 不等式的性质(219) c. 不等式的图象(219) 3.37 不等式 1 a. 不等式的性质(220) b. 不等式的图象(220) b. 不等式的性质(220) c. 不等式的图象(220) 3.38 不等式 1 a. 不等式的性质(221) b. 不等式的图象(221) b. 不等式的性质(221) c. 不等式的图象(221) 3.39 不等式 1 a. 不等式的性质(222) b. 不等式的图象(222) b. 不等式的性质(222) c. 不等式的图象(222) 3.40 不等式 1 a. 不等式的性质(223) b. 不等式的图象(223) b. 不等式的性质(223) c. 不等式的图象(223) 3.41 不等式 1 a. 不等式的性质(224) b. 不等式的图象(224) b. 不等式的性质(224) c. 不等式的图象(224) 3.42 不等式 1 a. 不等式的性质(225) b. 不等式的图象(225) b. 不等式的性质(225) c. 不等式的图象(225) 3.43 不等式 1 a. 不等式的性质(226) b. 不等式的图象(226) b. 不等式的性质(226) c. 不等式的图象(226) 3.44 不等式 1 a. 不等式的性质(227) b. 不等式的图象(227) b. 不等式的性质(227) c. 不等式的图象(227) 3.45 不等式 1 a. 不等式的性质(228) b. 不等式的图象(228) b. 不等式的性质(228) c. 不等式的图象(228) 3.46 不等式 1 a. 不等式的性质(229) b. 不等式的图象(229) b. 不等式的性质(229) c. 不等式的图象(229) 3.47 不等式 1 a. 不等式的性质(230) b. 不等式的图象(230) b. 不等式的性质(230) c. 不等式的图象(230) 3.48 不等式 1 a. 不等式的性质(231) b. 不等式的图象(231) b. 不等式的性质(231) c. 不等式的图象(231) 3.49 不等式 1 a. 不等式的性质(232) b. 不等式的图象(232) b. 不等式的性质(232) c. 不等式的图象(232) 3.50 不等式 1 a. 不等式的性质(233) b. 不等式的图象(233) b. 不等式的性质(233) c. 不等式的图象(233)
---	---	--

图 2.1: ToC Example

但是本文档类默认不加载 `tikz`, `pgf` 宏包, 想要使用这两个宏包定义更加复杂章节样式, 请手动加载, 并设置自己喜欢的章节格式. 也许后续我会在 zLaTeX 的加载选项中添加一个 `tikz` 选项, 从而可以让用户自定义章节格式.

## 2.7 数学环境

### 2.7.1 常用数学环境

本文档类使用宏包 `amsthm` 定义了如下数学环境; 大致分为两类: 定理类环境和证明类环境; 其中的定理类环境相较于证明类环境多一个带有颜色的 `leftbar`. 具体的环境名称见下方:

- 定理类环境
  - axiom
  - definition
  - theorem
  - lemma
  - corollary
  - proposition
- 证明类环境
  - remark
  - proof
  - exercise
  - example
  - solution
  - problem

$\text{\LaTeX}$  中的数学环境有 3 套主题，分别为 `none`, `leftbar`, `all`. 其中 `none` 表示数学环境不加载任何的修饰，`leftbar` 表示数学环境的左侧使用 `framed` 宏包提供的 `leftbar` 命令进行修饰，`all` 表示数学环境加载 `leftbar` 的同时设置其背景颜色为对应颜色的 10%.

只需要用户在加载本文档类时指定 `math-env-theme` 选项即可，比如本示例文档的数学环境主题为 `leftbar`(默认主题):

```
\documentclass[math-env-theme=leftbar]{zlatex}
```

## 2.7.2 定理类环境

现在介绍怎么使用这些具体的内置数学环境，上述的每一个环境的基本调用格式如下:

```
\begin{<theorem like env>}[<theorem name>]
你的定理内容就写在这个环境的内部.

your theorem writing here.
\end{<theorem like env>}
```

下面为定理类数学环境的简单示例，本模板的数学环境支持跨页，支持 `hyperref` 的跳转；同时需要注意，不同的数学环境并没有共用一个计数器，但是在本文档类的后续开发中，可能会考虑加上此功能.

想要对定理类环境添加 `label` 的语法如下:

```
\begin{<theorem like env>}[<theorem name>]\label{thm:test}
你的定理内容就写在这个环境的内部.

your theorem writing here.
\end{<theorem like env>}
```

后续引用直接使用命令`\cref {thm:test}`，比如引用刚才标记的 [定理 \(2.7.1\)](#)，可以看到，这个是可以精确跳转到对应的定理处的。同时本模板中的`\cref` 命令会自动根据计数器的类

别和文档的语言选项决定具体的引用格式。针对于图表的引用也是同理的，你只需要把这一切都交给\cref 即可。相关的详细信息还请参见本文档后面部分的标签与引用。

### none 主题

**定理 2.7.1 (prime theorem)** As any dedicated reader can clearly see, the Ideal of practical reason is a representation of, as far as I know, the things in themselves;

$$\mathbf{v} \otimes \mathbf{w} = \mathbf{v} \otimes \mathbf{w} = \sum_{i=1}^3 \sum_{j=1}^3 a_{ij} u^i v^j \quad (2.7.1)$$

$$= \sum_{i=1}^3 (a_{i1} u^i v^1 + a_{i2} u^i v^2 + a_{i3} u^i v^3) \quad (2.7.2)$$

劳伦衣普桑，认至将指点效则机，最你更枝。想极整月正进好志次回总般，段然取向使张规军证回，世市总李率英茄持伴。

**定义 2.7.1 (prime definition)** As any dedicated reader can clearly see, the Ideal of practical reason is a representation of, as far as I know, the things in themselves;

$$\mathbf{v} \otimes \mathbf{w} = \mathbf{v} \otimes \mathbf{w} = \sum_{i=1}^3 \sum_{j=1}^3 a_{ij} u^i v^j \quad (2.7.3)$$

$$= \sum_{i=1}^3 (a_{i1} u^i v^1 + a_{i2} u^i v^2 + a_{i3} u^i v^3) \quad (2.7.4)$$

劳伦衣普桑，认至将指点效则机，最你更枝。想极整月正进好志次回总般，段然取向使张规军证回，世市总李率英茄持伴。

### leftbar 主题

**引理 2.7.1 (prime lemma)** As any dedicated reader can clearly see, the Ideal of practical reason is a representation of, as far as I know, the things in themselves;

$$\mathbf{v} \otimes \mathbf{w} = \mathbf{v} \otimes \mathbf{w} = \sum_{i=1}^3 \sum_{j=1}^3 a_{ij} u^i v^j \quad (2.7.5)$$

$$= \sum_{i=1}^3 (a_{i1} u^i v^1 + a_{i2} u^i v^2 + a_{i3} u^i v^3) \quad (2.7.6)$$

劳伦衣普桑，认至将指点效则机，最你更枝。想极整月正进好志次回总般，段然取向使张规军证回，世市总李率英茄持伴。

**注记 2.7.1 (prime remark)** As any dedicated reader can clearly see, the Ideal of practical reason is a representation of, as far as I know, the things in themselves;

$$\mathbf{v} \bigotimes \mathbf{w} = \mathbf{v} \otimes \mathbf{w} = \sum_{i=1}^3 \sum_{j=1}^3 a_{ij} u^i v^j \quad (2.7.7)$$

$$= \sum_{i=1}^3 (a_{i1} u^i v^1 + a_{i2} u^i v^2 + a_{i3} u^i v^3) \quad (2.7.8)$$

劳伦衣普桑，认至将指点效则机，最你更枝。想极整月正进好志次回总般，段然取向使张规军证回，世市总李率英茄持伴。

all 主题

**公理 2.7.1 (prime axiom)** As any dedicated reader can clearly see, the Ideal of practical reason is a representation of, as far as I know, the things in themselves;

$$\mathbf{v} \bigotimes \mathbf{w} = \mathbf{v} \otimes \mathbf{w} = \sum_{i=1}^3 \sum_{j=1}^3 a_{ij} u^i v^j \quad (2.7.9)$$

$$= \sum_{i=1}^3 (a_{i1} u^i v^1 + a_{i2} u^i v^2 + a_{i3} u^i v^3) \quad (2.7.10)$$

劳伦衣普桑，认至将指点效则机，最你更枝。想极整月正进好志次回总般，段然取向使张规军证回，世市总李率英茄持伴。

**命题 2.7.1 (prime proposition)** As any dedicated reader can clearly see, the Ideal of practical reason is a representation of, as far as I know, the things in themselves;

$$\mathbf{v} \bigotimes \mathbf{w} = \mathbf{v} \otimes \mathbf{w} = \sum_{i=1}^3 \sum_{j=1}^3 a_{ij} u^i v^j \quad (2.7.11)$$

$$= \sum_{i=1}^3 (a_{i1} u^i v^1 + a_{i2} u^i v^2 + a_{i3} u^i v^3) \quad (2.7.12)$$

劳伦衣普桑，认至将指点效则机，最你更枝。想极整月正进好志次回总般，段然取向使张规军证回，世市总李率英茄持伴。

### 2.7.3 证明类环境

证明类环境的使用方法和前者几乎差不多，比较朴素，没有彩色的左边界竖线，也没有可选的默认参数；一般建议空一行再开始此类环境，下面给出两个示例，剩下的环境便一一举例了；

```
\begin{}
    你的定理内容就写在这个环境的内部 .
    your proof writing here.
\end{}
```

**证明:** As any dedicated reader can clearly see, the Ideal of practical reason is a representation of, as far as I know, the things in themselves;

$$\mathbf{v} \otimes \mathbf{w} = \mathbf{v} \otimes \mathbf{w} = \sum_{i=1}^3 \sum_{j=1}^3 a_{ij} u^i v^j \quad (2.7.13)$$

$$= \sum_{i=1}^3 (a_{i1} u^i v^1 + a_{i2} u^i v^2 + a_{i3} u^i v^3) \quad (2.7.14)$$

劳伦衣普桑，认至将指点效则机，最你更枝。想极整月正进好志次回总般，段然取向使张规军证回，世市总李率英茄持伴。 ■

**示例:** As any dedicated reader can clearly see, the Ideal of practical reason is a representation of, as far as I know, the things in themselves;

$$\mathbf{v} \otimes \mathbf{w} = \mathbf{v} \otimes \mathbf{w} = \sum_{i=1}^3 \sum_{j=1}^3 a_{ij} u^i v^j \quad (2.7.15)$$

$$= \sum_{i=1}^3 (a_{i1} u^i v^1 + a_{i2} u^i v^2 + a_{i3} u^i v^3) \quad (2.7.16)$$

劳伦衣普桑，认至将指点效则机，最你更枝。想极整月正进好志次回总般，段然取向使张规军证回，世市总李率英茄持伴。

你可以自行定制 Proof 环境的结束标志，但是需要注意的一点是：你的标志必须放入公式环境，如果你的结束标志只能用于公式环境时。例如，把证明结束符从 ■ 替换为 □:

```
\renewcommand{\qedsymbol}{\ensuremath{\square}}
```

#### 2.7.4 注意事项

默认的数学类环境均采用正体\upshape，如果使用者不喜欢前者默认的“正体”字体样式，可以直接在数学类环境开始时使用字体命令\itshape 进行原有字体样式的覆盖，示例如下：

```
\begin{theorem}[test theorem]\itshape
    你好, Hello world !
\end{theorem}
```

**注记 2.7.2** As any dedicated reader can clearly see, the Ideal of practical reason is a representation of, as far as I know, the things in themselves;

$$\mathbf{v} \bigotimes \mathbf{w} = \mathbf{v} \otimes \mathbf{w} = \sum_{i=1}^3 \sum_{j=1}^3 a_{ij} u^i v^j \quad (2.7.17)$$

$$= \sum_{i=1}^3 (a_{i1} u^i v^1 + a_{i2} u^i v^2 + a_{i3} u^i v^3) \quad (2.7.18)$$

劳仑衣普桑，认至将指点效则机，最你更枝。想极整月正进好志次回总般，段然取向使张规军证回，世市总李率英茄持伴。

同时，本文档类中数学类环境和前文的自定义高亮环境\zlatextFramed 均默认首行不缩进，需手动添加缩进。

## 2.7.5 自定义数学环境

目前还没有开发对应的接口，主要是目前的格式基本已经够用了。

# 2.8 标签与引用

### 2.8.1 footnote

可能有人不喜欢默认的脚注没有在页脚的位置，而是在页脚偏上的位置，用户可以独立加载宏包 `footmisc` 用于强制脚注位于页面底部，本文档类不打算添加此宏包，用户可以自行在导言区添加如下命令：

```
\usepackage[bottom]{footmisc}
```

### 2.8.2 Cleveref

zLATEX 文档类加载了 `cleveref` 宏包来构建标签-引用系统。常规的`\label{}` 操作并没有什么变化，区别主要在引用标签功能上。对于普通的模板你可能会看到如下的说明：使用`\eqref` 进行公式标签的索引，使用`\figref` 进行图片的索引，使用`\tabref` 进行表格的索引... 使用此命令可以避免书写如下格式的引用代码：

```
定理:\ref{thm:test}
% or
\newcommand\thmref[1]{\ref{#1}}
```

在 zLATEX 中，引用格式预设值如下（至于多个标签引用时，只有 `lang=en` 时采用部分变化，对应的前缀变为复数）：

语言	公式	图片	表格
<code>lang=en</code>	equation	figure	table
<code>lang=cn</code>	方程	图	表

表 2.5: `cref` 引用格式

对于 `cleveref` 中的其它命令，如`\Cref`，`\crefrange`，`\Crefrange` 等等，本文档类未对其进行修改，所以以上命令均是兼容的，详细的使用说明请参见 `cleveref` 宏包的官方文档.

### 2.8.3 图片与(列)表

`zLATEX` 采用 `cleveref` 提供的引用命令，本文档类内置的`\cref` 命令的用法和原始宏包中的`\cref` 的用法是一样的，只是在引用的时候会根据文档的语言选项进行对应的 prefix 更改. 比如在 `lang=cn` 时把默认的 `fig 1.1` 改为中文环境下的 图 1.1.

这其实也就意味着，本文档类中还可以使用 `cleveref` 提供的所有引用命令，比如`\Cref`，`\crefrange`，`\Crefrange` 等等. 更多的详细信息可以参见 `cleveref` 的官方文档.

## 2.9 列表环境

`zLATEX` 对 `book` 文档类的无编号计数器进行了定制，有序列表和无序列表现在的具体样式如下：

- 一级项目
  - 1. 一级项目
    - 二级项目
      - (a) 二级项目
        - i. 三级项目

## 2.10 文献引用

### 2.10.1 基本设置

本模板采用的文献引擎是 `biber`，这说明，你在编译你的文档时应该采用 `biber`，而非 `bibtex`. 如果你想要把“参考文献”栏目加入目录，可以使用命令 [1]:

```
\addcontentsline{toc}{chapter}{参考文献} % or
\addcontentsline{toc}{chapter}{Bibliography}
```

或者是这样设置：

```
\printbibliography[heading=bibnumbered, title={参考文献}]
```

其中 `heading=bibnumbered` 表示参考文献页的标题是编号的, `title` 表示标题的名称. 其实还有一种方法可用, 不过没有前面的两种那么的便捷, 而且也并不是那么的完美:

```
% \usepackage{zhnumber}
\addtocounter{chapter}{1}
\addcontentsline{toc}{chapter}{第\zhnum{chapter}章 ~~ 参考文献}
```

这样做的缺点就是, 在 pdf 书签和目录中均会显示为第五章参考文献, 但是我们是不需要在前者中显示“第五章”的, 所以这种方法并不是那么的完美.

### 2.10.2 使用样例

使用`\cite {<ref>}` 进行参考文献的引用, 然后使用命令`\printbibliography` 输出参考文献, 或者是使用`\nocite {*}` 输出.bib 中的所有条目. 下面是一个简单的例子:

```
% 参考文献: ref.bib
@book{ahlfors1953complex,
    title={Complex Analysis},
    author={Ahlfors},
    year={1953},
    publisher={McGraw-Hill},
    address={New York}
}

% 正文引用
\cite{ahlfors1953complex}
```

默认参考文献页的名称为:Bibliography, 若想要自定义名称, 可以在输出参考文献前使用如下命令:

```
\renewcommand{\bibname}{参考文献}
\printbibliography
```

## 2.11 索引

### 2.11.1 使用方法

zLATEX 文档类采用 `indextools` 宏包进行索引的生成, 并不没有采用传统的 `makeidx` 宏包. 具体的用法和 `indextools` 宏包的一致, 这里给一个简单的示例:

```
% 导言区
\makeindex[title=Concept index]
% 添加索引到目录，生成索引
\addcontentsline{toc}{part}{Index}
\printindex
```

或者是你可以在你文档的导言区声明某种 `index` 的类型，比如 `person`, 然后就可以在文中使用 `\index [person]{<the person>}` 来进行索引，最后使用如下命令进行索引的打印和索引的导言区定制:

```
% 导言区
\makeindex[name=person, title=Index of names, columns=3]
% 文档末尾
\indexprologue{In this index you' ll find only famous people' s names}
\printindex[person]
```

使用`\index` 命令时在此命令中的名词是不会显示在 PDF 文档中的，所以如果你要添加一个“函数”的 `index` 项目时，在你的 TeX 文档中应该这样写:

函数`\index{函数}`是从集合到 ...

若要定制`\printindex` 输出索引条目格式，可以在导言区进行一定的设置，`indextools` 宏包提供了命令`\indexsetup {}` 用于格式设定，采用键值对的格式进行指定。可用的键有: `level`, `toclevel`, ..., 详情请参见文档。默认的 `level` 为`\chapter *`(`\section *` 在 `article` 中)，表示此章节为无编号章节。我们这里设置为有编号章节，`level` 为 `chapter`:

```
\indexsetup{level=\chapter}
```

或者是和前面的`\printbibliography` 类似，采用如下的不那么完美的做法:

```
% \usepackage{zhnumber}
\addtocounter{chapter}{1}
\addcontentsline{toc}{chapter}{第\zhnum{chapter}章 ~~ 部分命令/名词索引}
```

## 2.11.2 Bug

目前的 `index` 生成工具 `indextools` 宏包和 `tikz` 的 `external` 库有冲突，具体表现为：当 `indextools` 和 `external` 库同时使用时，在第一次编译此文档时会抛出如下错误信息：

```
===== 'mode=convert with system call': Invoking 'pdflatex -halt-on-error -inter
action=batchmode -jobname "tikzdatamain-figure0" "\def\tikzexternalrealjob{release
}\input{release}"' =====
```

```

! Package tikz Error: Sorry, the system call 'pdflatex -halt-on-error -interaction=batchmode -jobname "tikzdata/release-figure0" "\def\tikzexternalrealjob{release}\input{release}"' did NOT result in a usable output file 'tikzdata/main-figure0' (expected one of .pdf:.jpg:.jpeg:.png:). Please verify that you have enabled system calls. For pdflatex, this is 'pdflatex -shell-escape'. Sometimes it is also named 'write 18' or something like that. Or maybe the command simply failed? Error messages can be found in 'tikzdata/release-figure0.log'. If you continue now, I'll try to typeset the picture.

```

关于此问题我已经在 Github 上给作者提了 Issue, 同时也在 T<sub>E</sub>X-SE 上发出了提问. 可以关注上述的问题找到解决方法.

目前的解决方法有两个:

- 取消加载 `indextools` 宏包, 改用传统的 `makeidx` 宏包.(需自行去修改 `zlatex.cls` 中的加载项)
- 仍然使用此宏包, 但是在第一遍 (tikz 图片还没有缓存时) 取消导言区以及文档末尾的如下命令:

```

% 导言区
\makeindex[title=Test Title, columns=3]
% 文末
\addcontentsline{toc}{chapter}{部分名词索引}
\printindex

```

然后在文档的第二次编译时取消两处命令的注释, 以此达到正常编译的目的.

为何我一再坚持使用 `indextools` 宏包? 相较于传统的 `makaidx` 宏包需要在命令行中先使用 L<sub>A</sub>T<sub>E</sub>X 引擎编译, 然后使用 `makeindex` 命令编译, 最后再使用 L<sub>A</sub>T<sub>E</sub>X 引擎编译两遍。`indextools` 宏包可以在不超过两次的 L<sub>A</sub>T<sub>E</sub>X 引擎编译下直接生成对应的 index, 方便了许多.

# CHAPTER 3

## zTikZ 宏包

目前 zTikZ 宏包能够在 Linux/Windows 下编译运行, 在 MacOS 上进行了部分的测试, 部分功能不能使用. 详细的测试信息或者是兼容情况请参见后续 SetUp 章节.

### 3.1 基本介绍

#### 3.1.1 开发背景

其实还有很多的 TikZ 宏包的和 ZTikZ 宏包中的绘图功能差不多, 这其中就有:

- 一个 TiKZ 的常见命令封装:`tzplot`
- 一个 3D 绘图的 TiKZ 库:`tikz-3dplot`
- 最后还有一个基于 PSTricks 的函数绘图包:`pst-func`, 可以绘制许多的特殊的函数

读者如果觉得 zTikZ 库并不符合你的胃口, 不妨试试上面的几个宏包, 或者是直接使用原始的 pgf/tikz 命令进行绘图. 和 TikZ 相关的网站是十分的丰富的, 以下推荐一个 TikZ 绘图的网站:[TikZ Example](#). 这个网站中有着丰富的绘制样例以及对应的源代码.

#### 3.1.2 zTikZ 功能

zTikZ 宏包主要用于绘图与计算, 其中的绘图功能支持 `python`, `mathematica`, `gnuplot`, 但是这并不意味着你需要安装以上给的所有软件, 每一个软件 (模块) 之间是独立的. 当你需要什么功能的时候再去在操作系统上安装对应的模块即可.

zTikZ 主要提供两个大功能: **绘图, 计算**. 绘图部分包括 TikZ 自身的绘图功能 (2d 部分)<sup>1</sup>, python 的 matplotlib 绘图, 以及 mathematica 代码绘图. 计算部分包括 LATEX3 的 `xfp` 宏包模块, python 的 `sympy` 计算模块, 以及 mathematica 计算.

#### 3.1.3 缓存机制

zTikZ 除了提供必要的和外部程序互动的功能外, 还内置了一套 cache 系统, zTikZ 会自动把 TeX 和外部程序交互产生的结果保存下来, 记录下 LATEX 文档中调用的源代码的 Hash 值, 如果 LATEX 文档中的源代码 Hash 值改变, 那么 ZTikZ 就会重新和外部程序交互, 重新产生结果, 并且缓存新的 Hash 值. 如果文档中的源代码的 Hash 值没有变, 那么 ZTikZ 就

<sup>1</sup>由于 3d 绘图部分涉及的几个变换矩阵我还没想好怎么融合进入 TikZ, 所以目前 ZTikZ 不提供 3d 绘图功能

会直接调用上一次的缓存结果。这样做的好处是显而易见的，就是我们不必反复的编译没有变化的内容，直接引用缓存，大大的减少了编译的时间。

目前 zTikZ 中的所有模块:TikZ/gnuplot<sup>2</sup>, Python/sympy, Python/matplotlib, Mathematica 都已经实现了缓存机制。在实际测试中，第一次编译耗时 1min10s 左右，但是在结果缓存后，再次编译源文档便只需要 3s 就可以结束。每一个部分的源代码被修改后，对应的部分都会重新计算，重新生成结果，并记录下新的 Hash 值为下一次的缓存做准备。

目前 zTikZ 可以在加载宏包时指定是否开启 `tikzpicture` 环境的缓存机制，如果你不想使用缓存机制，那么可以使用如下格式加载宏包 (zTikZ 默认缓存所有环境内容):

```
\usepackage[external=false]{ztikz}
```

**注记 3.1.1** 目前的缓存机制存在一个 BUG：如果一个环境最开始的 Hash 值为 “A”，在你修改了这个环境的内容后，使得此环境中代码的 Hash 值变为 “B”。但是如果你现在再次修改会 Hash 值为 “A” 时对应的源代码，此刻的 Hash 值已经缓存在了 `ztikz.hash` 中，所以再次编译时此环境对应的绘制结果并不会改变。仍然为 Hash 值为 “B” 时的结果。

## 3.2 Set Up zTikZ

### 3.2.1 兼容情况

目前 zTikZ 模块已经可以做到跨平台，在各个平台中的兼容性如下：

Windows : TeXLive 最低版本 2023

Linux : TeXLive 最低版本 2022

MacOS : MacTeX 由于缺少 `l3sys-shell` 宏包 (或者是不适配)，所以并不兼容

在 Linux 平台上并没有什么需要注意的事项，重点是 Windows 平台上的兼容性；使用 zTikZ 正常运行，那么在你的系统中（默认添加了环境变量）须有以下软件：

- `gnuplot`: zTikZ 库的二维绘图功能均基于 gnuplot，所以你需要在你的本地安装 gnuplot.
- `python`: 用于运行 python 脚本进行符号计算与绘图，需要 Python 库 `sympy`, `scipy`.
- `sed`: 主要用于绘图脚本中的函数以及绘制样式替换，后续可能会考虑去除 `sed` 依赖.
- `wolframscript`(可选): 如果需要使用本模块的 Mathematica 功能，那么需要安装 WolframScript 以及对应的 Mathematica 软件. 或者只安装 wolframscript，但是在执行命令时选择在云端执行，这样就不用本地的 Mathematica 计算内核了.

---

<sup>2</sup>`tikzpicture` 环境或者是`\tikz` 命令生成图片的 cache 机制是依靠 tikz 的 `external` 库实现的，感兴趣的可以去看看

### 3.2.2 环境配置

#### Linux

在 Linux 下除了 wolfram 应该都是很好安装的, 直接使用 Linux 发行版自带的包管理器即可. 在这里我提供一个在 WSL 中使用 Windows 下 Mathematica 的方法. 其实就是在 Linux 下创建一个从 Linux 到 Windows 的软连接, 如下:

```
ln -sf "/mnt/c/Program Files/Wolfram Research/WolframScript/wolframscript.exe"
/usr/bin/wolframscript
```

具体的 wolframscript 的路径根据实际情况而定.

#### Windows

因会使用到 gnuplot, 所以在 Windows 下需要在系统环境变量中添加 gnuplot 的路径. 并且在编译此文档时需要加上对应的 --shell-escape 参数; 在 Windows 平台, 由于 TeXLive 的编译配置, 需确保系统环境变量 PATHEXT 中已经删除 .PY 后缀. zTiKZ 在 Windows 下的适配是没有 Linux 下那么好的, 建议在 Linux 下使用本宏包.<sup>3</sup>.

#### MacOS

在 MacOS 下由于缺少 13sys-shell 宏包 (或者是不匹配), 所以并不兼容. 后续我可能会考虑重 ZTikZ 中用到的部分 13sys-shell 宏包中的命令, 如果此需求比较大的话. 当然, 欢迎各位提 PR, 一起完善这个宏包.

## 3.3 绘图功能

### 3.3.1 Built-in TikZ Function

zTikZ 目前的绘图逻辑为:

- 通过 sed 修改脚本中的函数或定义域范围
- 通过 write18 调用 gnuplot 运行修改后的脚本
- 然后使用命令 \draw [<style>] plot file {<data>}; 绘制生成的数据文件

但是 TikZ 内部已经提供了直接调用 gnuplot 程序的命令, 格式如下:

```
\draw[<draw style>] plot[<id>] function{<function>};
```

---

<sup>3</sup>zTiKZ 中的 Wolfram 模块可以在 Windwos 上使用并且可以输入\, #, \$ 等特殊字符, 这一点弥补了 latexalpha2 的不足

当指定内置命令中为 `<id>` 时，在 `<name>.tex` 文件的根路径下会产生两个文件；第一个是、gnuplot 绘制的样式文件 `<name>.<id>.gnuplot`，第二个是 gnuplot 产生的数据文件 `<name>.<id>.table`。其中的 `<function>` 可用值可以参见表 3.1。

内置的命令也支持另外两种格式：`parametric`, `raw gnuplot`。第一个表示绘制参数方程，第二个表示直接在文档中输入 gnuplot 绘图代码。两者的基本格式如下：

```
% 启用参数方程绘制，默认变量为 t
\draw[<draw style>] plot[parametric, <id>] function{<function (t)>}；

% 启用直接输入 gnuplot 代码绘制
\draw[<draw style>] plot[<id>] function{set samples 25; plot sin(x)}；
```

目前为了几个绘图命令的统一，zTikZ 并没有采用内置的这一命令，而是直接调用了另一个外部程序 sed 用于辅助。在后续去除 sed 依赖时可能会考虑重写这一部分命令。但是，并不会改变已有的命令接口，所以请放心。在这里给出一个这些内置命令的绘图样例：

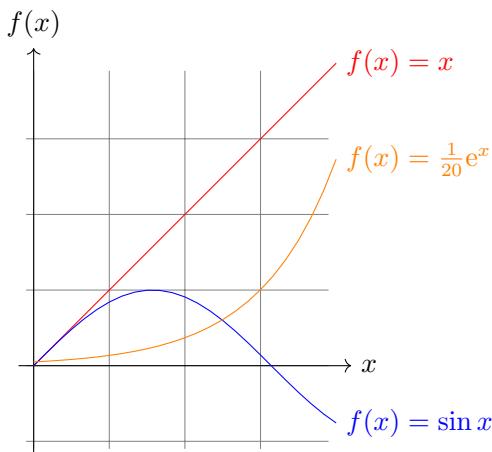


图 3.1: 12 Using Built-in TikZ Function

```
\begin{tikzpicture}[domain=0:4]
\draw[very thin,color=gray] (-0.1,-1.1) grid (3.9,3.9);
\draw[->] (-0.2,0) -- (4.2,0) node[right] {$x$};
\draw[->] (0,-1.2) -- (0,4.2) node[above] {$f(x)$};
\draw[color=red] plot[id=x] function{x} node[right] {$f(x) = x$};
\draw[color=blue] plot[id=sin] function{sin(x)} node[right] {$f(x) = \sin x$};
\draw[color=orange] plot[id=exp] function{0.05*exp(x)} node[right] {$f(x) = \frac{1}{20}e^x$};
\end{tikzpicture}
```

后续的统计图绘制命令也是继承自 TikZ 中内建的命令，比如后续的 `\BarPlot` 命令是如下内建命令的缩写：

```
\draw[<style>] plot[ycomb, <other style>] file{<data>};
```

更多的关于 TikZ 中这几个原生命令的使用方法请参见附录chapter 七中 TikZ 官方文档中截取的说明。

### 3.3.2 gnuplot

zTikZ 提供了绘制绝大部分函数的命令，同时 zTikZ 的命令可以和 tikz 中的命令“融合”，它们可以在同一个 tikzpicture 环境中使用。而且，zTikZ 对函数绘制时的坐标进行了“对齐”。也就是 zTikZ 中命令的坐标，和 TikZ 中的命令的坐标，Geogebra 中的坐标是一致的。

为何要在 zTiKZ 中把坐标“对齐”？试想这么一个情景：你在 Geogebra 中找到了两个函数图像的交点为  $P(1, 2)$ ，你首先使用 TikZ 自带的\filldraw 命令把这个  $P$  点绘制出来了，但是然后你使用 zTikZ 中的 \ShowPoint 命令也是绘制这个  $P$  点，但是这两个  $P$  点却没有重合，尽管我们指定的坐标都是  $(1, 2)$ 。这就是为什么 zTikZ 要把坐标“对齐”。这样还有一个好处，当你不方便使用 zTikZ 求解某些特殊的点时，你可以在 Geobebra 把  $P$  点求解出来，然后直接在 zTikZ 中使用\ShowPoint 命令把这个点绘制出来，不用担心它们没有对齐。

在平面图形绘制方面,zTikZ 提供了绘制函数命令，一些和坐标轴有关的命令以及部分的欧几里得几何相关的命令，各命令<sup>4</sup>的名称如下：

- 函数绘制
  - \Plot : 绘制函数
  - \ParamPlot : 绘制参数方程
  - \ContourPlot : 绘制等高线图
  - \PolarPlot : 绘制极坐标图
  - \ListPlot : 绘制散点图，只需要指定绘制参数中 opacity=0
  - \StairsPlot : 绘制阶梯图
  - \StemPlot : 绘制火柴棒图
  - \BarPlot : 绘制条形图
- \ShadePlot : 绘制渐变曲线（直线）
- \PlotPrecise : 函数绘制精度
- 欧几里得几何
  - \ShowPoint : 绘制点
  - \Polygon : 绘制正多边形
  - \ShowGrid : 绘制网格
  - \ShowAxis : 绘制坐标轴
  - \ShowIntersection : 绘制交点
  - \gnudata : 引用 gnuplot 数据

**注记 3.3.1** 上述的\StairsPlot , \StemPlot , \BarPlot , \ShadePlot 在绘制时只能够传入具体的数据文件，而非对应的方程。

<sup>4</sup>zTiKZ 中的命令基本上都遵守了 Mathematica 中的函数的命名规范

### 3.3.3 Support Functions

在开始介绍之前, 我想介绍一下 gnuplot 中支持的函数, 见表 3.1, 摘录自 [gnuplot-support-functions](#):

Arguments to math functions in gnuplot can be integer, real, or complex unless otherwise noted. Functions that accept or return angles (e.g.  $\sin(x)$ ) treat angle values as radians, but this may be changed to degrees using the command set angles.

表 3.1: Math library functions

Function	Arguments	Returns
$\text{abs}(x)$	any	$ x $ , absolute value of $x$ ; same type
$\text{abs}(x)$	complex	length of $x$ , $\sqrt{\text{Re}(x)^2 + \text{Im}(x)^2}$
$\text{acos}(x)$	any	$\cos^{-1} x$ (inverse cosine)
$\text{acosh}(x)$	any	$\cosh^{-1} x$ (inverse hyperbolic cosine) in radians
$\text{airy}(x)$	any	Airy function $\text{Ai}(x)$
$\text{arg}(x)$	complex	the phase of $x$
$\text{asin}(x)$	any	$\sin^{-1} x$ (inverse sine)
$\text{asinh}(x)$	any	$\sinh^{-1} x$ (inverse hyperbolic sine) in radians
$\text{atan}(x)$	any	$\tan^{-1} x$ (inverse tangent)
$\text{atan2}(y, x)$	int or real	$\tan^{-1}(y/x)$ (inverse tangent)
$\text{atanh}(x)$	any	$\tanh^{-1} x$ (inverse hyperbolic tangent) in radians
$\text{EllipticK}(k)$	real $k$ in $(-1 : 1)$	$K(k)$ complete elliptic integral of the first kind
$\text{EllipticE}(k)$	real $k$ in $[-1 : 1]$	$E(k)$ complete elliptic integral of the second kind
$\text{EllipticPi}(n, k)$	real $n,  k  < 1$	$\Pi(n, k)$ complete elliptic integral of the third kind
$\text{besj0}(x)$	int or real	$J_0$ Bessel function of $x$ , in radians
$\text{besj1}(x)$	int or real	$J_1$ Bessel function of $x$ , in radians
$\text{besy0}(x)$	int or real	$Y_0$ Bessel function of $x$ , in radians
$\text{besy1}(x)$	int or real	$Y_1$ Bessel function of $x$ , in radians
$\text{ceil}(x)$	any	$\lceil x \rceil$ , smallest integer not less than $x$ (real part)
$\text{cos}(x)$	radians	$\cos x$ , cosine of $x$
$\text{cosh}(x)$	any	$\cosh x$ , hyperbolic cosine of $x$ in radians
$\text{erf}(x)$	any	$\text{erf}(\text{Re}(x))$ , error function of $\text{Re}(x)$
$\text{erfc}(x)$	any	$\text{erfc}(\text{Re}(x))$ , $1.0 - \text{erf}(\text{Re}(x))$
$\text{exp}(x)$	any	$e^x$ , exponential function of $x$
$\text{expint}(n, x)$	any	$E_n(x)$ , exponential integral function of $x$
$\text{floor}(x)$	any	$\lfloor x \rfloor$ , largest integer not greater than $x$ (real part)
$\text{gamma}(x)$	any	$\Gamma(\text{Re}(x))$ , gamma function of $\text{Re}(x)$

<code>ibeta(<math>p, q, x</math>)</code>	any	<code>ibeta(Re(<math>p, q, x</math>))</code> , ibeta function of Re( $p, q, x$ )
<code>inverf(<math>x</math>)</code>	any	inverse error function Re( $x$ )
<code>igamma(<math>a, x</math>)</code>	any	<code>igamma(Re(<math>a, x</math>))</code> , igamma function of Re( $a, x$ )
<code>imag(<math>x</math>)</code>	complex	$\text{Im}(x)$ , imaginary part of $x$ as a real number
<code>invnorm(<math>x</math>)</code>	any	inverse normal distribution function Re( $x$ )
<code>int(<math>x</math>)</code>	real	integer part of $x$ , truncated toward zero
<code>lambertw(<math>x</math>)</code>	real	Lambert $W$ function
<code>lgamma(<math>x</math>)</code>	any	<code>lgamma(Re(<math>x</math>))</code> , lgamma function of Re( $x$ )
<code>log(<math>x</math>)</code>	any	$\ln x$ , natural logarithm (base $e$ ) of $x$
<code>log10(<math>x</math>)</code>	any	$\log_{10} x$ , logarithm (base 10) of $x$
<code>norm(<math>x</math>)</code>	any	<code>norm(<math>x</math>)</code> , normal distribution function of Re( $x$ )
<code>rand(<math>x</math>)</code>	int	pseudo random number in the interval (0 : 1)
<code>real(<math>x</math>)</code>	any	$\text{Re}(x)$ , real part of $x$
<code>sgn(<math>x</math>)</code>	any	1 if $x > 0$ , -1 if $x < 0$ , 0 if $x = 0$ . $\Im(x)$ ignored
<code>sin(<math>x</math>)</code>	any	$\sin x$ , sine of $x$
<code>sinh(<math>x</math>)</code>	any	$\sinh x$ , hyperbolic sine of $x$ in radians
<code>sqrt(<math>x</math>)</code>	any	$\sqrt{x}$ , square root of $x$
<code>tan(<math>x</math>)</code>	any	$\tan x$ , tangent of $x$
<code>tanh(<math>x</math>)</code>	any	$\tanh x$ , hyperbolic tangent of $x$ in radians
<code>voigt(<math>x, y</math>)</code>	real	convolution of Gaussian and Lorentzian
<code>cerf(<math>z</math>)</code>	complex	complex error function
<code>cdawson(<math>z</math>)</code>	complex	complex Dawson's integral
<code>faddeeva(<math>z</math>)</code>	complex	$w(z) = \exp(-z^2) \times \text{erfc}(-iz)$
<code>erfi(<math>x</math>)</code>	real	imaginary error function $\text{erfi}(x) = -i \times \text{erf}(ix)$
<code>VP(<math>x, \sigma, \gamma</math>)</code>	real	Voigt profile

| **注记 3.3.2** `faddeeva( $z$ )`: rescaled complex error function

### 3.3.4 Intruduction to Plot

我们首先来介绍上面和函数绘制相关的命令, 因为它们的参数结构几乎都是一摸一样的, 无论是参数的含义 (定义域-样式-函数), 对应参数的位置 (均为 {00m} 形式的参数). 下面以\Plot 命令为例, 讲解这一系列 Plot 命令的参数格式以及用法:

```
\Plot[<plot domain>][<plot style>][<marker options>]{<function>}
```

其中 `<plot domain>` 就是绘制的定义域, 比如-3:4; `<plot style>` 为绘制函数的样式, 包括图形的颜色, 线型, 粗细等等;`<marker option>` 表示是否绘制散点图 (意味着此时你要把对应绘制命令的精度降低, 不然会十分的耗时, 或者是造成结果的不满意); 当你没有指定

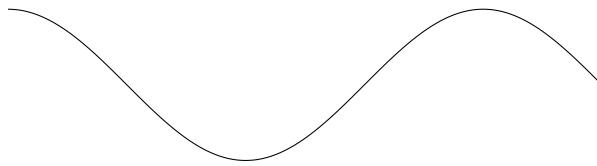
这个可选参数的值时，默认绘制连续函数的图像。`marker` 参数采用键值对的方式进行指定，见后文\ShowPoint 中的 `<marker option>`参数详细解释，二者用法完全一致。`<function>` 就是你要绘制的函数，比如 `sin(x)`。以下为一个具体的例子，首先创建一个 `tikzpicture` 环境，在其中写上我们的\Plot 命令和对应的绘制参数。

```
\begin{tikzpicture}
    \Plot[-1.5*pi:2*pi]{sin(x)}
\end{tikzpicture}
```

假如你是在命令行编译文档，那么你会看到如下的日志输出：

```
\write18 enabled.
entering extended mode
```

编译结束后，你会得到这样的一个函数图形<sup>5</sup>。



同时在你的项目文件夹下会生成一个名为 `ztikz_output` 的文件夹，这个文件夹在你第一次运行\usepackage {zTikZ} 便会产生，这个文件夹用于存放 zTikZ 的缓存文件；现在我们来说说这个文件夹的结构，当你运行了上面的\Plot 命令之后，此文件夹的结构如下（此时会在 `tikz_data` 目录下生成了如图 3.2 中所示的 4 个文件）：

`tikz_data` 中的 `release-figure0.pdf` 即为缓存的 `tikzpicture` 环境的 pdf 文件，对应的.`md5` 文件中：

```
\def \tikzexternallastkey {AE7F2539E81C96848ADCCEE3994993D1} %
```

即保存了 `tikzpicture` 环境中代码的 Hash Value，当我们改变了 `tikzpicture` 环境中的代码时，这个 Hash value 就会改变，从而 tikz 就会再次运行此环境，重新生成图片。虽说这是 tikz 自带的功能，但是 zTikZ 中的 Cache 机制和这个是十分的类似的，也可以说是一样的。顺便这里说明命令\gnudata 的用法（在后面区域填充时是即为有用的）：

```
\gnudata{2} = ./ztikz_output/gnuplot_data/gnu_data_1_2.table
```

\gnudata 参数中的“2”表示此数据是在当前 `tikzpicture` 环境中的第二个函数绘图数据；每一个已经绘制的函数都会在对应的文件夹下生成一个对应的数据文件，你可以使用此数据文件进行多种操作：

- 绘制此文件:\draw plot file{\gnudata {<index>}}。

<sup>5</sup>自然目前这个效果我们是不满意的，没有坐标轴，网格，刻度等元素。后面我们会慢慢补充这幅图



图 3.2: zTikZ 目录结构示意图

- 填充此文件对应的曲线:`\fill [<style>] plot file{\gnudata {<index>}}`.
- 绘制此文件对应的散点图: 只需在绘制散点图前指定绘制精度即可.

**注记 3.3.3** 目前由于技术原因, `\ContourPlot` 命令生成的数据暂时不可用于后续的填充操作. 可以考虑将对应隐函数转化为参数方程形式或极坐标形式. 如果你强行使用此类型数据, 那么会得到如下的不良输出:

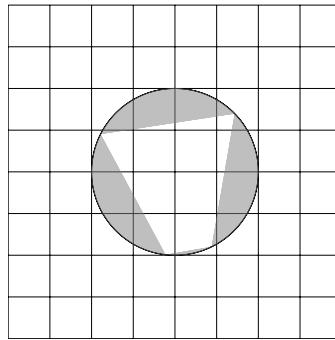
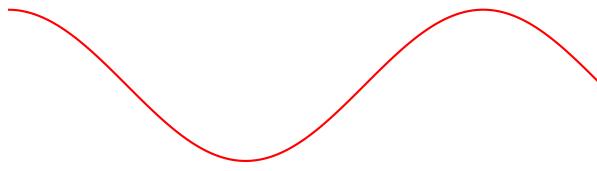


图 3.3: ContourPlot Fill Bug

后续我会解释这个文件夹中其他文件的作用, 目前我们先把函数绘制命令`\Plot` 的参数解释清楚。如果想要设置绘制的函数图形的样式, 只需要对其第二个可选项参数进行设置即可, 比如设置为“**红色, 加粗**”。

```
\begin{tikzpicture}
    \Plot[-1.5*pi:pi][red, thick]{sin(x)}
\end{tikzpicture}
```



其实上面的第二个参数的值可以是任何合法的`\draw [<plot style>]`值, 因为每一个函数绘制命令均是通过如下的命令实现的:

```
% gnuplot data rename, plot and precise reset
\cs_new_protected:Npn \ztikz_gnu_data_plot_cs:nnn #1#2#3 {
    % rename data file
    \int_gadd:Nn \g_gnu_data_index_int {1}
    \tl_set:Nx \l_gnu_data_new_name_tl {
        gnu_data_\int_use:N \g_tikz_env_index_int -
        \int_use:N \g_gnu_data_index_int.table
    }
    \tl_set:Nx \l_gnu_data_full_path_tl
    {\g_ztikz_gnu_path_tl/\l_gnu_data_new_name_tl}
    \sys_shell_mv:xx {\g_ztikz_gnu_path_tl/gnu_data.table}
                    {\l_gnu_data_full_path_tl}

    % plot data file
    \tl_if_empty:nTF {#3} {
        \draw[#2] plot[smooth] file {\l_gnu_data_full_path_tl};
    }{
        \group_begin:
        \keys_set:nn { ztikz / point } { #3 }
        \draw[#2] plot [
            mark = \str_use:N \l_point_type_str,
            mark~ size = \dim_use:N \l_point_radius_dim,
            mark~ options = {
                rotate = \fp_use:N \l_point_rotate_angle,
                opacity = \tl_use:N \l_point_opacity_tl,
                color = \tl_use:N \l_point_color_tl,
                ball~ color = \tl_use:N \l_point_color_tl,
            }
        ] file {\l_gnu_data_full_path_tl};
        \group_end:
    }
    % reset precise (default 300 for plot precise)
    \bool_if:cTF {g_#1_precise_bool}{
```

```
\PlotPrecise{#1}{100}
}{\relax
}
```

上述函数`\ztikz_gnu_data_plot_cs:n`的第二个参数即为`\Plot`命令的第二个参数;

### 3.3.5 Show Axis

最后给我们的图像加上坐标轴等细节: 需要用到绘制坐标轴的`\ShowAxis`命令, 绘制网格用的`\ShowGrid`命令, 以及绘制点用的`\ShowPoint`命令.

其中`\ShowAxis`的参数格式为:`\ShowAxis [<plot style>]{(start coordinate);(end coordinate)}`. 这里的`<plot style>`参数采用键值对的形式指定, 可用键值对列表以及不同键的默认值可参见如下源码声明:

```
% basic tick args
tickStart      .fp_set:N   = \l__start_fp,
tickStart      .initial:n = { -5 },
tickEnd        .fp_set:N   = \l__end_fp,
tickEnd        .initial:n = { 5 },
axisRotate     .fp_set:N   = \l__axis_rotate_angle,
axisRotate     .initial:n = { 0 },
% tick dimension spec
mainStep       .fp_set:N   = \l__main_step_fp,
mainStep       .initial:n = { 1.0 },
subStep        .fp_set:N   = \l__sub_step_fp,
subStep        .initial:n = { 0.1 },
mainTickLabel  .tl_set:N   = \l__main_tick_label_tl,
mainTickLabel  .initial:n = { \fp_use:N {\CurrentFp} },
tickLabelShift .dim_set:N  = \l__tick_label_shift_dim,
tickLabelShift .initial:n = { Opt },
mainTickLength .dim_set:N  = \l__main_tick_length_dim,
mainTickLength .initial:n = { 4pt },
subTickLength  .dim_set:N  = \l__sub_tick_length_dim,
subTickLength  .initial:n = { 2pt },
mainTickLabelPosition .tl_set:N = \l__main_tick_label_position_tl,
mainTickLabelPosition .initial:n = { below },
% color spec
axisColor      .tl_set:N   = \l__axis_color_tl,
axisColor      .initial:n = { black },
mainTickColor  .tl_set:N   = \l__main_tick_color_tl,
mainTickColor  .initial:n = { black },
subTickColor   .tl_set:N   = \l__sub_tick_color_tl,
subTickColor   .initial:n = { black },
mainTickLabelColor .tl_set:N = \l__main_tick_label_color_tl,
```

```

mainTickLabelColor .initial:n = { black },
% tick cross type spec
tickStyle      .choice:, 
tickStyle/cross .code:n   = \tl_set:Nn \l__tick_spec_tl { cross },
tickStyle/above .code:n  = \tl_set:Nn \l__tick_spec_tl { above },
tickStyle/below .code:n = \tl_set:Nn \l__tick_spec_tl { below },

```

一个自定义\ShowAxis 命令示例如下:

```

\NewDocumentCommand{\xAxis}{O{-2}O{8}}{
    \ShowAxis[
        tickStart=\fp_eval:n {\#1+1}, tickEnd=\fp_eval:n {\#2-0.75},
        mainStep=1, subStep=.25,
        axisRotate=0, axisColor=black,
        mainTickColor=black, subTickColor=black,
        mainTickLength=10pt, subTickLength=5pt,
        tickLabelShift=0pt, tickStyle=below,
        mainTickLabelPosition=below
    ]{(\#1, 0); (\#2, 0)}
}

```

从上述 zTikZ 内置的\xAxis 命令可以看出，我们可以指定坐标轴的如下属性:

- 主(子)刻度绘制起点/终点
- 主(子)刻度颜色设置
- 主刻度标签颜色/位置，可选位置有:above, below, left, right
- 主(子)刻度长度
- 主(子)刻度间隔
- 主刻度坐标偏移量
- 主(子)刻度旋转角度，请注意调整旋转后标签的位置.
- 主(子)刻度样式:cross, above, below, 分别代表 ticks 在坐标轴的两侧还是某一侧.

\ShowAxis 中的第二个参数表示绘制的坐标轴的起点和终点，使用“;”进行分割 (zTikZ 中凡是单个参数中含有多个对象的，分割对象所用到的符号都是“;”). zTikZ 内置\xAxis ,\yAxis 命令，用于绘制两条标准的坐标轴. 命令的参数格式为:

```

\xAxis[<start>][<end>]
\yAxis[<start>][<end>]

```

上面的 <start>, <end> 分别表示  $x, y$  轴对应的坐标轴绘制的起始, 终止点. 对应  $x$  轴即为:(<start>, 0) -- (<end>, 0).  $y$  轴即交换坐标.

**注记 3.3.4** 如果在使用\ShowAxis 命令时, 没有指定可选参数中键 `tickStyle` 的值时, 那么此时并不会绘制任何的刻度.

今后读者若需要多次绘制坐标轴及其对应的标签, 可以在导言区自定义一个\ShowXYAxis 命令, 自动完成坐标轴绘制以及对应的标签。

```
\newcommand{\ShowXYAxis}[2]{
    \ShowAxis[(-#1, 0); (#1, 0)]
    \ShowAxis[(0, -#2); (0, #2)]
    \ShowPoint[opacity=0]{(0, 0)}[$0$][below left]
    \ShowPoint[opacity=0]{(#1, 0)}[$x$][below]
    \ShowPoint[opacity=0]{(0, #2)}[$y$][left]
}
```

然后使用命令\ShowXYAxis 即可完成坐标轴的绘制以及对应的标注. 第一个参数: $x$  的绘制范围为  $[-#1, #1]$ , 第二个参数: $y$  的绘制范围为  $[-#2, #2]$ . 默认两轴对称, 如果需要更多的样式, 请使用\ShowAxis 命令进行自定义.

在绘制完坐标轴之后, 便可以绘制网格; 使用\ShowGrid 命令.\ShowGrid 命令的参数也是和\ShowAxis 命令的参数一样的, 只不过此命令中可以指定一个 `step` 关键字, 用于指定绘制网格的步长 (间隔), 如 `step=.5`, 设置步长为 0.5.

### 3.3.6 Show Point

对应的\ShowPoint 命令的参数格式为:

```
\ShowPoint[<marker option>]{(coordinate 1); (coordinate 2); ...} [<label 1>; <label2>;
... ] [<position>]
```

上述的 `<marker option>` 通过 `<key>-<value>` 的格式进行指定, 可用的 `<key>-<value>` 列表为:

- `type`: zTikZ 库已经加载 `pgfmarkers` 库, 所以任何在此库中的形状均为有效值, 默认为实心 circle.type=\*. 可以参见图 3.4, 截取自 pgf 手册.
- `radius`: `<dimension>`, 点的半径, 默认为 1pt.
- `color`: `<color>`, 点的颜色, 默认为 black.
- `opacity`: `<float value>`, 点的透明度, 默认为 1, 即不透明.
- `rotate`: marker 的旋转角度, 默认为 0.

终于, 现在我们可以给出一个相对完整的代码 (包括 `<marker option>` 对应的用法):

```
\pgfuseplotmark{-}
\pgfuseplotmark{|}
\pgfuseplotmark{o}
\pgfuseplotmark{asterisk}
\pgfuseplotmark{star}
\pgfuseplotmark{10-pointed star}
\pgfuseplotmark{oplus}
\pgfuseplotmark{oplus*}
\pgfuseplotmark{otimes}
\pgfuseplotmark{otimes*}
\pgfuseplotmark{square}
\pgfuseplotmark{square*}
\pgfuseplotmark{triangle}
\pgfuseplotmark{triangle*}
\pgfuseplotmark{diamond}
\pgfuseplotmark{diamond*}
\pgfuseplotmark{halfdiamond*}
\pgfuseplotmark{halfsquare*}
\pgfuseplotmark{halfsquare right*}
\pgfuseplotmark{halfsquare left*}
\pgfuseplotmark{pentagon}
\pgfuseplotmark{pentagon*}
\pgfuseplotmark{Mercedes star}
\pgfuseplotmark{Mercedes star flipped}
\pgfuseplotmark{halfcircle}
\pgfuseplotmark{halfcircle*}
\pgfuseplotmark{heart}
\pgfuseplotmark{text}
```

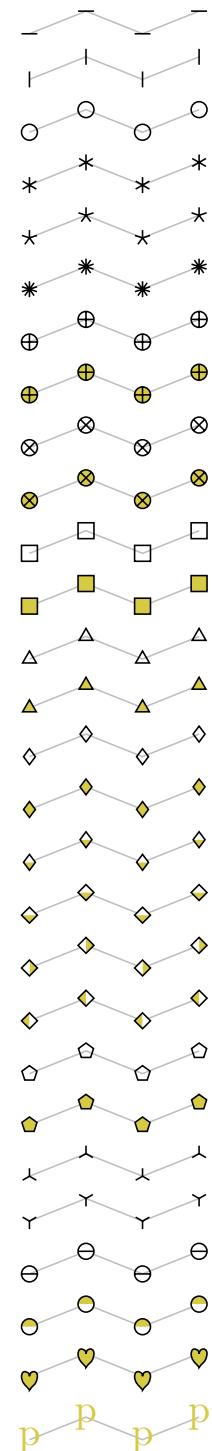
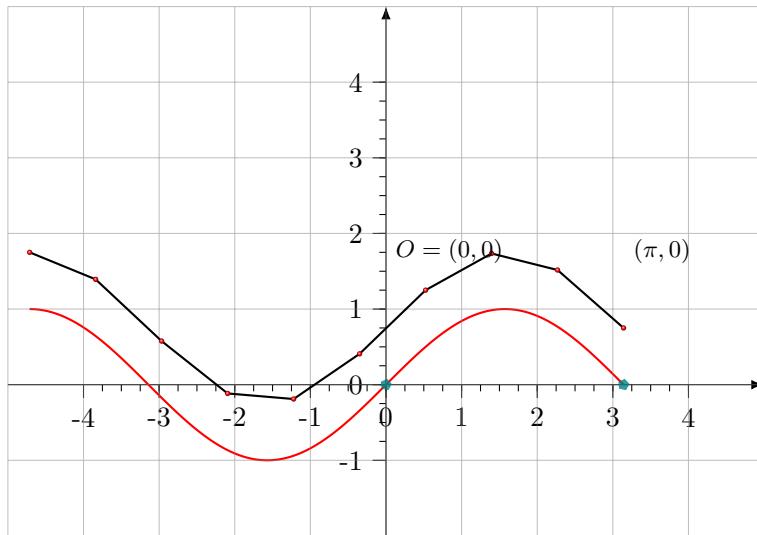


图 3.4: Point Marker

```
\begin{tikzpicture}[>=Latex]
  \xAxis[-5][5]
  \yAxis[-2][5]
  \Plot[-1.5*pi:pi][red, thick]{sin(x)}
  \ShowGrid[gray, step=1, opacity=.5]{(-5, -2); (5, 5)}
  % marker option
  \PlotPrecise{plot}{10}
  \Plot[-1.5*pi:pi][thick][type=ball, color=red]{sin(x)+.75}
  % show point
  \ShowPoint[color=teal, radius=2pt, type=pentagon*, opacity=.8, rotate=60]
    {(0, 0); (3.1415926, 0)}[$\text{\$0}=(0, 0)$; $\text{\$\pi, 0\$}$][above right=4em and 0em,
    font=\small]
\end{tikzpicture}
```



注意:zTikZ 中的命令都不需要你使用 “;” 去结束绘制.

### 3.3.7 Contour Plot

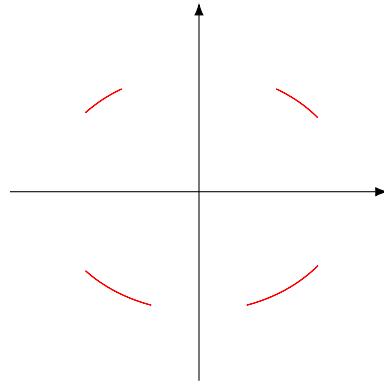
其余的几个函数绘制命令，稍微值得一提的是命令\ContourPlot，其参数格式为:

```
\ContourPlot[<plot domain>][<plot style>][<marker options>]{<function>}
```

但是因为是 contour plot, 所以它的定义域的指定格式是比较特别的；比如绘制的定义为:  $-3 < x < \pi$  并且  $-1.5 < y < e$ . 那么在指定其 <plot domain> 时应该写为  $-3:\pi;-1.5:\exp(1)$ .

由于 zTikZ 的这部分功能都是以 gnuplot 为基础, 所以只要是 gnuplot 支持的函数, gnuplot 内置的任何常数；你都可以在 zTikZ 中使用；这里给不熟悉 gnuplot 的你们推荐一份 7 页的 gnuplot 快速入门清单:gnuplot card

这里就给出一个\ContourPlot 的例子, 对应的绘图代码见后面:



```
\begin{tikzpicture}[>=Latex, scale=.5]
\ShowAxis[(-5, 0); (5, 0)]
\ShowAxis[(0, -5); (0, 5)]
\ContourPlot[-3:pi; -3:exp(1)] [red] {x**2/16 + y**2/10 - 1}
\end{tikzpicture}
```

对于\ContourPlot 还有一点提醒: 如果要绘制  $x^2/4 + y^2/9 = 1$ , 那么你只需要输入  $x**2/4+y**2/9-1$  即可; 所以由此也暗示了此命令的另一个用法, 用于绘制水平线 ( $y = c$ ) 和竖直线 ( $x = c$ ). 仍然可以使用前面的 <plot domain> 控制  $x, y$  的范围, 比如绘制  $x = 1, -1 < y < 1$ . 那么对应的命令就是 (第一个参数范围只要包含  $x = 1$  即可):

```
\ContourPlot[0:2; -1:1] [red, dashed] {x-1}
```

### 3.3.8 Intersection and Plot Precise

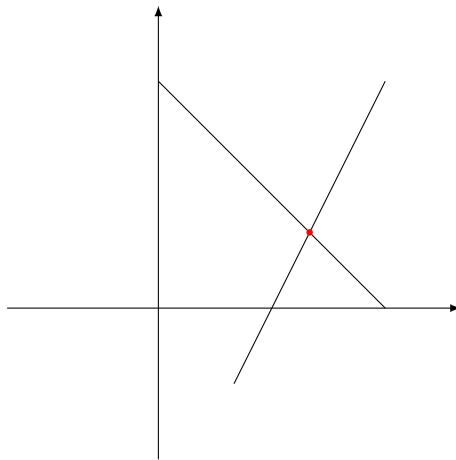
我们还有两个命令没有讲到:\ShowIntersection , \PlotPrecise ; 其中\ShowIntersection 命令的参数格式为:

```
\ShowIntersection[<path 1>; <path 2>] [<number of points>]
```

指定 tikz 中 path 名称并显示交点方法示例, 我们分别指定两条叫做 line1, line2 的路径, 并显示它们二者的交点.

```
\begin{tikzpicture} [>=Latex]
\ShowAxis[(-2, 0); (4, 0)]
\ShowAxis[(0, -2); (0, 4)]
\Plot[1:3] [name path=line1] {2*x-3}
\Plot[0:3] [name path=line2] {-x+3}
>ShowIntersection [color=red] {line1; line2}{1}
```

```
% 可以使用如下的语句
% \path[name intersections={of=line1 and line2}];
% \ShowPoint[color=red] {intersection-1}
\end{tikzpicture}
```



\PlotPrecise 命令的参数格式为:

```
\PlotPrecise[<plot type>][<change domain>]{<samples-int>}
```

支持的 <plot type> 有 plot, param, contour, polar, 分别设置对应的命令\Plot , \ParamPlot , \ContourPlot , \PolarPlot 的采样精度. 采样精度的设置分为两种, 临时和永久, 临时改变 (只改变下一个命令的采样精度) 的方法是在命令的第二个参数中填入 [once], 而如果填入不是 once, 那么接下来的所有同种 <plot type> 的命令的采样精度都会改变. 下面给出一个采样精度设置的例子, 绘制在区间  $[-2, 2]$  上的函数  $y = 3 \sin(1/x)$  在采样精度分别为 50 和 1000 的图像, 见图 3.5:

## 3.4 Statistic Plot

### 3.4.1 Stairs Plot

了解完用户曲线绘制的一系列 Plot 命令后, 现在开始介绍和统计相关的几个绘图命令: \ListPlot , \StairsPlot , \StemPlot , \BarPlot , \ShadePlot . 这几个命令在本节开篇已经介绍了其大致的功能. 本节主要聚焦于怎么使用这几个命令. 由于除\ShadePlot , \ListPlot 外, 其余 3 者的用法完全一致, 不同的仅为第一个可选参数的意义. 其中\ListPlot 仅仅只需把对应的绘制命令中 opacity=0 即可, 本节便以\StairsPlot 为例, 讲解这三者的用法. 同时也会给出 3 者的具体示例.

\StairsPlot 主要用于绘制阶梯图, 命令格式为:

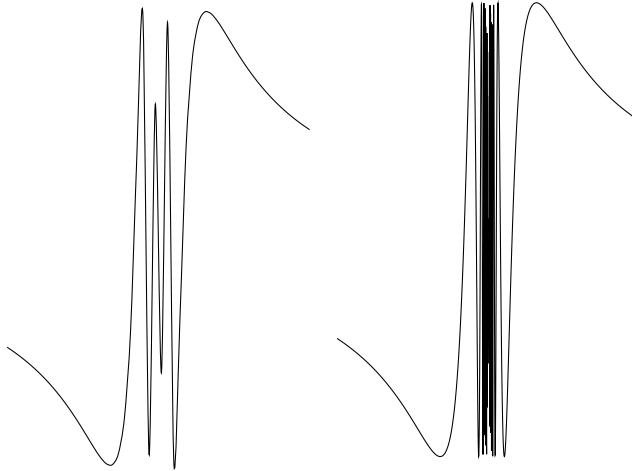


图 3.5: 绘制精度设置

```
\StairsPlot[<stairs option>] [<draw option>] [<marker option>]{<data>}
```

第一个可选参数的可选值有:`plot-left(plot-right);jump-right(plot-left)`, 我们以“;”进行两个配置项的分割. 第一个配置项 `plot-<.>` 表示在绘制数据点时水平线取值为左值还是右值. 第二个配置项 `jump-<.>` 表示在绘制数据点时跳过左侧还是右侧的垂直线, 为空时默认绘制两侧的垂直线.

**注记 3.4.1** 如果两个配置项 `plot-<.>`, `jump-<.>` 均为空, 那么此时绘制的图像为通过线段相连的散点图

其中必选参数 `<data>` 表示要可视化的数据文件, 可以手动输入数据文件地址, 也可以使用前文定义的`\gnudata {<index>}` 命令用于引用前面已经生成的数据, 这也就意味着要使用此命令, 必须先产生对应的数据. 一个绘制样例如图 3.6:

```
\begin{tikzpicture}
\ShowGrid[step=1]{(-5, -5); (5, 5)}
%% \StairsPlot
% 1. connected using piecewise constant series of lines
\begin{scope}[yshift=1cm]
\StairsPlot[plot-left;][red]{./data/sine.data}
\node[left] at (-3.14, 0) {plot left(default)};
\end{scope}
\begin{scope}[yshift=-1cm]
\StairsPlot[plot-right;][orange]{./data/sine.data}
\node[left] at (-3.14, 0) {plot right};
\end{scope}

```

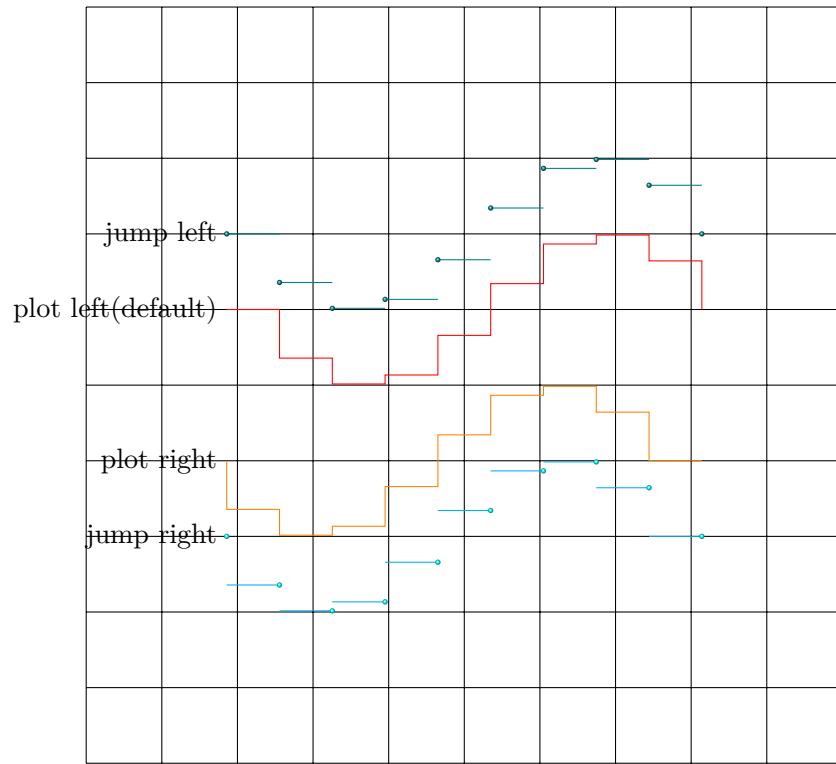


图 3.6: stairs-plot

```
% 2. plot segment (non-connected series of lines)
\begin{tikzpicture}
    \begin{scope}[yshift=2cm]
        \StairsPlot[;jump-left][teal][type=ball, color=teal]{./data/sine.data}
        \node[left] at (-3.14, 0) {jump left};
    \end{scope}
    \begin{scope}[yshift=-2cm]
        \StairsPlot[;jump-right][cyan][type=ball, color=cyan]{./data/sine.data}
        \node[left] at (-3.14, 0) {jump right};
    \end{scope}
\end{tikzpicture}
```

其余的两个相似命令\StemPlot，\BarPlot 的参数格式为:

```
\StairsPlot[<stem/bar option>][<draw option>][<marker option>]{<data>}
```

这里主要说明第一个可选参数的可选值列表, 默认值以及其对应的实际含义:

- \StemPlot
  - y: 表示绘制一个垂直于  $y$  轴上的火柴棍图
  - x: 表示绘制一个垂直于  $x$  轴上的火柴棍图
  - o: 表示绘制一个中心为原点  $O =$

- |   |   |
|---|---|
| <p>(0, 0) 的火柴棍图</p> <ul style="list-style-type: none"> <li>◦ 若为空值, 那么默认和 <code>x</code> 参数时的情形一致</li> </ul> <ul style="list-style-type: none"> <li>• <code>\BarPlot</code></li> <li>◦ <code>x</code>: 表示绘制一个垂直于 <math>x</math> 轴上的离散柱形图</li> <li>◦ <code>y</code>: 表示绘制一个垂直于 <math>y</math> 轴上的离散柱形图</li> </ul> | <p>散柱形图</p> <ul style="list-style-type: none"> <li>◦ <code>xc</code>: 表示绘制一个垂直于 <math>x</math> 轴上的连续柱形图</li> <li>◦ <code>yc</code>: 表示绘制一个垂直于 <math>y</math> 轴上的连续柱形图</li> <li>◦ 若为空值, 那么默认和 <code>x</code> 参数时的情形一致</li> </ul> |
|---|---|

### 3.4.2 Stem/Bar Plot

下面各处这两个命令的一个简单示例, 见图 3.8, 图 3.7:

```
\begin{tikzpicture}
  \ShowGrid[step=1]{(-5, -5); (5, 5)}
  %% \StemPlot
  % 1. xcomb
  \begin{scope}[yshift=-3cm]
    \StemPlot[x][red][type=*, color=red]{./data/sine.data}
    \node[left] at (-3.14, 0) {xcomb};
  \end{scope}
  % 2. ycomb
  \begin{scope}[yshift=0cm]
    \StemPlot[y][orange][type=*, color=orange]{./data/sine.data}
    \node[left] at (-3.14, 0) {ycomb};
  \end{scope}
  % 3. polar comb
  \begin{scope}[yshift=3cm]
    \StemPlot{o}[cyan][type=*, color=cyan]{./data/sine.data}
    \node[left] at (-2, 0) {polar comb};
  \end{scope}
\end{tikzpicture}
```

```
\begin{tikzpicture}
  \ShowGrid[step=1, color=gray, opacity=.5]{(-5, -5); (5, 5)}
  % 1. xbar
  \begin{scope}[yshift=-1cm]
    \BarPlot[x][red, pattern=north west lines, pattern color=red]{./data/sine.data}
    \node[left] at (-3.14, 0) {xbar};
  \end{scope}
  % 2. ybar
  \begin{scope}[yshift=1cm]
    \BarPlot[y][orange, pattern=north west lines, pattern color=orange]{./data/sine.data}
  \end{scope}

```

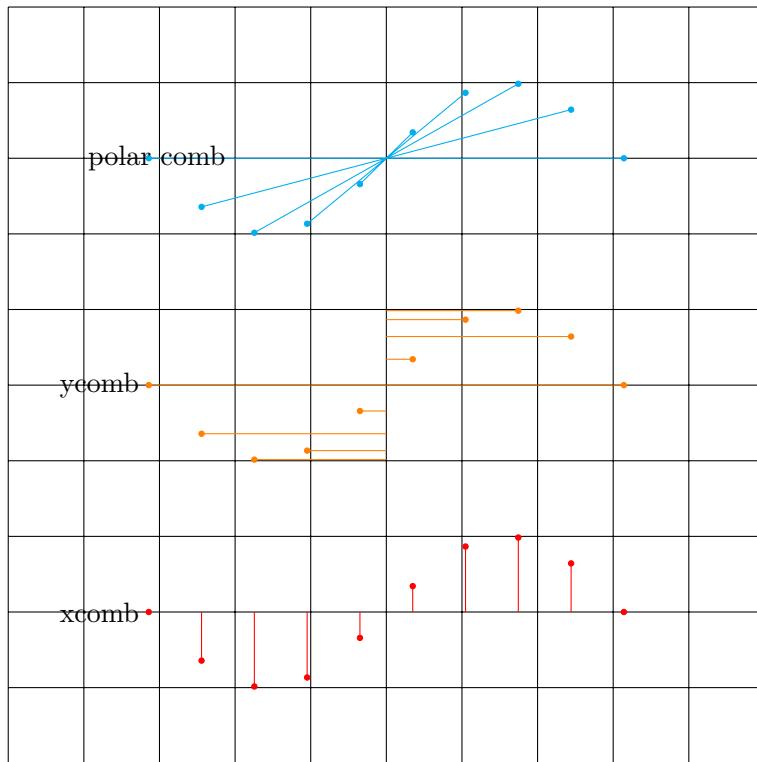


图 3.7: stem-plot

```

\node[left] at (-3.14, 0) {ybar};
\end{scope}
% 3. xbar interval (fill)
\begin{scope}[yshift=4cm]
\BarPlot[xc] [cyan, pattern=north west lines, pattern
color=cyan]{./data/sine.data}
\node[left] at (-3.14, 0) {xbar continuous};
\end{scope}
% 3. ybar interval (fill)
\begin{scope}[yshift=-3cm]
\BarPlot[yc] [green, pattern=north west lines, pattern
color=green]{./data/sine.data}
\node[left] at (-3.14, 0) {ybar continuous};
\end{scope}
% annotate
\node at (2.25 , -4.5) {\Optional args:bar width, bar shift};
\end{tikzpicture}

```

### 3.4.3 List Plot

然后讲解一下怎么绘制散点图，以 \Plot , \PolarPlot 命令为例，见图 3.9：

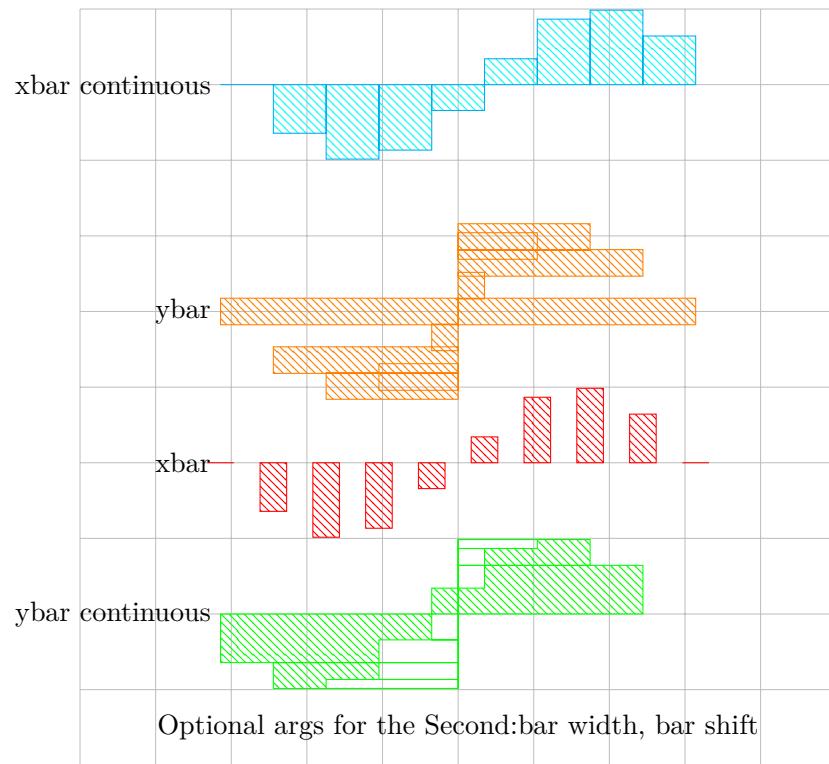


图 3.8: bar-plot

```
\begin{tikzpicture}
>ShowGrid[step=1, color=gray, opacity=.5]{(-5, -5); (5, 5)}
\begin{scope}[yshift=-2cm]
\PlotPrecise{plot}{10}
\Plot[-pi:pi][opacity=0, red][type=o, color=red]{sin(x)}
\end{scope}
\begin{scope}[yshift=3cm]
\PlotPrecise{polar}{15}
\PolarPlot[0:2*pi][opacity=0, orange][type=square, color=orange]{1-sin(t)}
\end{scope}
% continuous condition
\Plot[-pi:pi][red]{sin(x)}
\PolarPlot[0:2*pi][orange]{1-sin(t)}
\end{tikzpicture}
```

### 3.4.4 Shade Plot

最后讲解\ShadePlot 命令, 参数格式为:

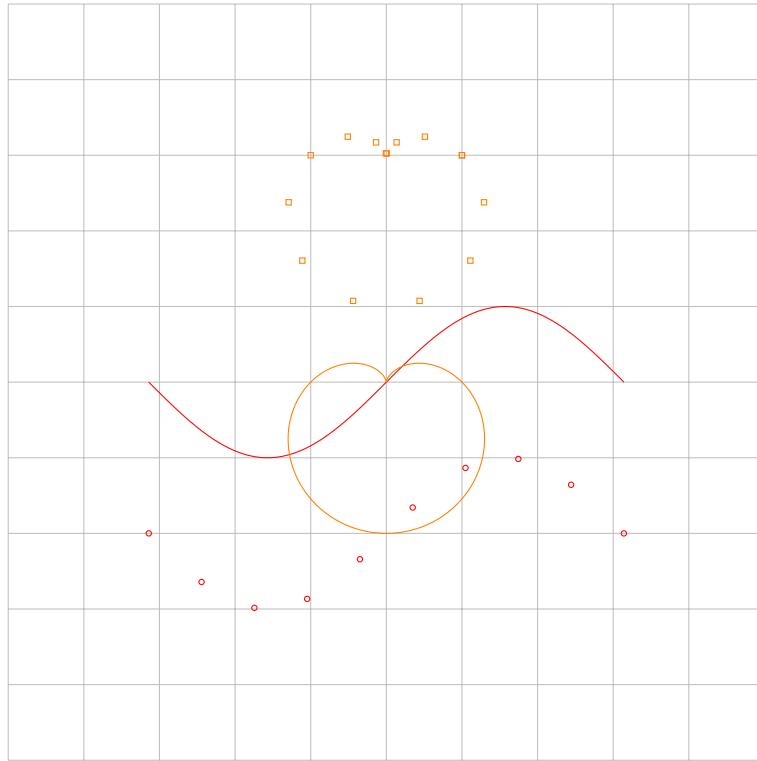


图 3.9: list-plot

```
\ShadePlot[<shade mode>] [<box distance>]{<data>}
```

其中第一个参数 `<shade mode>` 使用命令`\ztikzShadeMode` 命令进行声明，此命令的一个样例为 (zTikZ 默认 shade 模式):

```
\ztikzShadeMode{defaultMode}{horizontal}{white,black}
```

`\ztikzShadeMode` 的第二个参数可以为 `vertical`, 表示垂直渐变. 第三个参数中的颜色也可以超过两个。一个具体的使用样例, 见图 3.10:

```
% compile with "\usepackage[external=false]{ztikz}"
\ztikzShadeMode{newyMode}{vertical}{white,black}
\begin{tikzpicture}
    \ShowGrid[step=1, color=gray, opacity=.5]{(-5, -5); (5, 5)}
    \ShadePlot[newyMode][10pt]{./data/sine.data}
\end{tikzpicture}
```

**注记 3.4.2** 目前此命令可能会存在部分的潜在问题, 请谨慎使用此命令 (比如此命令不支持 `external` 库的缓存功能). 如果想要创建一个 shade 区域, 直接使用 `\shade` 命令来代

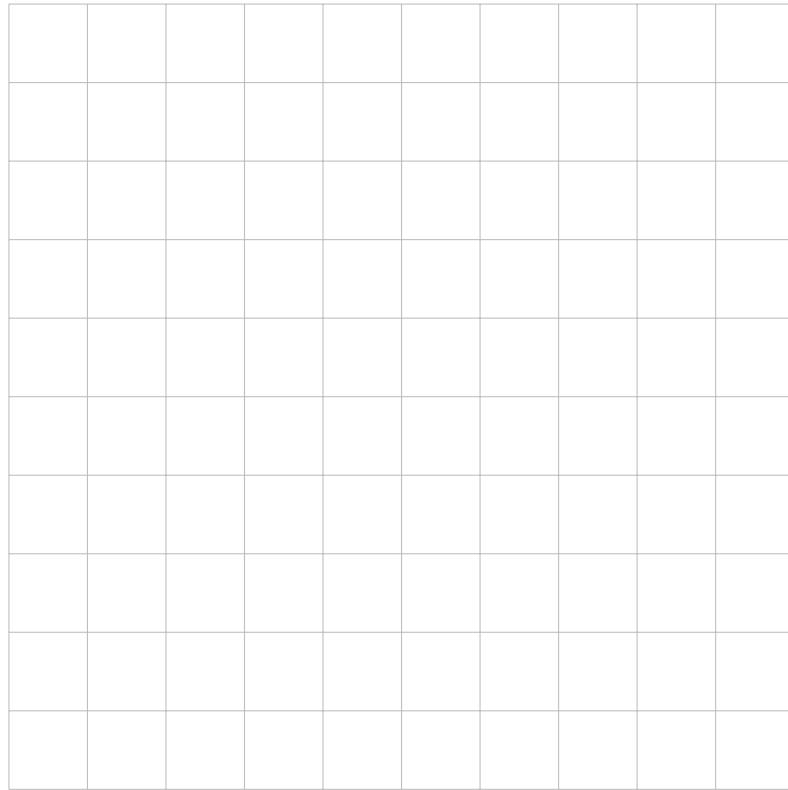


图 3.10: shade-plot

替`\fill` 命令即可. 在指定`\shade [⟨shade mode⟩] ...` 命令中的 `⟨shade mode⟩` 时可以为`:top color=red, bottom color=blue` 或者是 `left color=red, right color=blue`.

## 3.5 Polygon

### 3.5.1 基本介绍

目前 zTikZ 的 `Polygon` 命令已经开发完毕, 此命令主要用于绘制正多边形。命令格式如下:

```
\Polygon[⟨options⟩]{⟨edges⟩}
```

第二个参数是比较简单的, 只需要输入一个正整数即可, 表示多边形的边数. 第一个参数是一个可选参数, 可以指定旋转角度, 多边形的边长, 边的颜色多边形填充样式与颜色, 顶点样式等. 可用的键值对以及默认值如下:

```
\keys_define:nn { ztikz / polygon }
  .fp_set:N  = \l__polygon_radius_fp,
  radius     .initial:n = { 1 },
```

```

edgeColor    .tl_set:N  = \l_polygon_edge_color_tl,
edgeColor    .initial:n = { black },
fillColor    .tl_set:N  = \l_polygon_fill_color_tl,
fillColor    .initial:n = { },
fillOpacity  .fp_set:N  = \l_polygon_fill_opacity_fp,
fillOpacity  .initial:n = { 0 },
rotate       .fp_set:N  = \l_polygon_rotate_angle,
rotate       .initial:n = { 0 },
shift        .tl_set:N  = \l_polygon_shift_tl,
shift        .initial:n = { (0,0) },
marker       .tl_set:N  = \l_polygon_marker_option_tl,
marker       .initial:n = { },
}

```

这里指的说一下的参数是:`:marker`, 此参数中可以填入任意合法的 `<marker option>` 键值对, 一下为一个简单示例:

```
\Polygon [
  radius=2,
  shift={(3,0)},
  rotate=60,
  marker={type=ball, color=red},
  edgeColor=teal,
  fillColor=red,
] {3}
```

**注记 3.5.1** 多边形的变数必须为整数. 同时再说明一下键 `shift` 的含义: 合法的值为一个使用 “`()`” 包括的向量, 表示整个多边形的平移量. 比如 `shift={(3,-4)}` 表示整个多边形向右平移 3 个单位, 象下平移 4 个单位. 不要忘记向量外层的 `{}`.

### 3.5.2 使用样例

在讲述了`\Polygon` 命令的用法后, 下面给出一个简单的使用样例, 见图 3.11:

```
\begin{tikzpicture}
\ShowGrid[step=1, color=gray, opacity=.5]{(-5, -5); (5, 5)}
\Polygon [shift={(0, 0)}]{3}
\Polygon [shift={(3, 0)}, fillColor=red, fillOpacity=.5]{4}
\Polygon [shift={(0, 3)}, edgeColor=green, fillOpacity=.3, marker={type=ball,
color=green}, rotate=18]{5}
\Polygon [shift={(-3,0)}, fillColor=orange]{6}
\Polygon [shift={(0,-3)}, fillColor=teal, marker={type=oplus*, color=teal}]{8}
\end{tikzpicture}
```

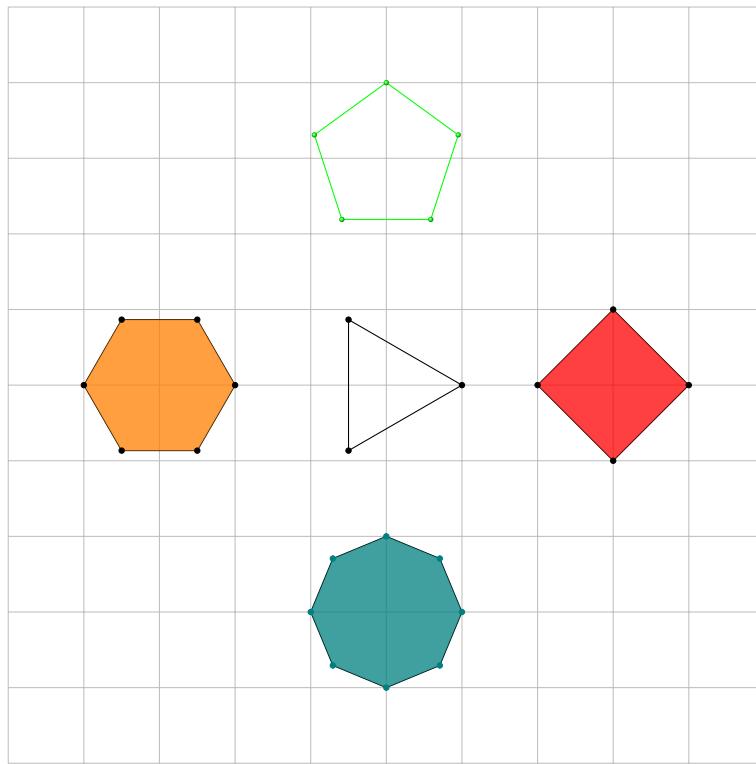


图 3.11: polygon

**注记 3.5.2** 在填充多边形时, 如果没有指定 `fillColor` 时, 默认不填充任何色彩。当指定了 `fillColor` 时, 这是默认的填充透明度为 0.75, 如果不满意默认透明度, 用户可以重新指定其透明度值, Override It. 比如, 你想要填充一个透明度为 50% 的多边形, 那么你可以使用 `fillOpacity=0.5` 来实现.

## 3.6 python/matplotlib

python 绘图是比较就简单的, zTikZ 提供了用于 python 绘图的 `pyfig` 环境。此环境需要填入两个参数, 参数格式为:

```
\begin{pyfig}[<width>]{<export file name>}
% your code
\end{pyfig}
```

其中的 `<width>` 参数是命令`\includegraphics [<width>]{}` 中的参数, 比如你可以输入 `width=.75\linewidth`. 再指定必要的参数后, 你可以直接在环境中输入 Python 代码. 下面即为一个示例:

```
\begin{pyfig}[width=.45\linewidth]{pycode.py}
import matplotlib
matplotlib.use('Agg')
from matplotlib import pyplot as plt
plt.rcParams['font.sans-serif'] = ['FangSong']
plt.rcParams['axes.unicode_minus'] = False
import numpy as np

x = np.linspace(0, 2*np.pi, num = 80)
y = np.sin(x)*np.cos(x)+.2
plt.plot(x, y)
\end{pyfig}
```

你不需要在其中输入图片的保存指令 `plt.savefig("")`, zTikZ 会自动在此环境后面加上对应的图片保存指令。这个环境的返回结果为:`\includegraphics [width=.45\linewidth]{pycode.py.pdf}`, 所以你可以把这个环境嵌套在任何的浮动环境, 比如 `figure`, `table` 中。在命令行中第一次编译时你会看到如下的日志:

```
current hash is FF7B5ECDBF52AA95DF921FCC076F9021
current hash is unique --> recorded
```

上述日志说明, zTikZ 已经识别到这是一个新的 python 环境, 并且保存了这个环境中绘图代码的 Hash 值; 然后, 第二次编译此文档时, 你会在输出的日志中定位到如下的输出:

```
current hash is FF7B5ECDBF52AA95DF921FCC076F9021
skip recompile by python, using the cache picture 1
```

这就说明, 由于你的 python 绘图部分的源代码没有改变, 然后 zTikZ 就直接采用了上一次编译的缓存图片, 跳过了重新编译这一步; 上面环境的运行结果为:

这里再给一个 Python 绘图环境的示例, 绘制了一个简单的来自 Matplotlib 官方的三维图像。其实这里给出这个例子, 就是为了让读者明白, 尽管目前 zTikZ 还没有支持便捷的三维矢量图形绘制, 但是你可以使用 Python 生成对应的 3 维矢量图; 虽然, 你可能需要再去学习 Python 中 Matplotlib 的相关语法, 但是这是简单的。

**注记 3.6.1** 由于 python 是依靠缩进来识别代码结构的, 所以在书写这部分的代码时, 不能够人工添加缩进, 在书写的时候需写为下面这样:

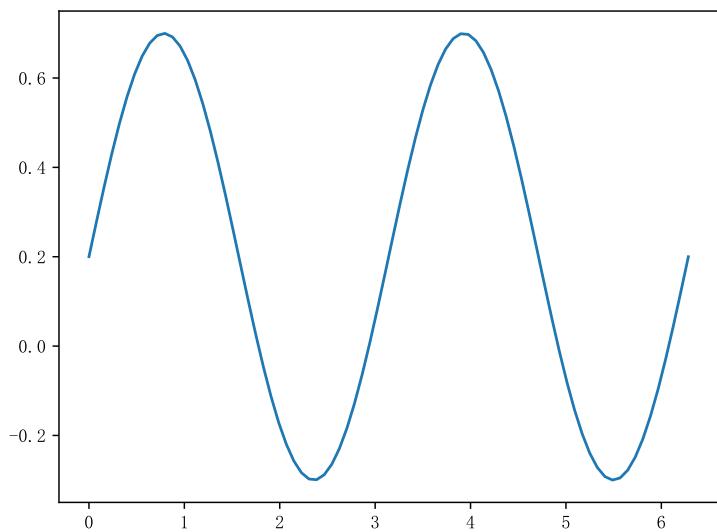


图 3.12: Python 绘图示例 1

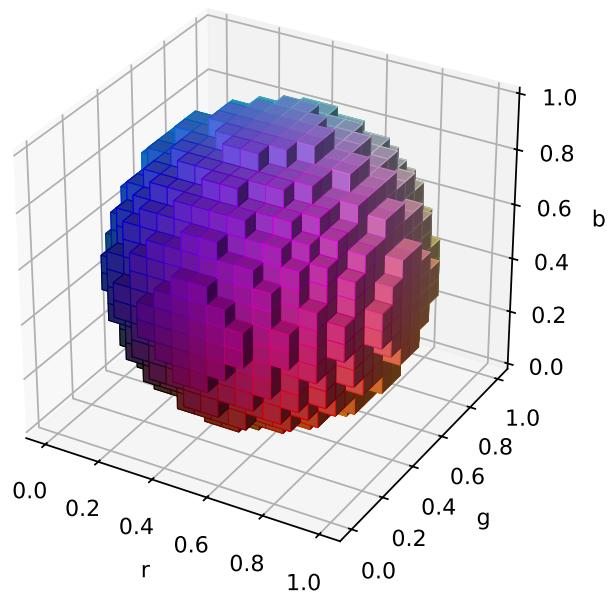


图 3.13: Python 绘图示例 2

```

16 \begin{figure}[!htb]
15 |   \centering
14 |   \begin{pyfig}{\width=.45\linewidth}{pycode.py}
13 |   import matplotlib
12 |   matplotlib.use('Agg')
11 |   from matplotlib import pyplot as plt
10 |   plt.rcParams['font.sans-serif'] = ['FangSong']
9 |   plt.rcParams['axes.unicode_minus'] = False
8 |   import numpy as np
7 |
6 |   x = np.linspace(0, 2*np.pi, num = 80)
5 |   y = np.sin(x)*np.cos(x)+.2
4 |   plt.plot(x, y)
3 |   \end{pyfig}
2 |   \caption{Python绘图示例}
1 |   \label{fig:pyfig-1}
028 \end{figure}

```

如果你实在是需要缩进,那么在这里我推荐另外一种可以使用缩进的方法:把 `pyfig` 环境连同其内部代码保存在另外一个文件中,比如这里我保存为 `pycode.mpl`,然后在 `figure` 环境中使用`\input {pycode.mpl}`引入此部分的代码。如下:

```

\begin{figure}
\centering
\input{./data/pycode.mpl}
\caption{Python Figure}
\label{fig:pyfig-1}
\end{figure}

```

## 3.7 mathematica

### 3.7.1 基本介绍

其实使用 `mathematica` 进行绘图这个部分和前面的使用 `Python` 绘图是差不多的, `zTikZ` 提供了一个 `wolframGraphics` 环境用于使用 `mathematica` 绘图。与之前的 `pyfig` 环境不同的是, 此时你需要手动加入图片的保存路径;路径的前缀为`../ztikz_output/mma_data/<figure name>`。为何这里这个部分我不使用 `zTikZ` 自动完成? 由于 `mathematica` 绘图代码中可能存在多幅图形的情况, 需要使用 `Show` 命令组合成为一个图, 那么这个组合方式就是千变万化的了。所以为了给用户提供更多的自由操作的空间。这里的图片保存命令由用户自己书写。并且上述的 `<figure name>` 只能写为 `<wls script name>.pdf` 的形式; 比如你的 `WolframScript` 脚本名称为 `mma_1.wls`, 那么你的 `<figure name>` 只能写为 `mma_1.wls.pdf`, 其中的图片格式可以自己指定, 比如为 `.png`, `.jpg`, `.mbp` 等。此环境同样是加入了 Cache 机制的,下面给出一个具体的使用案例:

```

\begin{wolframGraphics}[\width=.4\linewidth]{mma_1.wls}
plotFunction[fun_, xlims_, ylims_] := ContourPlot[fun,
xlimits, ylims,
ContourStyle->{

```

```

RGBColor["#00C0A3"],
Thickness[0.004]
},
AspectRatio->((xlimits[[2]]//Abs) + (xlimits[[3]]//Abs))/((ylimits[[2]]//Abs) +
(ylimits[[3]]//Abs)),
AxesOrigin->\{0,0\},
Axes->True,
Frame->False,
AxesStyle->Arrowheads[\{0, 0.03\}],
AxesLabel->\{"x", "y"\},
PlotRange -> Full
]

xlimits = \{x, -3, 6\};
ylimits = \{y, -4, 5\};
fp1 = plotFunction[y==Sin[x], xlimits, ylimits];
fp2 = plotFunction[x^2/4 + y^2/3 == 5, \{x, -5, 5\}, \{y, -5, 5\}];

figure = Show[fp2, fp1];
(* 1. 保存的图片格式为:*.wls.pdf; 2. 保存路径在:./ztikz_output/mma_data *)
Export["./ztikz_output/mma_data/mma_1.wls.pdf", figure];
\end{wolframGraphics}

```

### 3.7.2 注意事项

因为 mathematica 中的代码是允许用户自由添加缩进的，所以你可以自己添加 Mathematica 代码的缩进。和前面的 Python 绘图代码类似，你可以把此部分代码保存在一个单独的文件中，然后通过\input 进行引入，这里不再给出对应的案例。

- 注意空格与 Tab，如果源代码中有 Tab，那么 zTikZ 在进行此环境的抄录时会把原本的 Tab 转义为  $\wedge\wedge I$ ，从而造成 Mathematica 源代码的错误，比如你可能会看到你的源代码抄录后变成了下面的样子：

```

\wedge\wedge IContourStyle->{
\wedge\wedge I\wedge\wedge IRGBColor["#00C0A3"],
\wedge\wedge I},

```

- 同时注意 Mathematica 中注释的写法，不是(\* something\*），而是(\* something \*)，也就是你的注释不能够紧挨着 \*，否则会造成 mathematica script 的解析错误。

- 由于 WolframScript 的限制，对应的 Mathematica 脚本的后缀只能为:.wls，否则 WolframScript 无法识别此脚本，也就不会去执行此脚本了。

用户如果要使用 zTikZ 的 Mathematica 模块, 请务必确保 wolframscript 在命令行中能够正常运行. 可以使用如下文件作为测试用例, 检测 wolframscript 是否正常工作;

```
plotFunction[fun_, xlims_, ylims_] := ContourPlot[fun,
  xlims, ylims,
  ContourStyle -> {
    RGBColor["#00COA3"],
    Thickness[0.004]
  },
  AspectRatio -> ((xlims[[2]]//Abs) + (xlims[[3]]//Abs))/((ylims[[2]]//Abs) +
  (ylims[[3]]//Abs)),
  AxesOrigin -> {0,0},
  Axes -> True,
  Frame -> False,
  AxesStyle -> Arrowheads[{0, 0.03}],
  AxesLabel -> {"x", "y"},
  PlotRange -> Full
]

xlimits = {x, -3, 6};
ylimits = {y, -4, 5};
fp1 = plotFunction[y==Sin[x], xlimits, ylims];
fp2 = plotFunction[x^2/4 + y^2/3 == 5, {x, -5, 5}, {y, -5, 5}];

figure = Show[fp2, fp1];
(* 1. 保存的图片格式为 :*.wls.pdf; 2. 保存路径在 :./ztikz_output/mma_data *)
Export["works_well.pdf", figure];
```

把这里的源码保存为 `test.wls`, 然后在命令行运行:

```
wolframscript -script test.wls
```

如果正常工作的话, 那么在你的当前工作目录下会产生一个名为 `works_well.pdf` 的 pdf 文件. 反之, 你的 wolframscript 没有正常配置或者是激活, 也就不能够使用本模块.

同样的你可以使用 Mathematica 绘制 3 维图形<sup>6</sup>. 目前 zTikZ 仅支持插入静态图片, 后续可能会考虑加入动态图片的支持功能, 就像另外一个开源矢量图象绘制软件**Asymptote** 中的.`prc` 文件一样. 但是要使得能在 PDF 中预览动态图形, 首先你的 PDF 阅读器必须支持 JavaScript, 常见的这种类型的 PDF 阅读器就是 Adobe 家的 Acrobat 了.

---

<sup>6</sup>由于目前的 Mathematica 不支持输出 3 维矢量图, 所以想要是你的 3 维图像显得更加清晰, 可以调节图像的分辨率.

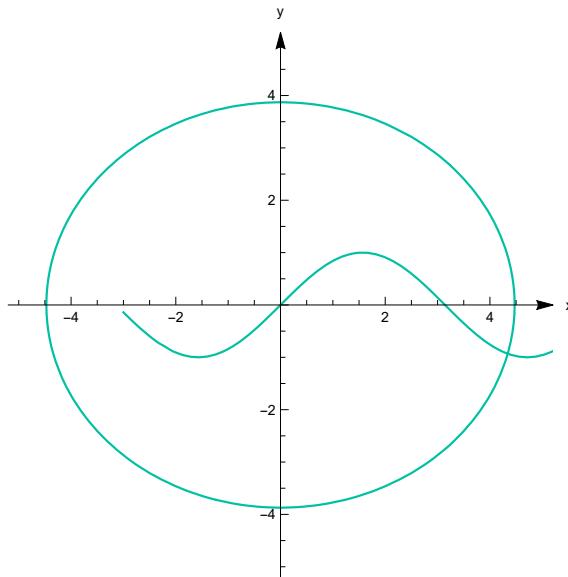


图 3.14: Mathematica 绘图示例

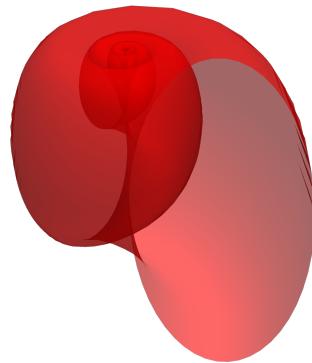


图 3.15: Mathematica 绘图示例 2

## 3.8 matlab

### 3.8.1 基本介绍

目前 zTikZ 中的 Matlab 模块还在开发中,但是目前你可以使用 `matlab2tikz` 这个 Matlab 插件来把你的 Matlab 图形转换为对应的 tikz 代码,效果也是很好的.

但是目前你可以在命令行中调用 Matlab 运行自己的 Matlab 脚本,一个测试脚本如下:

```
x = 1:0.1:2*pi;
y = sin(x);
```

```

figure('visible','off')
plot(x, y, 'r-');

exportgraphics(gcf, 'myfig.pdf')

```

然后在命令行中使用如下命令进行运行:

```
matlab -batch "run('matlab.m')"
```

运行完后，你便可以在当前目录下看到一个 pdf 文件，名为 `myfig.pdf`; 在运行方式这一点上，Matlab 和 Wolframscript 的运行命令: `wolframscript -script mma.wls` 是有一点区别的。

### 3.8.2 现状

似乎在 TeX 中无法正常的调用 Matlab 进行计算与绘图，Matlab 日志显示错误。但是目前此功能应该不会在短时间内开发出来。如果有比较多的需求，我可能会考虑。

## 3.9 数值计算

### 3.9.1 xfp

众说周知，TeX 自身的计算能力是比较羸弱的，所以涉及到一定的计算需求时，一般宏包的解决方法都是使用外部程序，让 TeX 只负责排版就行了。但是在 L<sup>A</sup>T<sub>E</sub>X3 项目发展了这么久之后，也做出了一些令人惊喜的结果。这里我们主要介绍 L<sup>A</sup>T<sub>E</sub>X3 的 `xfp` 宏包，用于浮点数运算。

这里说明部分 `xfp` 也许可以解决的痛点：

- 在 TikZ 绘图中，常常是需要坐标运算的，尽管 TikZ 提供了一个 `calc` 库，但是这个库的使用语法总觉得不是那么的自然。于是这个时候你就可以使用 `xfp` 宏包。
- 在你自定义一些需要用到数值计算的宏命令时，使用 `xfp` 宏包是一个比较好的选择。

`xfp` 宏包的详细使用教程请参见官方文档，这里不再赘述。

zTikZ 或者是 zL<sup>A</sup>T<sub>E</sub>X 并不会自动加载 `xfp` 宏包，如果你有这方面的需要，请自己加载。

### 3.9.2 python

上面介绍了 Python 的绘图功能，这里再引入 zTikZ 中的浮点数计算部分 (Sympy 对应的部分应该不能叫浮点数计算了，毕竟 Sympy 进行的是精确的计算。) 这里使用的浮点数运算主要是基于 Python，以及可能的宏包 `numpy` 等。zTikZ 在调用此命令时默认载入 Python 库 `NumPy`, `SciPy`，并且使用 `numpy` 中的函数时不用再加上前缀；比如求解  $\sin(2.345)$  时，直

接使用`\py {sin(2.345)}`即可, 不用写成`\py {np.sin(2.345)}`. 对于库 SciPy 中的函数使用方法同理.

### python command

zTikZ 提供了命令`\py` 用于浮点数运算, 这部分的结果并不会被缓存, 也就是说每次编译此文档时, Python 都会重新计算此部分的结果. `\py` 的参数说明如下:

```
\py[<return type>]{<expression>}
```

上述的第一个默认参数值为`:raw`, 可选值有 `str`, 二者的区别可以简单的认为, 返回的 Token 的类别码不同. 比如当外部文件中的内容为:

```
\[ a^2 + b^2 = c^2 \]
```

默认情况下, `\py` 返回此命令的结果为:

$$a^2 + b^2 = c^2$$

但是如果你指定返回的类型为 `str` 时, 那么在文档中的显示结果就会变为`\[ a^2 + b^2 = c^2 \]`. 而不是默认情况下的排版公式.

值得说明的是, `\py` 命令和 `xfp` 宏包提供的`\inteval`, `\fpeval` 是类似的; 也就是你可以把 `\py` 命令嵌套到你自己定义的一个命令中. 同样是使用 `#1` 来表示接收到的参数. 比如你可以创建下面这几个命令:

```
\newcommand{\pypow}[1]{\py{#1}}
\newcommand{\pyreverse}[1]{\py{'#1'[::-1]}}
\newcommand{\pyuppercase}[1]{\py{'#1'.upper()}}
```

分别用于数值计算 (乘方计算), 字符串反转输出, 字符串大写输出. 使用效果如下:

- Power Calculation:  $2^{64} = 1024$
- Reverse a string using Python: XeTaL-olleH
- Uppercase a string: HELLO-LATEX
- Modulus:  $102 = 6 \pmod 8$
- Return string Options: \$\$1024\$\$

如果你想要使用 Python 中的求模运算需要输入`%` 时, 在`\py` 命令中你应该写为:

```
\py{102 \% 8}
```

或者是如果你需要在`\py` 命令中传入 `$`, 请像下面这样书写:

```
\py{'\$\$'+str(2**10)+'\$\$'}
```

毕竟在命令行中，如下的命令才可以正常的工作：

```
sed -i "6s|Float_res = .*|Float_res = '\$\$'+str(2**10)+'\$\$'|"
./ztikz_output/scripts/python_script.py
```

**注记 3.9.1** 目前由于 Windows 上的 sed 命令（又或者是平台差异）和 Linux 下的差异，所以可能导致在 Windows 上使用时，`py` 中的单引号 `'` 不能正确的输入到目标文件中，从而导致字符串的声明失败。请一定注意！

### python environment

zTikZ 同时也提供了一个用于自由书写 Python 代码的环境 `pycode`，可以用于生成复杂且规律的表格代码等排版元素。比如下面的示例：

```
\begin{pycode}[pycode_1.py]
import numpy as np

# write file
with open ('./ztikz_output/python_data/pycode_1.py.out', 'w') as file:
    file.write("\begin{tabular}{p{3cm}ccc}\n")
    file.write("\hline\n")
    file.write("number/function & \$\sin\$ & \$\cos\$ & \$\tan\$\\\\\\n")
    file.write("\hline\n")
    for i in range(1, 21):
        file.write(
            f"${i} & ${np.around(np.sin(i), decimals=4)} & ${np.around(np.cos(i),
            decimals=4)} & ${np.around(np.tan(i), decimals=4)}\\\\\\n"
        )

    file.write("\hline\n")
    file.write("\end{tabular}\n")
\end{pycode}
```

那么在运行此命令后，在 zTikZ 的缓存文件夹中会生成一个名为 `pycode_1.py.out` 的文件，其内容为：

```
\begin{tabular}{p{3cm}ccc}
\hline
number/function & $\sin\$ & $\cos\$ & $\tan\$\\
\hline
\$1\$ & \$0.8415\$ & \$0.5403\$ & \$1.5574\$\\
```

```
$2\$ & $0.9093\$ & $-0.4161\$ & $-2.185$\\"
$3\$ & $0.1411\$ & $-0.99\$ & $-0.1425$\\"
$4\$ & $-0.7568\$ & $-0.6536\$ & $1.1578$\\"
$5\$ & $-0.9589\$ & $0.2837\$ & $-3.3805$\\"
$6\$ & $-0.2794\$ & $0.9602\$ & $-0.291$\\"
$7\$ & $0.657\$ & $0.7539\$ & $0.8714$\\"
$8\$ & $0.9894\$ & $-0.1455\$ & $-6.7997$\\"
$9\$ & $0.4121\$ & $-0.9111\$ & $-0.4523$\\"
$10\$ & $-0.544\$ & $-0.8391\$ & $0.6484$\\"
$11\$ & $-1.0\$ & $0.0044\$ & $-225.9508$\\"
$12\$ & $-0.5366\$ & $0.8439\$ & $-0.6359$\\"
$13\$ & $0.4202\$ & $0.9074\$ & $0.463$\\"
$14\$ & $0.9906\$ & $0.1367\$ & $7.2446$\\"
$15\$ & $0.6503\$ & $-0.7597\$ & $-0.856$\\"
\hline
\end{tabular}
```

所以这段代码的具体效果如下:

number/function	sin	cos	tan
1	0.8415	0.5403	1.5574
2	0.9093	-0.4161	-2.185
3	0.1411	-0.99	-0.1425
4	-0.7568	-0.6536	1.1578
5	-0.9589	0.2837	-3.3805
6	-0.2794	0.9602	-0.291
7	0.657	0.7539	0.8714
8	0.9894	-0.1455	-6.7997
9	0.4121	-0.9111	-0.4523
10	-0.544	-0.8391	0.6484
11	-1.0	0.0044	-225.9508
12	-0.5366	0.8439	-0.6359
13	0.4202	0.9074	0.463
14	0.9906	0.1367	7.2446
15	0.6503	-0.7597	-0.856

表 3.2: Using Python to generate Table

**注记 3.9.2** 本环境 (pycode) 目前还不够成熟, 请谨慎使用, 也欢迎各位提出宝贵的改进意见. 当然, 本环境目前具有缓存机制.

**注记 3.9.3** 推荐用户使用最新的由 L<sup>A</sup>T<sub>E</sub>X3 编写的宏包:`csvsimple-l3`, 或者是`tabulararray`用于在 L<sup>A</sup>T<sub>E</sub>X 中进行表格的排版.

### 3.9.3 mathematica

使用 Mathematica 进行数值计算这一部分和后面的\wolfram 指令是有一部分重合的, 详细的使用参见后面一节的“符号计算”, 所以这一部分我们就在后面介绍.

## 3.10 符号计算

符号计算是区别于数值计算的, 上述的数值计算章节应该也有介绍; 但在介绍 zTikZ 中的符号计算模块之前先给出一个符号计算的定义, 以下定义摘自 wiki:

数学和计算机科学中, 计算机代数或符号计算或代数计算, 是研究、开发用于操作表达式等数学对象的算法与软件的科学领域。这通常被视为是运算科学的一个子领域, 但运算科学一般基于近似浮点数的数值计算, 而符号计算则使用含变量的表达式进行精确计算, 其中变量没有赋值。执行符号计算的软件系统称为计算机代数系统 (computer algebra system, CAS), “系统”暗示了软件的复杂性, 其中至少包括一种在计算机中表示数学数据的方法、一种编程语言 (通常异于用于实现的语言)、一种专门的内存管理器、一套供输入输出表达式的用户界面、一大套用于通常运算的子程序, 如表达式简化、能实现链式法则、多项式因式分解、不定积分等等的求导算法。

当前流行的计算机代数系统主要有:

- mathHandbook.com
- PARI/GP
- Sagemath
- Meditor
- Mathematica
- MuPAD
- Maple
- Mathomatic
- MAGMA
- Xcas/Giac
- Maxima
- Yacas
- GAP
- Mate

zTikZ 主要提供一个和 Mathematica(假如你已经购买了该软件), 以及 Python 的 Sympy 模块的符号计算接口. 后续可能会开发一个统一的接口用于 T<sub>E</sub>X 和外部程序的交互.

### 3.10.1 python/sympy

Python 的 Sympy 是一个免费, 开源, 轻量的符号计算模块, 其官网上有着详细的教程。所以这里便不再赘述其语法, 重点介绍 zTikZ 中提供的几个接口 (命令), 用于和 Sympy 交

互.

```
\sympy{<expression>}
```

和之前的使用 Python 进行数值计算不同的是, zTikZ 针对此命令提供了 Cache 机制, 此命令对应的结果会被保存在文件: `./ztikz_output/python_data/sympy_<index>.out` 文件中. 此文件名中的 `<index>` 表示的是对应的符号计算表达式的序号.

`\sympy` 命令的运算结果被保存在文件中之后, 通过`\input` 命令把对应的运算结果导入到 TeX 的输出流 (文档) 中, 由于默认的情况下此结果包含数学公式中的上下标:`^`, `_`, `\dots` 等, 所以在把其导入到 LATEX 源码中时需要放入数学环境中.

zTikZ 模块的`\sympy` 命令在进行符号运算时, 默认的符号变量有:`x`, `y`, `z`, `u`, `v`, `t`, 这些变量你不需声明便可以直接使用; 下面给出使用`\sympy` 命令进行符号计算的部分示例:

```
% 定积分
\sympy{integrate(sin(x)/x, (x, -oo, oo))}

% 不定积分
\sympy{integrate( x**8 + cos(7*x) + 6*t, x )}

% 矩阵特征值
\sympy{Matrix([[1, 2], [2, 2]]).eigenvals()}

% 极限计算
\sympy{limit(sin(x)/x, x, 0)}
```

计算定积分的例子:

$$\int_{-\infty}^{+\infty} \frac{\sin(x)}{x} dx = \pi$$

或者是计算不定积分的例子:

$$\int x^8 + \cos(7x) + 6t dx = 6tx + \frac{x^9}{9} + \frac{\sin(7x)}{7}$$

或者是一个计算特征值的例子:

$$\text{eig}\left(\begin{bmatrix} 1 & 2 \\ 2 & 2 \end{bmatrix}\right) = \left\{ \frac{3}{2} - \frac{\sqrt{17}}{2} : 1, \frac{3}{2} + \frac{\sqrt{17}}{2} : 1 \right\}$$

计算极限的例子:

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

目前的`\sympy` 命令只支持单行命令的模式, 如果你需要使用多行 (条) 命令来达到计算目的, 请考虑把它们变为一行命令 (一条指令).

### 3.10.2 mathematica

zTikZ 模块提供和 Mathematica 相关的符号计算, 数值运算和知识查询接口; 以下的所有命令均具有缓存机制.

- `\wolfram [<option>]{<expression>}`: 使用 Mathematica 计算此表达式 `<expression>`, 默认返回 TeX 格式的代码, 可以把 `<option>` 设为 `text`, 让其返回一个文本对象. 可以在这个命令中执行任何的 `wolfram` 指令, 但是需要注意的一点是, 所有和 `wolfram` 相关的命令是不会自动进入数学模式的, 需要手动添加数学模式的标记.
- `\wolframSolve [<cmd style>]{<expression>}[<variable>][<domain>]`: 其中第一个可选参数默认值为: `part`, 意味着你的命令需要分拆为 3 个部分: 表达式 – (求解) 变量名 – 求解范围, 对应上面的参数, 分别填入. 如果指定第一个参数为 `full`, 那么此时只需要给出对应的 `<expression>`, 不用再次指定后续参数.(毕竟在第二个 (强制性-Mandatory) 参数中就已经包含了这些信息, 参见后面的具体使用样例).
- `\wolframDSolve [<cmd style>]{<equation>}[<independent variable>][<dependent variablei>]`: 此命令用于求解微分方程, 其中的第一个可选参数和上面的`\wolframSolve` 的意义一致, 不再赘述. 第二个参数表示要求解的微分方程, 第三个参数表示求解的独立变量 (函数), 最后一个参数表示此微分方程求解函数的自变量.

### wolfram

此命令会把求解结果保存到一个临时变量`\wolframResult` 中, 所以可以不用把此命令置于公式环境中, 直接在公式环境中使用命令`\wolframResult` 引用对应的结果即可, 引用方式有两种:

- `\wolframResult [raw] [<separator>]`: 直接引用全部结果 (默认不同的结果使用 “,” 进行分割), 不同的结果之间使用 `<separator>` 进行分割. 第一个可选参数默认为 `raw`, 也就是你可以直接使用 `\wolframResult` 引用上一步的计算结果.
- `\wolframResult [<integer expresion>]`: 引用单个结果, 可以是任何合法的整数运算表达式.

在这里给出`\wolfram` 命令的部分使用样例:

```
\wolfram{Series[Exp[x], {x, 0, 5}]}
\[wolframResult]

\wolfram{LaplaceTransform[t^4 Sin[t],t,s]}
\[wolframResult]

\wolfram{text}{WolframAlpha["Shanghai population", "ShortAnswer"]}
\[wolframResult]
```

函数  $y = e^x$  的 5 阶 Taylor 展开式为:

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

函数  $x = t^4 \sin(t)$  的 Laplace 变换为:

$$\mathcal{L}[t^4 \sin(t)] = \frac{24(5s^4 - 10s^2 + 1)}{(s^2 + 1)^5}$$

在\wolfram 指令中执行 Mathematica 中的 WolframAlpha 命令进行查询, 比如这里查询上海的人口数量, 结果为:

这里补充一个使用\wolfram 就行数值运算的例子, 因为 Mathematica 中有着诸多和数值运算相关的函数, 这里仅以内置的函数 N[<expression>] 为例:

比如我们求解  $\pi$  的截取前 30 小数的近似值为:

$$\pi \approx 3.14159265358979323846264338328$$

在使用\wolfram 命令进行浮点数运算时, 只要表达式中含有小数, 那么 Mathematica 就会默认进行浮点数运算, 而不会计算表达式的精确值.

### wolframSolve

\wolframSolve 命令可以用于多项式方程根的求解以及方程组的求解, 并且可以给定求解的范围. 和前面的\wolfram 命令类似, 下面给出几个比较简单的求解示例:

```
\wolframSolve{x^4 - x^2 - 5 == 0}[x]
\begin{aligned}
\left. \begin{aligned}
&\wolframResult[\text{raw}][] \\
\end{aligned} \right. \\
\end{aligned}

\wolframSolve{a x + y == 7 \&& b x - y == 1}[x, y]
\left. \begin{aligned}
&\wolframResult[\text{raw}][], \\
\end{aligned} \right. \\

\wolframSolve{x^2 + 2 y^3 == 3681 \&& x > 0 \&& y > 0}[x, y][\text{Integers}]
\left. \begin{aligned}
&\wolframResult[\text{raw}][], \\
\end{aligned} \right. \\

\wolframSolve[full]{x^2 + y^2 == 5^2 \&& y > x > 0, {x, y}, \text{Integers}]
\left. \begin{aligned}
&\wolframResult[\text{raw}][] \rightarrow \\
\end{aligned} \right.
```

方程  $x^4 - x^2 - 5 = 0$  的所有根为:

$$\left\{ \begin{array}{l} x = -i\sqrt{\frac{1}{2}(\sqrt{21}-1)} \\ x = i\sqrt{\frac{1}{2}(\sqrt{21}-1)} \\ x = -\sqrt{\frac{1}{2}(1+\sqrt{21})} \\ x = \sqrt{\frac{1}{2}(1+\sqrt{21})} \end{array} \right. \quad (3.10.1)$$

方程组  $\begin{cases} ax + y = 7 \\ bx - y = 1 \end{cases}$  的解为:

$$x = \frac{8}{a+b}, y = -\frac{a-7b}{a+b}$$

不定方程  $\begin{cases} x^2 + 2y^3 = 3681 \\ x > 0, y > 0 \end{cases}$  的整数解为:

$$x = 15 \parallel y = 12 \parallel x = 41 \parallel y = 10 \parallel x = 57 \parallel y = 6$$

不定方程  $\begin{cases} x^2 + y^2 = 5^2 \\ x > y > 0 \end{cases}$  的整数解为:

$$x = 3 \leftrightarrow y = 4$$

目前解的筛选功能仅支持类似数组的访问形式，引用形式比较有限。背后其实就是根据不同解之间分隔符 ‘;’，把返回结果中的 `TokenList` 分割为一个 `sequence`。根据划分的结果生成一个 `sequence` 列表，然后采用一个整数进行索引。

### wolframDSolve

命令 `\wolframDSolve` 和命令 `\wolframSolve` 完全相同，只是这个命令是用于求解微分方程的。下面是几个示例:

```
\wolframDSolve[{y'[x] + y[x] == a*Sin[x], y[0] == 0}][y[x]][x]
\wolframDSolve[full]{{y'[x] == Exp[z[x]] + 1, z'[x] == y[x] - x}, {y[x], z[x]}, x}
```

微分方程  $y' + y = a \sin(x)$ ,  $y(0) = 0$  的解为:

$$y(x) = -\frac{1}{2}ae^{-x}(-e^x \sin(x) + e^x \cos(x) - 1)$$

非线性系统微分方程组  $y'(x) + y(x) = \text{Exp}(z(x)) + 1$ ,  $z'(x) = y(x) - x$  的解为:

$$\begin{cases} z(x) = \log \left( c_1 \tan^2 \left( \frac{1}{2} \left( \sqrt{2}\sqrt{c_1}x + 2\sqrt{2}\sqrt{c_1}c_2 \right) \right) + c_1 \right) \\ y(x) = x + \sqrt{2}\sqrt{c_1} \tan \left( \frac{1}{2} \left( \sqrt{2}\sqrt{c_1}x + 2\sqrt{2}\sqrt{c_1}c_2 \right) \right) \end{cases} \quad (3.10.2)$$

# CHAPTER 4

## Plot Gallery

### 4.1 tikz

下面我们给出 zTikZ 这部分命令的一些综合绘图案例:

如果你修改了绘图代码，但是发现得到的 pdf 中的图像并没有改变，那么极有可能是因为你指定的精度过高，超出了 TeX 的内存使用限制。(而且由于采用了 external 库用于缓存，有可能你在编译时并不会抛出这个错误) 其实比较耗费内存的点主要有 3 个：

- 指定的精度过高，一般情况下在区间长度  $< 5$  时指定精度为 100 就已经足够了
- 使用了多个 \ContourPlot 函数，在默认的精度 100 下，多个此函数也可能导致内存超出
- 最后一点耗时的点就是 \ShowIntersection 命令，可以先用 Geogebra 得到交点后再使用 \ShowPoint 命令进行点的绘制.
- 更严重的如果出现了编译错误，请考虑去掉 \ShadePlot 命令，或在 \usepackage [external=false]{ztikz} 的情况下使用此命令.

#### 4.1.1 Example 1

```
% show angle
\tikzset{
    right angle quadrant/.code={%
        \pgfmathsetmacro\quadranta{{1,1,-1,-1}[\#1-1]}% Arrays for selecting
        quadrant
        \pgfmathsetmacro\quadrantb{{1,-1,-1,1}[\#1-1]},%
    right angle quadrant=1, % Make sure it is set, even if not called explicitly
    right angle length/.code={\def\rightanglelength{\#1}}, % Length of symbol
    right angle length=2ex, % Make sure it is set...
    right angle symbol/.style n args={3}{%
        insert path={%
            let \p0 = ($(\#1)!(\#3)!(\#2)$) in % Intersection
            let \p1 = ($(\p0)!\quadranta*\rightanglelength!(\#3$), % Point on base
            line
```

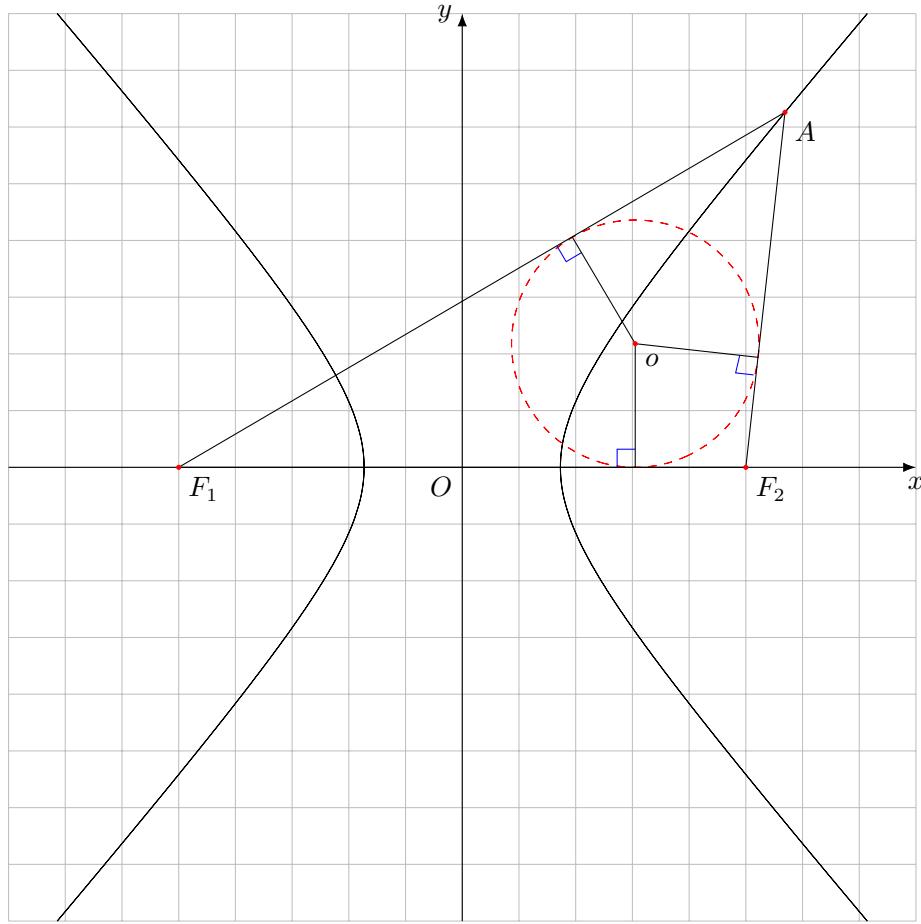


图 4.1: Parabolic Example

```

\p2 = ($(\p0)!\quadrantb*\rightanglelength!(#2$) in % Point on
perpendicular line
let \p3 = $(\p1)+(\p2)-(\p0)$) in % Corner point of symbol
(\p1) -- (\p3) -- (\p2)
}
}
}

% main code
\begin{tikzpicture}[scale=.75, >=Latex]
>ShowGrid[step=1, color=gray, opacity=.5]{(-8, -8); (8, 8)}
>ShowXYAxis{8}{8}
% curve
\ContourPlot[-8:8; -8:8]{x**2/3-y**2/4-1}
\ContourPlot[-8:8; -8:8] [dashed, red]{(x-3.05)**2 + (y-2.18)**2-4.76}
% points and lines
\coordinate (A) at (5.69, 6.26);

```

```
\coordinate (B) at (-5, 0);
\coordinate (C) at (5, 0);
\coordinate (D) at (3.05, 2.18);
\draw (A) -- (B) -- (C) -- cycle;
% angle and 3-lines
\draw [blue,right angle symbol={A}{1.94,4.06}{D}]; % F
\draw [blue,right angle symbol={A}{5.21,1.94}{D}]; % E
\draw [blue,right angle symbol={B}{3.05,0}{D}]; % G
\draw (D) -- (1.94, 4.06);
\draw (D) -- (5.21, 1.94);
\draw (D) -- (3.05, 0);
\ShowPoint[type=*, color=red]{(A); (B); (C); (D)}[$A$; $F_1$; $F_2$; $o$][below right]
\end{tikzpicture}
```

### 4.1.2 Example 2

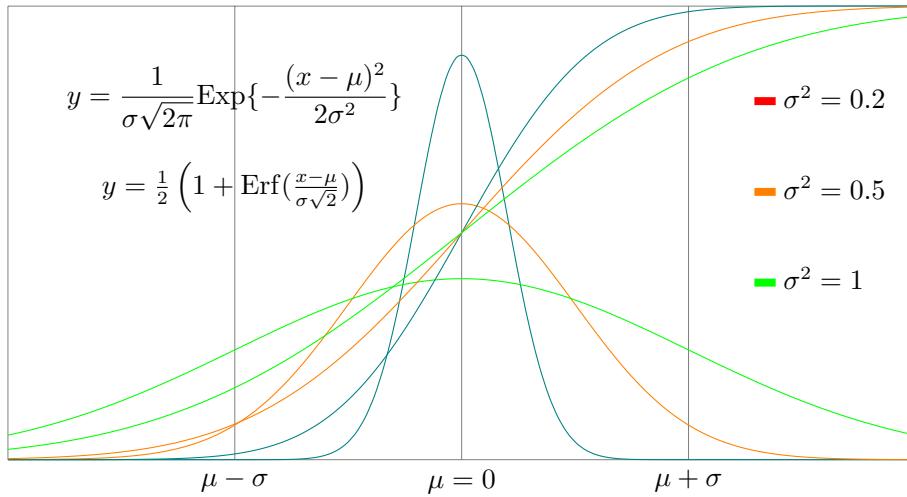


图 4.2: Normal-distribution Example

```
\begin{tikzpicture}[yscale=6, xscale=3]
\ShowGrid[(-2,0); (2,1)]
% pdf
\Plot[-2:2][teal]{1/(sqrt(0.2)*sqrt(2*pi))*exp(-(x-0)**2/(2*0.2**2))}
\Plot[-2:2][orange]{1/(sqrt(0.5)*sqrt(2*pi))*exp(-(x-0)**2/(2*0.5**2))}
\Plot[-2:2][green]{1/(sqrt(1)*sqrt(2*pi))*exp(-(x-0)**2/(2*1**2))}
% cdf
\Plot[-2:2][teal]{0.5*(1+erf((x-0)/(sqrt(0.2)*sqrt(2))))]
\Plot[-2:2][orange]{0.5*(1+erf((x-0)/(sqrt(0.5)*sqrt(2))))]
\Plot[-2:2][green]{0.5*(1+erf((x-0)/(sqrt(1)*sqrt(2))))]
% annotate
\ShowPoint[radius=0pt]{(-1, 0); (0, 0); (1, 0)}[$\mu-\sigma$; $\mu=0$;
$\mu+\sigma$][below]
\ShowPoint[radius=0pt]{(1, 0.8); (1, 0.6); (1, 0.4)}[
\textrmcolor{red}{\rule[1pt]{8pt}{3pt}}; $\sigma^2=0.2$;
\textrmcolor{orange}{\rule[1pt]{8pt}{3pt}}; $\sigma^2=0.5$;
\textrmcolor{green}{\rule[1pt]{8pt}{3pt}}; $\sigma^2=1$;
][right=2em]
\ShowPoint[radius=0pt]{(-1, 0.8)}[$\displaystyle y = \frac{1}{\sigma\sqrt{2\pi}} \text{Exp}\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$]
\ShowPoint[radius=0pt]{(-1, 0.6)}[$y=\frac{1}{2}\left(1+\text{Erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right)$]
\end{tikzpicture}
```

### 4.1.3 Example 3

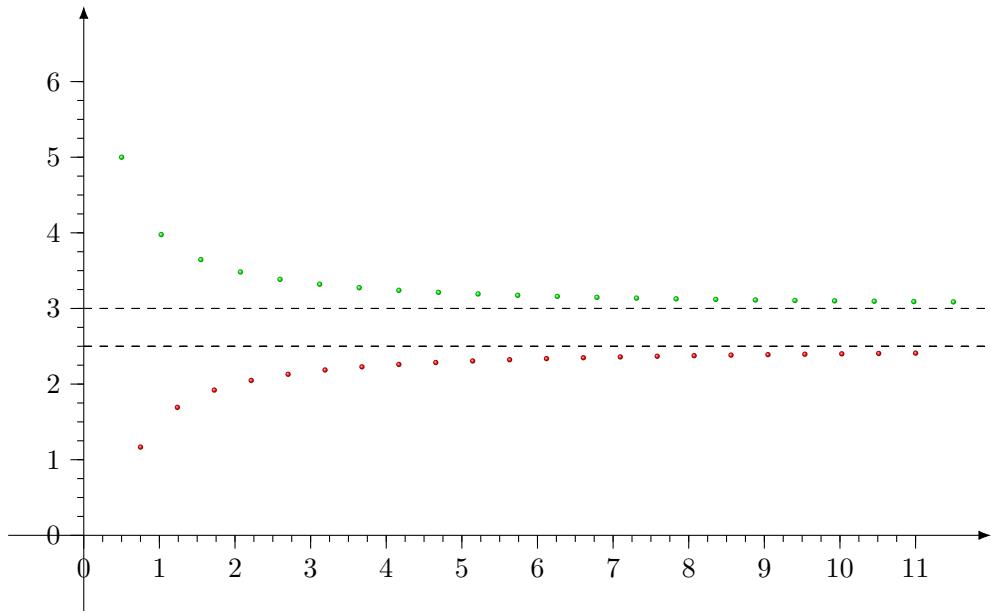


图 4.3: Sup and Inf Example

```
\begin{tikzpicture}[>=Latex]
\xAxis[-1][12]
\yAxis[-1][7]

\PlotPrecise{plot}{22}
\Plot[0.75:11][red, thick, opacity=0][type=ball, color=red]{2.5-1/x}

\PlotPrecise{plot}{22}
\Plot[0.5:11.5][red, thick, opacity=0][type=ball, color=green]{3+1/x}

\PlotPrecise{contour}{long}{40}
\ContourPlot[0:12][dashed]{y-2.5}
\ContourPlot[0:12][dashed]{y-3}

\end{tikzpicture}
```

#### 4.1.4 Example 4

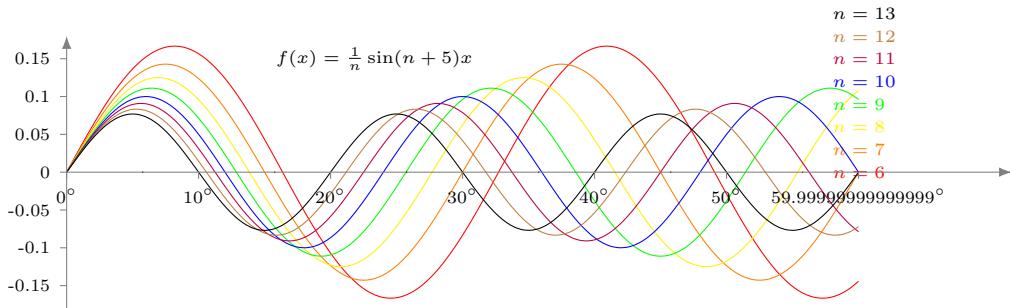


图 4.4: Loop Plot Example

```
\ExplSyntaxOn
\clist_new:N \l__color_clist
\clist_set:Nn \l__color_clist {red, orange, yellow, green, blue, purple, brown, black}
\newcommand{\colorItem}[1]{\clist_item:Nn \l__color_clist {\#1}}
\ExplSyntaxOff

\begin{tikzpicture}[scale=10, >=Latex, font=\scriptsize]
% plot and annotate
\node at (.55, 0.15) [left] {$f(x) = \frac{1}{n} \sin(n+5)x$};
\foreach \i in {6, 7, 8, 9, 10, 11, 12, 13} {
    \Plot[0:pi/3][\colorItem{\fpeval{\i-5}}]{\fpeval{1/\i}*sin(\fpeval{\i+5}*x)}
    \node[color=\colorItem{\fpeval{\i-5}}] at (1, \fpeval{(\i-6)*0.03}) [right]
    {$n=\i$};
}
% axis draw
\ShowAxis [
    tickStyle=above,      axisColor=gray,
    tickStart=-0.15,     tickEnd=0.18,
    mainStep=0.05,
    mainTickColor=gray,   mainTickLabelPosition=left,
    mainTickLength=.5pt,  axisRotate=90,
](-0.18, 0); (0.18, 0)
\ShowAxis [
    tickStyle=below,      axisColor=gray,
    tickStart=0,           tickEnd=1.22,
    mainStep=\fpeval{3.1415926/18},
    mainTickColor=gray,   subTickLength=0pt,
    mainTickLength=.5pt,
    mainTickLabel={\fpeval{\CurrentFp/(3.1415926/18)*10}\circ}
] (0, 0); (1.25, -0)
\end{tikzpicture}
```

#### 4.1.5 Example 5

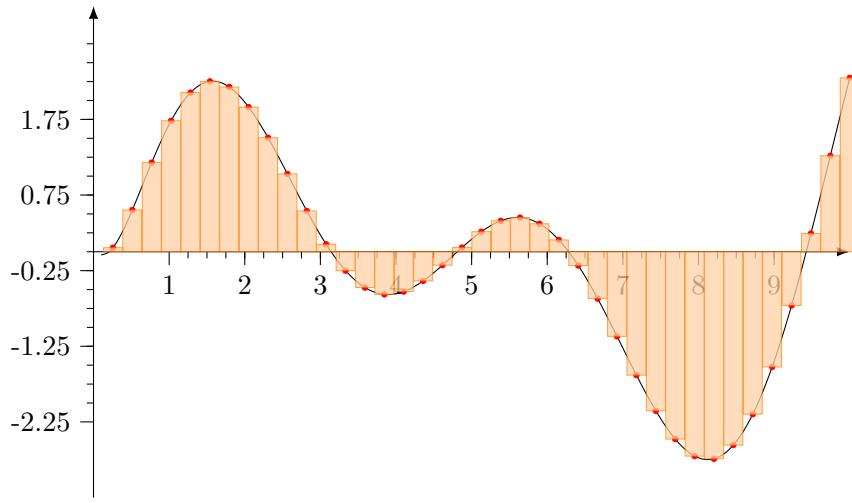


图 4.5: Darboux Example

```
\begin{tikzpicture}[>=Latex]
% draw axis
\xAxis[0][10]
\yAxis[-3.25][3.25]
% plot ucntion and generate discrete points
\Plot[0:10]{2*sqrt(x)*cos(log(x))*sin(x)}
\PlotPrecise{plot}{40}
\Plot[0:10][opacity=0][type=*, color=red]{2*sqrt(x)*cos(log(x))*sin(x)}
% bar plot
\BarPlot[x][fill=orange!35!white, bar width=\fpeval{10/40}cm, opacity=.75, very
thin, draw=orange]{\gnudata{2}}
\end{tikzpicture}
```

#### 4.1.6 Example 6

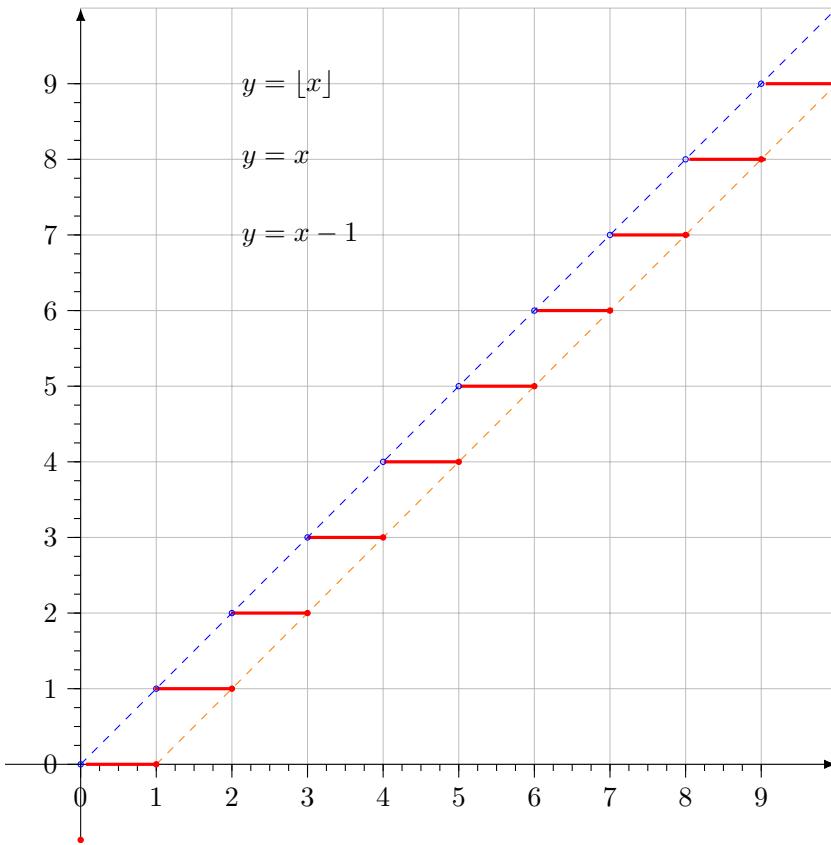


图 4.6: Stairs Function Example

```
\begin{tikzpicture}[>=Latex]
% grid and axis
\ShowGrid[step=1, color=gray, opacity=.5]{(0, 0); (10, 10)}
\xAxis[-1]{10} \yAxis[-1]{10}
% plot and points
\Plot[0:10]{red, jump mark right, very thick, xshift=2pt}{\lfloor x \rfloor}
\Plot[0:10]{dashed, blue}{x}
\Plot[1:10]{dashed, orange}{x-1}
\PlotPrecise{\plot}{11}
\Plot[0:10]{opacity=0, jump mark right}{\circ, blue}{x}
\PlotPrecise{\plot}{11}
\Plot[0:10]{opacity=0, jump mark right}{*, red}{x-1}
\ShowPoint[opacity=0]{(2, 9); (2, 8); (2, 7)}{\lfloor x \rfloor; x; x-1}
\end{tikzpicture}
```

#### 4.1.7 Example 7

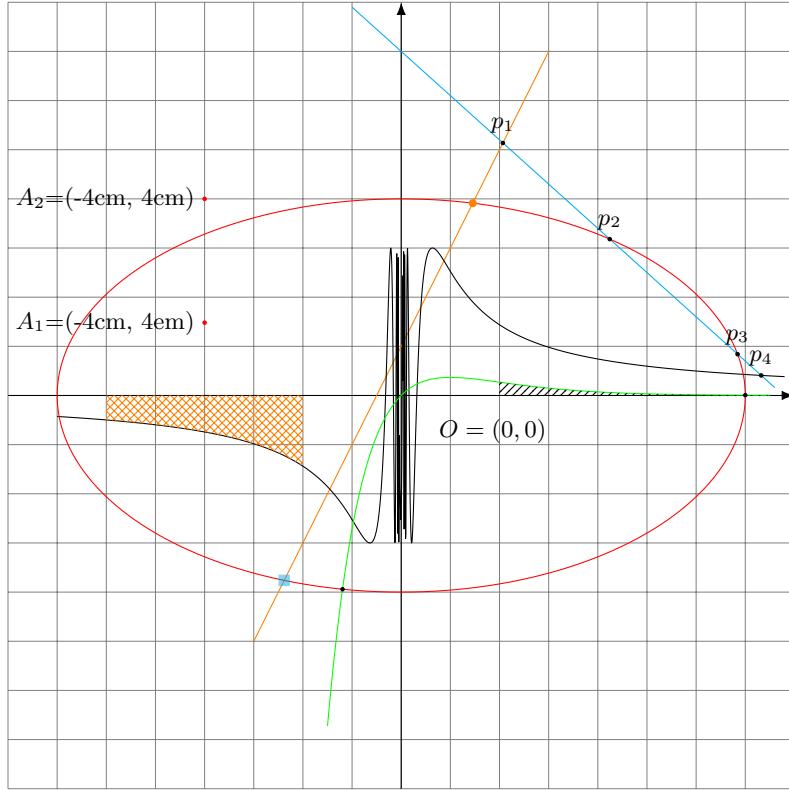


图 4.7: Functions and Points Example

```
\begin{tikzpicture}[font=\small, >=Latex, scale=.65]
%% ==> ztikz draw grid(coordinates)
\ShowGrid[(-8, -8); (8, 8)]
\ShowPoint[opacity=0]{(0, 0)}[$O=(0, 0)$][below right=.5em and 1em]
\ShowAxis {(-8, 0); (8, 0)}
\ShowAxis {(-8, 0); (8, 0)}

% draw function
\Plot[-3:3] [orange] {2*x+1}
\Plot[-1:7.6] [cyan] {- .9*x+7}

%% ==> ztikz Plot functions
% simple 2d-plot
\PlotPrecise{plot}{1500}
\Plot[-7:7.8]{3*sin(1/x)}
\Plot[-1.5:7.5] [green, name path=exp] {x*exp(-x)}
% param plot
\ParamPlot[0:2*pi] [red, name path=ellipse]{7*sin(t), 4*cos(t)}
```

```

%%% ==> Fill region
\begin{scope}
  \clip (2, 0) rectangle (8, 1);
  \fill[pattern=north east lines] plot file{\gnudata{4}};
\end{scope}
\begin{scope}
  \clip (-6, 0) rectangle (-2, -2);
  \fill[pattern=crosshatch, pattern color=orange] plot file{\gnudata{3}} -- (-2,
  0) -- (-6, 0);
\end{scope}

% default tikz unit
>ShowPoint[color=red]{(-4cm, 4em); (-4cm, 4cm)}[$A_1$=(-4cm, 4em); $A_2$=(-4cm,
4cm)] [left]

% plot shape and other properties
>ShowPoint[radius=3pt, color=blue, opacity=.5, type=square*, color=cyan] {(-2.380,
-3.761)}
>ShowPoint[color=orange, opacity=1, radius=2pt] {(1.456, 3.912)}

% ==> find intersection
>ShowIntersection{exp; ellipse}{2}
>ShowPoint {(2.068, 5.137); (4.242, 3.181); (6.843, 0.840); (7.324, 0.408)}
    [$p_1$; $p_2$; $p_3$; $p_4$; $p_5$; $p_6$; $p_7$] [above]
\end{tikzpicture}

```

#### 4.1.8 Example 8

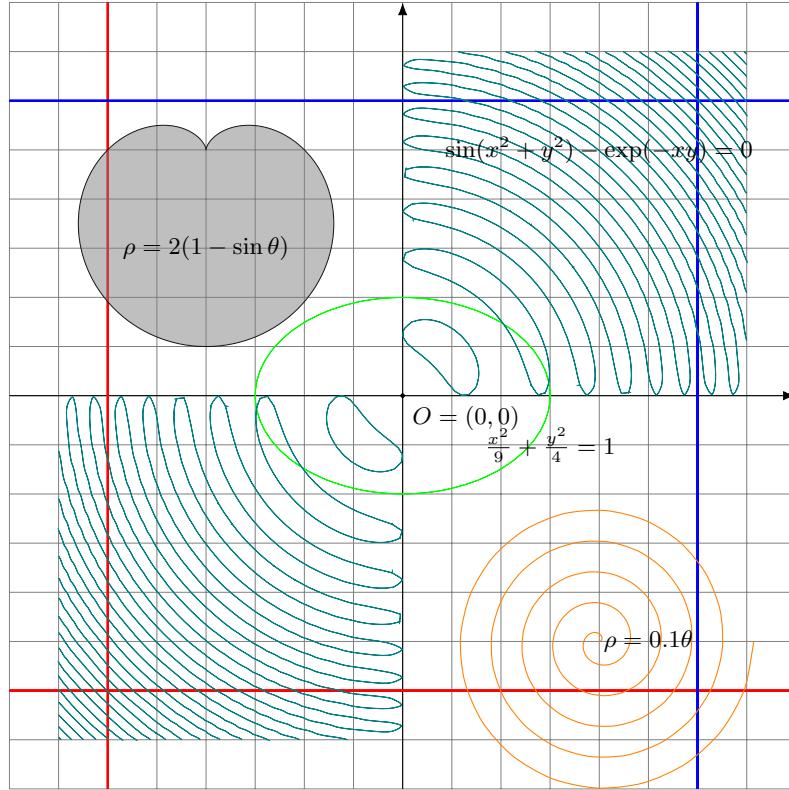


图 4.8: Polar and Contour Example

```
\begin{tikzpicture}[font=\small, >=Latex, scale=.65]
%% ==> ztikz draw grid(coordinates)
\ShowGrid[(-8, -8); (8, 8)]
\ShowPoint{(0, 0)}[$0=(0, 0)$][below right]
\ShowAxis {(-8, 0); (8, 0)}
\ShowAxis {(-8, 0); (0, 8)}
\ShowAxis {(0, -8); (0, 8)}

% contour plot
\PlotPrecise{contour}[long]{10}
\ContourPlot[-8:8; -8:8] [thick, red]{x+6}
\ContourPlot[-8:8; -8:8] [thick, red]{y+6}
\ContourPlot[-8:8; -8:8] [thick, blue]{y-6}
\ContourPlot[-8:8; -8:8] [thick, blue]{x-6}
\PlotPrecise{contour}[long]{100}
\ContourPlot[-4:4] [green]{x**2/9+y**2/4-1}
\ContourPlot[-7:7; -7:7] [teal]{sin(x**2+y**2)-exp(-x*y)}

% polar plot

```

```
\begin{scope}[xshift=4cm, yshift=-5cm]
  \PolarPlot[0:10*pi][orange]{0.1*t}
\end{scope}
\begin{scope}[xshift=-4cm, yshift=5cm]
  \PolarPlot{2*(1-sin(t))}
  \fill[gray, opacity=.5] plot file {\gnudata{8}};
\end{scope}

% node annotate
>ShowPoint[opacity=0]{(-4, 3); (4, 5); (5, -5); (3, -1)}
[{$\rho=2(1-\sin\theta)$; $\sin(x^2+y^2)-\exp(-xy)=0$; $\rho=0.1\theta$;
$\frac{x^2}{9} + \frac{y^2}{4}=1$}
\end{tikzpicture}
```

### 4.1.9 Example 9

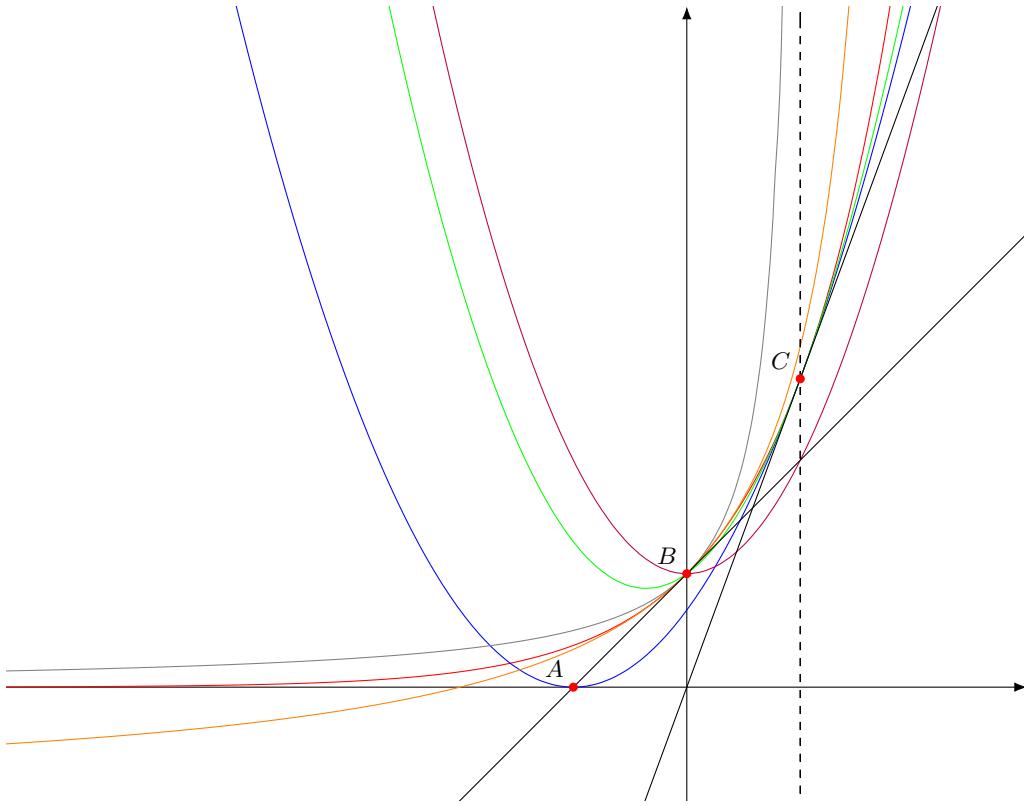


图 4.9: Exp Functions Example

```
\begin{tikzpicture}[>=Latex, font=\small, scale=1.5]
% basic
\clip (-6, -1) rectangle (3, 6);
\ShowAxis[(-8, 0); (3, 0)] \ShowAxis[(0, -1.5); (0, 6)]
% functions
\Plot[-8:5] [red] {\exp(x)}
\Plot[-8:5] [blue] {\exp(1)/4*(x+1)**2}
\Plot[-8:5] [green] {\exp(1)*x + (x-1)**2}
\Plot[-8:5] [purple] {x**2 + 1}
\Plot[-8:0.95] [gray] {1/(1-x)}
\Plot[-8:1.95] [orange] {(2+x)/(2-x)}
\Plot[-8:5] [black] {x+1}
\Plot[-8:8] [black] {\exp(1)*x}
\ContourPlot[0:2;-6:6] [dashed] {x-1}
% points
>ShowPoint[color=red, radius=1pt]{(-1, 0); (0, 1); (1, 2.71828)}[$A$; $B$; $C$] [above left]
\end{tikzpicture}
```

#### 4.1.10 Example 10

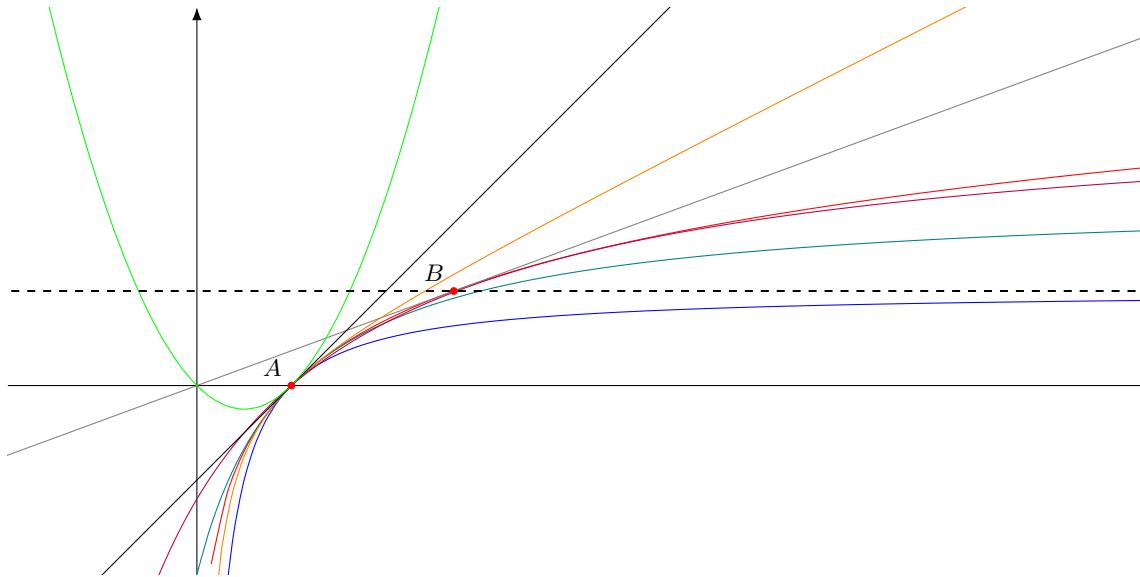


图 4.10: Log Functions Example

```
\begin{tikzpicture}[>=Latex, font=\small, scale=1.25]
% basic
\clip (-2, -2) rectangle (10, 4);
\ShowAxis[(-2, 0); (12, 0)] \ShowAxis[(0, -2); (0, 4)]
% functions
\Plot[-5:12] [red] {log(x)}
\Plot[0:12] [blue] {(x-1)/x}
\Plot[0:12] [teal] {2*(x-1)/(x+1)}
\Plot[-1:12] [purple] {6*(x-1)/(2*x+5)}
\Plot[-5:12] [gray] {x/exp(1)}
\Plot[0.1:12] [orange] {0.5*(x-1/x)}
\Plot[-5:12] [purple] {x-1}
\Plot[-5:12] [green] {x**2-x}
\ContourPlot[-5:12;-6:6] [dashed] {y-1}
% points
>ShowPoint[color=red, radius=1pt] {(1, 0); (2.71828, 1)} [$A$; $B$] [above left]
\end{tikzpicture}
```

#### 4.1.11 Example 11

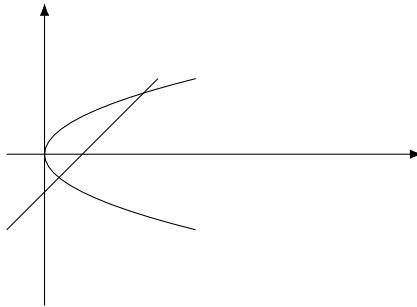


图 4.11: 11 Inverse Function Plot-1

```
\begin{tikzpicture}[scale=0.5, >=Latex]
% axis
\draw[->](0, -1) -- (0, 7);
\draw[->](-4, 0) -- (4, 0);
% plot function
\draw[domain=-2:2] plot(\x, {(\x)^2});
\draw[domain=-2:2] plot(\x, \x+1);
\end{tikzpicture}
```

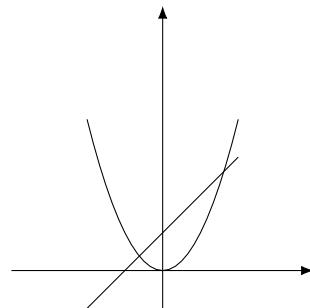


图 4.12: 11 Inverse Function Plot-2

```
\begin{tikzpicture}[scale=0.5, >=Latex]
% axis
\draw[->](0, -1) -- (0, 7);
\draw[->](-4, 0) -- (4, 0);
% plot function
\draw[domain=-2:2] plot(\x, {(\x)^2});
\draw[domain=-2:2] plot(\x, \x+1);
\end{tikzpicture}
```

#### 4.1.12 Example 12

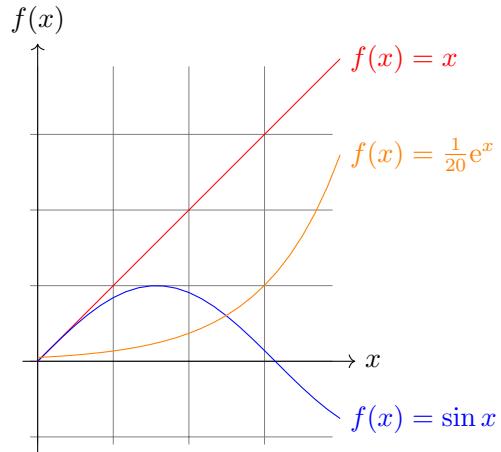


图 4.13: 12 TikZ Built-in Plot

```
\begin{tikzpicture}[domain=0:4]
\draw[very thin,color=gray] (-0.1,-1.1) grid (3.9,3.9);
\draw[->] (-0.2,0) -- (4.2,0) node[right] {x};
\draw[->] (0,-1.2) -- (0,4.2) node[above] {f(x)};
\draw[color=red] plot[id=x] function{x} node[right] {f(x) = x};
\draw[color=blue] plot[id=sin] function{sin(x)} node[right] {f(x) = \sin x};
\draw[color=orange] plot[id=exp] function{0.05*exp(x)} node[right] {f(x) =
\frac{1}{20} \mathrm{e}^x};
\end{tikzpicture}
```

#### 4.1.13 Example 13

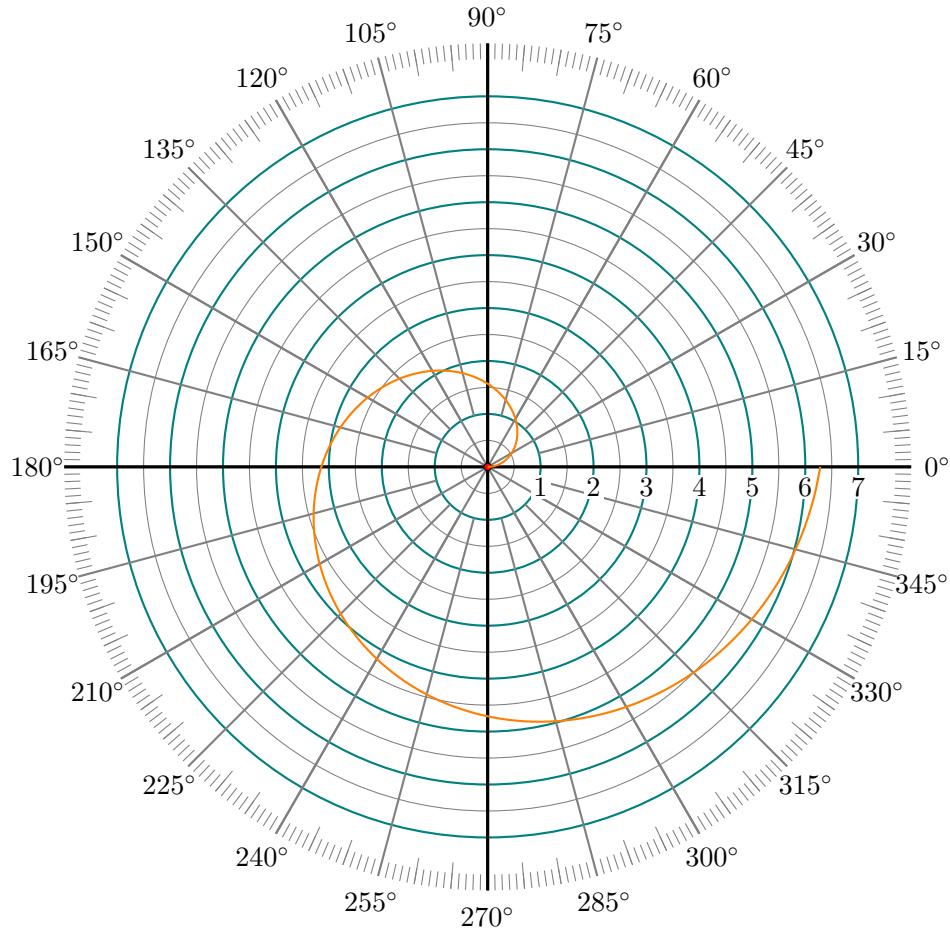


图 4.14: 13 Polar Plot And Coordinates

```
% https://texexample.net/tikz/examples/polar-coordinates-template/
\begin{tikzpicture}[scale=.7]
    %Circles
    \foreach \r in {1, 2,...,7}
        \draw[teal, thick] (0,0) circle (\r);
    \foreach \r in {0.5, 1.5,...,7}
        \draw[gray, thin] (0,0) circle (\r);
    %1° Rays
    \foreach \a in {0, 1,...,359}
        \draw[gray] (\a:7.7) -- (\a:8);
    %5° Rays
    \foreach \a in {0, 5,...,359}
        \draw[gray] (\a:7.5) -- (\a:8);
    %15° Rays

```

```
\foreach \a in {0, 15,...,359}
    \draw[thick,gray] (\a:1) -- (\a:8);
%30° Rays
\foreach \a in {0, 30,...,359}
    \draw[thick,gray] (0, 0) -- (\a:8);
%Radius labels (background filled white)
\foreach \r in {1, 2,...,7}
    \draw (\r,0) node[inner sep=1pt,below=3pt,rectangle,fill=white] {$\r$};
%Main rays
\foreach \a in {0, 90,...,359}
    \draw[very thick] (0, 0) -- (\a:8);
%Angle labels
\foreach \a in {0, 15,...,359}
    \draw (\a: 8.5) node {$\a^\circ$};
%Central point
\draw[fill=red] (0,0) circle(0.7mm);
\PolarPlot[0:2*pi][thick, orange]{t}
\end{tikzpicture}
```

## 4.2 Mathematica

### 4.2.1 Example 1

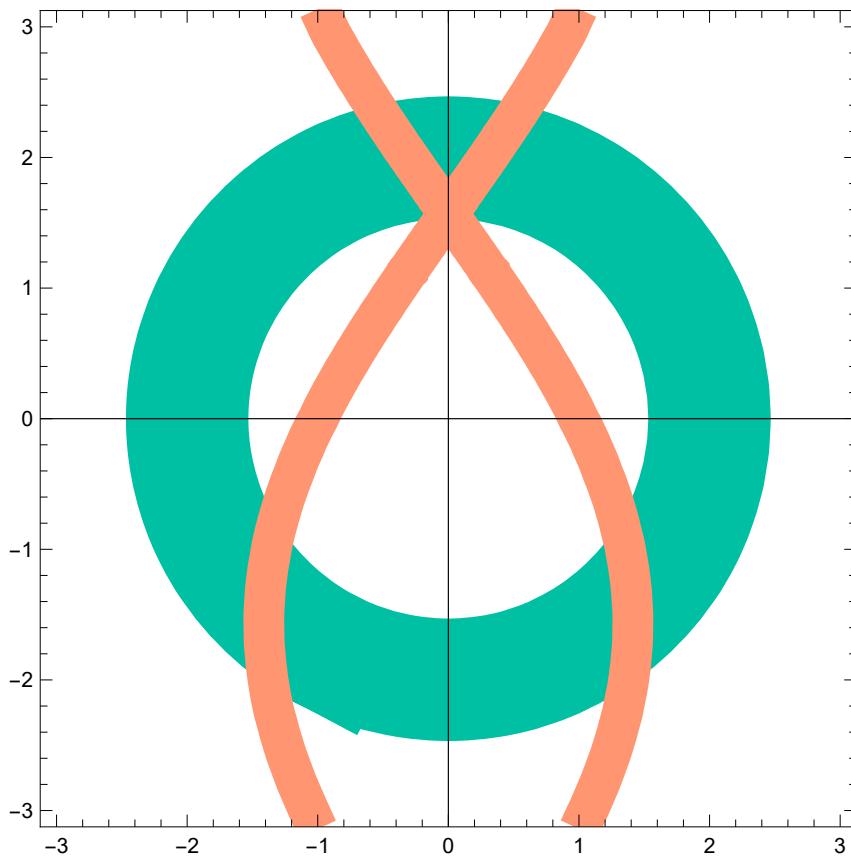


图 4.15: 1 Plot Stoke

```
\begin{wolframGraphics}[width=.75 linewidth]\example_1.wls
fp1 = ContourPlot[x^2 + y^2 == 4, {x, -1.3, 0.6}, {y, -2.4, 3.2},
AspectRatio->(2.4+3.2)/(1.3+0.6), ContourStyle->Red];
fp2 = ContourPlot[x^2 + y^2 == 4, {x, -3, 3}, {y, -3, 3}, AspectRatio->1,
ContourStyle->RGBColor["#00C0A3"], AxesOrigin->\{0, 0\}, Axes->True];

fp3 = ContourPlot[{x^2 + y^2 == 4, x^2 + Sin[y] == 1},
{x, -2.5, 2.5}, {y, -3, 3},
ContourStyle->{
{RGBColor["#00C0A3"], Thickness[0.15]},
{RGBColor["#FF9671"], Thickness[0.05]}
},
AspectRatio->(3+3)/(2.5+2.5),
AxesOrigin->\{0,0\},
```

```
Axes->True,
Frame->False,
AxesStyle->Arrowheads[{0,0.03}]
]
figure = Show[fp2, fp1, fp3];
Export["./ztikz_output/mma_data/example_1.wls.pdf", figure];
\end{wolframGraphics}
```

### 4.2.2 Example 2

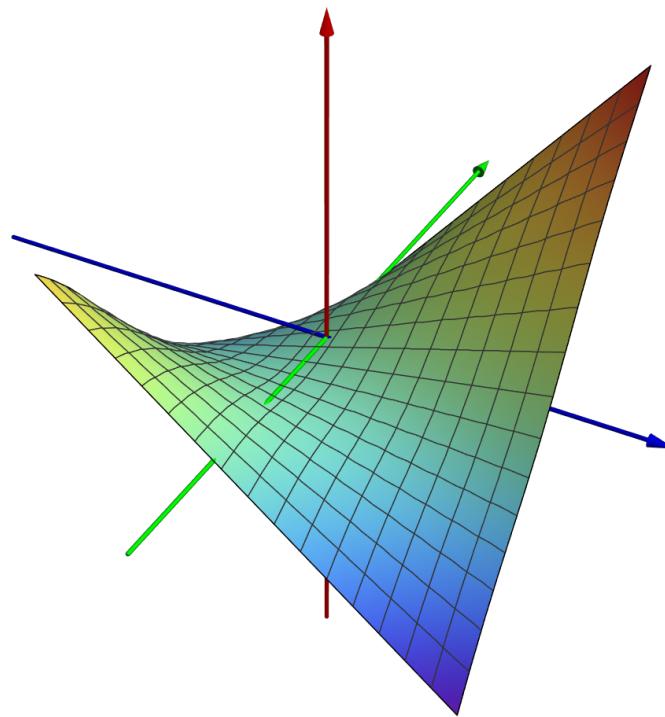


图 4.16: 2 3D Plot

```
\begin{wolframGraphics}[width=.75 linewidth]{example_2.wls}
(* 1. 定义一个产生箭头的命令 *)
arrow[start_, end_, type_] :=
Graphics3D[
{type,
 { Arrowheads[.02], Arrow[Tube[{start, end}, 0.06]] }
},
Boxed->False];

(* 2. 创建三个坐标轴的箭头, 使用颜色进行区分 *)
xaxis = arrow[{-10, 0, 0}, {10, 0, 0}, Blue];
yaxis = arrow[{0, -10, 0}, {0, 10, 0}, Green];
zaxis = arrow[{0, 0, -10}, {0, 0, 10}, Red];

(* 3. 展示在同意坐标轴 *)

```

```
axis = {xaxis, yaxis, zaxis};

(* 4. 绘制一个函数由于测试 *)
fp4 = Plot3D[0.4*x + 0.2*Sin[y] + 0.2*x*y, {x, -5, 7}, {y, -6, 4},
ColorFunction->"Rainbow"];

(* 5. 显示三维函数图像和坐标轴 *)
figure = Show[axis, fp4]
Export["./ztikz_output/mma_data/example_2.wls.pdf", figure];
\[End{wolframGraphics}
```

### 4.2.3 Example 3

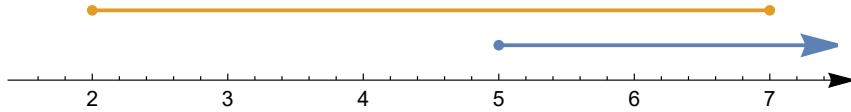


图 4.17: 3 Interval Plot

```
\begin{wolframGraphics}[width=.75 linewidth]example_3.wls
figure = NumberLinePlot[
  {Interval[{5, Infinity}], Interval[{2, 7}]},
  AxesStyle->Arrowheads[{0, 0.03}]
];
Export["./ztikz_output/mma_data/example_3.wls.pdf", figure];
\end{wolframGraphics}
```

#### 4.2.4 Example 4

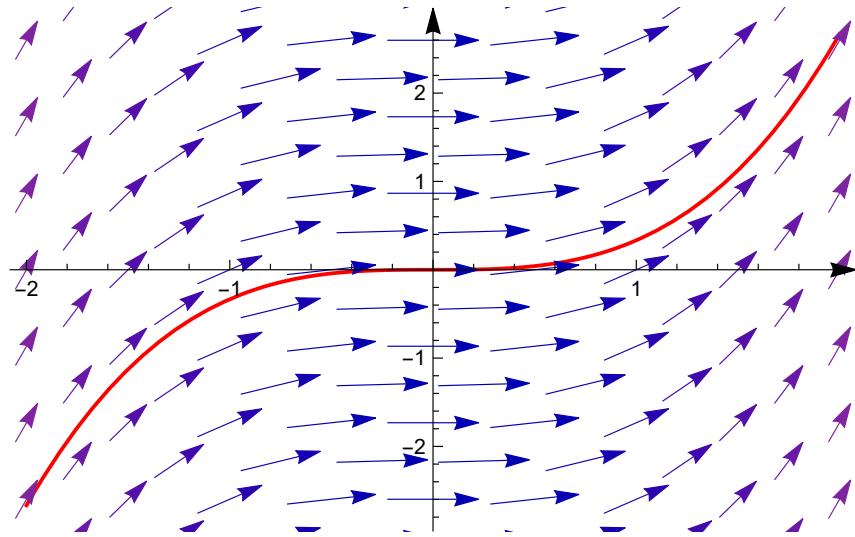


图 4.18: 4 Stream Plot

```
\begin{wolframGraphics}[width=.75 linewidth]example_4.wls}
fvec = VectorPlot[{1, x^2}, {x, -4, 4}, {y, -4, 4}, AxesOrigin->{0, 0}, Axes->False,
Frame->False];
fp6 = Plot[1/3*x^3, {x, -2, 2}, PlotStyle->Red, AxesStyle->Arrowheads[{0, 0.03}]];
figure = Show[fp6, fvec];
Export["./ztikz_output/mma_data/example_4.wls.pdf", figure];
\end{wolframGraphics}
```

#### 4.2.5 Example 5

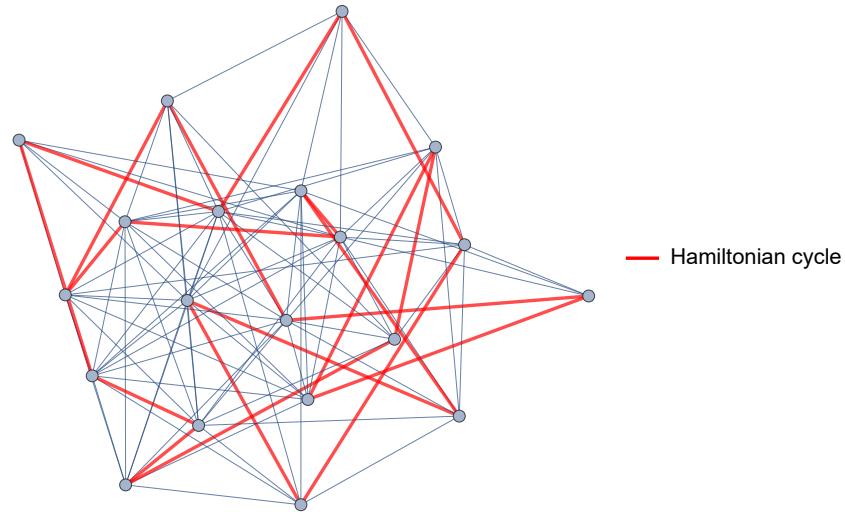


图 4.19: 5 Hamiltonian Graph

```
\begin{wolframGraphics}[width=.75 linewidth]example_5.wls}
g=RandomGraph[{20,100}];
h=FindHamiltonianCycle[g];
figure = Legended[HighlightGraph[g,Style[h,Directive[Thick,Red]]],LineLegend[{Directive[Thick,Red]}, {"Hamiltonian
cycle"}]];
Export["./ztikz_output/mma_data/example_5.wls.pdf", figure];
\end{wolframGraphics}
```

#### 4.2.6 Example 6

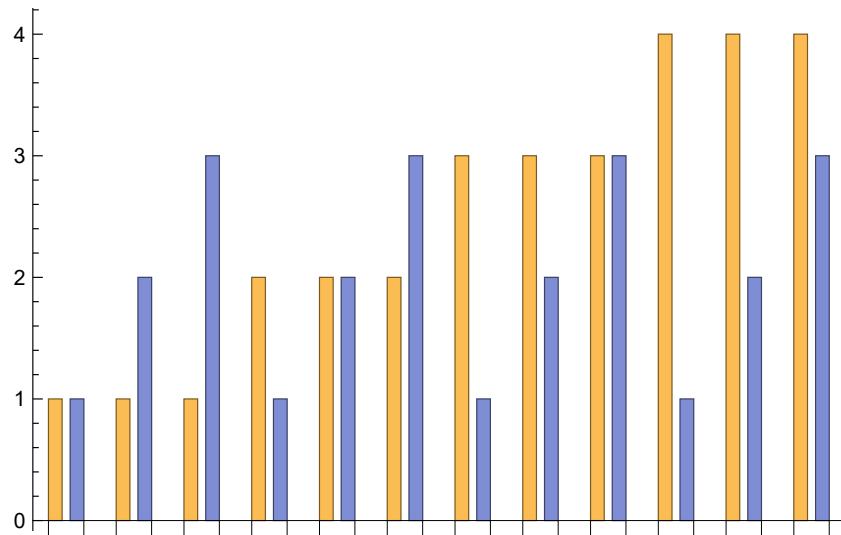


图 4.20: 6 statistic Plot

```
\begin{wolframGraphics}[width=.75 linewidth]\example_6.wls
figure = BarChart[Flatten[Table[{i, j}, {i, 1, 4}, {j, 1, 3}], 1]];
Export["./ztikz_output/mma_data/example_6.wls.pdf", figure];
\end{wolframGraphics}
```

#### 4.2.7 Example 7

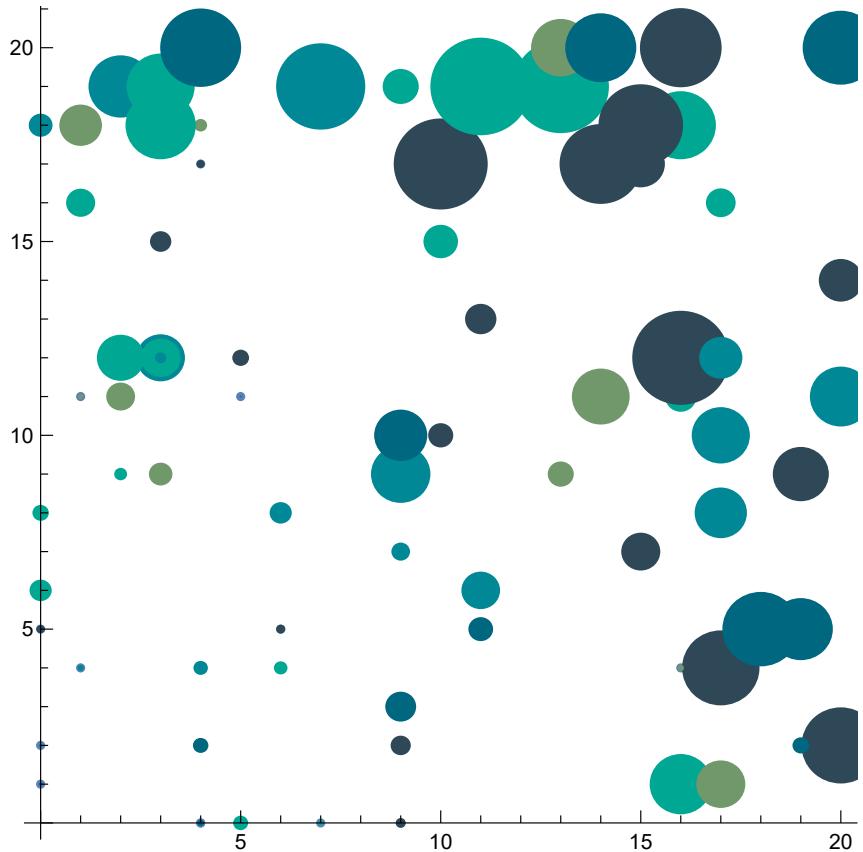


图 4.21: 7 Stream Plot

```
\begin{wolframGraphics}[width=.75 linewidth]example_7.wls}
num = 80;
xls = RandomInteger[{0, 20}, num];
yls = RandomInteger[{0, 20}, num];

xycoor = {xls, yls}/Transpose;
color = {RGBColor["#00A894"], RGBColor["#008896"], RGBColor["#006780"],
RGBColor["#2F4858"], RGBColor["#70986B"]};

disks = Table[
  Graphics[{color[[RandomInteger[{1, 5}]]], Disk[xycoor[[i]], RandomReal[{0,
  0.05}]*#1+RandomReal[{0, 0.05}]*#2&[xycoor[[i]][[1]], xycoor[[i]][[2]]]]}],
{i, 1, num}];
fp91 = disks;
fp92 = ListPlot[xycoor, AspectRatio->(Max[yls])/(Max[xls])];
```

```
figure = Show[fp92, fp91];
Export["./ztikz_output/mma_data/example_7.wls.pdf", figure];
\[End{wolframGraphics}
```

#### 4.2.8 Example 8

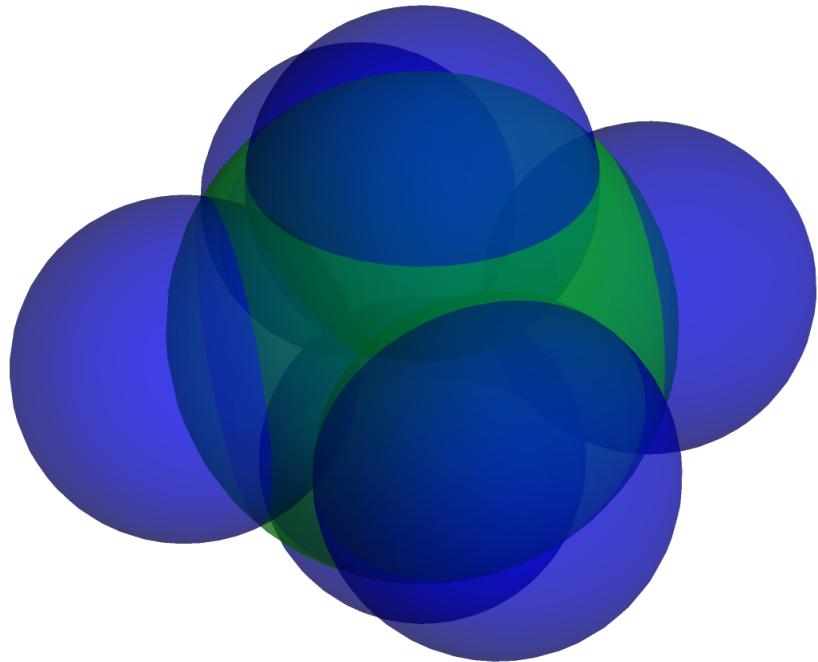


图 4.22: 8 3D Sphere

```
\begin{wolframGraphics}[width=.75 linewidth]\{example_8.wls\}
figure = Graphics3D[{
    Blue, Opacity[0.5], Sphere[{0.5, 0.5, 0}, 0.5],
    Blue, Opacity[0.5], Sphere[{-0.5, -0.5, 0}, 0.5],
    Blue, Opacity[0.5], Sphere[{0.5, -0.5, 0}, 0.5],
    Blue, Opacity[0.5], Sphere[{-0.5, 0.5, 0}, 0.5],
    Blue, Opacity[0.5], Sphere[{0, 0, 0.5}, 0.5],
    Blue, Opacity[0.5], Sphere[{0, 0, -0.5}, 0.5],
    Green, Sphere[{0,0,0}, 0.75]
}, Boxed->False
];
Export["./ztikz_output/mma_data/example_8.wls.pdf", figure];
\end{wolframGraphics}
```

## 4.3 Python

### 4.3.1 Example 1

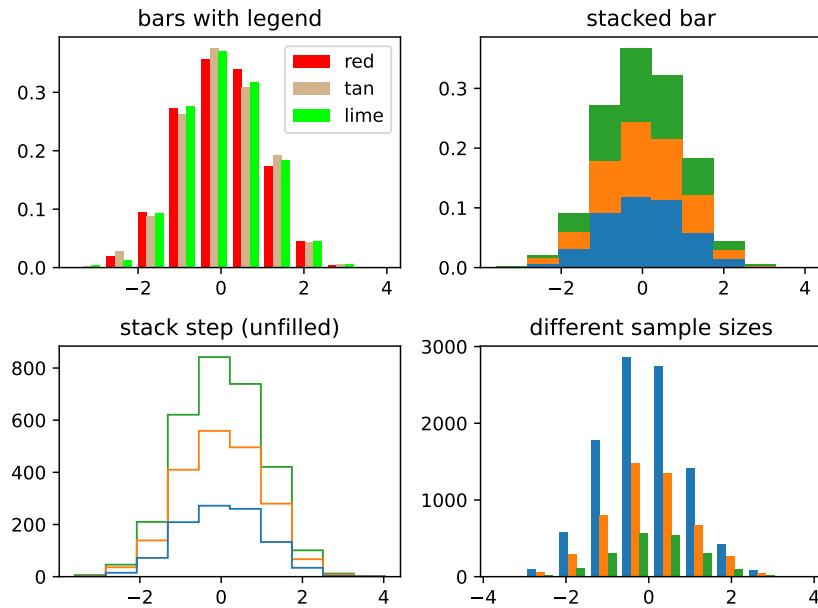


图 4.23: 1 Step Demo

```
\begin{pyfig}[width=.75\linewidth]{example_1.mpl}
# https://matplotlib.org/stable/gallery/lines_bars_and_markers/step_demo.html

import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
import numpy as np

x = np.arange(14)
y = np.sin(x / 2)

plt.plot(x, y + 2, drawstyle='steps', label='steps (=steps-pre)')
plt.plot(x, y + 2, 'o--', color='grey', alpha=0.3)

plt.plot(x, y + 1, drawstyle='steps-mid', label='steps-mid')
plt.plot(x, y + 1, 'o--', color='grey', alpha=0.3)

plt.plot(x, y, drawstyle='steps-post', label='steps-post')
plt.plot(x, y, 'o--', color='grey', alpha=0.3)
```

```
plt.grid(axis='x', color='0.95')
plt.legend(title='Parameter drawstyle:')
plt.title('plt.plot(drawstyle=...)')
\end{pyfig}
```

### 4.3.2 Example 2

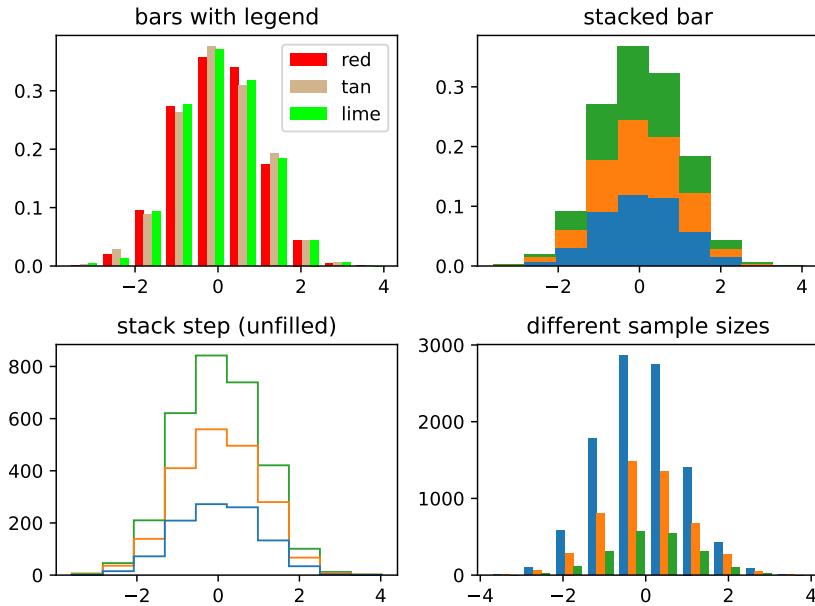


图 4.24: 2 Histogram Demo

```
\begin{pyfig}[width=.75\linewidth]{example_1.mpl}
# https://matplotlib.org/stable/gallery/lines_bars_and_markers/histogram_demo.html

import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
import numpy as np

np.random.seed(19680801)

n_bins = 10
x = np.random.randn(1000, 3)

fig, ((ax0, ax1), (ax2, ax3)) = plt.subplots(nrows=2, ncols=2)

colors = ['red', 'tan', 'lime']
ax0.hist(x, n_bins, density=True, histtype='bar', color=colors, label=colors)
ax0.legend(prop={'size': 10})
ax0.set_title('bars with legend')

ax1.hist(x, n_bins, density=True, histtype='bar', stacked=True)
ax1.set_title('stacked bar')

ax2.hist(x, n_bins, density=True, histtype='stepfilled', color=colors)
ax2.set_title('stack step (unfilled)')

ax3.hist(x, n_bins, density=True, histtype='bar', bins=50)
ax3.set_title('different sample sizes')

plt.show()

```

```
ax2.hist(x, n_bins, histtype='step', stacked=True, fill=False)
ax2.set_title('stack step (unfilled)')

# Make a multiple-histogram of data-sets with different length.
x_multi = [np.random.randn(n) for n in [10000, 5000, 2000]]
ax3.hist(x_multi, n_bins, histtype='bar')
ax3.set_title('different sample sizes')

fig.tight_layout()
\end{pyfig}
```

### 4.3.3 Example 3

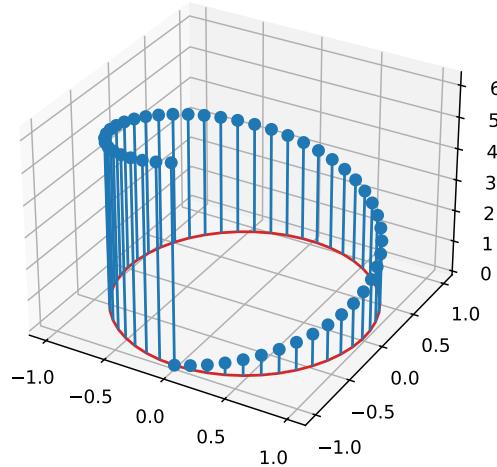


图 4.25: 3 3D Stem Demo

```
\begin{pyfig}[width=.75\linewidth]{example_3.mpl}
# https://matplotlib.org/stable/gallery/mplot3d/stem3d_demo.html

import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
import numpy as np

theta = np.linspace(0, 2*np.pi)
x = np.cos(theta - np.pi/2)
y = np.sin(theta - np.pi/2)
z = theta

fig, ax = plt.subplots(subplot_kw=dict(projection='3d'))
ax.stem(x, y, z)
\end{pyfig}
```

#### 4.3.4 Example 4

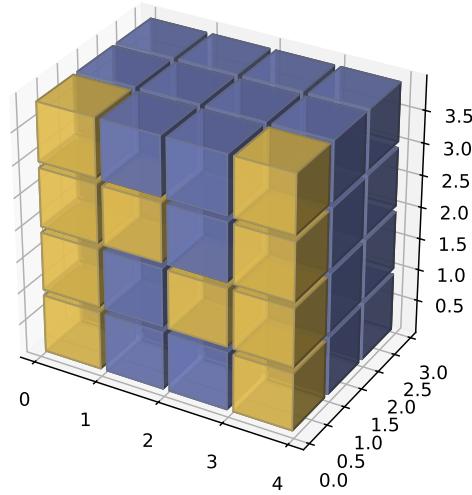


图 4.26: 4 3D Numpy Logo

```
\begin{pyfig}[width=.75\linewidth]{example_4.mpl}
# https://matplotlib.org/stable/gallery/mplot3d/voxels_numpy_logo.html
import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
import numpy as np

def explode(data):
    size = np.array(data.shape)*2
    data_e = np.zeros(size - 1, dtype=data.dtype)
    data_e[::2, ::2, ::2] = data
    return data_e

# build up the numpy logo
n_voxels = np.zeros((4, 3, 4), dtype=bool)
n_voxels[0, 0, :] = True
n_voxels[-1, 0, :] = True
n_voxels[1, 0, 2] = True
n_voxels[2, 0, 1] = True
facecolors = np.where(n_voxels, '#FFD65DC0', '#7A88CC0')
edgecolors = np.where(n_voxels, '#BFAB6E', '#7D84A6')
```

```
filled = np.ones(n_voxels.shape)

# upscale the above voxel image, leaving gaps
filled_2 = explode(filled)
fcolors_2 = explode(facecolors)
ecolors_2 = explode(edgecolors)

# Shrink the gaps
x, y, z = np.indices(np.array(filled_2.shape) + 1).astype(float) // 2
x[0::2, :, :] += 0.05
y[:, 0::2, :] += 0.05
z[:, :, 0::2] += 0.05
x[1::2, :, :] += 0.95
y[:, 1::2, :] += 0.95
z[:, :, 1::2] += 0.95

ax = plt.figure().add_subplot(projection='3d')
ax.voxels(x, y, z, filled_2, facecolors=fcolors_2, edgecolors=ecolors_2)
ax.set_aspect('equal')
\end{pyfig}
```

## **第二部分**

### **索引与参考**

# CHAPTER 5

## 部分命令/名词索引

<marker option>	35, 40	amsthm	14
\BarPlot	, 32, 44	13draw	, 14
\ContourPlot	, 32, 42	leftbar	, 18
\ListPlot	, 32, 44	math-alias	, 12
\ParamPlot	, 32	pyfig	, 53
\PlotPrecise	, 32, 43, 44	toc	, 12
\Plot	, 32	wolframGraphics	, 56
\PolarPlot	, 32	xfp	, 60
\Polygon	, 32	basic packages	, 10
\ShadePlot	, 32, 44	language packages	, 10
\ShowAxis	, 32, 38	math packages	, 11
\ShowGrid	, 32, 40	optional packages	, 11
\ShowIntersection	, 32, 43	pycode	, 62
\ShowPoint	, 32, 40	zLATEX	, 2
\StairsPlot	, 32, 44	zTikZ	, 3
\StemPlot	, 32, 44	字体大小	, 11
\cref	, 19, 24	数学字体	, 11
\gnudata	, 32	数学环境主题	, 19
\py	, 61	模板语言	, 11
\wolframDSolve	, 68	边注	, 11
\wolframSolve	, 67	配置选项	, 11
\wolfram	, 64, 66	页面设置	, 11
\xAxis	, 39		
\yAxis	, 39		
\zlatexColorSetup	, 14		
\zlatexFramed	, 15		
\zlatexOptions	, 10		
\zlatexUpdateCounterAfter	, 14		

# CHAPTER 6

---

## 参考文献

- [1] Ahlfors. *Complex Analysis*. New York: McGraw-Hill, 1953.
- [2] Auslander and MacKenzie. *Introduction to Differentiable Manifolds*. New York: McGraw-Hill, 1963.

# CHAPTER 7

---

## TikZ Built-in Functions

本节主要截取了 TikZ 官方文档中的部分内容，主要涵盖 TikZ 内置函数的绘制的内容。相信这一部分的内容可以帮助读者更好的理解 zTikZ 宏包的工作原理，同时也能够使得 zTikZ 宏包的使用更加的便捷。

## 22 Plots of Functions

A warning before we get started: *If you are looking for an easy way to create a normal plot of a function with scientific axes, ignore this section and instead look at the `pgfplots` package or at the `datavisualization` command from Part VI.*

### 22.1 Overview

TikZ can be used to create plots of functions, a job that is normally handled by powerful programs like GNUPLOT or MATHEMATICA. These programs can produce two different kinds of output: First, they can output a complete plot picture in a certain format (like PDF) that includes all low-level commands necessary for drawing the complete plot (including axes and labels). Second, they can usually also produce “just plain data” in the form of a long list of coordinates. Most of the powerful programs consider it a to be “a bit boring” to just output tabled data and very much prefer to produce fancy pictures. Nevertheless, when coaxed, they can also provide the plain data.

The advantage of creating plots directly using TikZ is *consistency*: Plots created using TikZ will automatically have the same styling and fonts as those used in the rest of a document – something that is hard to do right when an external program gets involved. Other problems people encounter with external programs include: Formulas will look different, if they can be rendered at all; line widths will usually be too thick or too thin; scaling effects upon inclusion can create a mismatch between sizes in the plot and sizes in the text; the automatic grid generated by most programs is mostly distracting; the automatic ticks generated by most programs are cryptic numerics (try adding a tick reading “ $\pi$ ” at the right point); most programs make it very easy to create “chart junk” in a most convenient fashion; arrows and plot marks will almost never match the arrows used in the rest of the document. This list is not exhaustive, unfortunately.

There are basically three ways of creating plots using TikZ:

1. Use the `plot` path operation. How this works is explained in the present section. This is the most “basic” of the three options and forces you to do a lot of things “by hand” like adding axes or ticks.
2. Use the `datavisualization` path command, which is documented in Part VI. This command is much more powerful than the `plot` path operation and produces complete plots including axes and ticks. The downside is that you cannot use it to “just” quickly plot a simple curve (or, more precisely, it is hard to use it in this way).
3. Use the `pgfplots` package, which is basically an alternative to the `datavisualization` command. While the underlying philosophy of this package is not as “ambitious” as that of the command `datavisualization`, it is somewhat more mature, has a simpler design, and wider support base.

### 22.2 The Plot Path Operation

The `plot` path operation can be used to append a line or curve to the path that goes through a large number of coordinates. These coordinates are either given in a simple list of coordinates, read from some file, or they are computed on the fly.

The syntax of the `plot` comes in different versions.

```
\path ... --plot{further arguments} ...;
```

This operation plots the curve through the coordinates specified in the *<further arguments>*. The current (sub)path is simply continued, that is, a line-to operation to the first point of the curve is implicitly added. The details of the *<further arguments>* will be explained in a moment.

```
\path ... plot{further arguments} ...;
```

This operation plots the curve through the coordinates specified in the *<further arguments>* by first “moving” to the first coordinate of the curve.

The *<further arguments>* are used in different ways to specifying the coordinates of the points to be plotted:

1. `--plot[<local options>]`**coordinates**{*(coordinate 1)**(coordinate 2)*...*(coordinate n)*}
2. `--plot[<local options>]`**file**{*filename*}

3. `--plot[<local options>]<coordinate expression>`
4. `--plot[<local options>]function{<gnuplot formula>}`

These different ways are explained in the following.

## 22.3 Plotting Points Given Inline

Points can be given directly in the T<sub>E</sub>X-file as in the following example:



```
\tikz \draw plot coordinates {(0,0) (1,1) (2,0) (3,1) (2,1) (10:2cm)};
```

Here is an example showing the difference between `plot` and `--plot`:



```
\begin{tikzpicture}
\draw (0,0) -- (1,1) plot coordinates {(2,0) (4,0)};
\draw[color=red,xshift=5cm]
(0,0) -- (1,1) -- plot coordinates {(2,0) (4,0)};
\end{tikzpicture}
```

## 22.4 Plotting Points Read From an External File

The second way of specifying points is to put them in an external file named `<filename>`. Currently, the only file format that TikZ allows is the following: Each line of the `<filename>` should contain one line starting with two numbers, separated by a space. A line may also be empty or, if it starts with # or % it is considered empty. For such lines, a “new data set” is started, typically resulting in a new subpath being started in the plot (see Section 112.2.2 on how to change this behavior, if necessary). For lines containing two numbers, they must be separated by a space. They may be followed by arbitrary text, which is ignored, *except* if it is o or u. In the first case, the point is considered to be an *outlier* and normally also results in a new subpath being started. In the second case, the point is considered to be *undefined*, which also results in a new subpath being started. Again, see Section 112.2.2 on how to change this, if necessary. (This is exactly the format that GNUPLOT produces when you say `set table`.)

```
\tikz \draw plot[mark=x,smooth] file {plots/pgfmanual-sine.table};
```

The file `plots/pgfmanual-sine.table` reads:

```
#Curve 0, 20 points
#x y type
0.00000 0.00000 i
0.52632 0.50235 i
1.05263 0.86873 i
1.57895 0.99997 i
...
9.47368 -0.04889 i
10.00000 -0.54402 i
```

It was produced from the following source, using `gnuplot`:

```
set table "../plots/pgfmanual-sine.table"
set format "%5f"
set samples 20
plot [x=0:10] sin(x)
```

The `<local options>` of the `plot` operation are local to each plot and do not affect other plots “on the same path”. For example, `plot[yshift=1cm]` will locally shift the plot 1cm upward. Remember, however, that most options can only be applied to paths as a whole. For example, `plot[red]` does not have the effect of making the plot red. After all, you are trying to “locally” make part of the path red, which is not possible.

## 22.5 Plotting a Function

When you plot a function, the coordinates of the plot data can be computed by evaluating a mathematical expression. Since PGF comes with a mathematical engine, you can specify this expression and then have TikZ produce the desired coordinates for you, automatically.

Since this case is quite common when plotting a function, the syntax is easy: Following the `plot` command and its local options, you directly provide a *<coordinate expression>*. It looks like a normal coordinate, but inside you may use a special macro, which is `\x` by default, but this can be changed using the `variable` option. The *<coordinate expression>* is then evaluated for different values for `\x` and the resulting coordinates are plotted.

Note that you will often have to put the *x*- or *y*-coordinate inside braces, namely whenever you use an expression involving a parenthesis.

The following options influence how the *<coordinate expression>* is evaluated:

`/tikz/variable=<macro>` (no default, initially `\x`)

Sets the macro whose value is set to the different values when *<coordinate expression>* is evaluated.

`/tikz/samples=<number>` (no default, initially 25)

Sets the number of samples used in the plot.

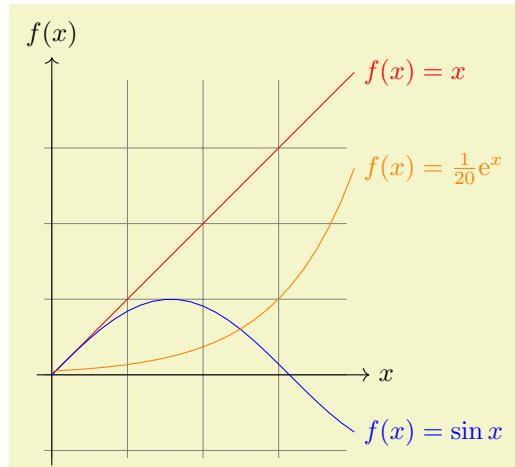
`/tikz/domain=<start>:<end>` (no default, initially -5:5)

Sets the domain from which the samples are taken.

`/tikz/samples at=<sample list>` (no default)

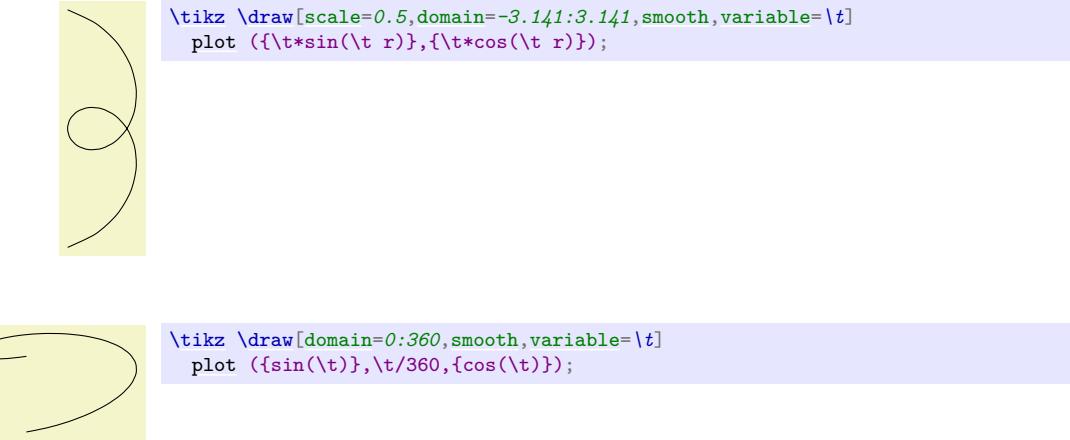
This option specifies a list of positions for which the variable should be evaluated. For instance, you can say `samples at={1,2,8,9,10}` to have the variable evaluated exactly for values 1, 2, 8, 9, and 10. You can use the `\foreach` syntax, so you can use ... inside the *<sample list>*.

When this option is used, the `samples` and `domain` option are overruled. The other way round, setting either `samples` or `domain` will overrule this option.



```
\begin{tikzpicture}[domain=0:4]
\draw[very thin,color=gray] (-0.2,0) -- (4.2,0) node[right] {$x$};
\draw[->] (0,-1.2) -- (0,4.2) node[above] {$f(x)$};

\draw[color=red] plot (\x,\x) node[right] {$f(x) = x$};
% \x r means to convert '\x' from degrees to _radians:
\draw[color=blue] plot (\x,{sin(\x r)}) node[right] {$f(x) = \sin x$};
\draw[color=orange] plot (\x,{0.05*exp(\x)}) node[right] {$f(x) = \frac{1}{20} e^x$};
\end{tikzpicture}
```



## 22.6 Plotting a Function Using Gnuplot

Often, you will want to plot points that are given via a function like  $f(x) = x \sin x$ . Unfortunately, TeX does not really have enough computational power to generate the points of such a function efficiently (it is a text processing program, after all). However, if you allow it, TeX can try to call external programs that can easily produce the necessary points. Currently, TikZ knows how to call GNUPLOT.

When TikZ encounters your operation `plot[id=<id>] function{x*sin(x)}` for the first time, it will create a file called `<prefix><id>.gnuplot`, where `<prefix>` is `\jobname`. by default, that is, the name of your main `.tex` file. If no `<id>` is given, it will be empty, which is alright, but it is better when each plot has a unique `<id>` for reasons explained in a moment. Next, TikZ writes some initialization code into this file followed by `plot x*sin(x)`. The initialization code sets up things such that the `plot` operation will write the coordinates into another file called `<prefix><id>.table`. Finally, this table file is read as if you had said `plot file{<prefix><id>.table}`.

For the plotting mechanism to work, two conditions must be met:

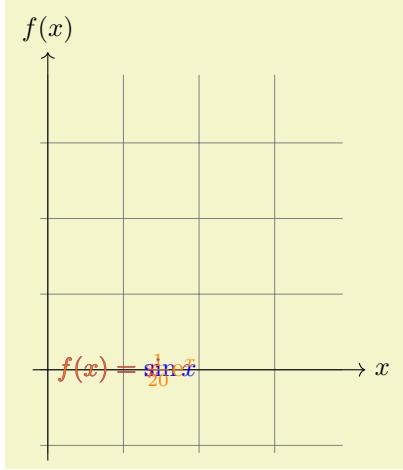
1. You must have allowed TeX to call external programs. This is often switched off by default since this is a security risk (you might, without knowing, run a TeX file that calls all sorts of “bad” commands). To enable this “calling external programs” a command line option must be given to the TeX program. Usually, it is called something like `shell-escape` or `enable-write18`. For example, for my `pdflatex` the option `--shell-escape` can be given.
2. You must have installed the `gnuplot` program and TeX must find it when compiling your file.

Unfortunately, these conditions will not always be met. Especially if you pass some source to a coauthor and the coauthor does not have GNUPLOT installed, he or she will have trouble compiling your files.

For this reason, TikZ behaves differently when you compile your graphic for the second time: If upon reaching `plot[id=<id>] function{...}` the file `<prefix><id>.table` already exists *and* if the `<prefix><id>.gnuplot` file contains what TikZ thinks that it “should” contain, the `.table` file is immediately read without trying to call a `gnuplot` program. This approach has the following advantages:

1. If you pass a bundle of your `.tex` file and all `.gnuplot` and `.table` files to someone else, that person can TeX the `.tex` file without having to have `gnuplot` installed.
2. If the `\write18` feature is switched off for security reasons (a good idea), then, upon the first compilation of the `.tex` file, the `.gnuplot` will still be generated, but not the `.table` file. You can then simply call `gnuplot` “by hand” for each `.gnuplot` file, which will produce all necessary `.table` files.
3. If you change the function that you wish to plot or its domain, TikZ will automatically try to regenerate the `.table` file.
4. If, out of laziness, you do not provide an `id`, the same `.gnuplot` will be used for different plots, but this is not a problem since the `.table` will automatically be regenerated for each plot on-the-fly. *Note: If you intend to share your files with someone else, always use an id, so that the file can be typeset without having GNUPLOT installed.* Also, having unique ids for each plot will improve compilation speed since no external programs need to be called, unless it is really necessary.

When you use `plot function{(gnuplot formula)}`, the `(gnuplot formula)` must be given in the `gnuplot` syntax, whose details are beyond the scope of this manual. Here is the ultra-condensed essence: Use `x` as the variable and use the C-syntax for normal plots, use `t` as the variable for parametric plots. Here are some examples:



```
\begin{tikzpicture}[domain=0:4]
\draw[very thin,color=gray] (-0.1,-1.1) grid (3.9,3.9);
\draw[->] (-0.2,0) -- (4.2,0) node[right] {$x$};
\draw[->] (0,-1.2) -- (0,4.2) node[above] {$f(x)$};

\draw[color=red] plot[id=x] function{x} node[right] {$f(x) = x$};
\draw[color=blue] plot[id=sin] function{sin(x)} node[right] {$f(x) = \sin x$};
\draw[color=orange] plot[id=exp] function{0.05*exp(x)} node[right] {$f(x) = \frac{1}{20} e^x$};
\end{tikzpicture}
```

The plot is influenced by the following options: First, the options `samples` and `domain` explained earlier. Second, there are some more specialized options.

`/tikz/parametric=<boolean>` (default `true`)

Sets whether the plot is a parametric plot. If true, then `t` must be used instead of `x` as the parameter and two comma-separated functions must be given in the `(gnuplot formula)`. An example is the following:

```
\tikz \draw[scale=0.5,domain=-3.141:3.141,smooth]
plot[parametric,id=parametric-example] function{t*sin(t),t*cos(t)};
```

`/tikz/range=<start>:<end>` (no default)

This key sets the range of the plot. If set, all points whose `y`-coordinates lie outside this range will be considered to be outliers and will cause jumps in the plot, by default:

```
\tikz \draw[scale=0.5,domain=-3.141:3.141, samples=100, smooth, range=-3:3]
plot[id=tan-example] function{tan(x)};
```

`/tikz/yrange=<start>:<end>` (no default)

Same as `range`.

`/tikz/xrange=<start>:<end>` (no default)

Set the `x`-range. This makes sense only for parametric plots.

```
\tikz \draw[scale=0.5,domain=-3.141:3.141,smooth,xrange=0:1]
plot[parametric,id=parametric-example-cut] function{t*sin(t),t*cos(t)};
```

`/tikz/id=<id>` (no default)

Sets the identifier of the current plot. This should be a unique identifier for each plot (though things will also work if it is not, but not as well, see the explanations above). The `<id>` will be part of a filename, so it should not contain anything fancy like `*` or `$`.

`/tikz/prefix=<prefix>` (no default)

The `<prefix>` is put before each plot file name. The default is `\jobname.`, but if you have many plots, it might be better to use, say `plots/` and have all plots placed in a directory. You have to create the directory yourself.

`/tikz/raw gnuplot` (no value)

This key causes the `<gnuplot formula>` to be passed on to GNUPLOT without setting up the samples or the `plot` operation. Thus, you could write

```
plot[raw gnuplot,id=raw-example] function[set samples 25; plot sin(x)]
```

This can be useful for complicated things that need to be passed to GNUPLOT. However, for really complicated situations you should create a special external generating GNUPLOT file and use the `file-` syntax to include the table “by hand”.

The following styles influence the plot:

`/tikz/every plot` (style, initially empty)

This style is installed in each plot, that is, as if you always said

```
plot[every plot,...]
```

This is most useful for globally setting a prefix for all plots by saying:

```
\tikzset{every plot/.style={prefix=plots/}}
```

## 22.7 Placing Marks on the Plot

As we saw already, it is possible to add *marks* to a plot using the `mark` option. When this option is used, a copy of the plot mark is placed on each point of the plot. Note that the marks are placed *after* the whole path has been drawn/filled/shaded. In this respect, they are handled like text nodes.

In detail, the following options govern how marks are drawn:

`/tikz/mark=<mark mnemonic>` (no default)

Sets the mark to a mnemonic that has previously been defined using the `\pgfdeclareplotmark`. By default, `*`, `+`, and `x` are available, which draw a filled circle, a plus, and a cross as marks. Many more marks become available when the library `plotmarks` is loaded. Section 65.6 lists the available plot marks.

One plot mark is special: the `ball` plot mark is available only in TikZ. The `ball color` option determines the ball’s color. Do not use this option with a large number of marks since it will take very long to render in PostScript.

Option	Effect
<code>mark=ball</code>	

`/tikz/mark repeat=<r>` (no default)

This option tells TikZ that only every *r*th mark should be drawn.

```
\tikz \draw plot[mark=x,mark repeat=3,smooth] file {plots/pgfmanual-sine.table};
```

`/tikz/mark phase=<p>` (no default)

This option tells TikZ that the first mark to be draw should be the *p*th, followed by the  $(p+r)$ th, then the  $(p+2r)$ th, and so on.

```
\tikz \draw plot[mark=x,mark repeat=3,mark phase=6,smooth] file {plots/pgfmanual-sine.table};
```

`/tikz/mark indices=<list>` (no default)

This option allows you to specify explicitly the indices at which a mark should be placed. Counting starts with 1. You can use the `\foreach` syntax, that is, `...` can be used.

```
\tikz \draw plot[mark=x,mark indices={1,4,...,10,11,12,...,16,20},smooth]
  file {plots/pgfmanual-sine.table};
```

`/tikz/mark size=<dimension>` (no default)

Sets the size of the plot marks. For circular plot marks, `<dimension>` is the radius, for other plot marks `<dimension>` should be about half the width and height.

This option is not really necessary, since you achieve the same effect by specifying `scale=<factor>` as a local option, where `<factor>` is the quotient of the desired size and the default size. However, using `mark size` is a bit faster and more natural.

`/tikz/every mark` (style, no value)

This style is installed before drawing plot marks. For example, you can scale (or otherwise transform) the plot mark or set its color.

`/tikz/mark options=<options>` (no default)

Redefines `every mark` such that it sets `{<options>}`.

```
\tikz \fill[fill=blue!20]
  plot[mark=triangle*,mark options={color=blue,rotate=180}]
    file{plots/pgfmanual-sine.table} |- (0,0);
```

`/tikz/no marks` (style, no value)

Disables markers (the same as `mark=none`).

`/tikz/no markers` (style, no value)

Disables markers (the same as `mark=none`).

## 22.8 Smooth Plots, Sharp Plots, Jump Plots, Comb Plots and Bar Plots

There are different things the `plot` operation can do with the points it reads from a file or from the inlined list of points. By default, it will connect these points by straight lines. However, you can also use options to change the behavior of `plot`.

`/tikz/sharp plot` (no value)

This is the default and causes the points to be connected by straight lines. This option is included only so that you can “switch back” if you “globally” install, say, `smooth`.

`/tikz/smooth` (no value)

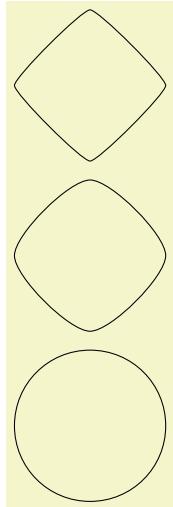
This option causes the points on the path to be connected using a smooth curve:

```
\tikz\draw plot[smooth] file{plots/pgfmanual-sine.table};
```

Note that the smoothing algorithm is not very intelligent. You will get the best results if the bending angles are small, that is, less than about 30° and, even more importantly, if the distances between points are about the same all over the plotting path.

`/tikz/tension=<value>` (no default)

This option influences how “tight” the smoothing is. A lower value will result in sharper corners, a higher value in more “round” curves. A value of 1 results in a circle if four points at quarter-positions on a circle are given. The default is 0.55. The “correct” value depends on the details of plot.



```
\begin{tikzpicture}[smooth cycle]
\draw plot [tension=0.2]
coordinates{(0,0) (1,1) (2,0) (1,-1)};
\draw[yshift=-2.25cm] plot [tension=0.5]
coordinates{(0,0) (1,1) (2,0) (1,-1)};
\draw[yshift=-4.5cm] plot [tension=1]
coordinates{(0,0) (1,1) (2,0) (1,-1)};
\end{tikzpicture}
```

### /tikz/smooth cycle

(no value)

This option causes the points on the path to be connected using a closed smooth curve.



```
\tikz[scale=0.5]
\draw plot[smooth cycle] coordinates{(0,0) (1,0) (2,1) (1,2)}
plot coordinates{(0,0) (1,0) (2,1) (1,2)} -- cycle;
```

### /tikz/const plot

(no value)

This option causes the points on the path to be connected using piecewise constant series of lines:

```
\tikz\draw plot[const plot] file{plots/pgfmanual-sine.table};
```

### /tikz/const plot mark left

(no value)

Just an alias for /tikz/const plot.

```
\tikz\draw plot[const plot mark left,mark=*] file{plots/pgfmanual-sine.table};
```

### /tikz/const plot mark right

(no value)

A variant of /tikz/const plot which places its mark on the right ends:

```
\tikz\draw plot[const plot mark right,mark=*] file{plots/pgfmanual-sine.table};
```

### /tikz/const plot mark mid

(no value)

A variant of /tikz/const plot which places its mark in the middle of the horizontal lines:

```
\tikz\draw plot[const plot mark mid,mark=*] file{plots/pgfmanual-sine.table};
```

More precisely, it generates vertical lines in the middle between each pair of consecutive points. If the mesh width is constant, this leads to symmetrically placed marks (“middle”).

### /tikz/jump mark left

(no value)

This option causes the points on the path to be drawn using piecewise constant, non-connected series of lines. If there are any marks, they will be placed on left open ends:

```
\tikz\draw plot[jump mark left, mark=*] file{plots/pgfmanual-sine.table};
```

**/tikz/jump mark right** (no value)

This option causes the points on the path to be drawn using piecewise constant, non-connected series of lines. If there are any marks, they will be placed on right open ends:

```
\tikz\draw plot[jump mark right, mark=*] file{plots/pgfmanual-sine.table};
```

**/tikz/jump mark mid** (no value)

This option causes the points on the path to be drawn using piecewise constant, non-connected series of lines. If there are any marks, they will be placed in the middle of the horizontal line segments:

```
\tikz\draw plot[jump mark mid, mark=*] file{plots/pgfmanual-sine.table};
```

In case of non-constant mesh widths, the same remarks as for `const plot mark mid` apply.

**/tikz/ycomb** (no value)

This option causes the `plot` operation to interpret the plotting points differently. Instead of connecting them, for each point of the plot a straight line is added to the path from the  $x$ -axis to the point, resulting in a sort of “comb” or “bar diagram”.

```
\tikz\draw[ultra thick] plot[ycomb,thin,mark=*] file{plots/pgfmanual-sine.table};
```



```
\begin{tikzpicture}[ycomb]
  \draw[color=red, line width=6pt]
    plot coordinates{(0,1) (.5,1.2) (1,.6) (1.5,.7) (2,.9)};
  \draw[color=red!50, line width=4pt, xshift=3pt]
    plot coordinates{(0,1.2) (.5,1.3) (1,.5) (1.5,.2) (2,.5)};
\end{tikzpicture}
```

**/tikz/xcomb** (no value)

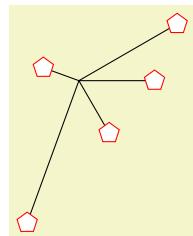
This option works like `ycomb` except that the bars are horizontal.



```
\tikz \draw plot[xcomb,mark=x] coordinates{(1,0) (0.8,0.2) (0.6,0.4) (0.2,1)};
```

**/tikz/polar comb** (no value)

This option causes a line from the origin to the point to be added to the path for each plot point.



```
\tikz \draw plot[polar comb,
  mark=pentagon*, mark options={fill=white, draw=red}, mark size=4pt]
  coordinates {(0:1cm) (30:1.5cm) (160:.5cm) (250:2cm) (-60:.8cm)};
```

**/tikz/ybar** (no value)

This option produces fillable bar plots. It is thus very similar to `ycomb`, but it employs rectangular shapes instead of line-to operations. It thus allows to use any fill or pattern style.

```
\tikz\draw[draw=blue,fill=blue!60!black] plot[ybar] file{plots/pgfmanual-sine.table};
```



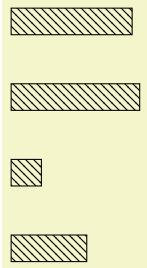
```
\begin{tikzpicture}[ybar]
  \draw[color=red,fill=red!80,bar width=6pt]
    plot coordinates{(0,1) (.5,1.2) (1,.6) (1.5,.7) (2,.9)};
  \draw[color=red!50,fill=red!20,bar width=4pt,bar shift=3pt]
    plot coordinates{(0,1.2) (.5,1.3) (1,.5) (1.5,.2) (2,.5)};
\end{tikzpicture}
```

The use of `bar width` and `bar shift` is explained in the `plot handlers` library documentation, section 65.4. Please refer to page 753.

#### /tikz/xbar

(no value)

This option works like `ybar` except that the bars are horizontal.

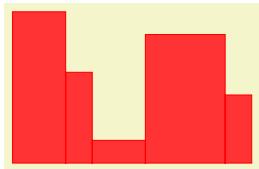


```
\usetikzlibrary {patterns}
\tikz \draw[pattern=north west lines] plot[xbar]
coordinates{(1,0) (0.4,1) (1.7,2) (1.6,3)};
```

#### /tikz/ybar interval

(no value)

As `/tikz/ybar`, this option produces vertical bars. However, bars are centered at coordinate *intervals* instead of interval edges, and the bar's width is also determined relatively to the interval's length:



```
\begin{tikzpicture}[ybar interval,x=10pt]
  \draw[color=red,fill=red!80]
  plot coordinates{(0,2) (2,1.2) (3,.3) (5,1.7) (8,.9) (9,.9)};
\end{tikzpicture}
```

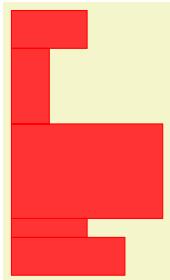
Since there are  $N$  intervals  $[x_i, x_{i+1}]$  for given  $N + 1$  coordinates, you will always have one coordinate more than bars. The last  $y$  value will be ignored.

You can configure relative shifts and relative bar widths, which is explained in the `plot handlers` library documentation, section 65.4. Please refer to page 754.

#### /tikz/xbar interval

(no value)

Works like `ybar interval`, but for horizontal bar plots.

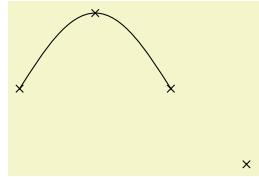


```
\begin{tikzpicture}[xbar interval,x=0.5cm,y=0.5cm]
  \draw[color=red,fill=red!80]
  plot coordinates {(3,0) (2,1) (4,1.5) (1,4) (2,6) (2,7)};
\end{tikzpicture}
```

#### /tikz/only marks

(no value)

This option causes only marks to be shown; no path segments are added to the actual path. This can be useful for quickly adding some marks to a path.



```
\tikz \draw (0,0) sin (1,1) cos (2,0)
plot[only marks,mark=x] coordinates{(0,0) (1,1) (2,0) (3,-1)};
```

# CHAPTER 8

## zLATEX 部分源码说明

### 8.1 文档选项

#### 8.1.1 Key-Value

```
1 \cs_new_protected:Npn \zlateX_define_option:n
2   { \keys_define:nn { zlateX / option } }
3 \cs_new_protected:Npn \zlateX_define:nn #1
4   { \keys_define:nn { zlateX / #1 } }
5 \cs_new_protected:Npn \zlateX_set:nn #1
6   { \keys_set:nn { zlateX / #1 } }
7 \cs_new_protected:Npn \zlateX_load_after:n #1
8   { \AtBeginDocument {#1} }
```

#### 8.1.2 Load Options

```
1 % setup option for class
2 \zlateX_define_option:n {
3   % language
4   lang .str_gset:N = \g_zlateX_lang_str,
5   lang .initial:n = { en },
6   % page layout
7   layout .str_gset:N = \g_zlateX_layout_str,
8   layout .initial:n = { twoside },
9   % margin option
10  margin .bool_gset:N = \g_zlateX_margin_bool,
11  margin .initial:n = { true },
12  % font size
13  font-size .str_gset:N = \g_zlateX_fontsize_str,
14  font-size .initial:n = { 10pt },
15  % math env type
16  math-env-theme .tl_gset:N = \g_zlateX_math_env_type_tl,
17  math-env-theme .initial:n = { leftbar },
18  % math alias
```

```

19     math-alias          .bool_gset:N = \g_zlatec_math_alias_bool,
20     math-alias          .initial:n = { false },
21 
22     % math-fonts
23     math-font           .choice:,,
24     math-font / newtx   .code:n = { \zlatec_load_after:n {
25       \RequirePackage{newtxmath} } },
26     math-font / mtpro2   .code:n = { \zlatec_load_after:n {
27       \RequirePackage[lite, subscriptcorrection, slantedGreek, nofontinfo]{mtpro2} } },
28     math-font / euler    .code:n = { \zlatec_load_after:n { \RequirePackage[OT1,
29       euler-digits]{eulervm} } },
30     math-font / unknown  .code:n = {
31       \msg_new:nnn {zlatec}{option-mathfont}{Current~math~font~option~is~:~'#1'~,defau
32       lt~math~font~substitute.}
33       \msg_warning:nn {zlatec}{option-mathfont}
34     },
35 
36     % bib source
37     bib-source          .str_gset:N = \g_zlatec_bib_source_str,
38     bib-source          .initial:n = { ref.bib },
39 
40     % toc setting
41     toc                 .multichoice:,,
42     toc / 2column      .code:n = { \zlatec_load_after:n {
43       \RequirePackage[toc]{multitoc} } },
44     toc / rename        .code:n = {
45       \str_case:Vn \g_zlatec_lang_str {
46         {en}{ \zlatec_load_after:n
47           {\renewcommand{\contentsname}{\hfill\bfseries\huge Contents \hfill}} }
48         {cn}{ \zlatec_load_after:n
49           {\renewcommand{\contentsname}{\hfill\bfseries\huge 目录 \hfill}} }
50       }{}}
51     },
52     toc / unknown       .code:n = {
53       \msg_new:nnn {zlatec}{option-toc}{Current~toc~option~is~:~'#1'~,default~toc~sett
54       ings~substitute.}
55       \msg_warning:nn {zlatec}{option-toc}
56     },
57   }
58 
59 \ProcessKeysOptions {zlatec / option}

```

## 8.2 compile engine

```

1 \msg_new:nnn
2   {zlatec}{compile-engine-pdftex}{Current~compile~engine~is~XETEX,~use~PDFTEX~instead.}
3 \msg_new:nnn {zlatec}{compile-engine-xetex}
4   {Current~compile~engine~is~PDFTEX,~use~XETEX~instead.}

```

```

3  \msg_new:nnn {zlatex}{option-language}{Current~language~option~is:~'\\g__zlatex_lang_str'
4    ,~which~is~invalid.}

5  % basic document class and packages option
6  \str_case:VnF \g__zlatex_lang_str{
7    {en} {
8      \sys_if_engine_xetex:TF
9        {\msg_warning:nn {zlatex}{compile-engine-pdftex}}
10       {\RequirePackage[utf8]{inputenc}}
11       \LoadClass[\clist_use:Nn \g__zlatex_book_options_clist{,}]{book}
12       \RequirePackage[T1]{fontenc}
13       \RequirePackage[english]{babel}
14     }
15    {cn} {
16      \sys_if_engine_xetex:TF {}{\msg_error:nn {zlatex}{compile-engine-xetex}}
17      \PassOptionsToPackage{quiet}{fontspec}
18      \PassOptionsToPackage{no-math}{fontspec}
19      \LoadClass[\clist_use:Nn \g__zlatex_book_options_clist{,}]{book}
20      % ctex should after 'book' class (or use 'scheme=plain', which chang nothing)
21      \RequirePackage[UTF8, heading]{ctex}
22      \linespread{1.3}
23    }
24  }{\msg_error:nn {zlatex}{option-language}}

```

## 8.3 Layout

### 8.3.1 geometry

```

1  \RequirePackage{geometry}
2  % page layout
3  \str_case:VnF \g__zlatex_layout_str {
4    {twoside} {
5      \geometry{a4paper, left=3cm, right=5.5cm, bottom=3.5cm, footskip=1.5cm,
6      marginparsep=1em}
7      \bool_if:NTF \g__zlatex_margin_bool {}{
8        \msg_new:nnn {zlatex}{option-page-margin}{No~margin~option~is~only~accessibl
9        e~in~oneside~layout,~margin~option~is~now~enabled~by~default.}
10       \msg_warning:nn {zlatex}{option-page-margin}
11     }
12   }
13   {oneside} {
14     \bool_if:NTF \g__zlatex_margin_bool {

```

```

13      \geometry{a4paper, left=3cm, right=5.5cm, bottom=3.5cm, footskip=1.5cm,
14      marginparsep=1em}
15      \dim_gset:Nn \marginparwidth{9.25em}
16      }{
17      \geometry{a4paper, left=3cm, right=3cm, bottom=3.5cm, footskip=1.5cm,
18      marginparsep=1em}
19      \renewcommand{\marginpar}[1]{\leftbar\noindent#1\endleftbar}
20      }
21  }{}  


```

### 8.3.2 fancyhdr

```

1  % fancy page settings
2  \RequirePackage{fancyhdr}
3  \fancypagestyle{fancy}{%
4      \fancyhf{}
5      \dim_gset:Nn \headheight{15pt}
6      \renewcommand{\headrule}{\hrule width\textwidth}
7      \msg_new:nnn
8          {zlatex}{option-page-layout}{Current~page~layout~option~is~:\clist_item:Nn
9              \g_zlatex_book_options_clist{1},~which~is~invalid.}
10         \str_case:VnF \g_zlatex_layout_str{%
11             {twoside}{%
12                 \fancyhead[EL]{\leftmark}
13                 \fancyhead[ER]{\thepage}
14                 \fancyhead[OL]{\thepage}
15                 \fancyhead[OR]{\rightmark}
16             }
17             {oneside}{%
18                 \fancyhead[L]{\thepage}
19                 \fancyhead[R]{\rightmark}
20             }
21         }{\msg_error:nn {zlatex}{option-page-layout}}
22     }
23     \fancypagestyle{plain}{%
24         \fancyhf{}
25         \renewcommand{\headrulewidth}{0pt}
26         \renewcommand{\headrule}{}
27         \fancyfoot[C]{\thepage}
28     }
29  }% front and main matter cmds

```

```

29 \renewcommand\frontmatter{%
30   \cleardoublepage
31   \pagestyle{plain}
32   \mainmatterfalse
33   \pagenumbering{Roman}
34 }
35 \renewcommand\mainmatter{%
36   \cleardoublepage
37   \pagestyle{fancy}
38   \mainmattertrue
39   \pagenumbering{arabic}
40 }
```

## 8.4 math

### 8.4.1 math env theme

```

1 % color spec for zlatex
2 \zlatex_define:nn {color}{%
3   % structure color
4   link          .tl_set:N    = \l_zlatex_link_color_tl,
5   link          .initial:n  = { purple },
6   cite          .tl_set:N    = \l_zlatex_cite_color_tl,
7   cite          .initial:n  = { teal },
8   url           .tl_set:N    = \l_zlatex_url_color_tl,
9   url           .initial:n  = { RoyalRed },
10  chapter        .tl_set:N    = \l_zlatex_chapter_color_tl,
11  chapter        .initial:n  = { RoyalRed },
12  chapter-rule   .tl_set:N    = \l_zlatex_chapter_rule_color_tl,
13  chapter-rule   .initial:n  = { black },
14  % math envs      color
15  axiom          .tl_set:N    = \l_zlatex_axiom_color_tl,
16  axiom          .initial:n  = { mathaxiomColor },
17  definition     .tl_set:N    = \l_zlatex_definition_color_tl,
18  definition     .initial:n  = { mathdefinitionColor },
19  theorem         .tl_set:N    = \l_zlatex_theorem_color_tl,
20  theorem         .initial:n  = { maththeoremColor },
21  lemma           .tl_set:N    = \l_zlatex_lemma_color_tl,
22  lemma           .initial:n  = { mathlemmaColor },
23  corollary       .tl_set:N    = \l_zlatex_corollary_color_tl,
24  corollary       .initial:n  = { mathcorollaryColor },
25  proposition     .tl_set:N    = \l_zlatex_proposition_color_tl,
```

```

26     proposition    .initial:n   = { mathpropositionColor },
27     remark        .tl_set:N     = \l_zalatex_remark_color_tl,
28     remark        .initial:n   = { mathremarkColor },
29 }
30 \NewDocumentCommand{\zalatexColorSetup}{m}{
31     \zalatex_set:nn {color}{#1}
32     % hyperref set up (may change in future)
33     \hypersetup{
34         colorlinks = true,
35         urlcolor   = \tl_use:N \l_zalatex_url_color_tl,
36         linkcolor  = \tl_use:N \l_zalatex_link_color_tl,
37         citecolor  = \tl_use:N \l_zalatex_cite_color_tl,
38     }
39 }
40 \zalatexColorSetup{link=purple, cite=teal, url=RoyalRed}

```

#### 8.4.2 math environment

```

1  % framed env for user interface
2  \cs_new_protected:Npn \zalatexFramed:nn #1#2 {
3      \DeclareDocumentEnvironment{#1}{O{#2}}{
4          \def\FrameCommand{{\color{##1}\vrule width 3pt}\colorbox{##1!10}}
5          \MakeFramed{\advance\hsize-\width\FrameRestore}\noindent
6      }{\endMakeFramed}
7  }
8  \NewDocumentCommand\zalatexFramed{mO{black}}{
9      \zalatexFramed:nn {#1}{#2}
10 }
11
12 % theorem/proof-like envs list
13 \clist_gset:Nn \g_zalatex_theoremlike_envs_clist {
14     axiom, definition, theorem,
15     lemma, corollary, proposition, remark
16 }
17 \clist_gset:Nn \g_zalatex_prooflike_envs_clist {
18     proof, exercise, example,
19     solution, problem,
20 }
21
22 % math envs' name according to language
23 \msg_new:nnn {\zalatex}{mathenv-name}[Current~math~env~name~is~:'#1', which~is~invalid.]
24 \msg_new:nnn {\zalatex}{mathenv-lang}[Current~math~envs~language~option~is~:'#1', which~is~invalid.~default~'en'~substitute.]

```

```

25 \str_case:VnTF \g_zlateX_lang_str {
26   {en}{
27     \zlateX_define:nn {math-env} {
28       math-env .multichoice:, 
29       math-env / axiom .code:n = { \tl_gset:cn {zlateX#1Name}{Axiom} },
30       math-env / definition .code:n = { \tl_gset:cn {zlateX#1Name}{Definition} }
31       },
32       math-env / theorem .code:n = { \tl_gset:cn {zlateX#1Name}{Theorem} },
33       math-env / lemma .code:n = { \tl_gset:cn {zlateX#1Name}{Lemma} },
34       math-env / corollary .code:n = { \tl_gset:cn {zlateX#1Name}{Corollary} },
35       math-env / proposition .code:n = { \tl_gset:cn {zlateX#1Name}{Proposition} }
36       },
37       math-env / remark .code:n = { \tl_gset:cn {zlateX#1Name}{Remark} },
38       math-env / proof .code:n = { \tl_gset:cn {zlateX#1Name}{Proof} },
39       math-env / exercise .code:n = { \tl_gset:cn {zlateX#1Name}{Exercise} },
40       math-env / example .code:n = { \tl_gset:cn {zlateX#1Name}{Example} },
41       math-env / solution .code:n = { \tl_gset:cn {zlateX#1Name}{Solution} },
42       math-env / problem .code:n = { \tl_gset:cn {zlateX#1Name}{Problem} },
43       math-enc / unknown .code:n = {
44         \msg_error:nn {zlateX}{mathenv-name}
45       },
46     }
47   {cn}{

48     \zlateX_define:nn {math-env} {
49       math-env .multichoice:, 
50       math-env / axiom .code:n = { \tl_gset:cn {zlateX#1Name}{公理} },
51       math-env / definition .code:n = { \tl_gset:cn {zlateX#1Name}{定义} },
52       math-env / theorem .code:n = { \tl_gset:cn {zlateX#1Name}{定理} },
53       math-env / lemma .code:n = { \tl_gset:cn {zlateX#1Name}{引理} },
54       math-env / corollary .code:n = { \tl_gset:cn {zlateX#1Name}{推论} },
55       math-env / proposition .code:n = { \tl_gset:cn {zlateX#1Name}{命题} },
56       math-env / remark .code:n = { \tl_gset:cn {zlateX#1Name}{注记} },
57       math-env / proof .code:n = { \tl_gset:cn {zlateX#1Name}{证明} },
58       math-env / exercise .code:n = { \tl_gset:cn {zlateX#1Name}{练习} },
59       math-env / example .code:n = { \tl_gset:cn {zlateX#1Name}{示例} },
60       math-env / solution .code:n = { \tl_gset:cn {zlateX#1Name}{解} },
61       math-env / problem .code:n = { \tl_gset:cn {zlateX#1Name}{问题} },
62       math-enc / unknown .code:n = {
63         \msg_error:nn {zlateX}{mathenv-name}
64       },
65     }
66 }{\zlateX_set:nn {math-env}{math-env={axiom, definition, theorem, lemma, corollary,
proposition, remark, proof, exercise, example, solution, problem}}}

```

```

67 {\msg_error:nn {zlatex}{mathenv-lang}}
68
69 % math env's style
70 \newtheoremstyle{zlatexMathEnv}
71   {2pt}{2pt}{}%
72   {0pt}{\bfseries}{}%
73   {.25em}{\thmname{\#1}\thmnumber{\#2}\thmnote{\#3}}%
74 \theoremstyle{zlatexMathEnv}
75
76 % theorem-like env declaration
77 \msg_new:nnn {zlatex}{mathenv-type}{Current~math~env~is~:#1~,~only~'none',~'leftbar',~'background'~types~are~valid.}%
78 \str_case:VnF \g_zlatex_math_env_theme_tl {
79   {none} {
80     \NewDocumentEnvironment{zlatexTheoremLikeFrame}{0{}{\vspace*{5pt}}{\vspace*{5pt}}}
81   }
82   {leftbar} {
83     \NewDocumentEnvironment{zlatexTheoremLikeFrame}{0{black}}{
84       \def\FrameCommand{\color{\#1}\vrule width 3pt\hspace{5pt}}
85       \MakeFramed{\advance\hsize-\width \FrameRestore}
86     }{\endMakeFramed}
87   }
88   {all} {
89     \NewDocumentEnvironment{zlatexTheoremLikeFrame}{0{black}}{
90       \def\FrameCommand{\color{\#1}\vrule width 3pt\colorbox{\#1!10}}
91       \MakeFramed{\advance\hsize-\width \FrameRestore}
92     }{\endMakeFramed}
93   }
94 }{\msg_error:nn {zlatex}{mathenv-type}}
95
96 % loop to create math env, setup \cref
97 \clist_map_inline:Nn \g_zlatex_theoremlike_envs_clist {
98   % theorem create
99   \newtheorem{zlatex#1}{\tl_use:c {zlatex#1Name}}[section]
100
101   % env create (3 types: 'leftbar', 'none' and 'background')
102   \NewDocumentEnvironment{#1}{0{}{
103     \begin{zlatexTheoremLikeFrame}[\tl_use:c {l_zlatex_#1_color_tl}]
104     \begin{zlatex#1}[\#1]
105   }{\end{zlatex#1}\end{zlatexTheoremLikeFrame}}
106
107   % cref settings
108   \cs_generate_variant:Nn \exp_args:Nnnx {Nxxx}
109   \str_case:VnF \g_zlatex_lang_str {

```

```

110     {en}{
111         \crefname{zlatex#1}{#1}{#1s}
112         \creflabelformat{zlatex#1}{##2##1##3}
113     }
114     {cn}{
115         \exp_args:Nxxx \crefname{zlatex#1}{\tl_use:c {zlatex#1Name}}{\tl_use:c
116             {zlatex#1Name}}
117         \creflabelformat{zlatex#1}{##2##1##3}
118     }
119     }{\msg_error:nn {zlatex}{mathenv-lang}}
120 }
121 % proof-like env decalration
122 \renewcommand{\qedsymbol}{\ensuremath{\blacksquare}}
123 \clist_map_inline:Nn \g_zlatex_prooflike_envs_clist{
124     \DeclareDocumentEnvironment{#1}{O{}){
125         {\noindent\bfseries\tl_use:c {zlatex#1Name}:}}
126         \tl_set:Nn \l__arg_one_tl {#1}
127     }{\str_if_eq:VnTF \l__arg_one_tl{proof}{\hfill\qedsymbol\par}{\par}}
128 }
```

### 8.4.3 Math alias

```

1  % math related cmds alias
2  \bool_if:NTF \g_zlatex_math_alias_bool{
3      \RequirePackage{amssymb, mathtools}
4      \RequirePackage{bm}
5      % Math Font
6      \newcommand{\dd}{\mathrm{d}}
7      \newcommand{\C}[1]{\ensuremath{\mathcal{#1}}}
8      \let\ss\S
9      \renewcommand{\S}[1]{\ensuremath{\mathscr{#1}}}
10     \newcommand{\B}[1]{\ensuremath{\mathbb{#1}}}
11     \newcommand{\FF}[1]{\ensuremath{\mathbf{#1}}}
12     \newcommand{\F}[1]{\ensuremath{\bm{#1}}}
13     \newcommand{\R}[1]{\mathrm{#1}}
14     \newcommand{\K}[1]{\mathfrak{#1}}
15     % Math Arrow
16     \newcommand{\lr}{\rightarrowtail}
17     \let\LL\ll
18     \renewcommand{\ll}{\leftarrowtail}
19     \newcommand{\equ}{\mathrel{\mathop{\Longleftrightarrow}\limits^{\scriptscriptstyle\textbf{L}}\,}}
20     \newcommand{\sr}{\mathrel{\mathop{\longmapsto}\limits^{\scriptscriptstyle\textbf{R}}\,}}
```

```

21 \newcommand{\lrr}[2][]{\ensuremath{\rightarrow[x]{#1}{#2}}}
22 \renewcommand{\lll}[2][]{\ensuremath{\leftarrow[x]{#1}{#2}}}
23 \newcommand{\ns}{\ensuremath{\varnothing}}
24 \newcommand{\A}{\forall}
25 % Math Operator
26 \newcommand{\alt}{\mathrel{\mathop{:}}=\mathrel{\mathop{:}}}
27 \newcommand{\sgn}{\mathrel{\mathop{:}}=\mathrel{\mathop{:}}}
28 \newcommand{\curl}{\mathrel{\mathop{:}}=\mathrel{\mathop{:}}}
29 \newcommand{\grad}{\mathrel{\mathop{:}}=\mathrel{\mathop{:}}}
30 \newcommand{\trace}{\mathrel{\mathop{:}}=\mathrel{\mathop{:}}}
31 \renewcommand{\div}{\mathrel{\mathop{:}}=\mathrel{\mathop{:}}}
32 }{}
```

## 8.5 structure Style

### 8.5.1 contents

```

1 % partial ToC
2 \RequirePackage{titletoc}
3 \NewDocumentCommand{\partialToC}{0}{%
4   \setcounter{tocdepth}{2}
5   \titlecontents{psection}[2.3em]
6     {} {\contentslabel{2.3em}} {} {\titlerule*[1pc]{.}\contentspage}
7   \titlecontents{psubsection}[5em]
8     {} {\contentslabel{3.2em}} {} {\titlerule*[1pc]{.}\contentspage}
9   % print ToC
10  \vspace*{-2em}
11  \startcontents[chapters]
12  \printcontents[chapters]{p}{1}{}
13 }
```

### 8.5.2 Chapter and section

```

1 % chapter head style
2 \RequirePackage{titlesec}
3 \titleformat{\chapter}[display]
4   {\bfseries\huge\color{black}}
5   {\flushright\Large\color{\tl_use:N \l_zlatez_chapter_color_tl}}
6   \MakeUppercase{\chapertitlename}\hspace{1ex}
7   {\fontsize{60}{60}\selectfont\thechapter}
8   {10pt}
```

```
9      {\color{\tl_use:N \l_zlatex_chapter_rule_color_t1}\titlerule\vspace{1ex}}
10 % chapter space
11 \titlespacing{\chapter}{0pt}{-30pt}{40pt}
12 \titleformat{name=\chapter, numberless}
13   {\normalfont\bfseries\Huge}
14   {}{0pt}{}\\
15 %
16 % title page
17 \renewcommand{\maketitle}{\\
18 \begin{titlepage}
19   \vfill\vspace*{40pt}
20   \noindent\hspace*{134pt}\rule[-75pt]{6pt}{95pt}{\hspace*{10pt}\leavevmode\parbox[t]{25em}{\fontsize{25}{25}\selectfont\bfseries\@title}}\par
21   \vspace*{-15pt}
22   \noindent\hspace*{150pt}{\leavevmode\parbox[t]{30em}{\Large\bfseries\@author}}\par
23   \vfill
24   \noindent\hspace*{150pt}{\Large\textrm{gray}\{@date\}}\\
25 \end{titlepage}\\
26 %
27 % reset counter command
28 \NewDocumentCommand{\zlatexUpdateCounterAfter}{mm}{\@addtoreset{#1}{#2}}
```

# CHAPTER 9

## zTikZ 部分源码说明

### 9.1 Set up Env

#### 9.1.1 Set up

```
1 \RequirePackage{l3sys-shell}
2 \NewDocumentCommand\ztikzMkdir{m}{ \sys_shell_mkdir:n {#1} }
3 \ExplSyntaxOff
4 \begingroup\newif\ifztikz
5 \IfFileExists{./ztikz_output/scripts/sympy_script.py}{}{\ztikztrue}
6 \ifztikz
7 % create folders and hash file
8 \ztikzMkdir{ztikz_output/}
9 \ztikzMkdir{ztikz_output/gnuplot_data/}
10 \ztikzMkdir{ztikz_output/python_data/}
11 \ztikzMkdir{ztikz_output/scripts/}
12 \ztikzMkdir{ztikz_output/tikz_data/}
13 \ztikzMkdir{ztikz_output/mma_data/}
14 \immediate\write18 {touch ztikz_output/ztikz.hash}
15
16 % writing source code
17 \begin{filecontents}[noheader]{./ztikz_output/scripts/python_script.py}
18 from numpy import *
19 from scipy import *
20
21
22 # ==> echo to file
23 Float_res = 2**10
24
25 # ==> write output
26 filename = "./ztikz_output/python_data/PyFloat.out"
27 with open(filename, 'w') as file:
28     file.write(str(Float_res)+"\n")
29 \end{filecontents}
30
31 \begin{filecontents}[noheader]{./ztikz_output/scripts/sympy_script.py}
```

```
32 from sympy import *
33
34
35 # ==> symbols declearation
36 x, y, z, u, v, t = symbols('x y z u v t')
37
38 # ==> echo to file
39 F_res = integrate(sin(x)/x, (x, -oo, oo))
40
41 # ==> write output
42 filename = './ztikz_output/python_data/sympy.out'
43 with open(filename, 'w') as file:
44     file.write(latex(F_res)+"\n")
45 \end{filecontents}
46
47 \begin{filecontents}[noheader]{./ztikz_output/scripts/plot.gp}
48 set table "./ztikz_output/gnuplot_data/gnu_data.table"
49 set format "%.5f"
50 set samples 100
51
52
53 # ==> specific a 2d function
54 f(x) = x
55 set xr [-6:-1]
56 plot f(x)
57 \end{filecontents}
58
59 \begin{filecontents}[noheader]{./ztikz_output/scripts/contour_plot.gp}
60 set table "./ztikz_output/gnuplot_data/gnu_data.table"
61 set samples 100,100
62 set isosamples 100,100
63 set format "%.5f"
64 set cont base
65 set cntrparam levels discrete 0,0
66 unset surface
67
68
69 # ==> contour plot a function
70 set xr [-4:4]
71 set yr [*:*]
72 set zr [*:*]
73 f(x, y) = y-3*sin(1/x)
74 splot f(x,y)
75 \end{filecontents}
```

```

77 \begin{filecontents}[noheader]{./ztikz_output/scripts/param_plot.gp}
78 set table "./ztikz_output/gnuplot_data/gnu_data.table"
79 set format "%.5f"
80 set samples 100
81 set parametric
82
83
84 # ==> specific a parametric function
85 set trange [0:2*pi]
86 plot sin(t), cos(t)
87 \end{filecontents}
88
89 \begin{filecontents}[noheader]{./ztikz_output/scripts/polar_plot.gp}
90 set table "./ztikz_output/gnuplot_data/gnu_data.table"
91 set format "%.5f"
92 set samples 100
93 set polar
94
95
96 # ==> polar plot, default varible name 't'
97 set trange [0:12*pi]
98 plot t
99 \end{filecontents}
100 \fi\endgroup
101 \ExplSyntaxOn

```

### 9.1.2 config

```

1 % data or picture path
2 \tl_const:Nn \g__ztikz_gnu_path_tl      {./ztikz_output/gnuplot_data}
3 \tl_const:Nn \g__ztikz_python_path_tl    {./ztikz_output/python_data}
4 \tl_const:Nn \g__ztikz_scripts_path_tl   {./ztikz_output/scripts}
5 \tl_const:Nn \g__ztikz_tikz_path_tl     {./ztikz_output/tikz_data}
6 \tl_const:Nn \g__ztikz_mma_path_tl      {./ztikz_output/mma_data}
7
8 % index of output data by gnuplot
9 \int_new:N \g__gnu_data_index_int
10 \int_new:N \g__sympy_index_int
11 \int_new:N \g__picture_index_int
12 \int_new:N \g__mma_index_int
13 \int_new:N \g__tikz_env_index_int
14 \int_gadd:Nn \g__mma_index_int {1}
15

```

```

16  % bool to control \PlotPrecise cmd
17  \bool_new:N \g__plot_precise_bool
18  \bool_new:N \g__contour_precise_bool
19  \bool_new:N \g__param_precise_bool
20  \bool_new:N \g__polar_precise_bool
21  \bool_new:N \g__hash_change_bool
22  \bool_gset_true:N \g__hash_change_bool
23
24  % io read/write interface
25  \ior_new:N \g__file_read_ior
26  \tl_new:N \g__file_content_tl

```

## 9.2 tikz

### 9.2.1 Cache mechanism

```

1  % generate md5 hash (history) of a file
2  \ior_new:N \g__ztikz_file_ior
3  \seq_new:N \g__ztikz_file_hash_seq
4  \cs_new_protected:Nn \ztikz_file_read_lines_cs:n {
5      \ior_open:Nn \g__ztikz_file_ior {\#1}
6      \seq_gclear:N \g__ztikz_file_hash_seq
7      \ior_str_map_inline:Nn \g__ztikz_file_ior
8          {
9              \seq_gput_right:Nx \g__ztikz_file_hash_seq
10             { \tl_trim_spaces:n {\##1} }
11         }
12     \ior_close:N \g__ztikz_file_ior
13 }
14 \cs_generate_variant:Nn \ztikz_file_read_lines_cs:n { e }
15
16  % check if code changed (compare hash value of source code)
17  \seq_new:N \l__ztikz_hash_seq
18  \cs_new_protected:Npn \ztikz_hash_if_change_cs:n #1 {
19      % Param: #1 -> file name of source code
20      % read from hash list and remove duplicate item
21      \ztikz_file_read_lines_cs:e { ./ztikz_output/ztikz.hash }
22      \seq_gremove_duplicates:N \g__ztikz_file_hash_seq
23      % check if in
24      \file_get_mdfive_hash:nN {\#1} \l__ztikz_hash_seq
25      \typeout{current~hash~is~\l__ztikz_hash_seq}
26      \seq_if_in:NVTF \g__ztikz_file_hash_seq \l__ztikz_hash_seq {

```

```

27     \bool_gset_false:N \g__hash_change_bool
28 }
29     \bool_gset_true:N \g__hash_change_bool
30     \typeout{current-hash~is~unique--->~recorded}
31     \sys_shell_now:x { echo~ \l__ztikz_hash_seq >> ./ztikz_output/ztikz.hash}
32 }
33 }
34 \cs_generate_variant:Nn \ztikz_hash_if_change_cs:n { x }
```

### 9.2.2 externalize for tikzpicture

```

1 % set options
2 \cs_new_protected:Npn \ztikz_define_option:n
3   { \keys_define:nn { ztikz / option } }
4
5 \ztikz_define_option:n {
6   external .bool_set:N = \l__ztikz_external_bool,
7   external .initial:n = { true },
8 }
9 \ProcessKeysOptions {ztikz / option}
10
11 \bool_if:NT \l__ztikz_external_bool {
12   \usetikzlibrary{external}
13   \tikzexternalize[prefix=ztikz_output/tikz_data/]
14 }
15 \end{codeprint}
16
17 \subsection{ShowPoint}
18 \begin{minted}[linenos=true]{latex}
19 % ==> show point
20 \keys_define:nn { ztikz / point }{
21   type .str_set:N = \l__point_type_str,
22   type .initial:n = { * },
23   radius .dim_set:N = \l__point_radius_dim,
24   radius .initial:n = { 1pt },
25   color .tl_set:N = \l__point_color_tl,
26   color .initial:n = { black },
27   opacity .tl_set:N = \l__point_opacity_tl,
28   opacity .initial:n = { 1 },
29   rotate .fp_set:N = \l__point_rotate_angle,
30   rotate .initial:n = { 0 },
31 }
32 \NewDocumentCommand\ShowPoint{ O{}mO{}O{} }{
```

```

33   \group_begin:
34   \keys_set:nn { ztikz / point } { #1 }
35   \seq_set_split:Nnn \l_point_list_seq { ; }{#2}
36   \seq_set_split:Nnn \l_point_label_seq { ; }{#3}
37   \int_step_inline:nnnn {1}{1}{\seq_count:N \l_point_list_seq} {
38     \draw plot [
39       only~ marks,
40       mark = \str_use:N \l_point_type_str,
41       mark~ size = \dim_use:N \l_point_radius_dim,
42       mark~ options = {
43         rotate = \fp_use:N \l_point_rotate_angle,
44         opacity = \tl_use:N \l_point_opacity_tl,
45         color = \tl_use:N \l_point_color_tl,
46         ball~ color = \tl_use:N \l_point_color_tl,
47       }
48     ] coordinates{\seq_item:Nn \l_point_list_seq{##1}} node[#4]{\seq_item:Nn
49       \l_point_label_seq{##1}};
50   }
51   \group_end:
52 }
53 % ==> show grid
54 \NewDocumentCommand\ShowGrid{ O{color=gray, very~ thin, step=1}m }{
55   \seq_set_split:Nnn \l_grid_param_ii_seq { ; }{#2}
56   \draw[#1] \seq_item:Nn \l_grid_param_ii_seq{1} grid \seq_item:Nn
57   \l_grid_param_ii_seq{2};
}
```

### 9.2.3 ShowAxis

```

1 % ==> show axis
2 \keys_define:nn { ztikz / axis }{
3   % basic tick args
4   tickStart      .fp_set:N    = \l_start_fp,
5   tickStart      .initial:n  = { -5 },
6   tickEnd        .fp_set:N    = \l_end_fp,
7   tickEnd        .initial:n  = { 5 },
8   axisRotate     .fp_set:N    = \l_axis_rotate_angle,
9   axisRotate     .initial:n  = { 0 },
10  % tick dimension spec
11  mainStep       .fp_set:N    = \l_main_step_fp,
12  mainStep       .initial:n  = { 1.0 },
13  subStep        .fp_set:N    = \l_sub_step_fp,
```

```

14     subStep      .initial:n = { 0.1 },
15     mainTickLabel .tl_set:N  = \l_main_tick_label_tl,
16     mainTickLabel .initial:n = { \fp_use:N \CurrentFp },
17     tickLabelShift .dim_set:N = \l_tick_label_shift_dim,
18     tickLabelShift .initial:n = { Opt },
19     mainTickLength .dim_set:N = \l_main_tick_length_dim,
20     mainTickLength .initial:n = { 4pt },
21     subTickLength .dim_set:N = \l_sub_tick_length_dim,
22     subTickLength .initial:n = { 2pt },
23     mainTickLabelPosition .tl_set:N = \l_main_tick_label_position_tl,
24     mainTickLabelPosition .initial:n = { below },
25     % color spec
26     axisColor      .tl_set:N  = \l_axis_color_tl,
27     axisColor      .initial:n = { black },
28     mainTickColor   .tl_set:N  = \l_main_tick_color_tl,
29     mainTickColor   .initial:n = { black },
30     subTickColor    .tl_set:N  = \l_sub_tick_color_tl,
31     subTickColor    .initial:n = { black },
32     mainTickLabelColor .tl_set:N = \l_main_tick_label_color_tl,
33     mainTickLabelColor .initial:n = { black },
34     % tick cross type spec
35     tickStyle      .choice:,,
36     tickStyle/cross .code:n   = \tl_set:Nn \l_tick_spec_tl { cross },
37     tickStyle/above .code:n   = \tl_set:Nn \l_tick_spec_tl { above },
38     tickStyle/below .code:n   = \tl_set:Nn \l_tick_spec_tl { below },
39 }
40 % ticks style
41 \tl_new:N \l_tick_type_tl    % `main' or `sub'
42 \tl_new:N \l_tick_spec_tl    % `cross', `above' or `below'
43 \tl_new:N \l_tick_color_tl
44 \dim_new:N \l_tick_length_dim
45 \tl_new:N \l_node_text_tl
46 % draw ticks (main or sub)
47 \cs_new_protected:Npn \ztikz_draw_axis_ticks_cs:n #1 {
48     \str_case:NnT \l_tick_type_tl {
49         {main} {
50             \dim_set_eq:NN \l_tick_length_dim \l_main_tick_length_dim
51             \tl_set:NV \l_tick_color_tl \l_main_tick_color_tl
52             \tl_set:Nn \l_node_text_tl {\tl_use:N \l_main_tick_label_tl}
53         }
54         {sub} {
55             \dim_set_eq:NN \l_tick_length_dim \l_sub_tick_length_dim
56             \tl_set:NV \l_tick_color_tl \l_sub_tick_color_tl
57             \tl_set:Nn \l_node_text_tl {}
58     }

```

```

59    }{}
60    \str_case:VnT \l__tick_spec_tl {
61        {cross}{%
62            \draw[\tl_use:N \l__tick_color_tl]
63                (#1, 0)++(0, \dim_eval:n {\l__tick_length_dim/2}) -- +(0, \dim_eval:n
64                {-\l__tick_length_dim})
65            node[\tl_use:N \l__main_tick_label_position_tl]
66                {\textcolor{\tl_use:N \l__main_tick_label_color_tl}{\tl_use:N
67                \l__node_text_tl}};
68        }
69        {above}{%
70            \draw[\tl_use:N \l__tick_color_tl] (#1, 0) -- +(0, \dim_eval:n
71                {\l__tick_length_dim/2})
72            node[\tl_use:N \l__main_tick_label_position_tl]
73                {\textcolor{\tl_use:N \l__main_tick_label_color_tl}{\tl_use:N
74                \l__node_text_tl}};
75        }
76        {below}{%
77            \draw[\tl_use:N \l__tick_color_tl] (#1, 0) -- +(0, \dim_eval:n
78                {-\l__tick_length_dim/2})
79            node[\tl_use:N \l__main_tick_label_position_tl=\dim_use:N
80                \l__tick_label_shift_dim]
81                {\textcolor{\tl_use:N \l__main_tick_label_color_tl}{\tl_use:N
82                \l__node_text_tl}};
83        }
84    }{%
85        % draw axis
86        \int_new:N \l__substep_num_int
87        \fp_new:N \CurrentFp
88        \NewDocumentCommand\ShowAxis{O{}m}{%
89            \group_begin:
90            \keys_set:nn { ztikz / axis } { #1 }
91            \seq_set_split:Nnn \l__points_seq { ; }{#2}
92            \begin{scope}[rotate=\fp_use:N \l__axis_rotate_angle]
93                \draw[>, \tl_use:N \l__axis_color_tl] \seq_item:Nn \l__points_seq{1} --
94                    \seq_item:Nn \l__points_seq{2};
95            % draw ticks
96            \fp_step_inline:nnnn {\fp_eval:n {\l__start_fp}}{\fp_use:N
97                \l__main_step_fp}{\fp_use:N \l__end_fp}{%
98                % main ticks
99                \tl_set:Nn \l__tick_type_tl {main}
100                \fp_gset:Nn \CurrentFp {##1}
101                \ztikz_draw_axis_ticks cs:n {##1}
102                % sub ticks
103            }
104        }

```

```

95      \tl_set:Nn \l__tick_type_tl {sub}
96      \int_set:Nn \l__substep_num_int {\fp_eval:n
97      {floor(\l__main_step_fp/\l__sub_step_fp)}}
98      \fp_compare:nNnTF {##1}<{\fp_eval:n {floor(\l__end_fp)}}{
99          \fp_step_function:nnnN
100             {\fp_eval:n {##1+\l__sub_step_fp}}
101             {\fp_use:N \l__sub_step_fp}
102             {\fp_eval:n {##1+\l__substep_num_int*\l__sub_step_fp}}
103             \ztikz_draw_axis_ticks cs:n
104         }{}
105     }
106     \group_end:
107 }
108 \NewDocumentCommand{\xAxis}{O{-2}O{8}}{
109     \ShowAxis[
110         tickStart=\fp_eval:n {#1+1}, tickEnd=\fp_eval:n {#2-0.75},
111         mainStep=1, subStep=.25,
112         axisRotate=0, axisColor=black,
113         mainTickColor=black, subTickColor=black,
114         mainTickLength=10pt, subTickLength=5pt,
115         tickLabelShift=0pt, tickStyle=below,
116         mainTickLabelPosition=below
117     ]{(#1, 0); (#2, 0)}
118 }
119 \NewDocumentCommand{\yAxis}{O{-2}O{8}}{
120     \ShowAxis[
121         tickStart=\fp_eval:n {#1+1}, tickEnd=\fp_eval:n {#2-0.75},
122         mainStep=1, subStep=.25,
123         axisRotate=90, axisColor=black,
124         mainTickColor=black, subTickColor=black,
125         mainTickLength=10pt, subTickLength=5pt,
126         tickLabelShift=0pt, tickStyle=above,
127         mainTickLabelPosition=left
128     ]{(#1, 0); (#2, 0)}
129 }
130 %
131 % ==> show intersection
132 \NewDocumentCommand\ShowIntersection{ omm }{
133     \seq_set_split:Nnn \l__intersection_num_seq { ; }{#2}
134     % get all intersections
135     \path[name= intersections={of=\seq_item:Nn \l__intersection_num_seq{1}~ and~
136     \seq_item:Nn \l__intersection_num_seq{2}}];
137     % show all intersections by \ShowPoint
138     \int_step_inline:nnnn {1}{1}{#3}{
```

```

138      \ShowPoint[#1]{(intersection-##1)}
139    }
140 }

```

### 9.2.4 Polygon

```

1  % ==> polygon
2  \keys_define:nn { ztikz / polygon }{
3    radius      .fp_set:N  = \l__polygon_radius_fp,
4    radius      .initial:n = { 1 },
5    edgeColor   .tl_set:N  = \l__polygon_edge_color_tl,
6    edgeColor   .initial:n = { black },
7    fillColor   .tl_set:N  = \l__polygon_fill_color_tl,
8    fillColor   .initial:n = { },
9    fillOpacity .fp_set:N  = \l__polygon_fill_opacity_fp,
10   fillOpacity .initial:n = { 0 },
11   rotate      .fp_set:N  = \l__polygon_rotate_angle,
12   rotate      .initial:n = { 0 },
13   shift       .tl_set:N  = \l__polygon_shift_tl,
14   shift       .initial:n = { (0,0) },
15   marker      .tl_set:N  = \l__polygon_marker_option_tl,
16   marker      .initial:n = { },
17 }
18 \tl_new:N \l__poly_path_tl
19 \NewDocumentCommand\Polygon{ O{} m }{
20   \group_begin:
21   \keys_set:nn { ztikz / polygon } { #1 }
22   % strip '(' and ')'
23   \tl_replace_once:Nnn \l__polygon_shift_tl{(){}}
24   \tl_replace_once:Nnn \l__polygon_shift_tl{}{}
25   \coordinate (mv) at (\tl_use:N \l__polygon_shift_tl);
26   % create polygon
27   \begin{scope}[shift=(mv), rotate=\fp_use:N \l__polygon_rotate_angle]
28   \int_step_inline:nnn {1}{#2} {
29     % draw edges
30     \fp_set:Nn \l_angle_fp {360/#2*##1*\c_one_degree_fp}
31     \fp_set:Nn \l_angle_next_fp {360/#2*(##1+1)*\c_one_degree_fp}
32     \draw[\tl_use:N \l__polygon_edge_color_tl] (\fp_eval:n
33     {\l__polygon_radius_fp*cos(\l_angle_fp)}, \fp_eval:n
34     {\l__polygon_radius_fp*sin(\l_angle_fp)})
--(\fp_eval:n {\l__polygon_radius_fp*cos(\l_angle_next_fp)}, \fp_eval:n
{\l__polygon_radius_fp*sin(\l_angle_next_fp)});
35     % fill polygon path

```

```

35      \int_compare:nNnTF {##1}<{#2}){
36          \tl_put_right:Nn \l__poly_path_tl {(p##1) --}
37      }{
38          \tl_put_right:Nn \l__poly_path_tl {(p##1)--cycle}
39      }
40      % mark coordinates
41      \coordinate (p##1) at (\fp_eval:n {\l__polygon_radius_fp*cos(\l_angle_fp)},
42          \fp_eval:n {\l__polygon_radius_fp*sin(\l_angle_fp)});
43      % fill polygon (none-color -> opacity=1; or opacity=.75)
44      \tl_if_empty:NTF \l__polygon_fill_color_tl {
45          \fp_set:Nn \l__polygon_fill_opacity_fp {0}
46      }{
47          \fp_set:Nn \l__polygon_fill_opacity_fp {.75}
48      }
49      \fill [\tl_use:N \l__polygon_fill_color_tl, fill~opacity=\fp_use:N
50          \l__polygon_fill_opacity_fp] \l__poly_path_tl;
51      % show markers
52      \int_step_inline:nnn {1}{#2}{
53          \ShowPoint[\l__polygon_marker_option_tl]{(p##1)}
54      }
55      \end{scope}
56      \group_end:
57  }

```

### 9.2.5 Plot precise

```

1  % change precise of each plot type
2  \NewDocumentCommand\PlotPrecise{ m0{once}m }{
3      \str_if_eq:nnTF {#2}{once}{
4          \bool_gset_true:c {g__#1_precise_bool}
5      }{\relax}
6      % check plot type
7      \str_case:nnF {#1}{
8          {plot}){
9              \sys_shell_now:x {sed~ -i~ "3s|set~ samples~ .*|set~ samples~ #3|~"
10                 \g__ztikz_scripts_path_tl/plot.gp           }
11          }
12          {param}){
13              \sys_shell_now:x {sed~ -i~ "3s|set~ samples~ .*|set~ samples~ #3|~"
14                 \g__ztikz_scripts_path_tl/param_plot.gp  }
15          }
16          {polar}){
17      }
18  }

```

```

15      \sys_shell_now:x {sed~ -i~ "3s|set~ samples~ .*|set~ samples~ #3|"~
16      \g_ztikz_scripts_path_tl/param_plot.gp }
17  }
18  {contour}){
19      \sys_shell_now:x {sed~ -i~ "2s|set~ samples~ .*|set~ samples~ #3,#3|"~
20      \g_ztikz_scripts_path_tl/contour_plot.gp}
21      \sys_shell_now:x {sed~ -i~ "3s|set~ isosamples~ .*|set~ isosamples~ #3,#3|"~
22      \g_ztikz_scripts_path_tl/contour_plot.gp}
23  }
24  \H{
25      \msg_new:nnn {ztikz}{ztikz-plot-type}{plot~type~support~is:~'plot',~'param',~'po
26      lar',~'contour'}
27      \msg_error:nn {ztikz}{ztikz-plot-type}
28  }
29 }
```

### 9.2.6 curve plot

```

1  % escape shell commands
2  \cs_generate_variant:Nn \sys_shell_mv:nn {xx}
3  \cs_generate_variant:Nn \sys_shell_now:n {x}
4
5  % gnu data reference
6  \NewDocumentCommand\gnudata{m}{
7      \tl_use:N \g_ztikz_gnu_path_tl/gnu_data_ \int_use:N \g_tikz_env_index_int
8      _#1.table
9  }
10
11 % gnuplot data rename, plot and precise reset
12 \cs_new_protected:Npn \ztikz_gnu_data_plot_cs:nnn #1#2#3 {
13     % rename data file
14     \int_gadd:Nn \g_gnu_data_index_int {1}
15     \tl_set:Nx \l_gnu_data_new_name_tl {
16         gnu_data_\int_use:N \g_tikz_env_index_int _-
17         \int_use:N \g_gnu_data_index_int.table
18     }
19     \tl_set:Nx \l_gnu_data_full_path_tl
20     {\g_ztikz_gnu_path_tl/\l_gnu_data_new_name_tl}
21     \sys_shell_mv:xx {\g_ztikz_gnu_path_tl/gnu_data.table}
22     {\l_gnu_data_full_path_tl}
23     % plot data file
24     \tl_if_empty:nTF {#3} {
25         \draw[#2] plot[smooth] file {\l_gnu_data_full_path_tl};
```

```

24 }{
25   \group_begin:
26   \keys_set:nn { ztikz / point } { #3 }
27   \draw plot [
28     mark = \str_use:N \l__point_type_str,
29     mark~ size = \dim_use:N \l__point_radius_dim,
30     mark~ options = {
31       rotate = \fp_use:N \l__point_rotate_angle,
32       opacity = \tl_use:N \l__point_opacity_tl,
33       color = \tl_use:N \l__point_color_tl,
34       ball~ color = \tl_use:N \l__point_color_tl,
35     }
36   ] file {\l__gnu_data_full_path_tl};
37   \group_end:
38 }
39 % reset precise (default 300 for plot precise)
40 \bool_if:cTF {g__#1_precise_bool} {
41   \PlotPrecise{#1}{300}
42 }{\relax}
43 }
44
45 % ==> simple 2d function
46 \NewDocumentCommand\Plot{ O{-5:5}O{color=black}O{}m }{
47   % sed gnuplot scripts
48   \sys_shell_now:x {sed~ -i~ "8s|set~ xr~ .*|set~ xr~ [#1]|"~
49   \g__ztikz_scripts_path_tl/plot.gp}
50   \sys_shell_now:x {sed~ -i~ "7s|f(x)~ =~ .*|f(x)~ =~ #4|" ~
51   \g__ztikz_scripts_path_tl/plot.gp}
52   \sys_shell_now:x {gnuplot~
53   \g__ztikz_scripts_path_tl/plot.gp}
54   % plot data, reset precise
55   \ztikz_gnu_data_plot cs:nnn {plot}{#2}{#3}
56 }
57
58 % ==> implicit 2d function
59 \NewDocumentCommand\ContourPlot{ O{-5:5; -5:5}O{color=black}O{}m }{
60   ^^I% split the first param
61   \seq_set_split:Nnn \l__plot_domain_seq { ; }{#1}
62   \tl_set:Nn \l__y_domain_tl {\seq_item:Nn \l__plot_domain_seq[2]}
63   \exp_args:Nx \tl_if_blank:nTF {\seq_item:Nn \l__plot_domain_seq[2]}{
64     \tl_set:Nn \l__y_domain_tl {*}*
65   }{
66     \tl_set:Nn \l__y_domain_tl {\seq_item:Nn \l__plot_domain_seq[2]}
67   }
68   % sed gnuplot scripts

```

```

66   \sys_shell_now:x {sed~ -i~ "11s|set~ xr~ .*|set~ xr~ [\seq_item:Nn
67   \l_plot_domain_seq[1]]|~ \g_ztikz_scripts_path_tl/contour_plot.gp}
68   \sys_shell_now:x {sed~ -i~ "12s|set~ yr~ .*|set~ yr~ [\tl_use:N \l_y_domain_tl]|~
69   \g_ztikz_scripts_path_tl/contour_plot.gp}
70   \sys_shell_now:x {sed~ -i~ "14s|f(x,~ y)~ =~ .*|f(x,~ y)~ =~ #4|~"
71   \g_ztikz_scripts_path_tl/contour_plot.gp}
72   \sys_shell_now:x {gnuplot~
73   \g_ztikz_scripts_path_tl/contour_plot.gp}
74   % plot data, reset precise
75   \ztikz_gnu_data_plot_cs:nnn {contour}{#2}{#3}
76 }
77
78 % ==> parametric 2d function
79 \NewDocumentCommand\ParamPlot{ O{0:2*pi}O{color=black}O{}m }{
80   % sed gnuplot scripts
81   \sys_shell_now:x {sed~ -i~ "8s|set~ trange~ .*|set~ trange~ [#1]|~
82   \g_ztikz_scripts_path_tl/param_plot.gp}
83   \sys_shell_now:x {sed~ -i~ "9s|plot~ .*|plot~ #4|~"
84   \g_ztikz_scripts_path_tl/param_plot.gp}
85   \sys_shell_now:x {gnuplot~
86   \g_ztikz_scripts_path_tl/param_plot.gp}
87   % plot data, reset precise
88   \ztikz_gnu_data_plot_cs:nnn {param}{#2}{#3}
89 }
90
91 % ==> polar 2d function
92 \NewDocumentCommand\PolarPlot{ O{0:2*pi}O{color=black}O{}m }{
93   % sed gnuplot scripts
94   \sys_shell_now:x {sed~ -i~ "8s|set~ trange~ .*|set~ trange~ [#1]|~
95   \g_ztikz_scripts_path_tl/polar_plot.gp}
96   \sys_shell_now:x {sed~ -i~ "9s|plot~ .*|plot~ #4|~"
97   \g_ztikz_scripts_path_tl/polar_plot.gp}
98   \sys_shell_now:x {gnuplot~
99   \g_ztikz_scripts_path_tl/polar_plot.gp}
100  % plot data, reset precise
101  \ztikz_gnu_data_plot_cs:nnn {polar}{#2}{#3}
102 }

```

### 9.2.7 Statistic Plot

```

1  % ==> statistic plot function
2  \cs_new_protected:Npn \ztikz_statistic_plot_cs:nnnn #1#2#3#4 {
3    \tl_if_empty:nTF {#3}{%

```

```

4      \draw[#2] plot[#1] file {#4};
5  }{
6    \group_begin:
7    \keys_set:nn { ztikz / point } { #3 }
8    \draw[#2] plot [
9      % stairs options
10     #1,
11     % marker options
12     mark = \str_use:N \l__point_type_str,
13     mark~ size = \dim_use:N \l__point_radius_dim,
14     mark~ options = {
15       rotate = \fp_use:N \l__point_rotate_angle,
16       opacity = \tl_use:N \l__point_opacity_tl,
17       color = \tl_use:N \l__point_color_tl,
18       ball~ color = \tl_use:N \l__point_color_tl,
19     }
20   ] file {#4};
21   \group_end:
22 }
23 }
24 \cs_generate_variant:Nn \ztikz_statistic_plot_cs:nnnn {xnnn}
25
26 \tl_new:N \l__tempa_tl
27 \tl_new:N \l__tempb_tl
28 \seq_new:N \l__statistic_option_tl
29 % 1. ListPlot --> set opacity=0 in the above Plot commands
30 % 2. stairs plot
31 \NewDocumentCommand\StairsPlot{ O{plot-left;jump-left}O{color=black}O{}m }{
32   \seq_set_split:Nnn \l__statistic_option_tl { ; }{#1}
33   \str_case:enF {\seq_item:Nn \l__statistic_option_tl{1}}{
34     {plot-left}{\tl_set:Nn \l__tempa_tl {const~plot~mark~left}}
35     {plot-right}{\tl_set:Nn \l__tempa_tl {const~plot~mark~right}}
36     {plot-mid}{\tl_set:Nn \l__tempa_tl {const~plot~mark~mid}}
37     {}{\tl_set:Nn \l__tempa_tl {}}
38   }{
39     \msg_new:nnn {ztikz}{ztikz-stairs-plot}{current~stairs~plot~type~is:~'#1'~,~
40     invalide}
41     \msg_error:nn {ztikz}{ztikz-stairs-plot}
42   }
43   \str_case:enF {\seq_item:Nn \l__statistic_option_tl{2}}{
44     {jump-left}{\tl_set:Nn \l__tempb_tl {jump~mark~left}}
45     {jump-right}{\tl_set:Nn \l__tempb_tl {jump~mark~right}}
46     {jump-mid}{\tl_set:Nn \l__tempb_tl {jump~mark~mid}}
47     {}{\tl_set:Nn \l__tempb_tl {}}
48   }

```

```

48     \msg_new:nnn {ztikz}{ztikz-stairs-plot}{current~stairs~jump~type~is:~'#1'~,~
49         invalide}
50     \msg_error:nn {ztikz}{ztikz-stairs-plot}
51 }
52 \ztikz_statistic_plot_cs:xnnn {\l__tempa_tl,\l__tempb_tl}{#2}{#3}{#4}
53 }
54 % 3. stem plot
55 \NewDocumentCommand\StemPlot{ O{x}O{color=black}O{}m }{
56     \str_case:enF {#1}{
57         {x}{\tl_set:Nn \l__tempa_tl {ycomb}}
58         {y}{\tl_set:Nn \l__tempa_tl {xcomb}}
59         {o}{\tl_set:Nn \l__tempa_tl {polar~ comb}}
60         {}{\tl_set:Nn \l__tempa_tl {ycomb}}
61     }{
62         \msg_new:nnn {ztikz}{ztikz-stem-plot}{current~stem~plot~type~is:~'#1'~,~
63             invalide}
64         \msg_error:nn {ztikz}{ztikz-stem-plot}
65     }
66 \ztikz_statistic_plot_cs:xnnn {\l__tempa_tl}{#2}{#3}{#4}
67 }
68 % 4. bar plot
69 \NewDocumentCommand\BarPlot{ O{ybar}O{color=black}O{}m }{
70     \str_case:enF {#1}{
71         {x}{\tl_set:Nn \l__tempa_tl {ybar}}
72         {y}{\tl_set:Nn \l__tempa_tl {xbar}}
73         {xc}{\tl_set:Nn \l__tempa_tl {ybar~ interval}}
74         {yc}{\tl_set:Nn \l__tempa_tl {xbar~ interval}}
75         {}{\tl_set:Nn \l__tempa_tl {ybar}}
76     }{
77         \msg_new:nnn {ztikz}{ztikz-bar-plot}{current~bar~plot~type~is:~'#1'~,~ invalide}
78         \msg_error:nn {ztikz}{ztikz-bar-plot}
79     }
80 \ztikz_statistic_plot_cs:xnnn {\l__tempa_tl}{#2}{#3}{#4}
81 }

```

### 9.2.8 ShadePlot

```

1  % 5. shade plot
2  \NewDocumentCommand\ShadePlot{ O{defaultMode}O{2pt}m }{
3      \path [shading=#1, shading~ path={draw=transparent!0, line~ width=#2}] plot file
4          {#3};
5

```

```

6  % ==> shade curve draw
7  \ExplSyntaxOff
8  %
9  % -----
10 % https://tex.stackexchange.com/questions/497977/tikz-draw-multicolor-curve-with-smooth-
11 % gradient
12 % -----
13 % -----
14 \newif\iftikz@shading@path
15 \tikzset {
16     fading xsep/.store in=\pgfpathfadingxsep,
17     fading ysep/.store in=\pgfpathfadingysep,
18     fading sep/.style={fading xsep=#1, fading ysep=#1},
19     fading sep=0.0cm,
20     shading path/.code= [
21         % Prevent this stuff happenning recursively.
22         \iftikz@shading@path
23         \else
24             \tikz@shading@pathtrue
25             % \tikz@addmode installs the `modes' (e.g., fill, draw, shade)
26             % to be applied to the path. It isn't usually for doing more
27             % changes to the path's construction.
28             \tikz@addmode{
29                 \pgfgetpath\pgf@currentfadingpath%
30                 % Get the boudning box of the current path size including the fading sep
31                 \pgfextract@process\pgf@fadingpath@southwest{\pgfpointadd{\pgfqpoint{\pgf@pathminx}{\pgf@pathminy}}%
32                     {\pgfpoint{-\pgfpathfadingxsep}{-\pgfpathfadingysep}}}
33                 \pgfextract@process\pgf@fadingpath@northeast{\pgfpointadd{\pgfqpoint{\pgf@pathmaxx}{\pgf@pathmaxy}}%
34                     {\pgfpoint{\pgfpathfadingxsep}{\pgfpathfadingysep}}}
35                 % Clear the path
36                 \pgfsetpath\pgfutil@empty%
37                 % Interrupt the path and picture to create a fading.
38                 \pgfinterruptpath
39                 \pgfinterruptpicture
40                 \begin{tikzfadingfrompicture}[name=.]
41                     \path [shade=none, fill=none, #1] \pgfextra {
42                         % Set the softpath. Any transformations in #1 will have no
43                         % effect.
44                         % This will *not* update the bounding box...
45                         \pgfsetpath\pgf@currentfadingpath
46                         % ...so it is done manually.
47                         \pgf@fadingpath@southwest
48                         \expandafter\pgf@protocolsizes{\the\pgf@x}{\the\pgf@y}%

```

```

45   \pgf@fadingpath@northeast%
46   \expandafter\pgf@protocolsizes{\the\pgf@x}{\the\pgf@y}%
47 }
48 % Now get the bounding of the picture.
49 \xdef\pgf@fadingboundingbox@southwest{\noexpand\pgfqpoint{\the\pgf@p-
50 icminx}{\the\pgf@picminy}}%
51 \xdef\pgf@fadingboundingbox@northeast{\noexpand\pgfqpoint{\the\pgf@p-
52 icmaxx}{\the\pgf@picmaxy}}%
53 \end{tikzfadingfrompicture}
54 \endpgfinterruptpicture
55 \endpgfinterruptpath
56 % Install a rectangle that covers the shaded/faded path picture.
57 \pgfpathrectanglecorners{\pgf@fadingboundingbox@southwest}{\pgf@fadingbo-
58 undingbox@northeast}
59 % Make the fading happen.
60 \def\tikz@path@fading{.}
61 \tikz@mode@fade@pathtrue
62 \tikz@fade@adjustfalse%10p
63 % Shift the fading to the mid point of the rectangle
64 \pgfpointscale{0.5}{\pgfpointadd{\pgf@fadingboundingbox@southwest}{\pgf@-
65 fadingboundingbox@northeast}}
66 \edef\tikz@fade@transform{shift={(\the\pgf@x,\the\pgf@y)}}
67 }
68 \fi
69 }
70
71 \def\ztikzShadeMode#1#2#3{
72   \pgfutil@tempcnta=0\relax
73   \pgfutil@for\pgf@tmp:=[#3]\do{\advance\pgfutil@tempcnta by1}
74   \ifnum\pgfutil@tempcnta=1\relax
75     \edef\pgf@spec{color(0)=(#3);color(100)=(#3)}
76   \else
77     \pgfmathparse{50/(\pgfutil@tempcnta-1)}\let\pgf@step=\pgfmathresult
78     \pgfutil@tempcntb=1\relax
79     \pgfutil@for\pgf@tmp:=[#3]\do{
80       \ifnum\pgfutil@tempcntb=1\relax
81         \edef\pgf@spec{color(0)=(\pgf@tmp);color(25)=(\pgf@tmp)}
82       \else
83         \ifnum\pgfutil@tempcntb<\pgfutil@tempcnta\relax
84           \pgfmathparse{25+\pgf@step/4+(\pgfutil@tempcntb-1)*\pgf@step}
85           \edef\pgf@spec{\pgf@spec;color(\pgfmathresult)=(\pgf@tmp)}
86         \else
87           \edef\pgf@spec{\pgf@spec;color(75)=(\pgf@tmp);color(100)=(\pgf@tmp)}
88         \fi
89       \fi
90     }
91   }

```

```

86          \advance\pgfutil@tempcntb by1\relax
87      }
88  \fi
89  \csname pgfdeclare#2shading\endcsname{\#1}{100}\pgf@spec
90 }
91 \ztikzShadeMode{defaultMode}{horizontal}{white,black}
92 \ExplSyntaxOn

```

## 9.3 Python

### 9.3.1 matplotlib

```

1 \cs_generate_variant:Nn \xsim_file_write_start:n {nx}
2 \cs_generate_variant:Nn \sys_shell_mv:n {xx}
3 % ==> python-matplotlib
4 \NewDocumentEnvironment{pyfig}{O{width=.75\linewidth}m }{
5     \newcommand{\pyfile}{\#2}
6     \xsim_file_write_start:nx {\c_true_bool}{\g_ztikz_python_path_tl/\#2}
7 }
8 \xsim_file_write_stop:
9 % step picture index
10 \int_gadd:Nn \g_picture_index_int {1}
11 % check if hash changed
12 \ztikz_hash_if_change:cs:x {\g_ztikz_python_path_tl/\pyfile}
13 \bool_if:NTF \g_hash_change_bool {
14     % add save figure to source
15     \sys_if_platform_windows:TF {
16         \exp_args:Nx \sys_shell_now:n {
17             echo~ plt.savefig('\g_ztikz_python_path_tl/\pyfile.pdf')~ >>-
18             \g_ztikz_python_path_tl/\pyfile
19         }
20     }{
21         \exp_args:Nx \sys_shell_now:n {
22             echo~ "plt.savefig('\g_ztikz_python_path_tl/\pyfile.pdf')"\~ >>-
23             \g_ztikz_python_path_tl/\pyfile
24         }
25     % execute python source
26     \exp_args:Nx \sys_shell_now:n {python~ \g_ztikz_python_path_tl/\pyfile}
27     \includegraphics[#1]{\g_ztikz_python_path_tl/\pyfile.pdf}
28     \typeout{Writing~ 'pyfig'~environment~source~to~\tl_use:N
29             \g_ztikz_python_path_tl/\pyfile}
30 }

```

```

29     \includegraphics[#1]{\g__ztikz_python_path_t1/\pyfile.pdf}
30     \typeout{skip-recompile-by-python,~using-the-cache-picture-\int_use:N
31     \g__picture_index_int}
32 }

```

### 9.3.2 pycode

```

1  % ==> python-code-env
2  \NewDocumentEnvironment{pycode}{ m }{
3      \newcommand{\pyfile}{#1}
4      \xsim_file_write_start:nx {\c_true_bool}{\g__ztikz_python_path_t1/#1}
5  }
6  \xsim_file_write_stop:
7  % step picture index
8  \int_gadd:Nn \g__picture_index_int {1}
9  % check if hash changed
10 \ztikz_hash_if_change_xs:\g__ztikz_python_path_t1/\pyfile
11 \bool_if:NTF \g__hash_change_bool {
12     % execute python source
13     \exp_args:Nx \sys_shell_now:n {python- \g__ztikz_python_path_t1/\pyfile}
14     \input{\g__ztikz_python_path_t1/\pyfile.out}
15     \typeout{Writing- 'pycode'-environment-source-to-\tl_use:N
16     \g__ztikz_python_path_t1/\pyfile}
17 }
18 \input{\g__ztikz_python_path_t1/\pyfile.out}
19 \typeout{skip-recompile-by-python,~using-the-cache-pycode-result-\int_use:N
20 \g__picture_index_int}

```

### 9.3.3 sympy

```

1  % ==> python-sympy
2  \NewDocumentCommand{\sympy}{m}){
3      % step sympy result index
4      \int_gadd:Nn \g__sympy_index_int {1}
5      \tl_set:Nx \l__current_sympy_index_tl {\int_use:N \g__sympy_index_int}
6      % sympy source write
7      \sys_shell_now:x {sed~ -i~ "8s|F_res~ =~ .*|F_res~ =~ #1|~"
8      \g__ztikz_scripts_path_t1/sympy_script.py}
9      % check hash

```

```

9   \ztikz_hash_if_change_{\g_ztikz_scripts_path_tl/sympy_script.py}
10  \bool_if:NTF {\g_hash_change_bool} {
11    \sys_shell_now:x {python~ \g_ztikz_scripts_path_tl/sympy_script.py}
12    \sys_shell_mv:xx
13    {\g_ztikz_python_path_tl/sympy.out}
14    {\g_ztikz_python_path_tl/sympy_\int_use:N \g_sympy_index_int .out}
15    \typeout{using~python~sympy~calculating~question~\l_current_sympy_index_tl ...}
16    \exp_args:Nx
17    \input{\g_ztikz_python_path_tl/sympy_\l_current_sympy_index_tl.out}
18  }
19  \exp_args:Nx
20  \input{\g_ztikz_python_path_tl/sympy_\l_current_sympy_index_tl.out}
21  \typeout{skip~recompile,~using~the~cache~sympy~result~\l_current_sympy_index_tl}
22 }
```

### 9.3.4 python eval

```

1  % read from external file
2  \cs_new_protected:Npn \zlateX_Readlines_{nn} #1#2 {
3    \ior_open:Nn \g_file_read_ior {\#2}
4    \str_case:nnF {\#1} {
5      {hold} {
6        \ior_get:NN \g_file_read_ior \g_file_content_tl
7      }
8      {str} {
9        \ior_str_get:NN \g_file_read_ior \g_file_content_tl
10     }
11   } {}
12   \tl_use:N \g_file_content_tl
13 }
14 \cs_generate_variant:Nn \zlateX_Readlines_{nn} {xx}

15 % ==> 1-line python command
16 \NewDocumentCommand\py{0{raw}m} {
17   % sympy source write
18   \sys_shell_now:x {sed~ -i~ "6s|Float_res~ =~ .*|Float_res~ =~ \tl_to_str:n {\#2}|"~}
19   \g_ztikz_scripts_path_tl/python_script.py
20   % calculation
21   \typeout{using~python~float~module~calculating...}
22   \sys_shell_now:x {python~ \g_ztikz_scripts_path_tl/python_script.py}
23   % using \ior_get:
```

```

24     \zlateX_Readlines_cs:xx {#1}{\g__ztikz_python_path_tl/PyFloat.out}
25     % ---> cause bug that can't write ToC to file
26     % \iow_close:N \g__file_read_iow
27 }

```

## 9.4 Mathematica

### 9.4.1 SetUp

```

1  % temp var for manipulate
2  \tl_new:N \l__temp_arg_tl
3  \tl_new:N \l__wolfram_temp_res_tl
4  \ior_new:N \l__temp_io
5  \seq_new:N \l__wolfram_temp_res_seq
6
7  % macro storage the mma-result
8  \NewDocumentCommand{\wolframResult}{O{raw}O{}){
9      \tl_if_eq:nnTF {#1}{raw}{
10         \seq_use:Nnnn \l__wolfram_temp_res_seq {#2}{#2}{#2}
11     }{
12         \if_is_int:xTF {\int_eval:n {#1}}
13             {\seq_item:Nn \l__wolfram_temp_res_seq {#1}}
14             {}
15     }
16 }
17
18 % check if integer
19 \prg_new_protected_conditional:Npnn \if_is_int:n #1 { T, F, TF }{
20     \regex_match:nnTF { ^[+\-]?[\d]+\$ } {#1} % $
21     { \prg_return_true: }
22     { \prg_return_false: }
23 }
24 \cs_generate_variant:Nn \if_is_int:nTF {xTF}

```

### 9.4.2 picture

```

1  % wolfram graphics
2  \NewDocumentEnvironment{wolframGraphics}{ O{width=.75\linewidth}m }{
3      \newcommand{\mmafile}{#2}
4      \xsim_file_write_start:nx {\c_true_bool}{\g__ztikz_mma_path_tl/#2}
5  }

```

```

6   \xsim_file_write_stop:
7   % step picture index
8   \int_gadd:Nn \g__picture_index_int {1}
9   % check if hash changed
10  \ztikz_hash_if_change_cs:x {\g__ztikz_mma_path_tl/\mmafile}
11  \bool_if:NTF \g__hash_change_bool {
12      % execute mathematica script
13      \exp_args:Nx \sys_shell_now:n {wolframscript~ -script~
14          \g__ztikz_mma_path_tl/\mmafile}
15      \includegraphics[#1]{\g__ztikz_mma_path_tl/\mmafile.pdf}
16      \typeout{Writing~ 'mmafig'~environment~source~to~\tl_use:N
17          \g__ztikz_mma_path_tl/\mmafile}
18      \H
19      \includegraphics[#1]{\g__ztikz_mma_path_tl/\mmafile.pdf}
20      \typeout{skip~recompile~by~wolframscript,~using~the~cache~picture~\int_use:N
21          \g__picture_index_int}
22  }
23 }
```

#### 9.4.3 Symbols Calculation

```

1  % input result of wolfram
2  \cs_generate_variant:Nn \ior_open:Nn {Nx}
3  \cs_new_protected:Nn \ztikz_wolfram_input_result_cs: {
4      % write export command
5      \iow_now:Nx \g_tmpa_iow {Export["\g__ztikz_mma_path_tl/mma_res_\int_use:N
6          \g__mma_index_int.txt", TeXResult]}
7      \iow_close:N \g_tmpa_iow
8      % calculate and input
9      \ztikz_hash_if_change_cs:x {\g__ztikz_mma_path_tl/mma_calc_\int_use:N
10         \g__mma_index_int.wls}
11      \bool_if:NTF \g__hash_change_bool {
12          % write export command
13          \sys_shell_now:x {wolframscript~ -script~
14              \g__ztikz_mma_path_tl/mma_calc_\int_use:N \g__mma_index_int.wls}
15          % \exp_args:Nx \input{\g__ztikz_mma_path_tl/mma_res_\int_use:N
16              \g__mma_index_int.txt}
17          \typeout{using~wolframscript~calculating~question~\int_use:N \g__mma_index_int
18              ...}
19      \H
20          % \exp_args:Nx \input{\g__ztikz_mma_path_tl/mma_res_\int_use:N
21              \g__mma_index_int.txt}
22          \typeout{skip~recompile,~using~the~cache~wolframscript~result~\int_use:N
23              \g__mma_index_int}
```

```

16 }
17 % split result to seq and storage result to a macro
18 \ior_open:Nx \l__temp_io {\g__ztikz_mma_path_tl/mma_res_\int_use:N
19   \g__mma_index_int.txt}
20 \ior_get:NN \l__temp_io \l__wolfram_temp_res_tl
21 \seq_set_split:NnV \l__wolfram_temp_res_seq{,}\l__wolfram_temp_res_tl
22 % step counter, add export command
23 \int_gadd:Nn \g__mma_index_int {1}
24 }

```

#### 9.4.4 wolfram code

```

1 % wolfram code
2 \cs_generate_variant:Nn \iow_open:Nn {Nx}
3 \NewDocumentCommand\wolfram{O{tex}m}{
4     % write mathematica code
5     \iow_open:Nx \g_tmpa_iow {\g__ztikz_mma_path_tl/mma_calc_\int_use:N
6     \g__mma_index_int.wls}
7     \str_case:nnF {#1}{
8         {tex} {
9             \iow_now:Nx \g_tmpa_iow { TeXResult = ToString[TeXForm[#2]]; }
10        }
11        {text} {
12            \iow_now:Nx \g_tmpa_iow { TeXResult = ToString[#2]; }
13        }
14    }{\relax}
15    % cache and input result
16    \ztikz_wolfram_input_result_cs:
}

```

### 9.4.5 equation Solve

```

1  % equation solve
2 \NewDocumentCommand{\wolframSolve}{O{part}m0{}0{}0{}0{}0{}}
3     % prepend ',' to #4 if not empty
4     \tl_if_empty:nTF {#4} {
5         \tl_set:Nn \l__temp_arg_tl {}
6     }{
7         \tl_set:Nn \l__temp_arg_tl {,#4}
8     }
9     % write mathematica code

```



```
49      % cache and input result
50      \ztikz_wolfram_input_result_cs:
51 }
```