

LaTeX 概要

2023 年 1 月 3 日

前言

很早就对 \LaTeX 工具有所耳闻，但是对这个功能强大的排版系统有所畏惧，认为其学习曲线必定非常高。目前最好的中文教材应该是 [1]，但该书不是太适合入门。对于初学者来说，需要大量可以立即编译的实例。比较好的入门教材是 [2] 和 [3]。

除了自己配置各个宏包外，也有许多现成的模板和资源可以使用。**Overleaf** 是一个在线的 \LaTeX 编辑器，同时也有各种模板可供调用。**ElegantBook** 是一个非常优美和完整的中文排版模板。

compositors 宏包旨在将一些平时做笔记时常用的配置集成在一起可供重复利用，其风格是非常简约的。

目录

前言	iii
插图	ix
表格	xi
代码清单	xiii
算法清单	xv
第 1 章 \TeX 的安装与配置	1
1.1 \TeX 版本	1
1.2 安装 \TeX	1
1.2.1 下载安装脚本	1
1.2.2 创建目录	2
1.2.3 通过图形界面安装	2
1.2.4 通过命令行安装	2
1.2.5 添加路径到 PATH	3
1.3 CTAN 镜像使用帮助	3
第 2 章 \LaTeX Workshop 编辑器配置	5
2.1 VSCode settings.json 文件	5
2.2 Build with Recipe	7
第 3 章 字体	9
3.1 字体的基本知识	9
3.1.1 介绍	9
3.1.2 字体的分类	10
3.1.3 中文字体	11
3.2 下载和安装字体	12

3.2.1	Debian	12
3.2.2	macOS	12
3.3	修改环境变量 <code>OSFONTDIR</code>	12
3.3.1	Debian	12
3.3.2	macOS	13
3.3.3	将字体	13
3.4	中文字体	14
3.4.1	罗马字体族 <code>Roman family</code>	14
3.4.2	无衬线字体族 <code>Sans serif family</code>	14
3.4.3	打字机字体族 <code>Typewriter family</code>	15
3.4.4	中文字体族的定义与使用	15
第 4 章	源代码排版	17
4.1	<code>verbatim</code> 环境与 <code>\verb</code> 命令	17
4.2	<code>listings</code> 包	18
4.2.1	<code>lstlisting</code> 环境	18
4.2.2	<code>\lstinline</code> 命令	18
4.2.3	<code>\lstinputlisting</code> 命令	18
4.2.4	自定义语言	19
4.3	<code>minted</code> 包	19
4.3.1	支持的语言和风格	19
4.3.2	<code>minted</code> 环境	20
4.3.3	<code>\mint</code> 命令	20
4.3.4	<code>\mintinline</code> 命令	21
4.3.5	<code>\inputminted</code> 命令	21
4.3.6	<code>listing</code> 环境	21
4.3.7	Troubleshooting	22
4.4	<code>latexcode</code> 环境与 <code>\inputlatexcode</code> 命令	22
第 5 章	算法排版	25
5.1	伪代码与 Pascal 代码风格的算法	25
5.2	跨页显式算法	27
5.2.1	<code>\algstore</code> 和 <code>\algrestore</code> 命令	28
5.2.2	<code>breakablealgorithm</code> 环境	30

第 6 章 定理与证明的排版	33
6.1 使用 \mathcal{S} amsthm 包	33
6.2 例题与解的排版	35
第 7 章 数学公式	37
7.1 Inline 公式	37
7.1.1 math 环境	37
7.1.2 $\backslash(\dots\backslash)$ 环境	37
7.1.3 $\$ \dots \$$ 环境	38
7.2 Displayed 公式	38
7.2.1 displaymath 环境	38
7.2.2 $\backslash[\dots\backslash]$ 环境	39
7.2.3 $\$ \$ \dots \$ \$$ 环境	40
7.3 自动编号: equation 环境	40
7.4 多行公式	42
7.4.1 multline 环境	42
7.4.2 gather 环境	42
7.4.3 align 环境	43
7.4.4 split 环境	43
7.4.5 flalign 环境	44
7.4.6 alignat 环境	44
7.4.7 aligned, gathered, alignedat 环境	45
7.5 文本	46
7.6 Cases	47
第 8 章 Beamer 演示文稿	49
8.1 Beamer 主题与颜色	49
8.2 设计演示文稿	49
参考文献	51

插图

表格

3.1 中文字体族	16
-----------------	----

代码清单

4.1 C/C++ 语言	18
4.2 hello.c 文件	19
4.3 JavaScript 语言	19
4.4 hello.c 文件	22

算法清单

1	Euclid's algorithm(pseudocode)	26
2	Euclid's algorithm(pascal)	27
3	The Bellman-Kalaba algorithm(Part I)	29
4	The Bellman-Kalaba algorithm(Part II)	29
5	The Bellman-Kalaba algorithm	31

第 1 章 \TeX 的安装与配置

1.1 \TeX 版本

\TeX 官方版本是 \TeX Live, 还有各种发行版¹: Windows 平台的 MiKTeX 和 macOS 平台的 MacTeX. 后面几个版本不如官方版本稳定, 推荐安装官方版本.

- 在 Debian 中安装 \TeX Live

不要使用 apt 安装, 这个安装的是 Debian 社区维护的版本, 使用 tlmgr 时会出现一些问题, 不能自动连到最新的仓库会报错, 如: Debian 10 对应的是 texlive 2018, 而 TUG 最新的仓库对应的是 texlive 2019.

- 在 macOS 中安装 \TeX Live

尽管 TUG 推荐安装 MacTeX, 但是使用 install-tl 安装脚本可以保证与 Linux 安装一致.

1.2 安装 \TeX

1.2.1 下载安装脚本

官方主页: <https://www.ctan.org/>, 国内镜像网站²:

- <https://mirror.bjtu.edu.cn/CTAN/systems/texlive/tlnet>
- <https://mirrors.aliyun.com/CTAN/systems/texlive/tlnet>
- <http://mirrors.cloud.tencent.com/CTAN/systems/texlive/tlnet>

国内从 [北京交通大学镜像](#) 下载较快, 下载其中的 install-tl.zip, 并解压缩, 该文件包含了各平台 (Linux, macOS, Windows) 的安装脚本.

注意: 不能只下载 install-tl 运行, 否则出错:

¹中文的 CTEX 套装已不再更新

²北京交通大学镜像比清华大学镜像快.

```
Can't locate TeXLive/TLUtils.pm in @INC (you may need to install the
  TeXLive::TLUtils module) (@INC contains: ./tlpkg /usr/local/Cellar/
  perl/5.32.1_1/lib/perl5/site_perl/5.32.1/darwin-thread-multi-2level
  /usr/local/Cellar/perl/5.32.1_1/lib/perl5/site_perl/5.32.1 /usr/local/
  Cellar/perl/5.32.1_1/lib/perl5/5.32.1/darwin-thread-multi-2level
  /usr/local/Cellar/perl/5.32.1_1/lib/perl5/5.32.1
  /usr/local/lib/perl5/site_perl/5.32.1) at ./install-tl line 150.
BEGIN failed--compilation aborted at ./install-tl line 154.
```

使脚本成为可执行文件:

```
chmod +x install-tl
```

1.2.2 创建目录

```
sudo mkdir /usr/local/texlive
sudo chown USERNAME /usr/local/texlive
```

1.2.3 通过图形界面安装

运行 `install-tl` 脚本, 为了避免 \TeX 在编译 `tex` 文件时报错而临时下载各种包, 建议安装过程中选择完整安装 `full scheme`, 占用磁盘空间约 8G.

```
./install-tl --repository
  https://mirror.bjtu.edu.cn/CTAN/systems/texlive/tlnet
```

1.2.4 通过命令行安装

同样也可以通过命令行安装:

```
$ ./install-tl --no-gui --repository
  https://mirror.bjtu.edu.cn/CTAN/systems/texlive/tlnet
```

1.2.5 添加路径到 PATH

安装结束后将 T_EX Live 命令行目录添加到 PATH, macOS 修改 `~/.bash_profile`; Debian 修改 `/etc/profile` (对所有用户生效):

- 2020 版 for macOS

```
export PATH=/usr/local/texlive/2020/bin/x86_64-darwin:$PATH
```

- 2021 版 for macOS(Apple 发布了 M1 处理器)

```
export PATH=/usr/local/texlive/2021/bin/universal-darwin:$PATH
```

- Debian

```
export PATH=/usr/local/texlive/2021/bin/x86_64-linux:$PATH
```

1.3 CTAN 镜像使用帮助

CTAN (The Comprehensive T_EX Archive Network) 镜像源可以使用 T_EX Live 管理器 `tlmgr` 更改.

在命令行中执行:

```
tlmgr option repository
└─ https://mirrors.tuna.tsinghua.edu.cn/CTAN/systems/texlive/tlnet
```

即可永久更改镜像源.

如果只需要临时切换, 可以用如下命令:

```
tlmgr update --all --repository
└─ https://mirrors.tuna.tsinghua.edu.cn/CTAN/systems/texlive/tlnet
```

其中的 `update --all` 指令可根据需要修改.

第 2 章 \LaTeX Workshop 编辑器配置

\TeX 自带编辑器 \TeX Works, 而使用 [Visual Studio Code](#) + [\$\text{\LaTeX}\$ Workshop 插件](#)也是非常方便的.

2.1 VSCode settings.json 文件

To open the User settings:

1. Open the command palette (either with F1 or Ctrl+Shift+P)
2. Type "open settings"

You are presented with two options, choose Open Settings (JSON) Which, depending on platform, is one of:

- Windows `%APPDATA%\Code\User\settings.json`
- macOS `$HOME/Library/Application Support/Code/User/settings.json`
- Linux `$HOME/.config/Code/User/settings.json`

The Workspace settings will be in a workspaceName.code-workspace file where you saved it, and the Folder settings will be in a .vscode folder if and when it has been created.

```
"latex-workshop.latex.recipe.default": "lastUsed",
"latex-workshop.latex.recipes": [
  {
    "name": "latexmk",
    "tools": [
      "latexmk"
    ]
  },
]
```

```
{
  "name": "xelatex and bibtex",
  "tools": [
    "xelatex",
    "bibtex",
    "xelatex",
    "xelatex"
  ]
},
"latex-workshop.latex.tools": [
  {
    "name": "latexmk",
    "command": "latexmk",
    "args": [
      "-xelatex",
      "-synctex=1",
      "-interaction=nonstopmode",
      "-file-line-error",
      "-shell-escape",
      "%DOC%"
    ]
  },
  {
    "name": "xelatex",
    "command": "xelatex",
    "args": [
      "-synctex=1",
      "-interaction=nonstopmode",
      "-file-line-error",
      "-shell-escape",
      "%DOC%"
    ],
    "env": {}
  },
],
```

```

{
  "name": "bibtex",
  "command": "bibtex",
  "args": [
    "%DOCFILE%"
  ],
  "env": {}
}
],
"latex-workshop.latex.autoClean.run": "onBuilt",
"latex-workshop.latex.clean.fileTypes": [
  "*.aux", "*.bbl", "*.pyg",
  "*.idx", "*.ind", "*.lof",
  "*.lot", "*.out", "*.toc",
  "*.acn", "*.acr", "*.alg",
  "*.glg", "*.glo", "*.gls",
  "*.fls", "*.log", "*.fdb_latexmk",
  "*.snm", "*.synctex", "*.synctex.gz",
  "*.nav", "*.xdv", "*.ilg",
  "*.blg", "*.lol", "*.cpt"
],

```

2.2 Build with Recipe

Build with recipe:

macOS: fn+F1 -> Search 'Build with recipe' -> Select the recipe

默认的 recipe 是使用 latexmk, 为了处理中文, 配置编译器为 xelatex. 对于普通的文档可行, 但是遇到有文献目录的就无法自动生成.

含有文献目录的 tex 需要多次编译:

```

xelatex [options] filename.tex # 生成 filename.aux
bibtex filename.aux           # 生成 filename.bbl
xelatex [options] filename.tex
xelatex [options] filename.tex

```

关于中间文件的删除

1. 设置 onBuilt 就清除, 或者
2. macOS: option+command+c 不太灵敏¹

¹快捷键查看: VSCode 左边最下面齿轮 -> Keyboard Shortcuts

第 3 章 字体

TeX 排版过程中要注意正确配置字体，否则会出现 'Font not found!' 错误。常见的错误主要是：不同平台的字体不一样，如 macOS 平台编译通过的文件在 Debian 平台就可能出现错误；或者字体所在目录不在搜索路径上¹。

3.1 字体的基本知识

3.1.1 介绍

现代 TEX 引擎（包括 XETEX、LuaTEX 和 ApTEX）已经全面支持使用 OpenType 字体，因而可以很方便地实现类似 Microsoft Word、Adobe InDesign 等软件的效果。

OpenType 字体格式由微软和 Adobe 联合开发，它有以下特点：

- 字体编码基于 Unicode
- 描述字体轮廓时，既可以用三次 Bézier 曲线（PostScript 轮廓），也可以用二次 Bézier 曲线（TrueType 轮廓）
- 支持合字（也叫连字，ligature）、小型大写字母（small caps）、多种数字样式、上下标、上下文替换等高级字体排印功能，并可用于复杂语言排版
- 在 Windows、macOS、Linux 等多种平台下均可使用

OpenType 字体的文件扩展名可为.otf、.ttf、.ttc，它们的区别如下：

- .otf：单个字体，使用 PostScript 轮廓
- .ttf：单个字体，使用 TrueType 轮廓

¹关于 TeX 中西字体排版，请参考：

- [在 TeX 中使用 OpenType 字体（一）](#)
- [在 TeX 中使用 OpenType 字体（二）](#)
- [在 TeX 中使用 OpenType 字体（三）](#)

- .ttc: 字体集合 (TrueType/OpenType Collection), 在单个文件中打包多个字体, 允许使用 TrueType 或 PostScript 轮廓

LATEX 中使用 OpenType 字体, 主要依靠下列宏包:

- fontspec: 通用字体选取
- CTeX: LATEX 中文排版框架, 依据引擎的不同, 底层分别利用 xeCJK 宏包 (XETEX) 和 luatexja 宏包 (LuaTEX) 来处理
- unicode-math: 实验性的 Unicode 数学排版功能

考虑到 ApTeX 仍处于开发状态, 尚未提供良好的用户接口, 因此下文仅适用于 XETEX 和 LuaTEX 引擎。

3.1.2 字体的分类

大体上说, 西文字体可以分为衬线体 (serif font) 和无衬线体 (sans-serif font) 两大类。

衬线体

所谓“衬线”, 是指笔画末端的一种装饰细节, 一般认为起源于古罗马的石刻拉丁字母。衬线据说有引导视线的作用, 因而往往被用作书籍、文章等的正文字体。根据产生年代以及风格样式, 又可以细分为几类。

无衬线体

顾名思义, 无衬线体就是指没有衬线的字体, 它在近现代才得到广泛的发展与运用。早些年, 屏幕分辨率远达不到印刷质量, 衬线等细节很难反映出来, 因此无衬线体在屏幕、网页显示上大行其道。

等宽字体

等宽字体与比例字体相对, 其中的所有字母、符号均有相同的宽度。严格来说, 等宽字体并不能单独作为一个分类, 但由于其用法比较特殊 (现代主要用于计算机程序的排版), 这里还是把它单独列出来。

变体

直立、正体的字型通常被称为罗马体 (Roman type)。除此之外, 一套完整的字体往往还具有若干其他字型。

- 意大利体 (italic type): 字形稍向右倾, 笔画带有手写体的风格。它不是罗马体的简单倾斜
- 斜体 (oblique/slant type): 大部分是原字形的简单倾斜, 少数设计精良的字体 (如 Univers、Helvetica 等) 也会再额外进行视觉修正
- 字重 (weight): 表示字体的粗细, 除了常规体 (regular), 还有粗体 (bold)、细体 (light), 以及半粗、半细、超粗、超细、特粗、特细等。现代字体也流行以数字区分不同字重, 典型例子如 Univers
- 小型大写 (small caps): 顾名思义, 形状是大写字母, 但尺寸略小 (一般接近小写字母)。西文中, 全大写字母组成的单词 (如 WHO) 常用小型大写的形式

3.1.3 中文字体

中文字体原则上应该与西文字体平行列出。但出于历史原因, 现代可用的中文字体数量远小于西文字体, 也更有细致的分类方案。

正文中常用的中文字体主要有宋、黑、楷、仿四种。括号中给出的是对应的日文名称²。

- 宋体 (明朝体): 实际上诞生于明朝, 因而也叫明体。特点是笔画硬朗、横细竖粗, 笔画末端带有类似「衬线」的装饰三角形。宋体习惯用于正文排版, 适合与衬线的西文字体搭配
- 黑体 (ゴシック体): 笔画粗细变化较小, 横竖对比也较小, 常与无衬线西文字体搭配
- 楷体 (楷書体): 来源于传统书法, 字形端庄, 对应与西文字体中的手写体
- 仿宋 (宋朝体): 源自于宋朝的刻书字体, 但实际成形于民国初年。特点是兼有宋体的结构与楷体的笔画, 较为清秀挺拔

CTeX 宏默认的字库配置基于以下逻辑:

- 宋体对应到 `\rmfamily` 的 `\upshape`
- 黑体 (或微软雅黑、苹方等现代黑体) 对应到 `\sffamily` 的 `\upshape`
- 楷体对应到 `\rmfamily` 的 `\itshape`
- 仿宋对应到 `\ttfamily` 的 `\upshape`
- 如果以上某种字体有相应的粗体, 则将其对应到 `\bfseries`, 如果没有则不做特殊处理; 特别地, 加粗宋体如果不存在的话, 则会改用黑体 (但不使用现代黑体)

²方正书宋简体不能正确显示这些日文名称, 此处使用了 Noto Serif SC 字体。

- 最后使用 `\setCJKfamilyfont` 命令定义额外的字体命令，如 `\songti`、`\heiti` 等；此时如果有隶书和圆体，也会用同样方式定义字体命令

3.2 下载和安装字体

texlive 自带的中文字体比较有限，如其自带 Fandol 楷体，该字库的“仓”误为“仑”，故不用。

可以从方正字体³下载方正字体：

<https://www.foundertype.com/index.php/FindFont/index>

从 Google Fonts 下载 Noto Sans SC 和 Noto Serif SC 字体：

<https://fonts.google.com/noto/fonts>

3.2.1 Debian

字体可以安装在下列目录中：

- `/usr/share/fonts`
- `/usr/local/share/fonts`
- `~/.fonts`

3.2.2 macOS

字体可以安装在下列目录中：

- `/System/Library/Fonts`
- `/Library/Fonts`
- `~/Library/Fonts`

3.3 修改环境变量 OSFONTDIR

修改 `/usr/local/texlive/2021/texmf.cnf` (根据实际安装的 texlive 版本)，添加：

3.3.1 Debian

³其它选择：Adobe Fonts 或 Google Fonts

```
OSFONTDIR = /usr/share/fonts//;/usr/local/share/fonts//;~/.fonts//
```

3.3.2 macOS

```
OSFONTDIR = /System/Library/Fonts//;/Library/Fonts//;~/Library/Fonts//
```

3.3.3 将字体

TeX 使用本地字库，或者在系统字体目录中安装相应的字体。其实，TeX 可以实现直接读取字体文件（如存储在项目的 `fonts/` 目录下）来进行文档的渲染工作，也就是项目本身含有字体文件⁴。

```
% 设置英文字体
\setmainfont[
  Path=fonts/,
  BoldFont=NotoSerif-Bold.otf,
  ItalicFont=NotoSerif-RegularItalic.otf,
  BoldItalicFont=NotoSerif-BoldItalic.otf,
]{NotoSerif-Regular.otf}
\setsansfont[
  Path=fonts/,
  BoldFont=NotoSans-Bold.otf,
  ItalicFont=NotoSans-RegularItalic.otf,
  BoldItalicFont=NotoSans-BoldItalic.otf,
]{NotoSans-Regular.otf}
\setmonofont[
  Path=fonts/
]{NotoSans-Regular.otf}

% 设置中文字体
\setCJKmainfont[
  Path=fonts/,
  BoldFont=NotoSerifSC-Bold.otf,
```

⁴注意：中文字体文件都比较大，会使得项目非常臃肿

```

        ItalicFont=NotoSerifSC-Regular.otf,
        BoldItalicFont=NotoSerifSC-Bold.ttf
    ]{NotoSerifSC-Regular.otf}
\setCJKsansfont[
    Path=fonts/,
    BoldFont=NotoSansSC-Bold.otf,
    ItalicFont=NotoSansSC-Regular.otf,
    BoldItalicFont=NotoSansSC-Bold.ttf
]{NotoSansSC-Regular.otf}
\setCJKmonofont[
    Path=fonts/
]{NotoSansSC-Regular.otf}

```

3.4 中文字体

中文字体与西文字体有很大区别，中文一般没有西文对应的成套字体，如粗体、斜体等。衬线（Serif）是字形笔画的起始段与末端的装饰细节部分，有衬线体如 Times New Roman，无衬线体（Sans serif）如 Arial。中文同样也有衬线体（如宋体）和无衬线体（如黑体）。

3.4.1 罗马字体族 Roman family

这是正文的默认字体，也可以使用 `\textrm{文字内容}` 或 `{\rmfamily 文字内容}` 重新设置文字的字体。

- Normal: 世界自由、正义与和平的基础 Lorem ipsum dolor sat amet
- Bold: 世界自由、正义与和平的基础 **Lorem ipsum dolor sat amet**
- Italic: 世界自由、正义与和平的基础 *Lorem ipsum dolor sat amet*
- BoldItalic: 世界自由、正义与和平的基础 ***Lorem ipsum dolor sat amet***

3.4.2 无衬线字体族 Sans serif family

使用 `\textsf{文字内容}` 或 `{\sffamily 文字内容}` 设置文字的字体。

- Normal: 世界自由、正义与和平的基础 Lorem ipsum dolor sat amet
- Bold: 世界自由、正义与和平的基础 **Lorem ipsum dolor sat amet**

- Italic: 世界自由、正义与和平的基础 *Lorem ipsum dolor sat amet*
- BoldItalic: 世界自由、正义与和平的基础 ***Lorem ipsum dolor sat amet***

3.4.3 打字机字体族 Typewriter family

通常代码抄录使用该字体族, 使用 `\texttt{文字内容}` 或 `{\ttfamily 文字内容}` 设置文字的字体。

- Normal: 世界自由、正义与和平的基础 Lorem ipsum dolor sat amet
- Bold: 世界自由、正义与和平的基础 **Lorem ipsum dolor sat amet**
- Italic: 世界自由、正义与和平的基础 *Lorem ipsum dolor sat amet*
- BoldItalic: 世界自由、正义与和平的基础 ***Lorem ipsum dolor sat amet***

3.4.4 中文字体族的定义与使用

使用 `\setCJKfamily{字体名称}{字体文件名}` 定义中文字体族⁵, 如:

```
\setCJKfamilyfont{宋体}{方正书宋简体.OTF}
\setCJKfamilyfont{圆体}{方正准圆简体.OTF}
\setCJKfamilyfont{仿宋}{方正仿宋简体.OTF}
\setCJKfamilyfont{楷体}{方正楷体简体.OTF}
\setCJKfamilyfont{黑体}{方正黑体简体.OTF}
\setCJKfamilyfont{隶书}{方正隶书简体.OTF}
\setCJKfamilyfont{行楷}{方正行楷简体.OTF}
\setCJKfamilyfont{行书}{字酷堂松雪行书简体.OTF}
```

使用 `\CJKfamily{字体名称}{文字内容}` 设置文字的字体。

⁵这里字体名称采用了中文, 也可以使用字母, 如 songti, fangsong 等

表 3.1: 中文字体族

字体名称	示例
宋体	世界自由、正义与和平的基础
圆体	世界自由、正义与和平的基础
仿宋	世界自由、正义与和平的基础
楷体	世界自由、正义与和平的基础
黑体	世界自由、正义与和平的基础
隶书	世界自由、正义与和平的基础
行楷	世界自由、正义与和平的基础
行书	世界自由、正义与和平的基础

第 4 章 源代码排版

4.1 verbatim 环境与\verb 命令

verbatim 环境适合比较简单场合下使用，无法实现代码的高亮显示，如：

```
\begin{verbatim}
#include <stdio.h>

int main(int argc, char **argc) {
    printf("Hello, world\n");
    return 0;
}
\end{verbatim}
```

```
#include <stdio.h>

int main(int argc, char **argc) {
    printf("Hello, world\n");
    return 0;
}
```

\verb 命令可以使用行内的代码显示，如：\verb|message = "hello,world"|。

实际上使用{\ttfamily message = "hello,world"}也可以达到类似的效果；并且\verb 命令不能出现章节标题之中。

4.2 listings 包

4.2.1 lstlisting 环境

```
\begin{lstlisting}[language=C,caption={C/C++ 语言},numbers=left]
/* Hello, world */
int main(int argc, char **argv) {
    printf("Hello, world\n");
    return 0;
}
\end{lstlisting}
```

代码 4.1: C/C++ 语言

```
1  /* Hello , world */
2  int main(int argc, char **argv) {
3      printf("Hello , world\n");
4      return 0;
5  }
```

4.2.2 \lstinline 命令

\lstinline 命令可以使用行内的代码显示,如:\lstinline|message = "hello,world"|

4.2.3 \lstinputlisting 命令

显示代码文件, 文件路径为相对于主文件的路径:

```
\lstinputlisting[language=C,caption={hello.c 文
件},numbers=left]{snippets/codelistings/helloworld.c}
```

代码 4.2: hello.c 文件

```
1 #include <stdio.h>
2
3 int main(int argc, char **argv) {
4     printf("Hello, world\n");
5     return 0;
6 }
```

4.2.4 自定义语言

listings 包没有提供 JavaScript 语言高亮显示，但是可以自定义（详见模板）。

```
\begin{lstlisting}[language=js,caption={JavaScript 语言}]
// 向控制台输出 'hello, world'
const message = 'hello, world'
console.log(message)
\end{lstlisting}
```

代码 4.3: JavaScript 语言

```
1 // 向控制台输出 'hello, world'
2 const message = 'hello, world'
3 console.log(message)
```

其它的扩展见: [Adding Keywords to Existing Language for Listings Package](#)

4.3 minted 包

minted 包需要借助 [pygments](#) 来渲染代码。minted 整体风格比 listings 略胜一筹。

4.3.1 支持的语言和风格

首先，安装 pygments:

```
pip3 install pygments
```

查看支持的语言：

```
pygmentize -L lexers
```

查查支持的风格：

```
pygmentize -L styles
```

此外，可以通过网站 [Pygments website](#) 或 [Syntax Highlighter](#) 来查看各种代码渲染的风格¹。

4.3.2 minted 环境

```
\begin{minted}{c}
int main(int argc, char **argv) {
    printf("Hello, world\n");
    return 0;
}
\end{minted}
```

```
int main(int argc, char **argv) {
    printf("Hello, world\n");
    return 0;
}
```

4.3.3 \mint 命令

对于单行的代码，可以使用 `\mint` 命令²，如：`\mint{js}|message = "hello, world"|`。

¹注意：部分风格与一些特殊字符有冲突，比较可靠的风格是 `xcode`。

²注意：不是在行内显示代码，而是另起一段显示。

4.3.4 \mintinline 命令

可以使用 \mintinline 命令来在行内显示代码,如:\mintinline{js}|message = "hello, world"|。

4.3.5 \inputminted 命令

显示代码文件, 文件路径为相对于主文件的路径:

```
\inputminted{c}{snippets/codelistings/helloworld.c}
```

```
#include <stdio.h>

int main(int argc, char **argv) {
    printf("Hello, world\n");
    return 0;
}
```

4.3.6 listing 环境

minted 提供 listing 环境来作为一个浮动体来包裹代码。

```
\begin{listing}[H]
  \caption{hello.c 文件}
  \inputminted{c}{snippets/codelistings/helloworld.c}
\end{listing}
```

代码 4.4: hello.c 文件

```
#include <stdio.h>

int main(int argc, char **argv) {
    printf("Hello, world\n");
    return 0;
}
```

4.3.7 Troubleshooting

在设置 `bgcolor` 后 `\mintinline` 不换行

在设置 `bgcolor` 后,对于较长的一行代码使用 `\mintinline` 时不换行,见 [Github issue 194: breaklines doesn't work with mintinline when other options are set](#) 和 [Stackoverflow: breaklines doesn't work with mintinline](#)。

The documentation mentions this as a limitation of `bgcolor`. The standard ways to put a background behind inline text don't work with line wrapping.

代码的换页

见 [Stackoverflow: Pagebreak for minted in figure](#) 和 [Code spanning over two pages with minted, inside listing with caption](#)。

一些字符出现红色方框

见 [Minted red box around greek characters](#) 和 [Code validation in minted package? How to disable it?](#)。

解决方法,使用下面的风格: `xcode`, `igor`, `rvt`。

4.4 latexcode 环境与 `\inputlatexcode` 命令

为了显示 \LaTeX 代码和现实渲染结果³,可以采用:

- `latexcode` 环境: 工作过程是将代码输出到文件,读取并渲染该文件

³实现类似 `tcblisting` 环境的功能

- `\inputlatexcode` 命令：直接读取并渲染该文件

显示并渲染代码：

```
\begin{latexcode}[colorbox]
{\LaTeX} is a high-quality typesetting system; it includes features designed
  for the production of technical and scientific documentation.
\end{latexcode}
```

显示并渲染代码文件，渲染结果带有 `colorbox`（这是默认参数）：

```
\inputlatexcode[colorbox]{.tex 文件路径}
```

显示并渲染代码文件，渲染结果不带 `colorbox`⁴：

```
\inputlatexcode[nobox]{.tex 文件路径}
```

⁴`algorithm` 与 `tcolorbox` 存在冲突，请使用 `nobox` 参数

第 5 章 算法排版

TeX 中算法或伪代码的排版可以选择下面两种方案的一种¹:

- 使用 `algpseudocode` 或 `algorithms` 或 `algorithmic` 包（三选一）排版算法主体部分，使用 `algorithm` 包设置算法标题
- 使用 `algorithm2e` 包

注意：上述宏包不能同时使用，否则将产生很多错误和冲突。各宏包的主要功能²：

algorithm float wrapper for algorithms.

algorithmic first algorithm typesetting environment.

algorithmicx second algorithm typesetting environment.

algpseudocode layout for algorithmicx.

algorithm2e third algorithm typesetting environment.

5.1 伪代码与 Pascal 代码风格的算法

在导言区加入：

```
\usepackage{algorithm}
\usepackage{algpseudocode}
\usepackage{algpascal}
```

¹<https://www.overleaf.com/learn/latex/Algorithms>

²<https://tex.stackexchange.com/questions/229355/algorithm-algorithmic-algorithmicx-algorithm2e-algpseudocode-confused>

```

1 \alglanguage{pseudocode}
2 \begin{algorithm}
3   \caption{Euclid's algorithm(pseudocode)}\label{euclid-pseduocode}
4   \begin{algorithmic}[1]
5     \Procedure{Euclid}{$a,b$}\Comment{The g.c.d. of a and b}
6       \State $r$ \gets $a\bmod b$
7       \While{$r\neq 0$}\Comment{We have the answer if r is 0}
8         \State $a$ \gets $b$
9         \State $b$ \gets $r$
10        \State $r$ \gets $a\bmod b$
11      \EndWhile
12      \State \textbf{return} $b$\Comment{The gcd is b}
13    \EndProcedure
14  \end{algorithmic}
15 \end{algorithm}

```

Algorithm 1 Euclid's algorithm(pseudocode)

1:	procedure EUCLID(a, b)	▷ The g.c.d. of a and b
2:	$r \leftarrow a \bmod b$	
3:	while $r \neq 0$ do	▷ We have the answer if r is 0
4:	$a \leftarrow b$	
5:	$b \leftarrow r$	
6:	$r \leftarrow a \bmod b$	
7:	end while	
8:	return b	▷ The gcd is b
9:	end procedure	

```

1 \alglanguage{pseudocode}
2 \begin{algorithm}
3   \caption{Euclid's algorithm(pascal)}\label{euclid-pascal}
4   \begin{algorithmic}[1]
5     \Procedure{Euclid}{$a,b$}\Comment{The g.c.d. of a and b}
6       \State $r := a\bmod b$

```

```

7      \While{$r\not= 0$}\Comment{We have the answer if r is 0}
8      \State $a := b$
9      \State $b := r$
10     \State $r := a\bmod b$
11     \EndWhile\label{euclidendwhile}
12     \State \textbf{return} $b$\Comment{The gcd is b}
13 \EndProcedure
14 \end{algorithmic}
15 \end{algorithm}

```

Algorithm 2 Euclid's algorithm(pascal)

1:	procedure EUCLID(a, b)	▷ The g.c.d. of a and b
2:	$r := a \bmod b$	
3:	while $r \neq 0$ do	▷ We have the answer if r is 0
4:	$a := b$	
5:	$b := r$	
6:	$r := a \bmod b$	
7:	end while	
8:	return b	▷ The gcd is b
9:	end procedure	

注 使用时注意以下问题:

1. 算法排版, algorithm 与 tcolorbox 存在冲突, 即 tcolorbox 环境不能包含 algorithm 环境;
2. 当同时引入 algpseudocode 和 algpascal, 必须使用 \aglanguage{pseudocode} 或 \aglanguage{pascal}, 否则不能正确解析数学公式.

5.2 跨页显式算法

如果算法较长, 则需要分页显示, 有两种方法:

1. \agstore 和 \agrestore 命令将算法分成几个部分
2. 自定义 breakablealgorithm 环境³

³<https://tex.stackexchange.com/questions/33866/algorithm-tag-and-page-break>

5.2.1 `\algstore` 和 `\algrestore` 命令

```

1  \alglanguage{pseudocode}
2  % 算法第一部分
3  \begin{algorithm}
4  \caption{The Bellman-Kalaba algorithm(Part I)}
5  \begin{algorithmic}[1]
6  \Procedure {BellmanKalaba}{ $G$ ,  $u$ ,  $l$ ,  $p$ }
7    \ForAll  $\{v \in V(G)\}$ 
8      \State  $l(v) \leftarrow \infty$ 
9    \EndFor
10   \State  $l(u) \leftarrow 0$ 
11   \Repeat
12     \For  $\{i \leftarrow 1, n\}$ 
13       \State  $\min \leftarrow l(v_i)$ 
14       \For  $\{j \leftarrow 1, n\}$ 
15         \If  $\{\min > e(v_i, v_j) + l(v_j)\}$ 
16           \State  $\min \leftarrow e(v_i, v_j) + l(v_j)$ 
17           \State \Comment For some reason we need to break here!
18 \algstore{bkbreak}
19 \end{algorithmic}
20 \end{algorithm}
21
22 % 算法第二部分
23 \begin{algorithm}
24 \caption{The Bellman-Kalaba algorithm(Part II)}
25 \begin{algorithmic}[1]
26 \algrestore{bkbreak}
27   \State  $p(i) \leftarrow v_j$ 
28   \EndIf
29   \EndFor
30   \State  $l'(i) \leftarrow \min$ 
31 \EndFor
32 \State  $\text{changed} \leftarrow 1 \neq l'$ 
33 \State  $l \leftarrow l'$ 

```

```

34   \Until{ $\neg$  changed}
35 \EndProcedure
36 \end{algorithmic}
37 \end{algorithm}

```

Algorithm 3 The Bellman-Kalaba algorithm(Part I)

```

1: procedure BELLMANKALABA( $G, u, l, p$ )
2:   for all  $v \in V(G)$  do
3:      $l(v) \leftarrow \infty$ 
4:   end for
5:    $l(u) \leftarrow 0$ 
6:   repeat
7:     for  $i \leftarrow 1, n$  do
8:        $min \leftarrow l(v_i)$ 
9:       for  $j \leftarrow 1, n$  do
10:        if  $min > e(v_i, v_j) + l(v_j)$  then
11:           $min \leftarrow e(v_i, v_j) + l(v_j)$ 
12:

```

▷ For some reason we need to break here!

Algorithm 4 The Bellman-Kalaba algorithm(Part II)

```

13:      $p(i) \leftarrow v_j$ 
14:   end if
15: end for
16:    $l(i) \leftarrow min$ 
17: end for
18:    $changed \leftarrow l \neq l$ 
19:    $l \leftarrow l$ 
20: until  $\neg changed$ 
21: end procedure

```

5.2.2 breakablealgorithm 环境

```

1  \alanguage{pseudocode}
2  \begin{breakablealgorithm}
3  \caption{The Bellman-Kalaba algorithm}
4  \begin{algorithmic}[1]
5  \Procedure {BellmanKalaba}{$G$, $u$, $l$, $p$}
6    \ForAll {$v$ \in $V(G)$}
7      \State $l(v)$ \leftarrow $\infty$
8    \EndFor
9    \State $l(u)$ \leftarrow 0$
10   \Repeat
11     \For {$i$ \leftarrow 1, $n$}
12       \State $\min$ \leftarrow $l(v_i)$
13       \For {$j$ \leftarrow 1, $n$}
14         \If {$\min > e(v_i, v_j) + l(v_j)$}
15           \State $\min$ \leftarrow $e(v_i, v_j) + l(v_j)$
16           \State $p(i)$ \leftarrow $v_j$
17         \EndIf
18       \EndFor
19       \State $l'(i)$ \leftarrow $\min$
20     \EndFor
21     \State $\text{changed}$ \leftarrow 1 \not= $l'$ $
22     \State $l$ \leftarrow $l'$ $
23   \Until{ $\neg$ $\text{changed}$ }
24 \EndProcedure
25 \Statex
26 \Procedure {FindPathBK}{$v$, $u$, $p$}
27   \If {$v = u$}
28     \State \textbf{Write} $v$
29   \Else
30     \State $w$ \leftarrow $v$
31     \While {$w \not= u$}
32       \State \textbf{Write} $w$
33       \State $w$ \leftarrow $p(w)$

```

```

34     \EndWhile
35   \EndIf
36 \EndProcedure
37 \Statex
38 \Procedure {FindPathBK}{ $v$ ,  $u$ ,  $p$ }
39   \If { $v = u$ }
40     \State \textbf{Write}  $v$ 
41   \Else
42     \State  $w \leftarrow v$ 
43     \While { $w \neq u$ }
44       \State \textbf{Write}  $w$ 
45       \State  $w \leftarrow p(w)$ 
46     \EndWhile
47   \EndIf
48 \EndProcedure
49 \end{algorithmic}
50 \end{breakablealgorithm}

```

Algorithm 5 The Bellman-Kalaba algorithm

```

1: procedure BELLMANKALABA( $G, u, l, p$ )
2:   for all  $v \in V(G)$  do
3:      $l(v) \leftarrow \infty$ 
4:   end for
5:    $l(u) \leftarrow 0$ 
6:   repeat
7:     for  $i \leftarrow 1, n$  do
8:        $min \leftarrow l(v_i)$ 
9:       for  $j \leftarrow 1, n$  do
10:        if  $min > e(v_i, v_j) + l(v_j)$  then
11:           $min \leftarrow e(v_i, v_j) + l(v_j)$ 
12:           $p(i) \leftarrow v_j$ 
13:        end if
14:      end for
15:       $l(i) \leftarrow min$ 
16:    end for

```

```

17:       $changed \leftarrow l \neq l$ 
18:       $l \leftarrow l$ 
19:  until  $\neg changed$ 
20: end procedure

```

```

21: procedure FINDPATHBK( $v, u, p$ )
22:   if  $v = u$  then
23:     Write  $v$ 
24:   else
25:      $w \leftarrow v$ 
26:     while  $w \neq u$  do
27:       Write  $w$ 
28:        $w \leftarrow p(w)$ 
29:     end while
30:   end if
31: end procedure

```

```

32: procedure FINDPATHBK( $v, u, p$ )
33:   if  $v = u$  then
34:     Write  $v$ 
35:   else
36:      $w \leftarrow v$ 
37:     while  $w \neq u$  do
38:       Write  $w$ 
39:        $w \leftarrow p(w)$ 
40:     end while
41:   end if
42: end procedure

```

第 6 章 定理与证明的排版

6.1 使用 $\mathcal{A}\mathcal{M}\mathcal{S}$ amsthm 包

在 compositors 宏包中定义了 theorem, lemma, proposition, corollary, definition, example, remark, remark*, solution, solution*, proof 环境。

```
\begin{definition}[Hypotenuse]
  The longest side of a triangle with a right angle is called the
  ~ \emph{hypotenuse}.
\end{definition}
```

定义 6.1 (Hypotenuse) The longest side of a triangle with a right angle is called the *hypotenuse*.

```
\begin{remark}
  The other sides are called \emph{catheti}, or \emph{legs}.
\end{remark}
```

注 1 The other sides are called *catheti*, or *legs*.

```
\begin{theorem}[Pythagoras]
  \label{pythagoras}
  In any right triangle, the square of the hypotenuse equals the sum of the
  ~ squares of the other sides.
\end{theorem}
```

定理 6.1 (Pythagoras) In any right triangle, the square of the hypotenuse equals the sum of the squares of the other sides.

```
\begin{proof}
  The proof has been given in Euclid's Elements,
  Book 1, Proposition 47. Refer to it for details.
  The converse is also true, see lemma \ref{converse}.
\end{proof}
```

证明 The proof has been given in Euclid's Elements, Book 1, Proposition 47. Refer to it for details. The converse is also true, see lemma 6.1. □

```
\begin{lemma}
  \label{converse}
  For any three positive numbers  $x$ ,  $y$ ,
  and  $z$  with  $x^2 + y^2 = z^2$ , there is a triangle with side lengths
   $x$ ,  $y$  and  $z$ . Such triangle has a right angle, and the
  hypotenuse has the length  $z$ .
\end{lemma}
```

引理 6.1 For any three positive numbers x , y , and z with $x^2 + y^2 = z^2$, there is a triangle with side lengths x , y and z . Such triangle has a right angle, and the hypotenuse has the length z .

```
\begin{remark*}
  This is the converse of theorem \ref{pythagoras}.
\end{remark*}
```

注 This is the converse of theorem 6.1.

6.2 例题与解的排版

```
\begin{example}
  求方程  $x^2 - x - 12 = 0$  的根.
\end{example}

\begin{solution*}
  对方程左边进行因式分解:
  \begin{gather*}
    (x-4)(x+3) = 0 \\
    \intertext{因此,}
    x-4=0 \quad \text{或} \quad x+3=0 \\
    \intertext{所以, 方程的根是:}
    x_1 = 4, \quad x_2 = -3
  \end{gather*}
\end{solution*}
```

例 6.1 求方程

$$x^2 - x - 12 = 0$$

的根.

解 对方程左边进行因式分解:

$$(x-4)(x+3) = 0$$

因此,

$$x-4=0 \quad \text{或} \quad x+3=0$$

所以, 方程的根是:

$$x_1 = 4, \quad x_2 = -3$$

第 7 章 数学公式

7.1 Inline 公式

由于输入便捷, \dots 更受欢迎.

7.1.1 `math` 环境

使用 `math` 环境:

```
1 一元二次方程
2 \begin{math}
3   ax^2 + bx + c = 0
4 \end{math}
5 的根是
6 \begin{math}
7   x_{1,2} = \cfrac{-b \pm \sqrt{b^2-4ac}}{2a}.
8 \end{math}
```

一元二次方程 $ax^2 + bx + c = 0$ 的根是 $x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$.

7.1.2 `\(...\)` 环境

使用 `\(...\)` 环境:

```
1 一元二次方程
2 \(\
```

```

3      ax^2 + bx + c = 0
4  \)
5  的根是
6  \ (
7      x_{1,2} = \cfrac{-b \pm \sqrt{b^2-4ac}}{2a}.
8  \)

```

一元二次方程 $ax^2 + bx + c = 0$ 的根是 $x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$.

7.1.3 $\$ \dots \$$ 环境

使用 $\$ \dots \$$ 环境:

```

1 一元二次方程
2  $
3      ax^2 + bx + c = 0
4  $
5  的根是
6  $
7      x_{1,2} = \cfrac{-b \pm \sqrt{b^2-4ac}}{2a}.
8  $

```

一元二次方程 $ax^2 + bx + c = 0$ 的根是 $x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$.

7.2 Displayed 公式

推荐使用 $\backslash[\dots \backslash]$, 在处理间距更好一些.

7.2.1 displaymath 环境

使用 `displaymath` 环境:

```

1 一元二次方程
2 \begin{displaymath}
3   ax^2 + bx + c = 0
4 \end{displaymath}
5 的根是
6 \begin{displaymath}
7   x_{1,2} = \cfrac{-b \pm \sqrt{b^2-4ac}}{2a}.
8 \end{displaymath}

```

一元二次方程

$$ax^2 + bx + c = 0$$

的根是

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

7.2.2 \[...\] 环境

使用 \[...\] 环境：

```

1 一元二次方程
2 \[
3   ax^2 + bx + c = 0
4 \]
5 的根是
6 \[
7   x_{1,2} = \cfrac{-b \pm \sqrt{b^2-4ac}}{2a}.
8 \]

```

一元二次方程

$$ax^2 + bx + c = 0$$

的根是

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

7.2.3 `$$...$$` 环境

使用 `$$...$$` 环境：

```

1 一元二次方程
2  $$
3  ax^2 + bx + c = 0
4  $$
5  的根是
6  $$
7  x_{1,2} = \cfrac{-b \pm \sqrt{b^2-4ac}}{2a}.
8  $$

```

一元二次方程

$$ax^2 + bx + c = 0$$

的根是

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

7.3 自动编号: `equation` 环境

`equation`, `align` 等环境默认是自动编号的, 如果不需要编号则使用 `equation*`, `align*` 等环境。对应多行公式内容, 如果不需要编号则使用 `\\notag`。

```

1 The quadratic equation
2 \begin{equation}
3 ax^2 + bx + c = 0,

```



```

4   \label{quad}
5   \end{equation}
6   where \((a, b)\) and \((c)\) are constants and \((a \neq 0)\),
7   has two solutions for the variable \((x)\):
8   \begin{equation}
9     \label{root}
10    x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.
11  \end{equation}
12
13  If the \emph{discriminant} \((\Delta)\) with
14  \[
15  \Delta = b^2 - 4ac
16  \]
17  is zero, then the equation (\ref{quad}) has a double solution: (\ref{root})
18  becomes
19  \[
20  x = -\frac{b}{2a}.

```

The quadratic equation

$$ax^2 + bx + c = 0, \quad (7.1)$$

where a, b and c are constants and $a \neq 0$, has two solutions for the variable x :

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. \quad (7.2)$$

If the *discriminant* Δ with

$$\Delta = b^2 - 4ac$$

is zero, then the equation (7.1) has a double solution: (7.2) becomes

$$x = -\frac{b}{2a}.$$

7.4 多行公式

7.4.1 multline 环境

```

1 % The first line is left-aligned, the last line right-aligned, and all other
  ~ lines in between are centered
2 \begin{multline}
3   \sum = a + b + c + d + e \\
4         + f + g + h + i + j \\
5         + k + l + m + n
6 \end{multline}

```

$$\begin{aligned} \sum &= a + b + c + d + e \\ &\quad + f + g + h + i + j \\ &\quad + k + l + m + n \end{aligned} \quad (7.3)$$

7.4.2 gather 环境

```

1 % All lines are centered
2 \begin{gather}
3   (a + b)^2 = a^2 + 2ab + b^2 \\
4   (a + b) \cdot (a - b) = a^2 - b^2
5 \end{gather}

```

$$(a + b)^2 = a^2 + 2ab + b^2 \quad (7.4)$$

$$(a + b) \cdot (a - b) = a^2 - b^2 \quad (7.5)$$

7.4.3 align 环境

```

1 % The lines are aligned at marked relation signs
2 \begin{align}
3   (a + b)^3 &= (a + b)(a+b)^2 && \\
4             &= (a + b)(a^2 + 2ab + b^2) && \\
5             &= a^3 + 3a^2b + 3ab^2 + b^3
6 \end{align}
7
8 % multiple alignments
9 \begin{align}
10  x^2 + y^2 &= 1 && \\
11  x^3 + y^3 &= 1 && \\
12  x &= \sqrt{1-y^2} && \\
13  x &= \sqrt[3]{1-y^3} && \\
14 \end{align}

```

$$(a+b)^3 = (a+b)(a+b)^2 \quad (7.6)$$

$$= (a+b)(a^2 + 2ab + b^2) \quad (7.7)$$

$$= a^3 + 3a^2b + 3ab^2 + b^3 \quad (7.8)$$

$$x^2 + y^2 = 1 \qquad x^3 + y^3 = 1 \quad (7.9)$$

$$x = \sqrt{1-y^2} \qquad x = \sqrt[3]{1-y^3} \quad (7.10)$$

$$(7.11)$$

7.4.4 split 环境

```

1 % Similar to align, but within another math environment, thus unnumbered.
2 \begin{equation}
3   \label{perfect-square}

```

```

4 \begin{split}
5 (a+b)^2 &= a(a+b)+b(a+b) \\
6 &= a^2+ab+ba+b^2 \\
7 &= a^2+b^2+2ab \\
8 \end{split}
9 \end{equation}

```

$$\begin{aligned}
 (a+b)^2 &= a(a+b)+b(a+b) \\
 &= a^2+ab+ba+b^2 \\
 &= a^2+b^2+2ab
 \end{aligned}
 \tag{7.12}$$

7.4.5 flalign 环境

```

1 % Similar to align with more than one column, but the columns are flushed to
  - the left and the right margin, respectively.
2 \begin{flalign}
3 x &= t & x &= 2 \\
4 y &= 2t & y &= 4 \\
5 \end{flalign}

```

$$\begin{array}{ll}
 x = t & x = 2 \quad (7.13) \\
 y = 2t & y = 4 \quad (7.14)
 \end{array}$$

7.4.6 alignat 环境

```

1 % Alignment at several places, each has to be marked by &.
2 \begin{alignat}{5}
3 & a_1 & & + a_3 & = 10

```

```

4      & &+ a_2 & &      & &+ a_4 & &= 20 \\
5      & a_1 & &+ a_2 & &+ a_3 & &+ a_4 & &= 30
6 \end{alignat}

```

$$a_1 + a_3 = 10 \quad (7.15)$$

$$+ a_2 + a_4 = 20 \quad (7.16)$$

$$a_1 + a_2 + a_3 + a_4 = 30 \quad (7.17)$$

```

1 \begin{alignat}{2}
2   x &= \sin t & \quad & \text{horizontal direction} \\
3   y &= \cos t & \quad & \text{vertical direction} \\
4 \end{alignat}

```

$$x = \sin t \quad \text{horizontal direction} \quad (7.18)$$

$$y = \cos t \quad \text{vertical direction} \quad (7.19)$$

注意: && 实际上是& &。这里增加了一列水平间距,使得文字能够左对齐。

7.4.7 aligned, gathered, alignedat 环境

Used for an aligned block within a math environment. This can be displayed math or in-line math.

```

1 \begin{equation}
2 \begin{aligned}
3   x + y + z &= 0 \\
4   y - z &= 1 \\
5 \end{aligned}
6 \end{equation}

```

$$\begin{aligned}x + y + z &= 0 \\ y - z &= 1\end{aligned}\tag{7.20}$$

7.5 文本

To insert some text into a formula, standard LaTeX provides the `\mbox` command. `amsmath` offers further commands:

`\text{words}`: inserts text within a math formula. The size is adjusted according to the current math style, that is, `\text` produces smaller text within subscripts or superscripts.

`\intertext{text}`: suspends the formula, the text follows in a separate paragraph, then the multi-line formula is resumed, keeping the alignment. Use it for longer text.

```

1 % intertext
2 \begin{align}
3   ax^2 + bx + c &= 0
4   \intertext{因为  $a \neq 0$ , 两边同时除以  $a$ , 得}
5   x^2 + \frac{b}{a}x + \frac{c}{a} &= 0
6   \intertext{配方得}
7   \left( x + \frac{b}{2a} \right)^2 - \frac{b^2 - 4ac}{4a^2} &= 0
8   \intertext{即}
9   \left( x + \frac{b}{2a} \right)^2 &= \frac{b^2 - 4ac}{4a^2}
10 \end{align}

```

$$ax^2 + bx + c = 0 \quad (7.21)$$

因为 $a \neq 0$, 两边同时除以 a , 得

$$x^2 + \frac{b}{a}x + \frac{c}{a} = 0 \quad (7.22)$$

配方得

$$\left(x + \frac{b}{2a}\right)^2 - \frac{b^2 - 4ac}{4a^2} = 0 \quad (7.23)$$

即

$$\left(x + \frac{b}{2a}\right)^2 = \frac{b^2 - 4ac}{4a^2} \quad (7.24)$$

7.6 Cases

```

1 \begin{gather}
2   \mathcal{S}(x) =
3   \begin{cases}
4     -1 & x < 0 \\
5     0 & x = 0 \\
6     1 & x > 0
7   \end{cases}
8 \end{gather}

```

$$\mathcal{S}(x) = \begin{cases} -1 & x < 0 \\ 0 & x = 0 \\ 1 & x > 0 \end{cases} \quad (7.25)$$

第 8 章 Beamer 演示文稿

8.1 Beamer 主题与颜色

- Beamer 内置主题与颜色预览 [Beamer Theme Matrix](#)
- Metropolis 主题 [Beamer Theme Metropolis](#)
- Solarized 颜色 [Beamer Color Theme Solarized](#)

Beamer 中的 `\pause`, `\pausesections` 和列表动画会产生额外的页面, 这点在打印文稿时需注意.

8.2 设计演示文稿

设计演示文件时应注意:

- Keep time constraints in mind; a frame per minute is a good rule of thumb.
- Use few sections, logically split in subsections; it is better to avoid subsubsections.
- Use self-explanatory titles for sectioning and frames.
- Bulleted lists help to keep things simple.
- Consider avoiding numbering references; one rarely cares about a reference to theorem 2.6 during a talk.
- Don't disrupt the reading flow with footnotes.
- Graphics, such as diagrams, help the audience with visualization.
- Slides should support your talk, not the other way round. Did you already bear with a presentation where the speaker just read aloud the text from the slides and used fancy transition effects? You can do it better.

参考文献

- [1] 刘海洋. *LaTeX*入门. 电子工业出版社, 2013.
- [2] Stefan Kottwitz. *LaTeX Beginner's Guide*. Packt Publishing, 2011.
- [3] Stefan Kottwitz. *LaTeX Cookbook*. Packt Publishing, 2015.

