zTEX 宏集介绍

Eureka zongpingding5(at)outlook(dot)com

由于本人时间有限, 目前此宏集的开发暂停.

September 21, 2025

1 简介

1.1 为何叫 zTEX?

为何宏集名称里有 'z' 这个前缀, 这也许应是许多用户想知道的问题? 可能的原因:

(1) 看到 LATEX3 开发团队用 "x" 来作为他们开发的一系列宏包前缀, 比如 xparse, xcoffins, xfp 等。我便不能再使用 "x" 这一前缀了. 这个时候, 突然想到了一个字母 – "z". 一方面 "x → y → z", 有了 "x", 才有 "z" (zTEX 全部基于 LATEX3 进行开发; 可以说, 没有 LATEX3, 就没有今天的 zTEX). 那么 "y" 去哪里了? 当作为用户的你 (you) 加入 zTEX 使用者阵营后, 就有 "y" 了.

(2) 你将 'z' 逆时针旋转 90°, 就可以得到 "阿列夫 – ℵ": 我希望 zTEX 宏集能够有进一步 (无限) 拓展的可能; 这个宏集在设计之初, 便一直坚持可拓展性这一原则. 普通用户可以使用用户层面的命令, 模板制作者可以使用 zTEX 提供的编程接口. 尽管 "ℵTEX" 这个目标有些不切实际, 但是万一实现了呢?

(3) 也许是看到了 TikZ 中的 "z", 于是便以 'z' 为本系列宏集的前缀了.

最开始的 zTEX 宏集仅包含一个基本的 zlatex.cls 文档类, 而且原来的名称叫做 "πLATEX"; 后面我又想基于 TikZ 开发一个绘图宏包, 用于实现常见平面图形的绘制以及外部程序的交互; 再后来发现 beamer 用起来很不方便, 便开发了 slide 库; 随着开发的不断深入, 我发现我已经在 ztex.cls 中写了很多十分有用的宏了, 于是我把这些宏分化了出来, 得到了 ztool 宏包, 得到了 thm, cmd, font, ... 这些模块, 以及 slide, alias, thm ... 这些库; 最终, zTEX bundle 诞生了.

1.2 为何用 zTEX?

为什么要用我这个 zTEX 宏集? zTikZ 中负责和外部程序交互的那几个模块现在处于一种比较尴尬的境地, 用户如果会用这些程序, 那么你可以单独使用这些程序调整图片的所有细节, 最后在 LATEX 中插入该图片. 如果用户不会使用这些外部拓展程序, 那么用户不仅需要先学习该程序的用法, 还需要学习 zTikZ 宏集中对应命令的 LATEX 语法; 这无疑是增加了用户的负担!

用户可以再思考这样一个问题: 我已经会用 LATEX 自己写模板了, 为什么还要用别人的模版? 我如果不会用 LATEX 写模板, 花费了大量的时间去了解一个庞大且复杂的模板的使用细节, 那么我为何不花费这些时间自己去学习 LATEX, 这样更能做出满足自己需求的模板? 最后还可以进一步推出: 我为什么一定要用 TEX 或 LATEX 呢? 用 Word, Indesign 这些成熟的软件, 甚至是手写, 难道就不能写一篇规范的论文/笔记吗?

所以为什么 Knuth 老爷子要花费十年的时间去开发 TEX 呢?

上述的一系列推论正确吗? 仔细想一想, 上面的推导其实不都是正确的. 前一个条件并不一定是充分的, 或者说我们使用了一个假命题 (关系) 去得到了另一个命题 (关系).

根据基础的逻辑知识: 定义汇集 $R \vee S$ 为两关系 R, S 的逻辑析取, 定义汇集 $\neg R$ 为关系 R 的逻辑否定. 从而我们就可以定义所谓的 “逻辑蕴含” 关系 $\Rightarrow$, 即记号 $R \Rightarrow S$, 前者其实是如下的关系汇集:

$$S \vee (\neg R)$$

注记 1.1 其实有 $\neg, \vee$ 这两个基础的符号就已经能表示出很多的关系了; 比如逻辑合取记号: $R \wedge S$, 它其实就是: $\neg[(\neg R) \vee (\neg S)]$. 在规定逻辑公理后, 就可以用它们来说明常用的 “三段论, 双重否定” 等逻辑推理了. 比如我们常用的逆否命题就是说: 关系 $(R \Rightarrow S) \Rightarrow ((\neg S) \Rightarrow (\neg R))$ 是真的.

在我们定义了关系 “真” 后, 如果关系 $R \Rightarrow S$ 是真的, 那么:

- 当关系 R 为真的, 关系 S 必然是真的, 也就是我们得到了一个 “真” 的结论;
- 但如果 R, S 同时为假, 关系 $R \Rightarrow S$ 也是真的. 而此时我们的结论并不是 “真的”, 也就是结论并不成立.

可以认为我们用一个假命题导出了另一个假命题, 下面说明 zTeX 值得你去用, 我将要如何去说服你呢?

让 “$R \Rightarrow S$” 中的命题 “R” 为假就好了. zTeX 的上手难度相较于默认的 LaTeX 要低一点, 达到同样的排版效果, 你所花费的时间更少. 故上述 “花费同样时间” 这一个命题为假, 即 “zTeX 值得你用” 这一命题成立. 你也许可以用其它的方式来反驳我, 但至少我找到了一个论据来说服我自己, 也找到了我开发这个宏集的初心.

1.3 项目维护

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

GitHub : https://github.com/zongpingding/zTeX_bundle

Gitlab : https://gitlab.com/zongpingding/zTeX_bundle

Gitee : https://gitee.com/zongpingding/zTeX_bundle

项目中包含: ztex 文档类, zTikZ 宏包, 以及 ztool 宏包的源码与用户手册. zTeX 宏集以 lppl 协议开源, 欢迎各位对源代码进行修改与二次分发. 若用户在使用此宏集的过程中发现任何的 Bug, 或想提出改进意见, 请在 Github 上提 Issue 或直接提交 PR.

请不要在 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 仓库中的内容同步到这两处. 后续的开发过程中, 三者不会同步更新, 请以 Github 仓库为准.

本项目为完全免费、纯属兴趣驱动（为爱发电）之作。对于任何使用本模板所引发的严重后果，我概不负责。我非常乐意帮助大家解决问题，但在提问之前，请务必先了解 LaTeX 的提问规范.

当前宏集的稳定版本于半年之前发布, 最新的开发版请切换到 “dev” 分支; 本手册适用于当前最新的开发版. 请到: Release 界面 下载.

1.4 基本组成

zTEX 宏集包含如下内容:

- ztex 文档类;
- ztool 宏包;
- ztikz 宏包;
- zslide 宏包 (不推荐使用).

zTEX 宏集独立实现了一个 ztool 宏包, 它是 zTEX 宏集中各文档类或宏包的基础. 此宏包中包含原来已被废弃的 l3sys-shell 中的所有命令. 除此之外, ztool 提供了 box 操作, 文件 IO 以及基本图形绘制相关的函数. 在 ztool 的协助下, zTEX 能够避免或减少命令行 `-shell-escape` 参数或其它相关宏包的调用 (如 robust-externalize 宏包).

ztex 文档类对标 memoir, koma-script 宏集, 用于生成书籍或演示文稿. 尽管在 zTEX 中, 直接将 `layout/slide` 选项置为 `true` 即可生成演示文档, 但该库目前很不成熟, 在严肃场合中, 推荐使用原始的 beamer 或 ctexbeamer 文档类.

zTikZ 宏包提供了绘制平面图形以及调用外部程序的接口[1]. zslide 宏包是自己临时设计的一套 beamer 主题, 还未进行常规测试, 请谨慎使用.

从本介绍文档即可看出，本模板整体风格较为朴素，未采用华丽的配色方案或精致的页面设计。然而，在长时间尝试和调试 LaTeX 模板的过程中，我逐渐发现这种简洁质朴的风格最符合广大 LaTeX 用户的使用习惯与审美偏好. 若你更倾向于精美的排版风格，亦可参考其他的模板，如 ElegantLaTeX、BeautyLaTeX 等.

1.5 用户手册

普通 LaTeX 用户可跳过本文档的 “节 (3)”. 该部分主要记录了我对本模板设计思路的说明，以及个人在编写 LaTeX 过程中的一些体会，对模板或宏包的实际使用并无直接帮助。若你希望了解 ztex 文档类的具体用法，请参阅 `zlatex_interface.pdf`; 若需了解 ztikz 宏包的使用方法，请参阅 `ztikz_interface.pdf`. 目前 zslide 宏包尚无详细文档，仅提供了示例文件 `zslide_manual.pdf` 供用户参考. ztool 宏包主要为模板的开发者准备, 普通用户无需阅读.

[1] 众所周知, 在 LaTeX 中绘图是一件十分痛苦的事, 于是乎你会看到很多书籍或笔记中的图形都是手绘或截图, 并非矢量图

2 安装使用

2.1 在线模板

为了让部分用户可以直接使用到 zTeX，免去“繁杂”的环境配置. 我已将本模板部署在 TeXPage 上，地址为: TeXPgae zTeX Project，直接打开此地址即可体验. 由于技术原因，zTikZ 请在本地体验.

2.2 本地安装

zTeX 宏集目前还未上传 CTAN, 因为还没有开发完成. 本文档类使用的部分 LaTeX3 命令在老版本的 TeXLive 下并不存在, 若用户的 TeXLive 版本过低，则可能无法正常使用本宏集. 目前 zTeX 文档类在各平台的兼容情况为:

Windows : TeXLive 最低版本 2024

Linux : TeXLive 最低版本 2024

MacOS : MacTeX 还未测试

因 zTeX 还未传入 CTAN(未来可能会考虑), 所以想要使用此文档类，只有如下两种方法:

- 把此宏集 – `ztex` 目录中的所有内容放入当前项目文件夹下;
- 在命令行运行命令: `kpsewhich -var-value=TEXMFHOME`, 在 Windows 上这个路径一般是: `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. 在此路径下新建文件夹 `tex/latex/ztex`; 此文件夹对应的路径我们记为 ⟨zTeX⟩, 随后把 `ztex` 目录中的所有内容放入 ⟨zTeX⟩ 下即可.

在本手册后续，我们使用 ⟨zTeX⟩ 表示本宏集的根目录.

NOTE: 如果用户不需要使用 alias 库, 那么一些比较老 TeXLive 也能运行此宏集.

3 开发过程

本模板的设计经历了较长时间的积累与迭代。最初接触 LaTeX 时，我只是将常用的宏整理进一个 .sty 文件中，误以为这便是一个宏包 (实际上它称得上是一个宏包). 随后接触到了 ElegantLaTeX 系列模板，并曾使用其中的 elegantbook 文档类撰写笔记。然而，随着使用深入，我逐渐发现模板默认的样式并不完全符合个人需求，许多细节希望能够自行定制。遗憾的是，当时对 LaTeX 的理解尚浅，面对复杂的模板源码无从下手 (打开任何一个模板，映入眼帘的源码对于我来说与一堆乱码无异)。后续通过查阅资料、阅读相关文章，逐步积累经验，渐渐熟悉了 LaTeX 中的各种命令与机制，才最终开始着手本模板的独立设计.

ZTeX 的第一版基本是在 elegantbook 文档类的基础上修改而成，仅在字体、配色等方面做了一些简单调整。然而，随着功能的不断叠加，模板逐渐变得混乱，代码结构也变得难以维护[2]。其中，键值对接口的实现对于我来说尤为困难。以文档类语言切换功能为例，当时通过 \ifdefstring 实现，以下是当初的相关代码片段：

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

代码的书写过程颇为繁琐。当时模板仍以 article 文档类为基础，缺乏许多 book 文档类中内置的计数器与章节结构，不得不自行声明相关命令。然而，自定义的命令常与其他宏包不兼容，尤其是在集成 hyperref 宏包时问题频出。由于计数器定义不规范，导致跳转功能异常。例如，使用 \label 时，所激活的跳转目标往往并非正确的章节位置，目录中的链接也存在类似问题，使用体验大打折扣。

另一方面，初代 ZTeX 文档类完全基于 LaTeX2ε 构建，许多宏展开相关的代码写的不仅繁琐，逻辑也很混乱。当时经验有限，模板中的大多数解决方案都借鉴 (抄袭) 自 TeX-StackExchange 上的回答，导致整个模板虽然“能跑”，但对其中许多命令的具体作用并不真正理解，并不清楚这些“解决方法”会不会产生一些不为人知的副作用.

[2] 事实上，最初 ztex 与 ztikz 宏包是写在一起的，整体结构非常凌乱.

3.1 ztex

后来，我将 ztikz 宏包从原有的 ztex 文档类中剥离出来，并使用 LaTeX3 对原始文档类和 ztikz 进行了重构。zTeX 文档类默认基于 article 文档类构建，同时也支持加载其他文档类。此阶段的开发理念发生了显著变化：在添加任何的配置前，我都会事先明确其提供的功能，了解该配置需要的依赖，这一配置对已有的代码或宏包有无影响，...，然后再自行编写代码实现。由此，zTeX 的开发正式开始了．事实证明，基于 LaTeX3 的重构极大提升了代码的清晰度和整体开发效率。以下为当时 ztex 文档类选项的相关声明：

```
\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}
```

看起来确实清爽了许多，但很快我意识到，这样的实现方式在实际使用中仍不够灵活。问题在于：当需传递给子文档类的选项较多时，必须逐一声明大量键值对；而当整个文档类中键值对数量庞大时，维护成本显著增加。为了解决这一问题，我引入了 l3keys 提供的元键机制（.meta:nn）。其核心作用在于：通过模块化管理各类键值对，实现层级式组织与调用，从而提升代码的可读性与扩展性。以下是当时 ztex 文档类中键值接口的实现代码：

```
\zlatex_define_option:n {
  % zlatex language
  lang              .str_gset:N   =  \g__zlatex_lang_str,
  lang              .initial:n    =  { en },
  % class and options
  class             .str_gset:N   = \g__zlatex_subclass_type_str,
  class             .initial:n    = { book },
  classOption       .clist_gset:N = \g__zlatex_subclass_option_clist,
  classOption       .initial:n    = { oneside, 10pt },
  % zlatex options meta key
  layout            .meta:nn      = {zlatex / layout}{#1},
  mathSpec          .meta:nn      = {zlatex / mathSpec}{#1},
  font              .meta:nn      = {zlatex / font}{#1},
}
```

为了轻松处理子文档类选项的加载问题，我引入了 ⟨classOption⟩ 这个键．

3.2 ztikz

开发宏包 ztikz 也花了我很多的时间，ztikz 从最开始的一个小宏包变成了一个拥有众多拓展库的庞然大物．这段时间，我为 ztikz 宏包开发了 cache，python，gnuplot，wolfram 和 l3draw 库．这些库可以先通过下面的命令进行声明：

```
\ProvidesExplFile{ztikzmodule.cache.tex}{2024/06/15}
  {1.0.0}{cache~module~for~ztikz}
```

然后在主宏包 ztikz 中使用如下命令进行调用：

```
\cs_new_nopar:Npn \g__ztikz_load_module:n #1
  {
    \clist_map_inline:nn {#1}
      { \file_if_exist_input:nF {modules/ztikzmodule.##1.tex}{} }
  }
\NewDocumentCommand\ztikzLoadModule{m}
  {
    \g__ztikz_load_module:n {#1}
  }
```

划分出 ztikz 的库后，宏包使用者只需通过如下的命令就可以轻松调用：

```
\ztikzLoadModule{cache, python}
```

而且，将一个宏包划分为一个个的库来开发这一行为，不仅可以方便宏包的使用者，更让宏包的开发者可以聚焦于单个库的开发，这极大地提高了我的开发效率．

在开发 ztikz 的 cache 库时，我遇到了数不清的困难，包括但不限于：

- 怎么将一个环境中的内容不加改变地输出到外部文件中？
- 怎么为每一个需要缓存的内容“打”上一个唯一的“身份标签”？
- 为什么同样都是字符串，但是 string 和 token list 在 `\tl_if_eq:nn` 中就是判断为不相等？
- 怎么调用上一次的缓存结果？
- 怎么临时忽略缓存机制，或强制调用上一次的缓存结果？
- 怎么提供对应的编程接口？
- ...

虽然，上述的问题目前均已解决，但目前的 cache 库仍有缺陷：

- 无法去除 tikz 的 externalize 库依赖，我自己还没有能力自己写一个 externalize 库出来．

- 无法提供与 Matlab 的交互接口.
- cache 库提供的普通用户接口仍然过于复杂.
- ...

3.3 ztool

大概是开发到中后期的时候, 我发现我在 ztex 或 ztikz 中定义了大量与此宏包无关的宏, 比如 “TEX 盒子操作”, “shell-escape”, “文件 IO 操作”; 然后我便把这些宏分离到了 ztool 宏包中. 上面的这些功能几乎时没有什么关联的, 后面我更是在 ztool 宏包内将它们划分为了下面的这几个部分:

- shell-escape,
- file-io,
- box,
- zdraw;

它们之间互不干扰, 用户在使用时仅需加载其需要的部分即可; 比如用户需要使用 file-io 中的一个宏, 他只需要使用如下的命令:

```
42 \ztoolloadlib{file-io}                                    42
```

此时, ztool 仅会加载 file-io 相关的宏, 其它部分的宏则不会被加载. ztool 实现这一机制同样是使用了上述方法 – 将 ztool 划分为一个个的库.

3.4 l3build

我之前完全没有接触过 LaTeX 相关的 “代码测试” 内容, 一个偶然的时间, 我发现了 l3build. 我们写的代码是需要测试的: 你需要确保后续开发的代码不会影响之前的代码, 怎么保证呢? 写好单元测试, 每次添加新功能后就跑一跑单元测试, 如果全部的测试都通过了, 那么你后续的开发是没问题的. 当然, 你的单元测试必须得写全面了.

最开始的自己很懒, 不想写测试, 觉得费时间, 多写一点代码不好吗? 但若你后续写的代码破坏了前面已有的功能, 这段代码就是没有意义的. 所以要勤于写单元测试!

4 宏集设计

4.1 设计参考

本系列自诞生以来始终由我个人独立开发，过程中借鉴了诸多优秀的文档类与宏包。其中，参考最多的是 CTEXart 文档类，它为本项目提供了主要的设计思路，该文档类完全基于 LATEX3 编写，在选项配置模块方面，它给了我很多启发。

ZTEX 宏集中的文档类或宏包的 Key-Value 接口先是参考了 TEX-StackExchange 上的相关讨论，然后再采用了 LATEX3 的 l3keys 模块实现。此方案的优点是显而易见的: 配置接口简洁明了、符合用户习惯、同时也便于模板的后续维护与扩展.

在后续的开发过程中，CUSTeX 宏集也为我带来了诸多启发，我参考了其中许多优秀的设计方案。尤其值得一提的是该项目将 “用户接口” 与 “编程接口” 进行区分的思想，对此宏集后续的开发影响颇深.

4.2 设计原则

说实话，这个标题可能有些夸大了 – “设计原则” 究竟指的是什么，我自己也不清楚。我只是希望我的模板看起来足够舒服而已。那怎样才能让一个模板 “看着舒服” 呢？我也无法给出明确答案。但至少，它应该与页边距、字体大小、字体样式等因素有关系。更进一步地说，这些因素并非彼此独立，而是相互制约、共同作用的。举例而言，当页边距增大、版心变小时，正文字体的大小也应随之调整，以维持整体的视觉平衡和可读性。

当时遇到了一个问题：一行设置多少个字符才合适？在查阅 TEX StackExchange 相关讨论后发现，对于英文文本来说，一行包含 65–90 个字母被认为是较为理想的范围，且常见的正文字体尺寸为 `10pt`、`11pt` 或 `12pt`。

至于页边距应如何设置，我参考了 elegantbook, ctexart 等文档类的设计，也逐渐总结出一些经验。起初，测量页面布局中的各项距离是非常不方便的，我都是动用尺子手动测量的。后来我发现了一个非常实用的宏包——fgruler，它可以在生成的 PDF 中直接显示页面布局的尺寸信息，且使用方法也非常简便：

```
43 \usepackage[hshift=0mm,vshift=0mm]{fgruler}                    43
```

当你在导言区加入上述配置后，生成 PDF 的每页都能看到如 图 (1) 这样的输出. 我终于摆脱使用尺子手动量这一方法了!

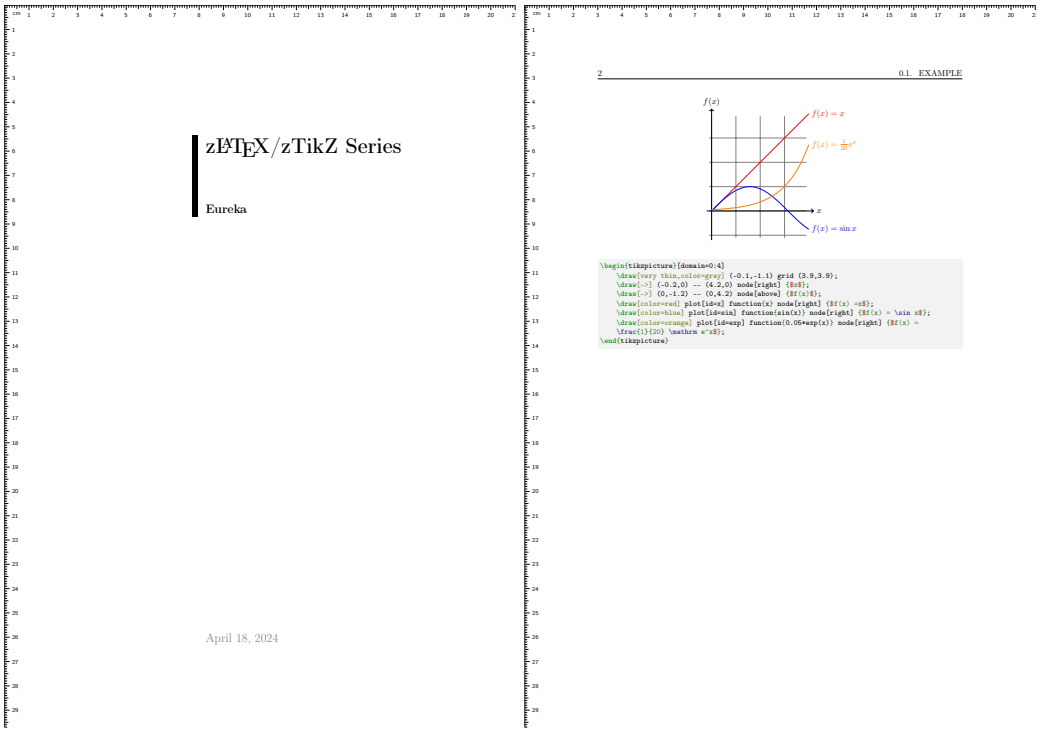


图 1: 页面布局示意图

在设计本宏集时，我始终在字体配置上有所犹豫：是否应将字体打包进模板？是否应

在模板中为用户设置默认字体？在本宏集的最初版本中，我尝试收集了一些免费的中英文字体，并直接放置在模板的文件夹中。然而，这种做法也带来了不少问题：

- 部分用户真的需要该字体吗？增加的字体会变成模板或用户的负担吗？
- 该字体可以随意传播吗？万一某个用户将该字体进行了商用？
- 部分中文字体包含的字形往往是不全的，怎么解决？
- ...

最终的处理办法：本宏集不打包任何的字体，但添加部分 TEXLive 内置字体配置；宏集本身提供字体设置的接口，但所有的字体定义与样式由用户指定．除此之外，ZTEX 还提供了数学字体配置接口，以供用户选用．

在开发 ZTEX 宏集的过程中，行距等排版细节也曾让我困扰许久。实际上，设计一个模板需要考虑的因素远比预期复杂，几乎每一个参数的设置都会相互影响。不过，在反复尝试与调整的过程中，我也逐渐总结出一条经验：对于一时把握不准的配置，就保留默认设置。

Be simple, be fool – 保持简单，反而更容易达到稳定和谐的效果。

尽管在开发过程中遇到了诸多困难，ZTEX 最终仍未烂尾，顺利完成并呈现在了大家面前。

4.3 无题

时至今日, 再次回头来看我的这个模板, 我反而有了一些其他的感受. 一个模板到底需要给用户定制什么东西? 到底需要给用户多大的自由空间 (配置选项)? 如果你的配置选项过多, 像 koma-script, Memoir 那样, 模板作者给用户处理了很多的细节, 提供了种类繁多的接口. 或者像部分简单的模板仅提供几个必要的设置和命令; 而且, 如果一个模板的说明文档都达到了上百页, 那么我作为一个用户为什么不自己学习做模板, 写一个适合自己的模板, 反而要话这部分时间来学习使用你的模板? 如果模板的配置选项过少, 那么用户又会觉得这个模板不够灵活. 所以, 到底什么样的一个模板设计才能够称得上是:**简单**, **灵活**, **易用**? 遗憾的是, 现在我也没有办法回答这个问题, 所以这个问题作为习题, 留给使用者回答了...

发展至今，LaTeX 宏集早已不再是一个简单的“文档类 + 绘图库 + 幻灯片”集合，这也使得它并不适合 LaTeX 初学者使用。在开发的过程中，我也逐渐意识到：很多时候，我们并不一定需要亲自设计一个模板。更合理的做法或许是 – 根据自己的需求，选择合适的功能性宏包，并通过它们提供的接口实现所需的功能。这种方式不仅更贴合实际使用场景，也减少了与其他宏包的兼容性问题，更无需投入大量时间去理解第三方模板的结构与细节.

实际上，article、book 等基础文档类，加上丰富的功能宏包，已经足以满足绝大多数排版需求。也许我们并不需要再去重复造一个模板的“轮子”。相比之下，我更认同将精力投入到基础性宏包的开发上，就如 pgf、l3draw 等优秀项目所做的那样 – 它们专注于提供一组底层的绘图或功能接口，将更高层的封装留给用户根据自身需求自行实现。

Happy LaTeXing !

>_<

5 文档指南

5.1 记号说明

本宏集的所有用户手册均遵守如下规范:

- 命令和键值对采用打字机字体;
- 键的默认值通过加粗标明, 并且与右侧蓝色文本一致;
- 所有命令排版格式为: `\cmd[oArg]{pArg}`;
- 所有键值排版格式为: ⟨key⟩ `= value`;

5.2 复制样例

LaTeX 宏集的所有用户手册均提供了大量示例及其对应的代码。为提升阅读体验，在排版过程中对部分代码抄录环境中的符号进行了格式上的调整。例如:

- 在示例代码中，换行符可能以 "↙" 表示，复制代码时请将该符号删除;
- 若示例中包含行号，请在复制后手动去除多余的行号;
- 此外，在后续的 Implementation 节中，部分代码因排版原因进行了换行，使用时请根据实际情况去除不必要的换行符，以确保代码能够正确编译。

5.3 键值指定

本系列中的大多数命令均采用键值对形式调用，因此，如果某个命令的可用键较多，而用户手册中的说明又较为模糊，用户可参考手册末尾 Implementation 部分中该命令的声明原型。该部分列出了该命令所支持的所有键及其默认值，有助于进一步理解和正确使用命令。下面以具体命令 \Polygon 为例，说明如何使用键值对接口：

```
% key-value setup
\keys_define:nn { ztikz / polygon }
  {
    radius       .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 = { white },
    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 = { },
  }
% command
\NewDocumentCommand\Polygon{ O{}m }
  {
    \group_begin:
    \keys_set:nn { ztikz / polygon } { #1 }
    ...
    \group_end:
  }
```

上述 \Polygon 命令解读：第一个参数为可选参数 (O 类型)，通过键值对进行指定. 可用的键有：⟨radius⟩, ⟨edgeColor⟩, ⟨fillColor⟩, ⟨fillOpacity⟩, ⟨rotate⟩, ⟨shift⟩, ⟨marker⟩ 等. 键 ⟨radius⟩ 接受一个浮点数 (参考后面的:“\fp_set:N”), 默认值为 1(参考后面的:“.initial:n = { 1 }”); 再比如, 键 ⟨edgeColor⟩ 可接受一个 tokenlist(参考后面的:“\tl_set:N”), 默认值为 “black” (参考后面的:“.initial:n = { black }”).

6.8.5 使用案例

最后附上一些复杂的目录格式定制示例, 涵盖目录样式设置以及目录数据提取, 可作为用户深度定制的参考. 下面这段代码展示了后续样例会用到的一些命令和依赖宏包:

```
% \usepackage{tikz}
% \usepackage{tikz-qtree}
% \usepackage[log]{tikz-qtree-patch}
% \usetikzlibrary { mindmap, trees }
\definecolor{SeaGreen}{rgb}{0.18, 0.55, 0.34}
\def\customtocthemecolor{SeaGreen}
\ExplSyntaxOn\makeatletter
\dim_new:N \linewidthfix
\def\tocupdatelinewidthfix{
  \dim_set:Nn \linewidthfix {\linewidth-2\fboxsep}
}
\def\ornamentcorner#1#2#3{
  \node[anchor=#1, rotate=#3, outer~sep=0pt, inner~sep=0pt] at (frame.#2)
  {\pgfornamenthan[width=.8cm, color=\customtocthemecolor]{10}};}
\def\ztochyperpage#1{\exp_args:Nne \hyper@link{link}{page.#1}{#1}}
\makeatother\ExplSyntaxOff
```

测试数据 本手册在排版目录时用到的测试数据: “data.toc”, “data_II.toc”, “data.ptoc”. 后者仅用于向读者展示 sect 模块中 keyval seq 的存储结构, 在 “dump-ptable = true” 时, ZTEX 会自动生成该文件.

data.toc

```
\contentsline {section}{{1}{AAA-1}}{1}{}
\contentsline {subsection}{{1.1}{BBB-1}}{1}{}
\contentsline {subsubsection}{{1.1.1}{CCC-1}}{1}{}
\contentsline {subsubsection}{{1.1.2}{CCC-2}}{1}{}
\contentsline {subsubsection}{{1.1.3}{CCC-3}}{1}{}
\contentsline {subsubsection}{{1.1.4}{CCC-4}}{1}{}
\contentsline {subsubsection}{{1.1.5}{CCC-5}}{1}{}
\contentsline {subsection}{{1.2}{BBB-2}}{1}{}
\contentsline {subsubsection}{{1.2.1}{CCC-6}}{1}{}
```

```
\contentsline {subsubsection}{{1.2.2}{CCC-7}}{1}{}
\contentsline {subsubsection}{{1.2.3}{CCC-8}}{1}{}
\contentsline {subsubsection}{{1.2.4}{CCC-9}}{1}{}
\contentsline {subsection}{{1.3}{BBB-3}}{1}{}
```

data_II.toc

```
\contentsline{section}{{1}{AAA-1}}{1}{}%
\contentsline{subsection}{{1}{BBB-1}}{1}{}%
\contentsline{subsubsection}{{1.1.1}{CCC-1}}{1}{}%
\contentsline{subsubsection}{{1.1.2}{CCC-2}}{2}{}%
\contentsline{subsubsection}{{1.1.3}{CCxxxC-3}}{3}{}%
\contentsline{subsubsection}{{1.1.4}{CCssgda\-shgfeC-4}}{100}{}%
\contentsline{subsubsection}{{1.1.5}{CsasfCC-5}}{123}{}%
\contentsline{subsubsection}{{1.1.6}{CasklklCC-x}}{999}{}%
\contentsline{subsection}{{2}{B   BB-2}}{1}{}%
\contentsline{subsubsection}{{1.2.1}{CCC-6}}{1}{}%
```

data.ptoc

```
class={section},name={1},title={AAA-1},page={1},anchor={section.1},raw={
\contentsline {section}{{1}{AAA-1}}{1}{}}
class={subsection},name={1.1},title={BBB-1},page={1},anchor={subsection.1.1},
raw={\contentsline {subsection}{{1.1}{BBB-1}}{1}{}}
class={subsubsection},name={1.1.1},title={CCC-1},page={1},
anchor={subsubsection.1.1.1},raw={\contentsline
{subsubsection}{{1.1.1}{CCC-1}}{1}{}}
class={subsubsection},name={1.1.2},title={CCC-2},page={1},
anchor={subsubsection.1.1.2},raw={\contentsline
{subsubsection}{{1.1.2}{CCC-2}}{1}{}}
class={subsubsection},name={1.1.3},title={CCC-3},page={1},
anchor={subsubsection.1.1.3},raw={\contentsline
{subsubsection}{{1.1.3}{CCC-3}}{1}{}}
class={subsubsection},name={1.1.4},title={CCC-4},page={1},
anchor={subsubsection.1.1.4},raw={\contentsline
{subsubsection}{{1.1.4}{CCC-4}}{1}{}}
```

```
class={subsubsection},name={1.1.5},title={CCC-5},page={1},
anchor={subsubsection.1.1.5},raw={\contentsline
{subsubsection}{{1.1.5}{CCC-5}}{1}{}}
class={subsection},name={1.2},title={BBB-2},page={1},anchor={subsection.1.2},
raw={\contentsline {subsection}{{1.2}{BBB-2}}{1}{}}
class={subsubsection},name={1.2.1},title={CCC-6},page={1},
anchor={subsubsection.1.2.1},raw={\contentsline
{subsubsection}{{1.2.1}{CCC-6}}{1}{}}
class={subsubsection},name={1.2.2},title={CCC-7},page={1},
anchor={subsubsection.1.2.2},raw={\contentsline
{subsubsection}{{1.2.2}{CCC-7}}{1}{}}
class={subsubsection},name={1.2.3},title={CCC-8},page={1},
anchor={subsubsection.1.2.3},raw={\contentsline
{subsubsection}{{1.2.3}{CCC-8}}{1}{}}
class={subsubsection},name={1.2.4},title={CCC-9},page={1},
anchor={subsubsection.1.2.4},raw={\contentsline
{subsubsection}{{1.2.4}{CCC-9}}{1}{}}
class={subsection},name={1.3},title={BBB-3},page={1},anchor={subsection.1.3},
raw={\contentsline {subsection}{{1.3}{BBB-3}}{1}{}}
```

案例:1 一个很简单的案例:

例 77

```
\begingroup
\ztocformat\subsection
  {
    format+=\color{teal},
    leader.sep=1pt,
    leader.raise=2.5pt,
    page.width=10pt
  }
\ztocgroupinsert{subsection,1,begin}%
  {%
    \begin{framed}%
    \pgfornament[width = 2cm,color = teal]{67}%
    \qquad\rule[-5em]{.5pt}{10em}%
    \begin{minipage}{.75\linewidth}%
  }
\ztocgroupinsert{subsection,1,end}%
  {%
    \end{minipage}%
    \end{framed}%
  }
\zlocaltoc{subsection}{4}
\endgroup
```

案例:2 下面这个案例清晰地展示了目录中这些 hook 的位置关系, BG:⟨*int*⟩ 和 EG:⟨*int*⟩ 代表目录条目自身局部组的开始和结束位置 (这些局部组目前已被移除 – 2025-09-12).

例 78

```
\ExplSyntaxOn
\int_new:N \g__test_toc_group_int
\AddToHook{ztoc/tocline/begin}
  {
    \fbox{BG:\int_use:N \g__test_toc_group_int}
  }
\AddToHook{ztoc/tocline/end}
  {
    \fbox{EG:\int_use:N \g__test_toc_group_int}
    \int_gincr:N \g__test_toc_group_int
  }
\cs_new:Npn \__debug_ztoc_hook:nn #1#2
  { \ztocgroupinsert{#1}{\fbox{\color{#2}\sffamily #1}} }

% begin/end hooks:
\__debug_ztoc_hook:nn {subsection,1,begin}{red}
\__debug_ztoc_hook:nn {subsection,1,end}{red}
\__debug_ztoc_hook:nn {subsubsection,5,begin}{gray}
\__debug_ztoc_hook:nn {subsubsection,5,end}{gray}
% new before/after hooks:
\__debug_ztoc_hook:nn {subsection,1,before}{blue}
\__debug_ztoc_hook:nn {subsection,1,after}{blue}
\__debug_ztoc_hook:nn {subsubsection,1,before}{teal}
\__debug_ztoc_hook:nn {subsubsection,1,after}{teal}
\__debug_ztoc_hook:nn {subsubsection,3,before}{purple}
\__debug_ztoc_hook:nn {subsubsection,3,after}{purple}
\ExplSyntaxOff
\zlocaltoc{subsection}{4}
```

subsection,1,before BG:0

EG:0 subsection,1,begin subsubsection,1,before BG:1

EG:1 subsubsection,1,after BG:2

EG:2 subsubsection,3,before BG:3

EG:3 subsubsection,3,after BG:4

EG:4 BG:5

EG:5 subsubsection,5,begin subsubsection,5,end BG:6

EG:6 subsection,1,end subsection,1,after

案例:3 这个案例相较之上一个案例更加复杂:

例 79

```
\begingroup\makeatletter
\tocupdatelinewidthfix
\def\customtocthemecolor{teal}
\ztocformat\subsection
  {
    explicit=true,
    code={%
      \noindent\fcolorbox{white}{\customtocthemecolor}{\hb@xt@\linewidthfix
        {\bfseries\large\color{white}\ztocname
          \hskip.75em\relax\ztoctitle\hfill\ztocpage}}%
    }
  }
\ztocformat\subsubsection{format+=\color{\customtocthemecolor}, name.before=
\S\;}
\ztocgroupinsert{subsection,1,begin}%
  {%
    \begin{tcolorbox}[
      enhanced, sharp corners,
      boxrule=.5pt, colback=white,
      left=20pt, colframe=\customtocthemecolor,
      overlay={
        \ornamentcorner{north west}{north west}{0}
        \ornamentcorner{north west}{south west}{90}
        \ornamentcorner{north west}{south east}{180}
        \ornamentcorner{north west}{north east}{270}
      }
    ]
    \pgfornament[width = 2.5cm,color = \customtocthemecolor]{8}%
    \kern-6pt\relax\begin{minipage}{.75\linewidth}%
  }
\ztocgroupinsert{subsection,1,end}%
```

```
  {%
    \end{minipage}%
    \end{tcolorbox}%
  }
\zlocaltoc{subsection}{4}
\makeatother\endgroup
```

6.1 font 模块 13

案例:4 我们可以使用 \ztocgroupinsert 命令复刻 titletoc 宏包所提供的样式:

例 80

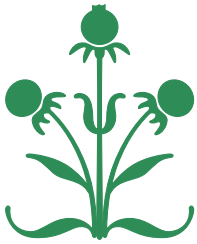
```
\makeatletter
\def\customssstyle#1{%
  \ztocgroupinsert{subsection,#1,begin}
    {\raggedright\hangindent2.3em\relax\noindent\hskip2.3em\relax 【}
  \ztocgroupinsert{subsection,#1,end}{】\par}
}
\makeatother
\customssstyle{1} \customssstyle{2}
\customssstyle{3} \customssstyle{4}
{
  \ztocformat\subsubsection{
    explicit=true,
    code={%
      \textit{\ztocname\enskip\ztoctitle\enskip
      (\ztochyperpage{\ztocpage})\ztociflast{}{;}\enskip}}
  }
  \ztocformat\subsection{
    explicit=true,
    code = {%
      \S\;\ztocname\enskip\ztoctitle
      \hskip1em\ztochyperpage{\ztocpage}\par}
  }
  \begin{multicols}{2}
    \zlocaltoc{subsection}{4, 11, 8, 6}
  \end{multicols}
}
```

案例: 5 这个案例相较之上一个案例更加复杂:

例 81

```
\begingroup\makeatletter
% \setcounter{tocdepth}{3}
\def\customtocthemecolor{SeaGreen}
\ztocformat\section
  {
    explicit=true,
    code={%
      \noindent\hskip1.5em\fcolorbox{white}{\customtocthemecolor}{\hb@xt@
        \dimexpr\linewidth-2\fboxsep-1.5em\relax
        {\bfseries\large\color{white}第\zhnumber{\ztocname}节
          \hskip.75em\relax\ztoctitle\hfill\ztocpage}}%
    }
  }
\ztocformat\subsection{
  name.before=\S\;, name.width=2.7em,
  format+=\color{\customtocthemecolor},
  leader.sep=2pt, page.width=1em,
  hyper.title=true,
  leader.content=\textcolor{\customtocthemecolor}{.},
}
\ztocgroupinsert{section,1,before}%
  {%
    \begin{tcolorbox}[
      enhanced, sharp corners,
      boxrule=.5pt, colback=white,
      left=20pt, colframe=\customtocthemecolor,
      overlay={
        \ornamentcorner{north west}{north west}{0}
        \ornamentcorner{north west}{south west}{90}
        \ornamentcorner{north west}{south east}{180}
        \ornamentcorner{north west}{north east}{270}
```

```
      }
    ]
    \pgfornament[width = 2.5cm,color = \customtocthemecolor]{10}%
    \begin{minipage}{.77\linewidth}%
  }
\ztocgroupinsert{section,1,begin}{%
  % \setlength{\columnsep}{0em}
  \begin{multicols}{2}}
\ztocgroupinsert{section,1,after}%
  {%
    \end{multicols}%
    \end{minipage}%
    \end{tcolorbox}%
  }
\zlocaltoc{section}{6}
\makeatother\endgroup
```



第六节 ZTEX 模块 12

案例: 6 这个案例复刻了 etoc 中的目录样式 (参见 etoc 宏包第 39 节):

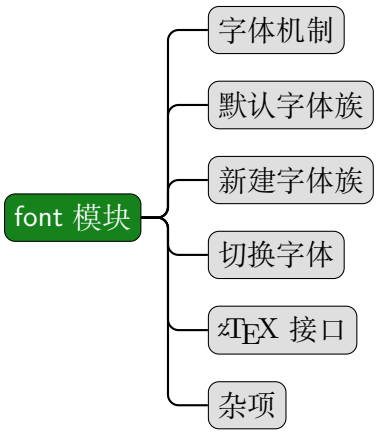
```
\ztocgroupinsert{section,1,before}
  {\begin{longtable}{|l|l|r|}}
\ztocgroupinsert{section,4,after}
  {\\ \hline\end{longtable}}
{
  \ztocformat\section{
    explicit=true,
    code = {%
      \ztociffirst{\kill\hline\multicolumn{3}{|c|}{\large\bfseries Table of Contents}\\\hline}{\\\hline}%
      \multicolumn{3}{||c||}{\bfseries 第\;\ztocname\;节\enskip\ztoctitle}%
    }
  }
  \ztocformat\subsection{
    explicit=true,
    code ={\\\hline
      \textbf{\S\ztocname} & \ztocname\enskip\ztoctitle & \ztochyperpage{\ztocpage}
    }
  }
  \ztocformat\subsubsection{
    explicit=true,
    code = {\\
      & \ztocname\enskip\ztoctitle & \ztochyperpage{\ztocpage}
    }
  }
  \zlocaltoc{section}{1, 2, 7, 9}
}
```

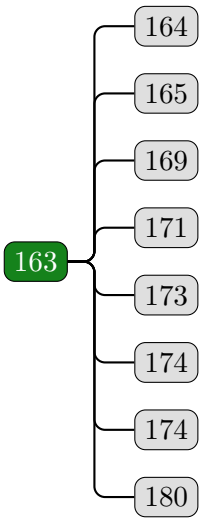
Table of Contents
第 1 节 基本介绍

案例: 7 这个案例也是受到 etoc 宏包的启发 – 将目录展示为思维导图的形式:

```
\definecolor{Green}{HTML}{15821b}
\tikzset{
  rootKey/.style={draw, rectangle, rounded corners, fill=Green, text=white},
}
\ExplSyntaxOn
\cs_new:Npn \__toc_tree:nn #1#2
  {
    \quark_if_no_value:nF {#1}
      {
        \Tree[.\node[rootKey]{#1};~#2 ]
      }
  }
\cs_generate_variant:Nn \__toc_tree:nn {oe}
\def\wraper#1{[.{\exp_not:n {#1}}~]}
\cs_new:Npn \ztoc_typeset_toctree:nnn #1#2#3
  {
    \ztoc_table_filter_key_byclass:nnnNN {#1}{#2}{#3}
      \g_ztoc_keyvaltoc_seq \g_tmpb_seq
    \seq_gpop_left:NN \g_tmpb_seq \secRoot
    \__toc_tree:oe { \secRoot }
      { \seq_map_function:NN \g_tmpb_seq \wraper }
  }
\def\TocTree#1#2#3
  {
    \exp_args:Nne \ztoc_typeset_toctree:nnn
      {#1}{#2}{#3}
  }
\ExplSyntaxOff

\noindent\begin{tikzpicture}[
    grow'=right,
    level distance=50pt,
```

```
    sibling distance=10pt,
    level 1/.style={level distance=25pt},
    every tree node/.style={anchor=west, draw, rectangle, rounded corners, fill=gray!25},
    every level 0 node/.style={anchor=east},
    edge from parent/.style={
      draw, thick, rounded corners,
      edge from parent path={
        (\tikzparentnode.east) -- +(10pt, 0pt)
        |- (\tikzchildnode.west)
      }
    },
  ]
  \TocTree{subsection}{4}{title}
  \begin{scope}[xshift=12cm]
    \TocTree{subsection}
      {\ztocindex{subsection}{\pkg{alias} 库}}
      {page}
  \end{scope}
\end{tikzpicture}
```



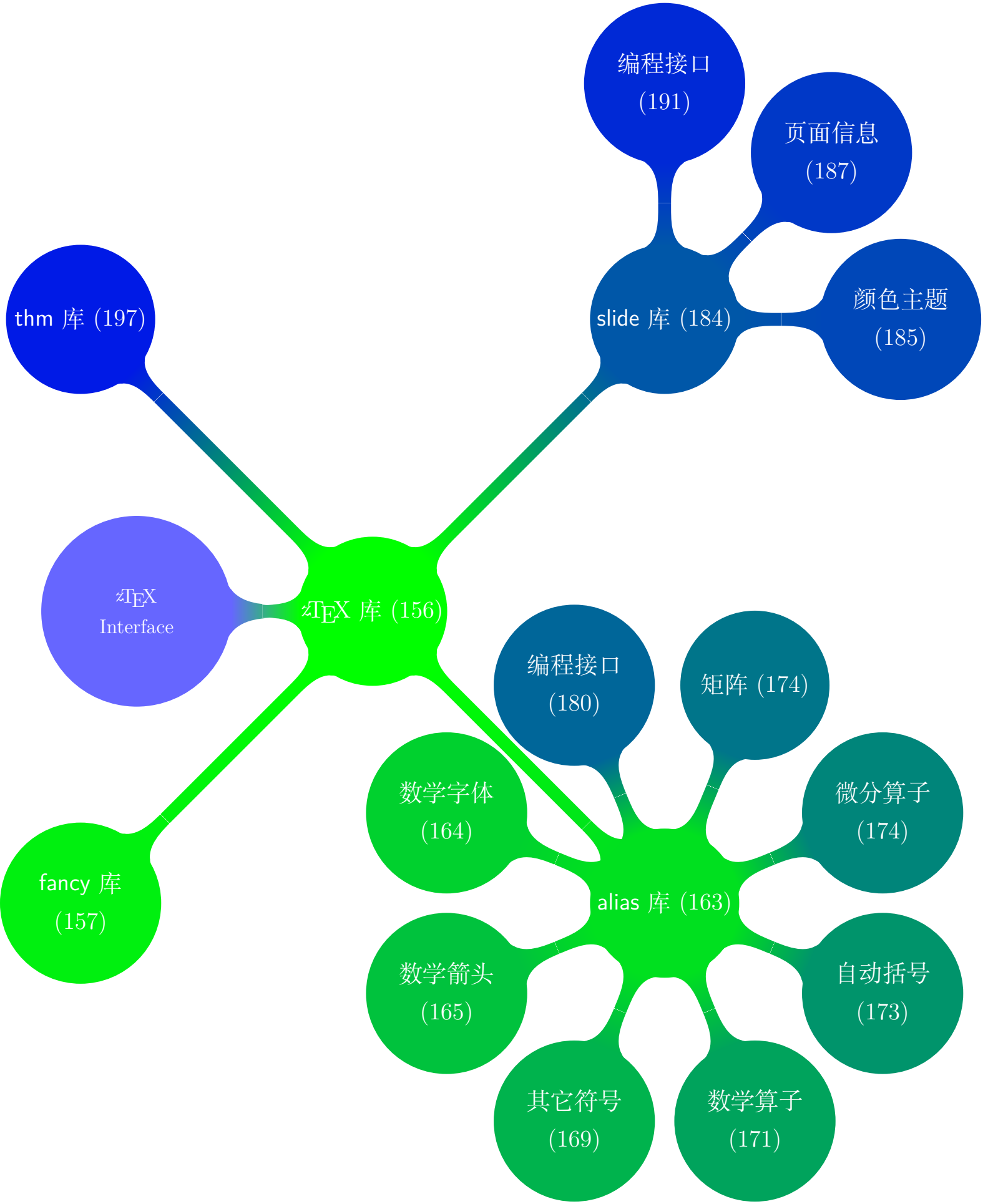


案例: 8 这个案列相较之上一个案例更加复杂, 它展示了 `\ztoc_group_parser:NNNnnnnn` 命令中 ⟨*step code*⟩ 的使用方法:

```
% \usetikzlibrary { mindmap, trees }
\ExplSyntaxOn
\int_new:N \g__mindmap_color_int
\seq_new:N \g__mindmap_data_seq
\seq_new:N \g__mindmap_keyvaldata_seq
\def\toclinecmd#1#2#3#4
  {
    % override hyperlink color.
    {\ztexlink{#1.#2}{\textcolor{white}{#3(#4)}}}
  }
\cs_new:Npn \__ztoc_gen_mindmap_data:nn #1#2
  {
    \ztoc_table_filter_byclass:nnNN
      { #1 }{ #2 }
      \g_ztoc_keyvaltoc_seq
      \g__mindmap_data_seq
    \ztoc_generate_keyvaltable_seq:NN
      \g__mindmap_data_seq
      \g__mindmap_keyvaldata_seq
    \ztoc_group_parser:NNNnnnnn \g__mindmap_keyvaldata_seq
      \toclinecmd \g_tmpa_tl
        {
          child [ concept~color = blue!
            \int_eval:n{\g__mindmap_color_int*6}!green
          ] \bgroup node[concept]
        }
      {}{}{\egroup}{\int_gincr:N \g__mindmap_color_int}
    % replace '\bgroup' and '\egroup' by '{' and '}'.
    \exp_args:NNo \str_set:Nn \l_tmpa_str { \g_tmpa_tl }
    \exp_args:NNne \str_replace_all:Nnn \l_tmpa_str
      { \bgroup }{ \char_generate:nn {123}{12} }
```

```
    \exp_args:NNne \str_replace_all:Nnn \l_tmpa_str
      { \egroup }{ \char_generate:nn {125}{12} }
    \tl_gset_rescan:Nno \g_tmpa_tl
      { \cctab_select:N \c_document_cctab }
      { \l_tmpa_str }
  }
\cs_new:Npn \__tkz_mind_map:n #1
  {
    \tl_if_empty:nF {#1}
      {
        \node[concept]
          {\color{white}\ztex{}~Interface}
          #1;
      }
  }
\newcommand{\mindmap}[2]
  {
    \__ztoc_gen_mindmap_data:nn
      { #1 }{ #2 }
    \exp_args:No \__tkz_mind_map:n
      { \g_tmpa_tl }
  }
\ExplSyntaxOff
\begin{tikzpicture}[
    mindmap, grow cyclic,
    text width=2cm, align=flush center,
    nodes={concept}, concept color=blue!60,
    root concept/.append style={
      text width=4cm, font=\Large
    },
    level 1 concept/.append style={
      font=\normalsize, color=white
    },
```

```
65      level 1/.append style={
66        level distance=5cm, sibling angle=45,
67        text width=3cm, font=\Large
68      },
69      level 2/.append style={
70        level distance=9cm, sibling angle=90,
71        text width=3cm, font=\Large
72      },
73      level 3/.append style={
74        level distance=5cm, sibling angle=45,
75        text width=3cm, font=\Large
76      },
77    ]
78    \mindmap{section}{7}
79  \end{tikzpicture}
```


编程接口
(191)
页面信息
(187)
颜色主题
(185)
slide 库 (184)
thm 库 (197)
zTEX
Interface
zTEX 库 (156)
fancy 库
(157)
编程接口
(180)
矩阵 (174)
数学字体
(164)
微分算子
(174)
alias 库 (163)
数学箭头
(165)
自动括号
(173)
其它符号
(169)
数学算子
(171)

案例: 9 下面这个案例展示了 \ztociffirst, \ztociflast 以及 \ztocdepth 等命令的使用方法:

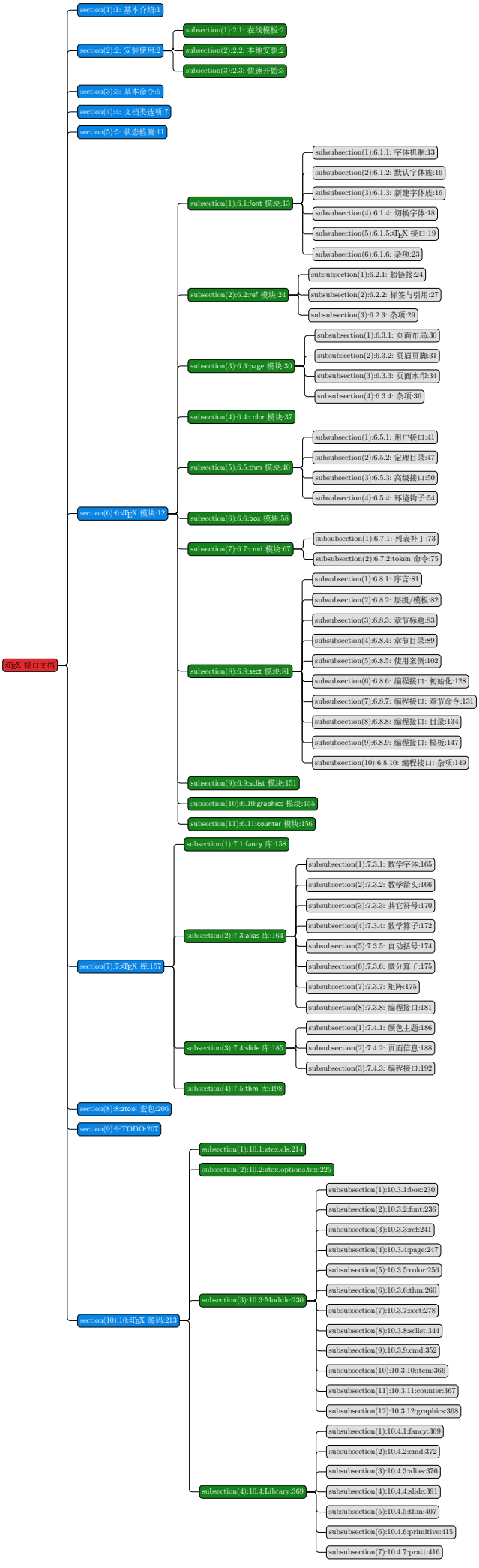
```
\newcommand{\nbraket}
  { \makebox[0pt]{\rule[-7.5pt]{.75pt}{19pt}} }
\newcommand{\ubraket}
  {
    \makebox[0pt]{
      \rule[-7pt]{.75pt}{10pt}
    }
    \makebox[0pt][l]{
      \kern-.375pt\raise3pt\hbox{\rule{3pt}{.75pt}}
    }
  }
\newcommand{\bbraket}
  {
    \makebox[0pt]{ \rule[3pt]{.75pt}{10pt} }
    \makebox[0pt][l]
      { \kern-.375pt\raise3pt
      \hbox{\rule{3pt}{.75pt}} }
  }
\newcommand{\showbar}
  {
    \prg_replicate:nn {\ztocdepth - 2}
      { \kern20pt \nbraket }
    \kern20pt \ztociffirst { \ubraket }
      { \ztociflast{\bbraket}{\nbraket} }
  }
\ztocformat{\section, \subsection, \subsubsection}
  {
    explicit=true,
    code = {
      \noindent
      \prg_replicate:nn {\ztocdepth-1}
```

```
        {
          \hskip 20pt\relax
        }
      \llap{\smash{\showbar}}
      \kern10pt
      \ztoctitle\par
    }
  }
\ExplSyntaxOff
{
  \ztocformat\subsection{ignore.name={10.}}
  \ztocformat\subsubsection{ignore.name={10.}}
  \multicolcontents{2}
}
```

案例: 10 下面这个案例相较之上一个案例更加复杂, 但它展示了 \ztoc_group_parser:NNNnnnnn 命令的强大和普适性:

```
% \usepackage{tikz}
% \usepackage{tikz-qtree}
% \usepackage{tikz-qtree-patch}
\definecolor{Red}{HTML}{e32b2d}
\definecolor{Green}{HTML}{15821b}
\definecolor{Blue}{HTML}{0984e3}
\tikzset{
  section/.style={draw, rectangle, rounded corners, fill=Red},
  subsection/.style={draw, rectangle, rounded corners, fill=Blue, text=white},
  subsubsection/.style={draw, rectangle, rounded corners, fill=Green, text=white},
}
\ExplSyntaxOn
% \def\toclinecmd#1#2#3#4{{#1:#2:#3:#4}~}
\def\toclinecmd#1#2#3#4{{\gparserclass(\gparserindex):#2:#3:#4}~}
\ztoc_group_parser:NNNnnnnn \g_ztoc_keyvaltoc_seq
  \toclinecmd \g_tmpa_tl {[.}{}{}{~]}{}
\def\TocTree
  {
    \exp_args:Nno \__toc_tree:nn
      {\ztex{}~接口文档}
      {\g_tmpa_tl}
  }
\cs_new:Npn \__toc_tree:nn #1#2
  {
    \quark_if_no_value:nF {#1}
      {
        \Tree[.\node[section]{#1};~#2 ]
      }
  }
\ExplSyntaxOff
```

```
31
32 \thispagestyle{empty}
33 \begin{tikzpicture}[
34     grow'=right, level distance=1cm,
35     sibling distance=10pt,
36     every tree node/.style = {
37       anchor=west, outer sep=0pt, draw, rectangle,
38       rounded corners, fill=gray!25
39     },
40     every level 1 node/.style = {
41       subsection
42     },
43     every level 2 node/.style = {
44       subsubsection
45     },
46     edge from parent/.style={
47       draw, thick, rounded corners,
48       edge from parent path={
49         (\tikzparentnode.east) -- +(.5cm, 0pt)
50         |- (\tikzchildnode.west)
51       },
52     },
53   ]
54   \TocTree
55 \end{tikzpicture}
```


LaTeX 接口文档
section(1):1: 基本介绍:1
section(2):2: 安装使用:2
subsection(1):2.1: 在线模板:2
subsection(2):2.2: 本地安装:2
subsection(3):2.3: 快速开始:3
section(3):3: 基本命令:5
section(4):4: 文档类选项:7
section(5):5: 状态检测:11
section(6):6:LaTeX 模块:13
subsection(1):6.1:font 模块:13
subsubsection(1):6.1.1: 字体机制:13
subsubsection(2):6.1.2: 默认字体族:16
subsubsection(3):6.1.3: 新建字体族:16
subsubsection(4):6.1.4: 切换字体:18
subsubsection(5):6.1.5:LaTeX 接口:19
subsubsection(6):6.1.6: 杂项:23
subsection(2):6.2:ref 模块:24
subsubsection(1):6.2.1: 超链接:24
subsubsection(2):6.2.2: 标签与引用:27
subsubsection(3):6.2.3: 杂项:29
subsection(3):6.3:page 模块:30
subsubsection(1):6.3.1: 页面布局:30
subsubsection(2):6.3.2: 页眉页脚:31
subsubsection(3):6.3.3: 页面水印:34
subsubsection(4):6.3.4: 杂项:36
subsection(4):6.4:color 模块:37
subsection(5):6.5:thm 模块:40
subsubsection(1):6.5.1: 用户接口:41
subsubsection(2):6.5.2: 定理目录:47
subsubsection(3):6.5.3: 高级接口:50
subsubsection(4):6.5.4: 环境钩子:54
subsection(6):6.6:box 模块:58
subsection(7):6.7:cmd 模块:67
subsubsection(1):6.7.1: 列表补丁:73
subsubsection(2):6.7.2:token 命令:75
subsection(8):6.8:sect 模块:81
subsubsection(1):6.8.1: 序言:81
subsubsection(2):6.8.2: 层级/模板:82
subsubsection(3):6.8.3: 章节标题:83
subsubsection(4):6.8.4: 章节目录:89
subsubsection(5):6.8.5: 使用案例:102
subsubsection(6):6.8.6: 编程接口: 初始化:128
subsubsection(7):6.8.7: 编程接口: 章节命令:131
subsubsection(8):6.8.8: 编程接口: 目录:134
subsubsection(9):6.8.9: 编程接口: 模板:147
subsubsection(10):6.8.10: 编程接口: 杂项:149
subsection(9):6.9:sclist 模块:151
subsection(10):6.10:graphics 模块:155
subsection(11):6.11:counter 模块:156
section(7):7:LaTeX 库:157
subsection(1):7.1:fancy 库:158
subsection(2):7.3:alias 库:164
subsubsection(1):7.3.1: 数学字体:165
subsubsection(2):7.3.2: 数学箭头:166
subsubsection(3):7.3.3: 其它符号:170
subsubsection(4):7.3.4: 数学算子:172
subsubsection(5):7.3.5: 自动括号:174
subsubsection(6):7.3.6: 微分算子:175
subsubsection(7):7.3.7: 矩阵:175
subsubsection(8):7.3.8: 编程接口:181
subsection(3):7.4:slide 库:188
subsubsection(1):7.4.1: 颜色主题:186
subsubsection(2):7.4.2: 页面信息:188
subsubsection(3):7.4.3: 编程接口:192
subsection(4):7.5:thm 库:198
section(8):8:ztool 宏包:206
section(9):9:TODO:207
section(10):10:LaTeX 源码:213
subsection(1):10.1:ztex.cls:214
subsection(2):10.2:ztex.options.tex:225
subsection(3):10.3:Module:230
subsubsection(1):10.3.1:box:230
subsubsection(2):10.3.2:font:236
subsubsection(3):10.3.3:ref:241
subsubsection(4):10.3.4:page:247
subsubsection(5):10.3.5:color:256
subsubsection(6):10.3.6:thm:260
subsubsection(7):10.3.7:sect:278
subsubsection(8):10.3.8:sclist:344
subsubsection(9):10.3.9:cmd:352
subsubsection(10):10.3.10:item:366
subsubsection(11):10.3.11:counter:367
subsubsection(12):10.3.12:graphics:368
subsection(4):10.4:Library:369
subsubsection(1):10.4.1:fancy:369
subsubsection(2):10.4.2:cmd:372
subsubsection(3):10.4.3:alias:376
subsubsection(4):10.4.4:slide:391
subsubsection(5):10.4.5:thm:407
subsubsection(6):10.4.6:primitive:415
subsubsection(7):10.4.7:pratt:416

TikZ Examples

Eureka

2025 年 5 月 31 日

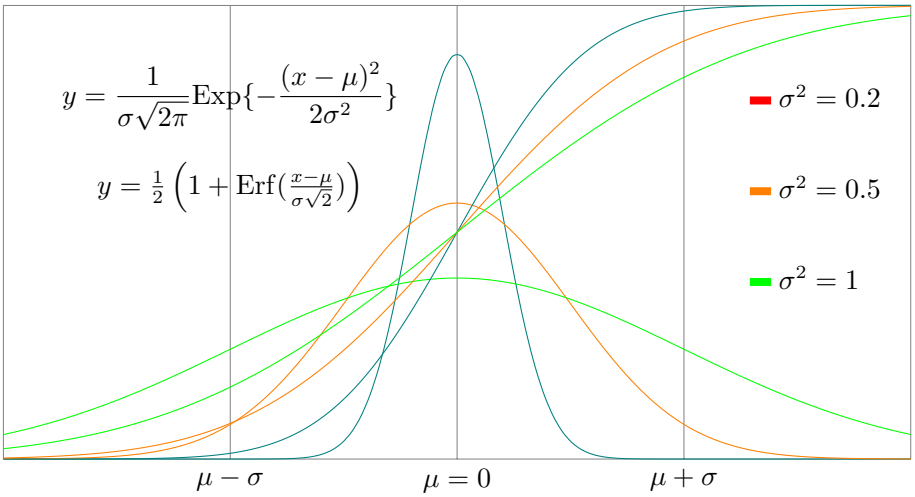
总目录

1 介绍

本文档展示了 TikZ 宏包中部分命令或环境的使用示例, 希望本文档可以帮助用户更好的掌握与使用 TikZ 宏集.

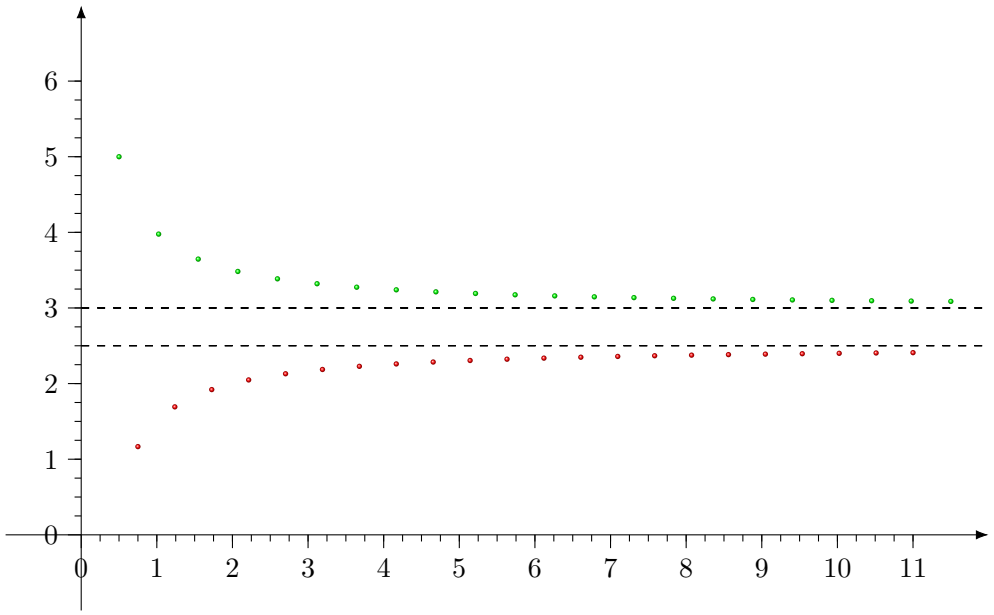
2 basic/gnuplot 库

2.1 案例 1



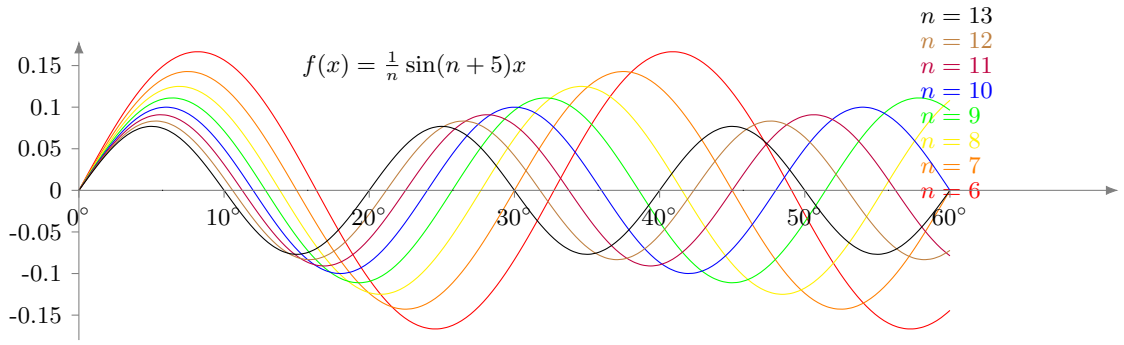

```
\begin{tikzpicture}[yscale=6, xscale=3]
  \ShowGrid{(-2,0); (2,1)}
  % pdf
  \Plot[domain=-2:2,style=teal]{1/(sqrt(0.2)*sqrt(2*pi))*exp(-(x-0)**2/(2*0.2**2))}
  \Plot[domain=-2:2,style=orange]{1/(sqrt(0.5)*sqrt(2*pi))*exp(-(x-0)**2/(2*0.5**2))}
  \Plot[domain=-2:2,style=green]{1/(sqrt(1)*sqrt(2*pi))*exp(-(x-0)**2/(2*1**2))}
  % cdf
  \Plot[domain=-2:2,style=teal]{0.5*(1+erf((x-0)/(sqrt(0.2)*sqrt(2))))}
  \Plot[domain=-2:2,style=orange]{0.5*(1+erf((x-0)/(sqrt(0.5)*sqrt(2))))}
  \Plot[domain=-2:2,style=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)}[
    \textcolor{red}{\rule[1pt]{8pt}{3pt}}\;$\sigma^2=0.2$;
    \textcolor{orange}{\rule[1pt]{8pt}{3pt}}\;$\sigma^2=0.5$;
    \textcolor{green}{\rule[1pt]{8pt}{3pt}}\;$\sigma^2=1$;
  ][right=2em]
  \ShowPoint[radius=0pt]
    {(-1, 0.8); (-1, 0.6)}
    [
      $\displaystyle y = \frac{1}{\sigma\sqrt{2\pi}}\mathrm{Exp}
        \{-\frac{(x-\mu)^2}{2\sigma^2}\}$;
      $y=\frac12\left(1+\mathrm{Erf}(\frac{x-\mu}{\sigma\sqrt{2}})\right)$
    ]
\end{tikzpicture}
```

2.2 案例 2



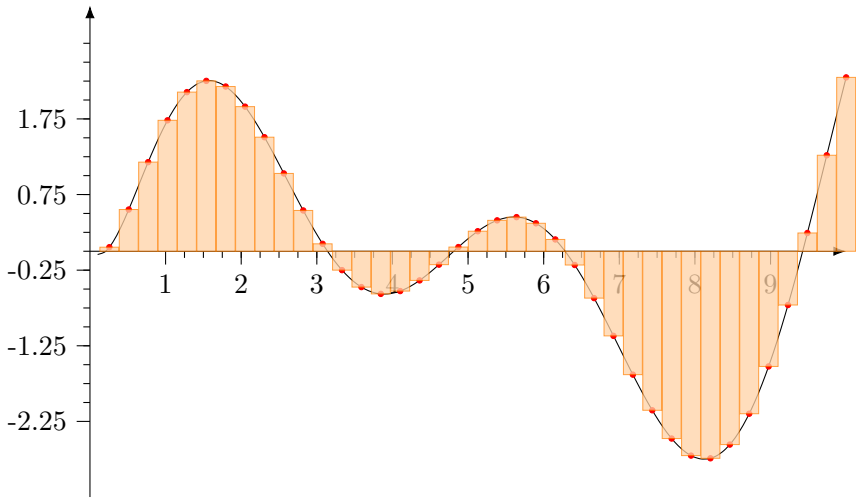

```
\begin{tikzpicture}[>=Latex]
  \xAxis[-1][12] \yAxis[-1][7]
  \PlotPrecise{plot}{22}
  \Plot[
    domain=0.75:11,
    style={red, thick, opacity=0},
    marker={type=ball, color=red}
  ]{2.5-1/x}
  \PlotPrecise{plot}{22}
  \Plot[
    domain=0.5:11.5,
    style={red, thick, opacity=0},
    marker={type=ball, color=green}
  ]{3+1/x}
  \PlotPrecise*{contour}{40}
  \ContourPlot[domain=0:12;, style={dashed}]{y-2.5}
  \ContourPlot[domain=0:12;, style={dashed}]{y-3}
\end{tikzpicture}
```

2.3 案例 3



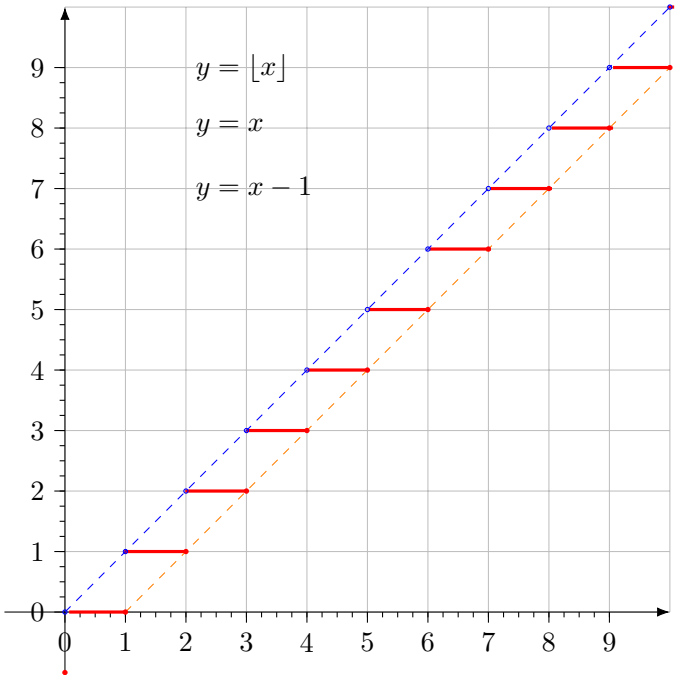

```
\ExplSyntaxOn
\clist_new:N \l__color_clist
\clist_set:Nn \l__color_clist {red, orange, yellow, green, blue, purple, brown, black}
\newcommand{\colorItem}[1]{\clist_item:Nn \l__color_clist {#1}}
\def\fptoint#1{\fp_to_int:n {#1}}
\ExplSyntaxOff
\begin{tikzpicture}[scale=11, >=Latex, font=\small]
  % 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[
      domain=0:pi/3,
      style=\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{pi/18},
    mainTickColor=gray, subTickLength=0pt,
    mainTickLength=.5pt,
    mainTickLabel={\fptoint{\CurrentFp/(pi/18)*10}$^\circ$}
  ]{(0, 0); (1.25, -0)}
\end{tikzpicture}
```

2.4 案例 4



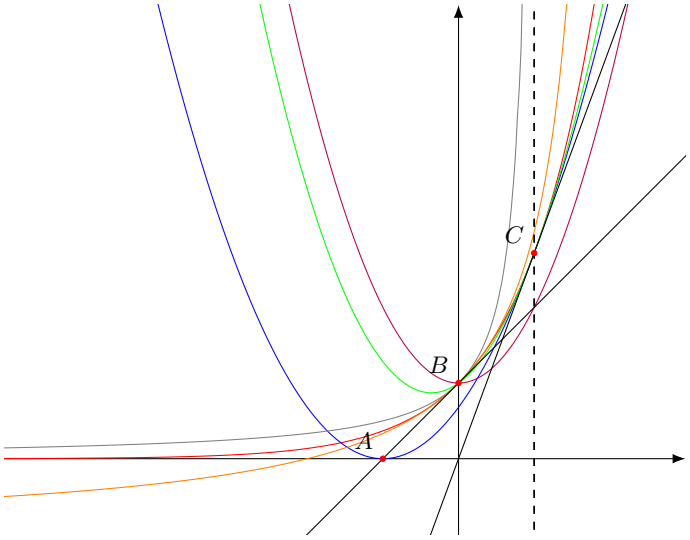

```
\begin{tikzpicture}[>=Latex]
  \xAxis[0][10] \yAxis[-3.25][3.25]
  \Plot[domain=0:10]{2*sqrt(x)*cos(log(x))*sin(x)}
  \PlotPrecise{plot}{40}
  \Plot[
    domain=0:10, style={opacity=0},
    marker={type=*, color=red}
  ]{2*sqrt(x)*cos(log(x))*sin(x)}
  \BarPlot[x][
    fill=orange!35!white,
    bar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

2.5 案例 5



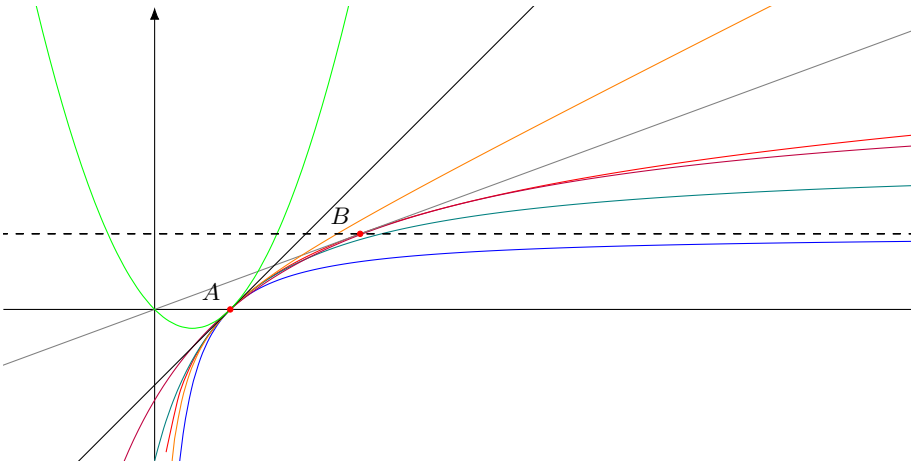

```
\begin{tikzpicture}[scale=.8, >=Latex]
  \ShowGrid[step=1, color=gray, opacity=.5]{(0, 0); (10, 10)}
  \xAxis[-1][10]  \yAxis[-1][10]
  \Plot[
    domain=0:10,
    style={red, jump mark right, very thick, xshift=2pt},
    marker={type=*, opacity=0}
  ]{floor(x)}
  \Plot[domain=0:10, style={dashed, blue}]{x}
  \Plot[domain=1:10, style={dashed, orange}]{x-1}
  \PlotPrecise{plot}{11}
  \Plot[
    domain=0:10,
    style={opacity=0, jump mark right},
    marker={type=o, color=blue}
  ]{x}
  \PlotPrecise{plot}{11}
  \Plot[
    domain=0:10,
    style={opacity=0, jump mark right},
    marker={type=*, color=red}
  ]{x-1}
  \ShowPoint[opacity=0]{(2, 9); (2, 8); (2, 7)}
    [$y=\lfloor x\rfloor$; $y=x$; $y=x-1$][right]
\end{tikzpicture}
```

2.6 案例 6



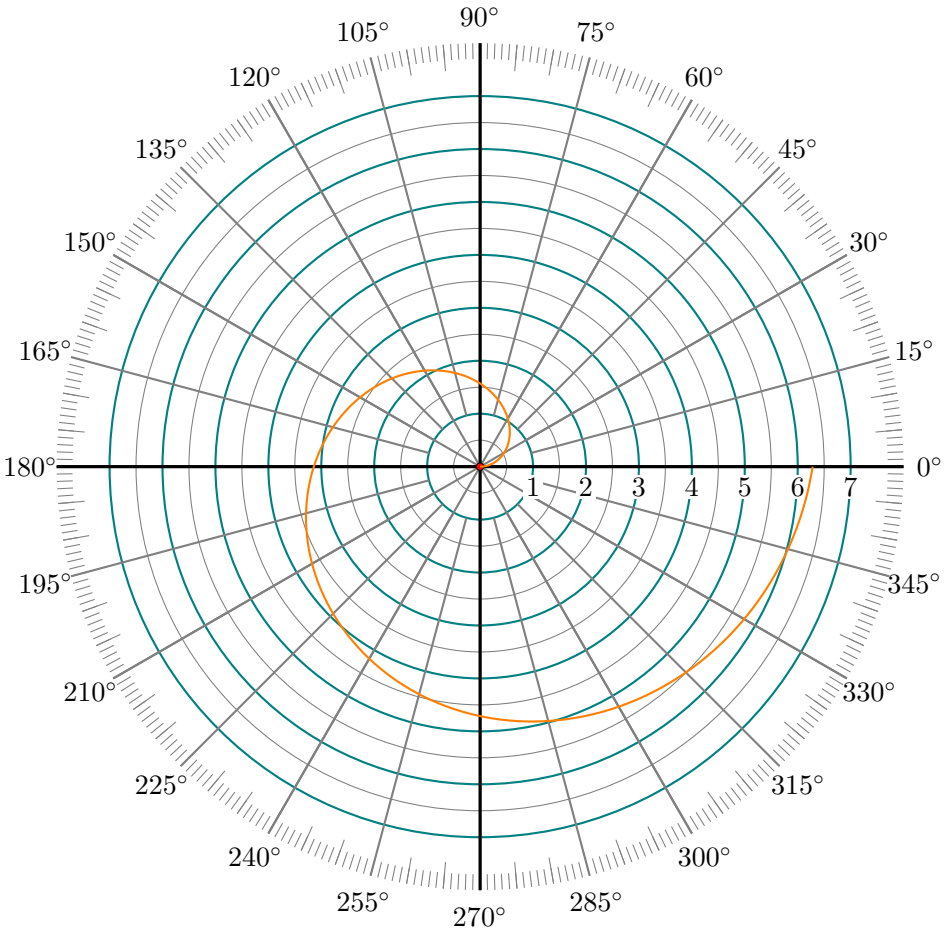

```
\begin{tikzpicture}[>=Latex, font=\small]
  \clip (-6, -1) rectangle (3, 6);
  \ShowAxis{(-8, 0); (3, 0)}  \ShowAxis{(0, -1.5); (0, 6)}
  \Plot[domain=-8:5,    style={red}]     {exp(x)}
  \Plot[domain=-8:5,    style={blue}]    {exp(1)/4*(x+1)**2}
  \Plot[domain=-8:5,    style={green}]   {exp(1)*x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

2.7 案例 7




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

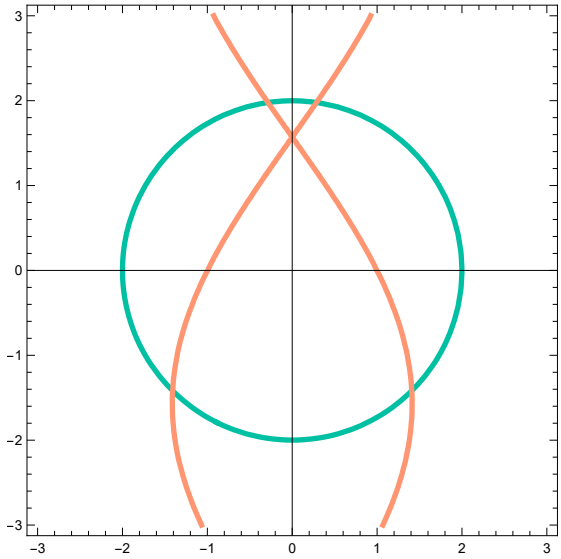
2.8 案例 8




```
% https://texample.net/tikz/examples/polar-coordinates-template/
\begin{tikzpicture}[scale=.7]
  \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);
  \foreach \a in {0, 1,...,359}   \draw[gray] (\a:7.7) -- (\a:8);
  \foreach \a in {0, 5,...,359}   \draw[gray] (\a:7.5) -- (\a:8);
  \foreach \a in {0, 15,...,359}  \draw[thick,gray] (\a:1) -- (\a:8);
  \foreach \a in {0, 30,...,359}  \draw[thick,gray] (0, 0) -- (\a:8);
  \foreach \r in {1, 2,...,7}
    \draw (\r,0) node[inner sep=1pt,below=3pt,rectangle,fill=white] {$\r$};
  \foreach \a in {0, 90,...,359}  \draw[very thick] (0, 0) -- (\a:8);
  \foreach \a in {0, 15,...,359}  \draw (\a: 8.5) node {$\a^\circ$};
  \draw[fill=red] (0,0) circle(0.7mm);
  \PolarPlot[domain=0:2*pi, style={thick, orange}]{t}
\end{tikzpicture}
```

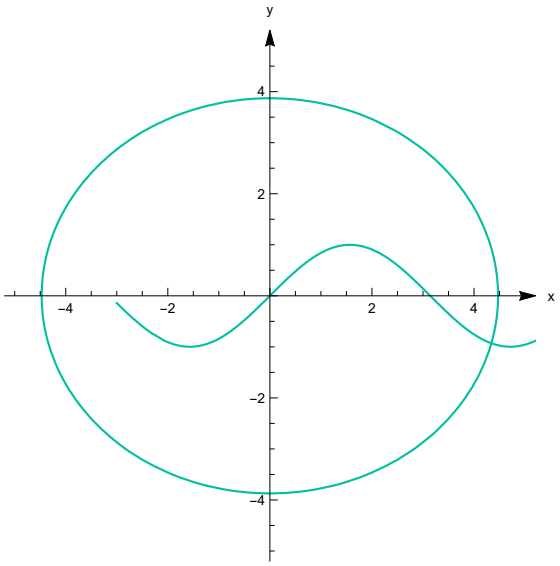
3 wolfram 库

3.1 案例 9



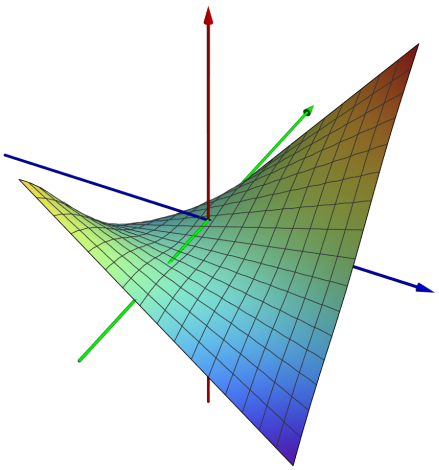

```
\begin{wolframGraphics}{wolframStroke}
fp1 = ContourPlot[
  x^2 + y^2 == 4, {x, -1.3, 0.6}, {y, -2.4, 3.2},
  AspectRatio->(2.4+3.2)/(1.3+0.6), ContourStyle->Red
];
fp2 = ContourPlot[
  x^2 + y^2 == 4, {x, -3, 3}, {y, -3, 3},
  AspectRatio->1, ContourStyle->RGBColor["#00C0A3"],
  AxesOrigin->{0, 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[0.01]},
    {RGBColor["#FF9671"], Thickness[0.01]}
  },
  AspectRatio->(3+3)/(2.5+2.5), AxesOrigin->{0,0},
  Axes->True, Frame->False,
  AxesStyle->Arrowheads[{0,0.01}]
]
FIGURE = Show[fp2, fp1, fp3];
\end{wolframGraphics}
\includegraphics[width=.5\linewidth]{\wolframOuputFile}
```

3.2 案例 10



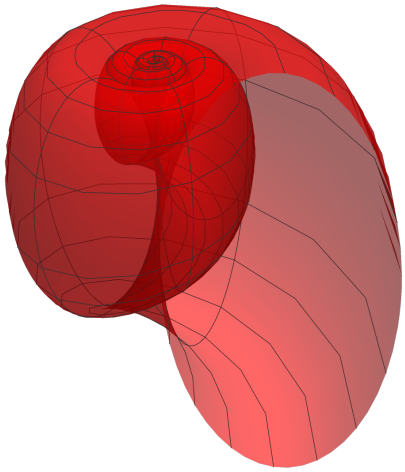

```
\begin{wolframGraphics}{wolfram2Dplot}
plotFunction[fun_, xlimits_, ylimits_] := ContourPlot[
  fun, xlimits, ylimits,
  ContourStyle->{
    RGBColor["#00C0A3"],
    Thickness[0.004]
  },
  AspectRatio->((xlimits[[2]]//Abs) + (xlimits[[3]]//Abs))
              /((ylimits[[2]]//Abs) + (ylimits[[3]]//Abs)),
  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];
\end{wolframGraphics}
\includegraphics[width=.5\linewidth]{\wolframOuputFile}
```

3.3 案例 11



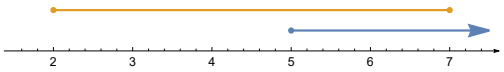
```
\begin{wolframGraphics}{wolfram3DAxis}
(* 1. 定义一个产生箭头的命令 *)
arrow[start_, end_, type_] := Graphics3D[
  { type,
    { Arrowheads[.02], Arrow[Tube[{start, end}, 0.06]]}
  }, Boxed->False
];
(* 2. 创建三个坐标轴的箭头，使用颜色进行区分 *)
xaxis = arrow[{-10, 0, 0}, {10, 0, 0}, Blue];
yaxis = arrow[{0, -10, 0}, {0, 10, 0}, Green];
zaxis = arrow[{0, 0, -10}, {0, 0, 10}, Red];
(* 3. 展示在同一坐标轴 *)
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]
\end{wolframGraphics}
\includegraphics[width=.5\linewidth]{\wolframOuputFile}
```

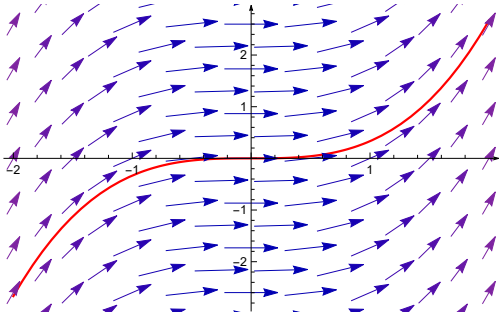
3.4 案例 12



```
\begin{wolframGraphics}{wolfram3DParametric}
FIGURE = ParametricPlot3D[
  {1.16^v*Cos[v]*(1+Cos[u]), -1.16^v*Sin[v]*(1+Cos[u]), -2 1.16^v*(1+Sin[u])},
  {u, 0, 2*Pi}, {v, -15, 6},
  PlotStyle->{Opacity[0.6],Red},
  PlotRange->All, PlotPoints->25,
  Axes->False, Boxed->False
];
\end{wolframGraphics}
\includegraphics[width=.4\linewidth]{\wolframOuputFile}
```

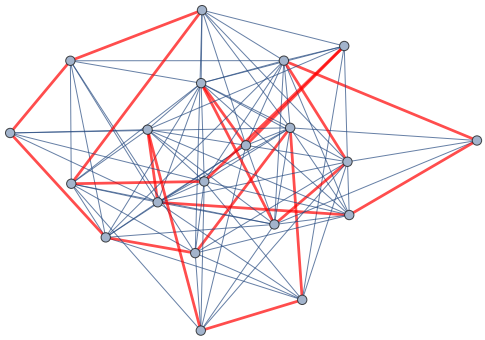
3.5 案例 13

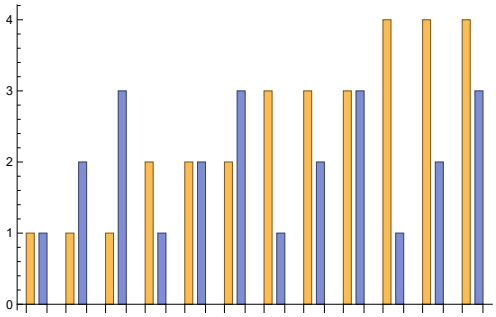





```
\begin{wolframGraphics}{wolframLine-I}
FIGURE = NumberLinePlot[
  { Interval[{5, Infinity}], Interval[{2, 7}] },
  AxesStyle->Arrowheads[{0, 0.01}]
];
\end{wolframGraphics}
\edef\mmaOutputTmp{\wolframOuputFile}

\begin{wolframGraphics}{wolframLine-II}
fvec = VectorPlot[
  {1, x^2}, {x, -4, 4}, {y, -4, 4},
  AxesOrigin->{0, 0}, Axes->False, Frame->False
];
fp = Plot[
  1/3*x^3, {x, -2, 2}, PlotStyle->Red,
  AxesStyle->Arrowheads[{0, 0.01}]
];
FIGURE = Show[fp, fvec];
\end{wolframGraphics}
\includegraphics[width=.45\linewidth]{\mmaOutputTmp}\qquad
\includegraphics[width=.45\linewidth]{\wolframOuputFile}
```

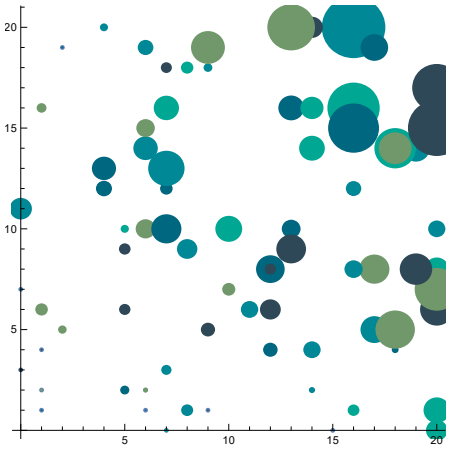
3.6 案例 14

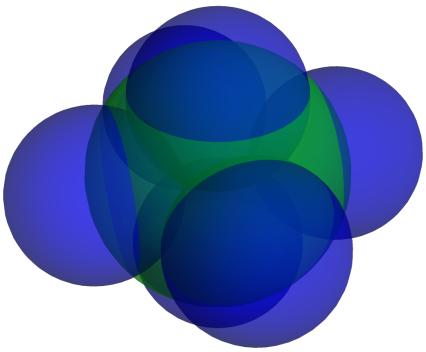





```
\begin{wolframGraphics}{wolframHamiltonian}
g = RandomGraph[{20, 100}];
h = FindHamiltonianCycle[g];
FIGURE = HighlightGraph[g, Style[h, Directive[Thick, Red]]];
\end{wolframGraphics}
\edef\mmaOutputTmp{\wolframOuputFile}

\begin{wolframGraphics}{wolframStatistic}
FIGURE = BarChart[Flatten[Table[{i, j}, {i, 1, 4}, {j, 1, 3}], 1]];
\end{wolframGraphics}
\includegraphics[width=.45\linewidth]{\mmaOutputTmp}\qquad
\includegraphics[width=.45\linewidth]{\wolframOuputFile}
```

3.7 案例 15

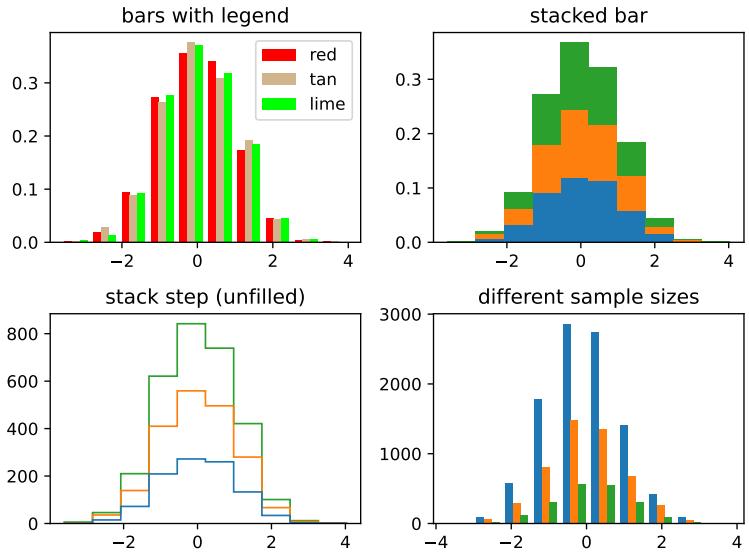




```
\begin{wolframGraphics}{wolfram2DBall}
xls = RandomInteger[{0, 20}, 80];
yls = RandomInteger[{0, 20}, 80];
xycoor = {xls, yls}//Transpose;
color = { RGBColor["#00A894"], RGBColor["#008896"], RGBColor["#006780"],
RGBColor["#2F4858"], RGBColor["#70986B"]};
fp1 = Table[
  Graphics[{ color[[RandomInteger[{1, 5}]]],
    Disk[xycoor[[i]], RandomReal[{0, 0.05}]*#1+RandomReal[{0,
    0.05}]*#2&[xycoor[[i]][[1]], xycoor[[i]][[2]]]]
  }], {i, 1, 80}
];
fp2 = ListPlot[xycoor, AspectRatio->(Max[yls])/(Max[xls])];
FIGURE = Show[fp2, fp1];
\end{wolframGraphics}
\edef\mmaOutputTmp{\wolframOuputFile}

\begin{wolframGraphics}{wolfram3DBall}
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
];
\end{wolframGraphics}
\includegraphics[width=.4\linewidth]{\mmaOutputTmp}\qquad
\includegraphics[width=.4\linewidth]{\wolframOuputFile}
```

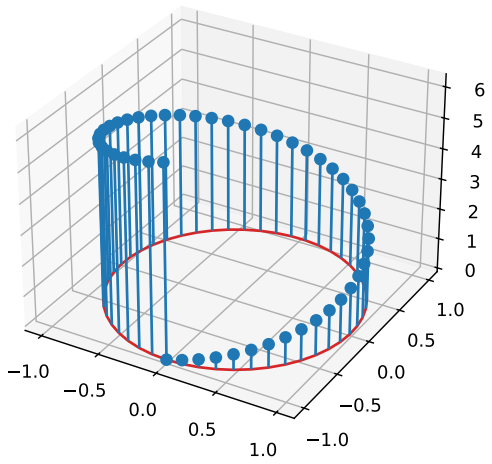
4 python 库

4.1 案例 16



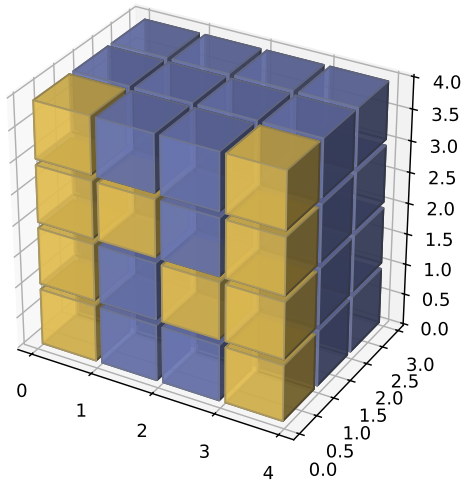

```
\begin{pyfig}{pyfigExampleA}{pyfig-A.pdf}
# https://matplotlib.org/stable/gallery/lines_bars_and_markers/histogram_demo.html
import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
import numpy as np

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, histtype='step', stacked=True, fill=False)
ax2.set_title('stack step (unfilled)')
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}
\includegraphics[width=.7\linewidth]{\pyfigOutputFile}
```

4.2 案例 17




```
\begin{pyfig}{pyfigExampleB}{pyfig-B.pdf}
# 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}
\includegraphics[width=.75\linewidth]{\pyfigOutputFile}
```

4.3 案例 18

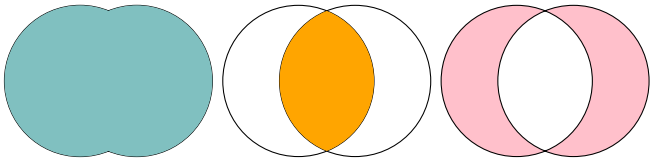



```
\begin{pyfig}{pyfigExampleC}{pyfig-C.pdf}
# https://matplotlib.org/stable/gallery/mplot3d/voxels_numpy_logo.html
import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
import numpy as np

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', '#7A88CCC0')
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}
\includegraphics[width=.75\linewidth]{\pyfigOutputFile}
```

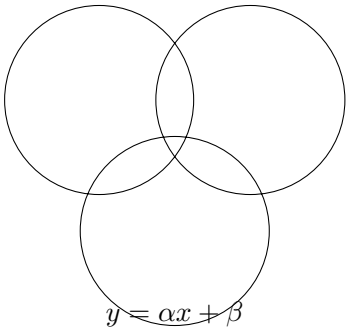
5 l3draw 库

5.1 案例 19



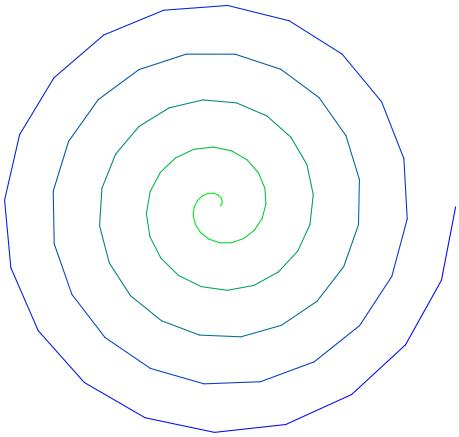
```
% union
\begin{Zdraw}
  \zxscale {0.5}  \zyscale {0.5}
  \zcirc {2cm, 0}{2cm}   \zcirc {3.5cm, 0}{2cm}
  \zusepath[draw, clip]  \zfcolor {teal!50}
  \zrect {-10cm, -10cm}{10cm, 10cm}
  \zusepath[fill]
\end{Zdraw}
% intersection
\begin{Zdraw}
  \zxscale {0.5} \zyscale {0.5}
  \zcirc {3.5cm, 0}{2cm} \zusepath[draw]
  \zcirc {2cm, 0}{2cm}   \zusepath[clip, draw]
  \zfcolor {orange}      \zcirc {3.5cm, 0}{2cm}
  \zusepath[fill]
\end{Zdraw}
% difference
\begin{Zdraw}
  \zxscale {0.5}       \zyscale {0.5}
  \zfevenodd           \zfcolor {pink}
  \zcirc {2cm, 0}{2cm} \zcirc {3.5cm, 0}{2cm}
  \zusepath[draw, fill]
\end{Zdraw}
```

5.2 案例 20




```
\begin{Zdraw}
  % draw circle
  \zxscale {.5} \zyscale {.5}
  \zcirc {-2cm, 0}{2.5cm}
  \zcirc {2cm, 0}{2.5cm}
  \zcirc {0, -2*sqrt(3)cm}{2.5cm}
  % add text
  \znewtext   \texta
  \zsetvtext  \texta {6em}{$y=\alpha x + \beta$\\ Hello~ world}
  \zscaletext \texta {2}{2}
  \zputtext   \texta {hc}{b}{0, -7cm}
  \zusepath[draw]
\end{Zdraw}
```

5.3 案例 21



```
\ExplSyntaxOn
% Data Source: https://tex.stackexchange.com/a/721052/294585
\ztool_read_file_as_seq:neN
  {\c_false_bool}{gradient.data}
  \l_tmpa_seq   % seq(without outer brace)={0, 0}, {0.03, 0.01}, ..., {3.14, 0}.
\cs_set:Npn \color_gradient:n #1
  { \color_select:n {blue!#1!green} }
\cs_generate_variant:Nn \color_gradient:n {e}

% Draw those segments
\draw_begin: \draw_cap_round:
\draw_xvec:n {1cm, 0}
\draw_yvec:n {0, 1cm}
\draw_path_moveto:n {\draw_point_vec:nn {0.785}{0}}
\int_step_inline:nnn {2}{\fp_eval:n {\seq_count:N \l_tmpa_seq-1}}
  {
    \seq_set_split:Nne \l_tmpb_seq {,}{\seq_item:Nn \l_tmpa_seq {#1}}
    \seq_set_split:Nne \l_tmpc_seq {,}{\seq_item:Nn \l_tmpa_seq {\fp_eval:n {#1+1}}}
    \color_gradient:e {\fp_eval:n {#1*100/\seq_count:N \l_tmpa_seq}}
    \draw_path_moveto:n {
      \draw_point_vec:nn {\seq_item:Nn \l_tmpb_seq {1}}
        {\seq_item:Nn \l_tmpb_seq {2}}
    }
    \draw_path_lineto:n {
      \draw_point_vec:nn {\seq_item:Nn \l_tmpc_seq {1}}
        {\seq_item:Nn \l_tmpc_seq {2}}
    }
    \draw_path_use_clear:n {draw}
  }
\draw_path_use_clear:n {draw} \draw_end:
\ExplSyntaxOff
```