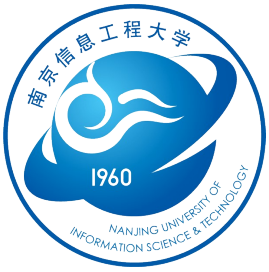
分类号：TP399　　单位代码：10300

密　级：　　学　　号：00000000000

南京信息工程大学

硕 士 学 位 论 文



论文题目：南京信息工程大学毕业论文 LaTeX 模板

申请人姓名：姓　名

指 导 教 师：指导老师

学 科 名 称：专业

研 究 方 向：研究方向

培 养 学 院：学院

提 交 时 间：2021 年 6 月 10 日

二〇二一 年 六 月

独创性声明

本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。本论文除了文中特别加以标注和致谢的内容外，不包含其他人或其他机构已经发表或撰写过的研究成果，也不包含为获得南京信息工程大学或其他教育机构的学位或证书而使用过的材料。其他同志对本研究所做的贡献均已在论文中作了声明并表示谢意。

学位论文作者签名：_____ 签字日期：_____

关于论文使用授权的说明

南京信息工程大学、国家图书馆、中国学术期刊（光盘版）杂志社、中国科学技术信息研究所的《中国学位论文全文数据库》有权保留本人所送交学位论文的复印件和电子文档，可以采用影印、缩印或其他复制手段保存论文，并通过网络向社会提供信息服务。本人电子文档的内容和纸质论文的内容相一致。除在保密期内的保密论文外，允许论文被查阅和借阅，可以公布（包括刊登）论文的全部或部分内容。论文的公布（包括刊登）授权南京信息工程大学研究生院办理。

□ 公开 □ 保密（_____ 年_____ 月）(保密的学位论文在解密后应遵守此协议)

学位论文作者签名：_____ 签字日期：_____

指 导 教 师 签 名：_____ 签字日期：_____

目　录

摘　要

本项目为非官方的南京信息工程大学硕博毕业大论文的模板，尽可能的做到不引入冗余的包和代码，考虑到有些同学对 LaTeX 更不熟悉，作者提供了详细的注释，并将该模板设为多文件的模式，初学者可以直接在分章节的文件中，填充文章内容，避免在单个文件中不会调整的尴尬。但是，必要的表格、图片和公式的排版知识仍是必须了解的，但是可以通过百度满足你的需求。

关键词：LaTeX，毕业论文模板，南京信息工程大学

Abstract

This project is an inofficial thesis template of Nanjing University of Information Science & Technology. Due to the deficiency of the LaTeX knowledge, this template is integrated from several existing templates with some optimization of the syntax. In addition, the template is set as multi files mode with a lot of comments for those who know little ablout LaTeX. However, syntax about table, graph and formula is necessary to learn except that you pay someone to help you.

Key words: LaTeX, Thesis Template, Nuist

第一章　基本使用

1.1 定义信息

我们在开始排版之间，需要导言区定义一些个人的信息来填充封面，也就是在 \begin{document} 之前，插入以下代码：

Listing 1.1　个人信息定义

```
\information{
  classification=中图分类号,
  studentno=学号,
  degree=学历,
  title=论文 \titlesep 题目,
  name=姓名,
  mentor=导师,
  mentorrank=导师职称,
  major=专业,
  research=研究方向,
  institute=学院,
  year=年,
  month=月,
  day=日,
  keywordscn=关键词1\keysepcn 关键词2\keysepcn 关键词3,
  keywordsen=KeyWord1\keysepen KeyWord2\keysepen KeyWord3
}
```

这样，我们就初步完成了封面信息的填充，注意，\titlesep 只能用于标题的换行，且只能用一次，一般标题长度足够了，\keysepcn 和 \keysepen 分别用于中英文关键词的分隔。

1.2 正文编写

如果排版时只在一个文件内，则修改时会十分不便，建议使用多文件排版，然后通过 \include{file} 引入。正文和致谢的引入位置在参考文献之前，这点需要注意，正确的

位置如下所示：

Listing 1.2　正文引入

```
% 正文开始
\mainmatter

% 具体正文章节
\include{chapters/chapter1}
\include{chapters/chapter2}
\include{chapters/chapter3}
\include{chapters/chapter4}
\include{chapters/chapter5}
\include{chapters/chapter6}
\include{chapters/chapter7}

% 致谢
\include{chapters/thanks}

% 参考文献开始
\addreference
```

在正文内部，可以分四个层级，例如：

Listing 1.3　章节命令使用

```
\chapter{一级标题}
\section{二级标题}
\subsection{三级标题}
\subsubsection{四级标题}
```

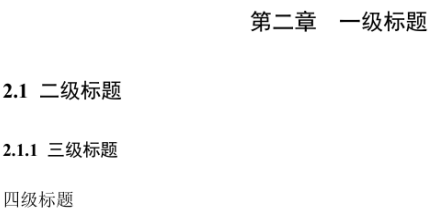


图 1-1　标题格式

1.3 常用命令

1.3.1 分页

分页包裹连续分页和奇数页分页，分别对应命令 \clearpage 和 \cleardoublepage，其中，\clearpage 在空白页不会显示页眉页脚

1.3.2 文本装饰

Listing 1.4 文本装饰

```
\CJKunderdot{汉字，下加点} \\
\CJKunderline{汉字，下划线} \\
\CJKunderdblline{汉字，双下划线} \\
\CJKunderwave{汉字，下波浪线} \\
\CJKsout{汉字，删除线} \\
\CJKxout{汉字，删除线}
```

汉字，下加点

汉字，下划线

汉字，双下划线

汉字，下波浪线

~~汉字，删除线~~

~~汉字，删除线~~

1.3.3 列表环境

LaTeX 提供了三种列表环境：编号的 enumerate 环境，不编号的 itemize 环境和使用关键字的 description 环境。在列表环境内部使用 \item 命令开始一个列表项，他可以带一个可选参数表示手动编号或关键字。

enumerate 环境使用数字自动编号：

Listing 1.5　enumerate 环境

```
\begin{enumerate}
  \item 中文
  \item English
  \item 日本語
\end{enumerate}
```

1. 中文
2. English
3. 日本語

itemize 环境不编号，但会在每个条目前加一个符号以示标记：

Listing 1.6　enumerate 环境

```
\begin{itemize}
  \item 中文
  \item English
  \item 日本語
\end{itemize}
```

- 中文
- English
- 日本語

description 环境总是使用 \item 命令的可选参数，把它作为条目关键字加粗显示：

Listing 1.7　enumerate 环境

```
\begin{description}
    \item[中文] 中国的语言文学
    \item[English] The language of England
    \item[日本語] 日本の言語
\end{description}
```

中文 中国的语言文学

English The language of England

日本語 日本の言語

这几个环境可以嵌套使用（至多四层），LaTeX 会自动处理不同层次的缩进和编号。

第二章　参考文献

2.1 参考文件格式

目前本模板仅有 GB/T 7714-2005 和 GB/T 7714-2015 样式，默认采用 GB/T 7714-2015 样式，如果需要使用其他格式，请自行寻找或编写 *.bst* 文件。

2.2 BibTeX 的使用

模板中有一个 *reference.bib* 文件，是存放参考文献信息的数据库，通过谷歌学术、微软学术或是百度学术，查找文献时，把其 BibTeX 的引用信息复制到这个文件里，在你需要引用的地方，按照如下方式引用：

Listing 2.1　BibTeX 引用参考文献

```
\cite{bibid}  % bibid 就是你复制引文信息的第一个参数
\cite{bibid1,bibid2,bibid2}  % 同时引用多个文献
```

第三章　数学排版

3.1 行内和行间公式

数学公式有两种排版方式：其一是与文字混排，称为行内公式；其二是单独列为一行排版，称为行间公式。

行内公式由一对 $ 符号包裹：

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

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

单独成行的行间公式在 LaTeX 里由 equation 环境包裹。equation 环境为公式自动生成一个编号，这个编号可以用 \label 和 \ref 生成交叉引用，amsmath 的 \eqref 命令甚至为引用自动加上圆括号；还可以用 \tag 命令手动修改公式的编号，或者用 \notag 命令取消为公式编号。

Listing 3.1　多行公式示例一

```
The Pythagorean theorem is:

\begin{equation}
  a^2 + b^2 = c^2 \label{pythagorean}
\end{equation}

Equation \eqref{pythagorean} is called `Gougu theorem' in Chinese.
```

The Pythagorean theorem is:

$$a^2 + b^2 = c^2 \tag{3.1}$$

Equation (3.1) is called ‘Gougu theorem’ in Chinese.

Listing 3.2　多行公式示例二

```
It's wrong to say

\begin{equation}

  1 + 1 = 3 \tag{dumb}

\end{equation}

or

\begin{equation}

  1 + 1 = 4 \notag

\end{equation}
```

It's wrong to say

$$1 + 1 = 3 \tag{dumb}$$

or

$$1 + 1 = 4$$

如果需要直接使用不带编号的行间公式，则将公式用命令 \[和 \] 包裹，与之等效的是 displaymath 环境。有的人更喜欢 equation* 环境，体现了带星号和不带星号的环境之间的区别。

Listing 3.3　多行公式示例三

```
\begin{equation*}

  a^2 + b^2 = c^2

\end{equation*}

For short:

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

Or if you like the long one:

\begin{displaymath}

  a^2 + b^2 = c^2

\end{displaymath}
```

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

For short:

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

Or if you like the long one:

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

我们通过一个例子展示行内公式和行间公式的对比。为了与文字相适应，行内公式在排版大的公式元素（分式、巨算符等）时显得很“局促”：

Listing 3.4　综合示例

```
In text:
$\lim_{n \to \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$.
In display:
\[
  \lim_{n \to \infty}
  \sum_{k=1}^n \frac{1}{k^2}
  = \frac{\pi^2}{6}
\]
```

In text: $\lim_{n\to\infty} \sum_{k=1}^{n} \frac{1}{k^2} = \frac{\pi^2}{6}$.

In display:

$$\lim_{n\to\infty} \sum_{k=1}^{n} \frac{1}{k^2} = \frac{\pi^2}{6}$$

3.2 数学模式

当用户使用 $ 开启行内公式输入，或是使用 \[命令、equation 环境时，LaTeX 就进入了数学模式。数学模式相比于文本模式有以下特点：

1. 数学模式中输入的空格被忽略。数学符号的间距默认由符号的性质（关系符号、运算符等）决定。需要人为引入间距时，使用 \quad 和 \qquad 等命令。
2. 不允许有空行（分段）。行间公式中也无法用 \\ 命令手动换行。排版多行公式需要用到其他公式环境。
3. 所有的字母被当作数学公式中的变量处理，字母间距与文本模式不一致，也无法生成单词之间的空格。

3.3 数学符号

3.3.1 一般符号

表 3-1　希腊字母

\Alpha，\Beta 等希腊字母符号不存在，因为它们和拉丁字母 A,B 等一模一样；小写字母里也不存在 \omicron，直接用拉丁字母 o 代替。

α	\alpha	θ	\theta	o	o	υ	\upsilon
β	\beta	ϑ	\vartheta	π	\pi	ϕ	\phi
γ	\gamma	ι	\iota	ϖ	\varpi	φ	\varphi
δ	\delta	κ	\kappa	ρ	\rho	χ	\chi
ϵ	\epsilon	λ	\lambda	ϱ	\varrho	ψ	\psi
ε	\varepsilon	μ	\mu	σ	\sigma	ω	\omega
ζ	\zeta	ν	\nu	ς	\varsigma		
η	\eta	ξ	\xi	τ	\tau		
Γ	\Gamma	Λ	\Lambda	Σ	\Sigma	Ψ	\Psi
Δ	\Delta	Ξ	\Xi	Υ	\Upsilon	Ω	\Omega
Θ	\Theta	Π	\Pi	Φ	\Phi		
$\varGamma$	\varGamma	$\varLambda$	\varLambda	$\varSigma$	\varSigma	$\varPsi$	\varPsi
$\varDelta$	\varDelta	$\varXi$	\varXi	$\varUpsilon$	\varUpsilon	$\varOmega$	\varOmega
$\varTheta$	\varTheta	$\varPi$	\varPi	$\varPhi$	\varPhi		

表 3-2　其他符号

…	\dots	⋯	\cdots	⋮	\vdots	⋱	\ddots
$\hbar$	\hbar	$\imath$	\imath	$\jmath$	\jmath	ℓ	\ell
$\Re$	\Re	$\Im$	\Im	$\aleph$	\aleph	$\wp$	\wp
$\forall$	\forall	$\exists$	\exists	$\mho$	\mho$^{\ell}$	∂	\partial
$'$	'	$\prime$	\prime	$\emptyset$	\emptyset	∞	\infty
∇	\nabla	$\triangle$	\triangle	$\Box$	\Box$^{\ell}$	$\Diamond$	\Diamond$^{\ell}$
$\bot$	\bot	$\top$	\top	$\angle$	\angle	$\surd$	\surd
$\diamondsuit$	\diamondsuit	$\heartsuit$	\heartsuit	$\clubsuit$	\clubsuit	$\spadesuit$	\spadesuit
$\neg$	\neg or \lnot	$\flat$	\flat	$\natural$	\natural	$\sharp$	\sharp

3.3.2 指数、上下标和导数

在 LaTeX 中一般用 ^ 和 _ 标明上下标，如果上下标不是一个字符需要使用花括号包裹，否则只对上下标符号后第一个字符起作用。

导数符号 ' (′) 是一类特殊的上标，可以适当连用表示多阶导数，也可以在其后连用上标：

```
$f(x) = x^2 \quad f'(x) = 2x \quad f''^{2}(x) = 4$
```

$$f(x) = x^2 \quad f'(x) = 2x \quad f''^{2}(x) = 4$$

3.3.3 分式和根式

分式使用 \frac{分子}{分母} 来书写。分式的大小在行间公式中是正常大小，而在行内被极度压缩。

In display style:

$$3/8 \qquad \frac{3}{8} \qquad \tfrac{3}{8}$$

In text style: $1\frac{1}{2}$ hours $1\frac{1}{2}$ hours

一般的根式使用 \sqrt{...}；表示 n 次方根时写成 \sqrt[n]{...}.

$$\sqrt{x} \Leftrightarrow x^{1/2} \quad \sqrt[3]{2} \quad \sqrt{x^2+\sqrt{y}}$$

特殊的分式形式，如二项式结构，由 amsmath 宏包的 \binom 命令生成：

```
Pascal's rule is
\[
  \binom{n}{k} =\binom{n-1}{k} + \binom{n-1}{k-1}
\]
```

Pascal's rule is

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

3.3.4 关系符

LaTeX 常见的关系符号如下表所示,同时还提供了自定义二元关系符的命令 \stackrel，用于将一个符号代价在原有的二元关系符之上：

```
\[
  f_n(x) \stackrel{*}{\approx} 1
\]
```

$$f_n(x) \stackrel{*}{\approx} 1$$

表 3-3　二元关系符

所有的二元关系符都可以加 \not 前缀得到相反意义的关系符，例如 \not= 就得到不等号（同 \ne）。

$<$	<	$>$	>	$=$	=
$\leq$	\leq or \le	$\geq$	\geq or \ge	$\equiv$	\equiv
$\ll$	\ll	$\gg$	\gg	$\doteq$	\doteq
$\prec$	\prec	$\succ$	\succ	$\sim$	\sim
$\preceq$	\preceq	$\succeq$	\succeq	$\simeq$	\simeq
$\subset$	\subset	$\supset$	\supset	$\approx$	\approx
$\subseteq$	\subseteq	$\supseteq$	\supseteq	$\cong$	\cong
$\sqsubset^{\ell}$	\sqsubset$^{\ell}$	$\sqsupset^{\ell}$	\sqsupset$^{\ell}$	$\bowtie$	\Join$^{\ell}$
$\sqsubseteq$	\sqsubseteq	$\sqsupseteq$	\sqsupseteq	$\bowtie$	\bowtie
$\in$	\in	$\ni$	\ni, \owns	$\propto$	\propto
$\vdash$	\vdash	$\dashv$	\dashv	$\models$	\models
$\mid$	\mid	$\parallel$	\parallel	$\perp$	\perp
$\smile$	\smile	$\frown$	\frown	$\asymp$	\asymp
$:$	:	$\notin$	\notin	$\neq$	\neq or \ne

表 3-4 $\mathcal{AMS}$ 二元关系符

⋖	\lessdot	⋗	\gtrdot	≒	\doteqdot
⩽	\leqslant	⩾	\geqslant	≓	\risingdotseq
⪕	\eqslantless	⪖	\eqslantgtr	≒	\fallingdotseq
≦	\leqq	≧	\geqq	≖	\eqcirc
⋘	\lll or \llless	⋙	\ggg	≗	\circeq
≲	\lesssim	≳	\gtrsim	≜	\triangleq
≲	\lessapprox	≳	\gtrapprox	≏	\bumpeq
≶	\lessgtr	≷	\gtrless	≎	\Bumpeq
⋚	\lesseqgtr	⋛	\gtreqless	∼	\thicksim
⪋	\lesseqqgtr	⪌	\gtreqqless	≈	\thickapprox
≼	\preccurlyeq	≽	\succcurlyeq	≊	\approxeq
≺	\curlyeqprec	≻	\curlyeqsucc	∽	\backsim
≾	\precsim	≿	\succsim	≌	\backsimeq
⪷	\precapprox	⪸	\succapprox	⊨	\vDash
⊆	\subseteqq	⊇	\supseteqq	⊩	\Vdash
∥	\shortparallel	⊃	\Supset	⊪	\Vvdash
◀	\blacktriangleleft	⊐	\sqsupset	∍	\backepsilon
▷	\vartriangleright	∵	\because	∝	\varpropto
▶	\blacktriangleright	⋐	\Subset	≬	\between
⊵	\trianglerighteq	⌢	\smallfrown	⋔	\pitchfork
◁	\vartriangleleft	∣	\shortmid	⌣	\smallsmile
⊴	\trianglelefteq	∴	\therefore	⊏	\sqsubset

3.3.5 算符

表 3-5　二元运算符

$+$	+	$-$	-		
$\pm$	\pm	$\mp$	\mp	$\triangleleft$	\triangleleft
$\cdot$	\cdot	$\div$	\div	$\triangleright$	\triangleright
$\times$	\times	$\setminus$	\setminus	$\star$	\star
$\cup$	\cup	$\cap$	\cap	$*$	\ast
$\sqcup$	\sqcup	$\sqcap$	\sqcap	$\circ$	\circ
$\vee$	\vee, \lor	$\wedge$	\wedge,\land	$\bullet$	\bullet
$\oplus$	\oplus	$\ominus$	\ominus	$\diamond$	\diamond
$\odot$	\odot	$\oslash$	\oslash	$\uplus$	\uplus
$\otimes$	\otimes	$\bigcirc$	\bigcirc	$\amalg$	\amalg
$\bigtriangleup$	\bigtriangleup	$\bigtriangledown$	\bigtriangledown	$\dagger$	\dagger
$\triangleleft$	\lhd$^\ell$	$\triangleright$	\rhd$^\ell$	$\ddagger$	\ddagger
$\trianglelefteq$	\unlhd$^\ell$	$\trianglerighteq$	\unrhd$^\ell$	$\wr$	\wr

表 3-6　$\mathcal{AMS}$ 二元运算符

$\dot{+}$	\dotplus	$\centerdot$	\centerdot		
$\ltimes$	\ltimes	$\rtimes$	\rtimes	$\divideontimes$	\divideontimes
$\Cup$	\doublecup	$\Cap$	\doublecap	$\smallsetminus$	\smallsetminus
$\veebar$	\veebar	$\barwedge$	\barwedge	$\doublebarwedge$	\doublebarwedge
$\boxplus$	\boxplus	$\boxminus$	\boxminus	$\circleddash$	\circleddash
$\boxtimes$	\boxtimes	$\boxdot$	\boxdot	$\circledcirc$	\circledcirc
$\intercal$	\intercal	$\circledast$	\circledast	$\rightthreetimes$	\rightthreetimes
$\curlyvee$	\curlyvee	$\curlywedge$	\curlywedge	$\leftthreetimes$	\leftthreetimes

∇(\nabla) 和 ∂(\partial) 也是常用的算符，虽然它们不属于二元算符。LaTeX 将数学函数的名称作为一个算符排版，字体为直立字体。

表 3-7　LaTeX 作为算符的函数名称一览

不带上下限的算符				
\sin	\arcsin	\sinh	\exp	\dim
\cos	\arccos	\cosh	\log	\ker
\tan	\arctan	\tanh	\lg	\hom
\cot	\arg	\coth	\ln	\deg
\sec	\csc			
带上下限的算符				
\lim	\limsup	\liminf	\sup	\inf
\min	\max	\det	\Pr	\gcd

```
\[
  \lim_{x \rightarrow 0}
  \frac{\sin x}{x}=1
\]
```

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

对于求模表达式，LaTeX 提供了 \bmod 和 \pmod 命令，前者相当于一个二元运算符，后者作为同余表达式的后缀：

```
$a\bmod b \\
x\equiv a \pmod{b}$
```

$$a \bmod b$$

$$x \equiv a \pmod{b}$$

amsmath 还允许用户在导言区用 \DeclareMathOperator定义自己的算符，其中带星号的命令定义带上下限的算符：

\DeclareMathOperator{\argh}{argh}
\DeclareMathOperator*{\nut}{Nut}

```
\[\argh 3 = \nut_{x=1} 4x\]
```

$$\operatorname{argh} 3 = \operatorname*{Nut}_{x=1} 4x$$

3.3.6 巨算符

积分号 $\int$(\int)、求和号 $\sum$ (\sum) 等符号称为巨算符。巨算符在行内公式和行间公式的大小和形状有区别。

```
In text:
$\sum_{i=1}^n \quad
\int_0^{\frac{\pi}{2}} \quad
\oint_0^{\frac{\pi}{2}} \quad
\prod_\epsilon $ \\
In display:
\[\sum_{i=1}^n \quad
\int_0^{\frac{\pi}{2}} \quad
\oint_0^{\frac{\pi}{2}} \quad
\prod_\epsilon \]
```

In text: $\sum_{i=1}^{n} \quad \int_0^{\frac{\pi}{2}} \quad \oint_0^{\frac{\pi}{2}} \quad \prod_{\epsilon}$

In display:

$$\sum_{i=1}^{n} \quad \int_0^{\frac{\pi}{2}} \quad \oint_0^{\frac{\pi}{2}} \quad \prod_{\epsilon}$$

巨算符的上下标位置可由 \limits 和 \nolimits 调整，前者令巨算符类似 lim 或求和算符 $\sum$，上下标位于上下方；后者令巨算符类似积分号，上下标位于右上方和右下方。

```
In text:
$\sum\limits_{i=1}^n \quad
\int\limits_0^{\frac{\pi}{2}} \quad
\prod\limits_\epsilon $ \\
In display:
\[\sum\nolimits_{i=1}^n \quad
\int\limits_0^{\frac{\pi}{2}} \quad
\prod\nolimits_\epsilon \]
```

In text: $\sum_{i=1}^{n} \int_{0}^{\frac{\pi}{2}} \prod_{\epsilon}$

In display:

$$\sum_{i=1}^{n} \int_{0}^{\frac{\pi}{2}} \prod_{\epsilon}$$

amsmath 宏包还提供了 \substack，能够在下限位置书写多行表达式；subarray 环境更进一步，令多行表达式可选择居中 (c) 或左对齐 (l)：

```
\[
\sum_{\substack{0\le i\le n \\
j\in \mathbb{R}}}
P(i,j) = Q(n)
\]
\[
\sum_{\begin{subarray}{l}
0\le i\le n \\
j\in \mathbb{R}
\end{subarray}}
P(i,j) = Q(n)
\]
```

$$\sum_{\substack{0\le i\le n \\ j\in\mathbb{R}}} P(i,j) = Q(n)$$

$$\sum_{\begin{subarray}{l} 0\le i\le n \\ j\in\mathbb{R} \end{subarray}} P(i,j) = Q(n)$$

表 3-8　巨算符

$\sum$	$\sum$	\sum	$\bigcup$	$\bigcup$	\bigcup	$\bigvee$	$\bigvee$	\bigvee
$\prod$	$\prod$	\prod	$\bigcap$	$\bigcap$	\bigcap	$\bigwedge$	$\bigwedge$	\bigwedge
$\coprod$	$\coprod$	\coprod	$\bigsqcup$	$\bigsqcup$	\bigsqcup	$\biguplus$	$\biguplus$	\biguplus
$\int$	$\int$	\int	$\oint$	$\oint$	\oint	$\bigodot$	$\bigodot$	\bigodot
$\bigoplus$	$\bigoplus$	\bigoplus	$\bigotimes$	$\bigotimes$	\bigotimes			
$\iint$	$\iint$	\iint	$\iiint$	$\iiint$	\iiint	$\iiiint$	$\iiiint$	\iiiint
$\int\cdots\int$	$\int\cdots\int$	\idotsint						

3.3.7 数学重音和上下括号

表 3-9　数学重音符号

最后一个 \wideparen 依赖 yhmath 宏包。

$\hat{a}$	\hat{a}	$\check{a}$	\check{a}	$\tilde{a}$	\tilde{a}
$\acute{a}$	\acute{a}	$\grave{a}$	\grave{a}	$\breve{a}$	\breve{a}
$\bar{a}$	\bar{a}	$\vec{a}$	\vec{a}	$\mathring{a}$	\mathring{a}
$\dot{a}$	\dot{a}	$\ddot{a}$	\ddot{a}	$\dddot{a}$	\dddot{a}
$\ddddot{a}$	\ddddot{a}				
$\widehat{AAA}$	\widehat{AAA}	$\widetilde{AAA}$	\widetilde{AAA}	$\wideparen{AAA}$	\wideparen{AAA}

表 3-10　作为重音的箭头符号

$\overrightarrow{AB}$	\overrightarrow{AB}	$\underrightarrow{AB}$	\underrightarrow{AB}
$\overleftarrow{AB}$	\overleftarrow{AB}	$\underleftarrow{AB}$	\underleftarrow{AB}
$\overleftrightarrow{AB}$	\overleftrightarrow{AB}	$\underleftrightarrow{AB}$	\underleftrightarrow{AB}

\overbrace 和 \underbrace 命令用来生成上/下括号，各自可带一个上/下标公式。

```
$\underbrace{\overbrace{(a+b+c)}^6
\cdot \overbrace{(d+e+f)}^7}
_\text{meaning of life} = 42$
```

$$\underbrace{\overbrace{(a+b+c)}^{6}\cdot\overbrace{(d+e+f)}^{7}=42}_{\text{meaning of life}}$$

3.3.8 箭头

常用的箭头包括 \rightarrow (→，或 \to)、\leftarrow（←，或 \gets）等。

表 3-11 箭头

$\leftarrow$	\leftarrow or \gets	$\longleftarrow$	\longleftarrow
$\rightarrow$	\rightarrow or \to	$\longrightarrow$	\longrightarrow
$\leftrightarrow$	\leftrightarrow	$\longleftrightarrow$	\longleftrightarrow
$\Leftarrow$	\Leftarrow	$\Longleftarrow$	\Longleftarrow
$\Rightarrow$	\Rightarrow	$\Longrightarrow$	\Longrightarrow
$\Leftrightarrow$	\Leftrightarrow	$\Longleftrightarrow$	\Longleftrightarrow
$\mapsto$	\mapsto	$\longmapsto$	\longmapsto
$\hookleftarrow$	\hookleftarrow	$\hookrightarrow$	\hookrightarrow
$\leftharpoonup$	\leftharpoonup	$\rightharpoonup$	\rightharpoonup
$\leftharpoondown$	\leftharpoondown	$\rightharpoondown$	\rightharpoondown
$\rightleftharpoons$	\rightleftharpoons	$\iff$	\iff
$\uparrow$	\uparrow	$\downarrow$	\downarrow
$\updownarrow$	\updownarrow	$\Uparrow$	\Uparrow
$\Downarrow$	\Downarrow	$\Updownarrow$	\Updownarrow
$\nearrow$	\nearrow	$\searrow$	\searrow
$\swarrow$	\swarrow	$\nwarrow$	\nwarrow
$\rightsquigarrow$	\leadsto$^{\ell}$		

amsmath 的 \xleftarrow 和 \xrightarrow 命令提供了长度可以伸展的箭头，并且可以为箭头增加上下标：

```
\[ a\xleftarrow{x+y+z} b \]

\[ c\xrightarrow[x<y]{a*b*c}d \]
```

$$a \xleftarrow{x+y+z} b$$

$$c \xrightarrow[x<y]{a*b*c} d$$

3.3.9 括号和定界符

表 3-12　定界符

amsmath 还定义了 \lvert、\rvert 和 \lVert、\rVert，分别作为 \vert 和 \Vert 对应的开符号（左侧）和闭符号（右侧）的命令。

(	(	)	)	↑	\uparrow	↓	\downarrow
[	[or \lbrack	]	] or \rbrack	⇑	\Uparrow	⇓	\Downarrow
{	\{ or \lbrace	}	\} or \rbrace	↕	\updownarrow	⇕	\Updownarrow
|	| or \vert	‖	\\| or \Vert	⌈	\lceil	⌉	\rceil
⟨	\langle	⟩	\rangle	⌊	\lfloor	⌋	\rfloor
/	/	\	\backslash				

表 3-13　用于行间公式的大定界符

⎛⎝	\lgroup	⎞⎠	\rgroup	⎰	\lmoustache
⏐	\arrowvert	‖	\Arrowvert	⎪	\bracevert
⎞⎠	\rmoustache				

使用 \left 和 \right 命令可令括号（定界符）的大小可变，在行间公式中常用。LaTeX 会自动根据括号内的公式大小决定定界符大小。\left 和 \right 必须成对使用。需要使用单个定界符时，另一个定界符写成 \left. 或 \right.。

```
\[1 + \left(\frac{1}{1-x^{2}}
\right)^3 \qquad
\left.\frac{\partial f}{\partial t}
\right|_{t=0}\]
```

$$1+\left(\frac{1}{1-x^2}\right)^3 \qquad \left.\frac{\partial f}{\partial t}\right|_{t=0}$$

有时我们不满意于 LaTeX 为我们自动调节的定界符大小。这时我们还可以用 \big、\bigg 等命令生成固定大小的定界符。更常用的形式是类似 \left 的 \bigl、\biggl 等，以及类似 \right 的 \bigr、\biggr 等（\bigl 和 \bigr 不必成对出现）。

```
$\Bigl((x+1)(x-1)\Bigr)^{2}$\\
$\bigl( \Bigl( \biggl( \Biggl( \quad
\bigr\} \Bigr\} \biggr\} \Biggr\} \quad
\big\| \Big\| \bigg\| \Bigg\| \quad
\big\Downarrow \Big\Downarrow
\bigg\Downarrow \Bigg\Downarrow$
```

$$\Big((x+1)(x-1)\Big)^2$$
$$\bigg(\Big(\Big(\Big(\quad\Big\}\Big\}\Big\}\bigg\} \quad \Big\|\Big\|\Big\|\Big\| \quad \Big\Downarrow\Big\Downarrow\Big\Downarrow\Big\Downarrow$$

3.4 多行公式

3.4.1 长公式折行

通常来讲应当避免写出超过一行而需要折行的长公式。如果一定要折行的话，习惯上优先在等号之前折行，其次在加号、减号之前，再次在乘号、除号之前。其它位置应当避免折行。

amsmath 宏包的 multline 环境提供了书写折行长公式的方便环境。它允许用 \\ 折行，将公式编号放在最后一行。多行公式的首行左对齐，末行右对齐，其余行居中。

```
\begin{multline}
  a + b + c + d + e + f
  + g + h + i \\
  = j + k + l + m + n\\
  = o + p + q + r + s\\
  = t + u + v + x + z
\end{multline}
```

$$
\begin{aligned}
a+b+c+d+e+f+g+h+i \\
&= j+k+l+m+n \\
&= o+p+q+r+s \\
&= t+u+v+x+z
\end{aligned} \tag{3.2}
$$

3.4.2 多行公式

更多的情况是，我们需要罗列一系列公式，并令其按照等号对齐。目前最常用的是 align 环境，它将公式用 & 隔为两部分并对齐。分隔符通常放在等号左边：

```
\begin{align}
  a & = b + c \\
    & = d + e
\end{align}
```

$$
\begin{aligned}
a &= b+c && (3.3) \\
&= d+e && (3.4)
\end{aligned}
$$

align 环境会给每行公式都编号。我们仍然可以用 \notag 去掉某行的编号。在以下的例子，为了对齐等号，我们将分隔符放在右侧，并且此时需要在等号后添加一对括号 {} 以产生正常的间距：

```
\begin{align}
a ={} & b + c \\
={} & d + e + f + g + h + i
+ j + k + l \notag \\
& + m + n + o \\
```

```
  ={} & p + q + r + s

  \end{align}
```

$$
\begin{aligned}
a &= b + c && (3.5)\\
&= d + e + f + g + h + i + j + k + l \\
&\quad + m + n + o && (3.6)\\
&= p + q + r + s && (3.7)
\end{aligned}
$$

align 还能够对齐多组公式，除等号前的 & 之外，公式之间也用 & 分隔：

```
  \begin{align}
  a &=1  &  b &=2   & c &=3   \\
  d &=-1 &  e &=-2  & f &=-5
  \end{align}
```

$$
\begin{aligned}
a &= 1 & b &= 2 & c &= 3 && (3.8)\\
d &= -1 & e &= -2 & f &= -5 && (3.9)
\end{aligned}
$$

如果我们不需要按等号对齐，只需罗列数个公式，gather 将是一个很好用的环境：

```
\begin{gather}
a = b + c \\
d = e + f + g \\
h + i = j + k \notag \\
l + m = n
\end{gather}
```

$$a = b + c \tag{3.10}$$

$$d = e + f + g \tag{3.11}$$

$$h + i = j + k$$

$$l + m = n \tag{3.12}$$

3.4.3 公用编号的多行公式

另一个常见的需求是将多个公式组在一起公用一个编号，编号位于公式的居中位置。为此，amsmath 宏包提供了诸如 aligned、gathered 等环境，与 equation 环境套用。以 -ed 结尾的环境用法与前一节不以 -ed 结尾的环境用法一一对应。我们仅以 aligned 举例：

```
\begin{equation}
        \begin{aligned}
                a &= b + c \\
                d &= e + f + g \\
                h + i &= j + k \\
                l + m &= n
        \end{aligned}
\end{equation}
```

$$\begin{aligned} a &= b + c \\ d &= e + f + g \\ h + i &= j + k \\ l + m &= n \end{aligned} \tag{3.13}$$

3.5 数组和矩阵

为了排版二维数组，LaTeX 提供了 array 环境，用法与 tabular 环境极为类似，也需要定义列格式，并用 \\ 换行。数组可作为一个公式块，在外套用 \left、\right 等定界符：

```
\[ \mathbf{X} = \left(
  \begin{array}{cccc}
  x_{11} & x_{12} & \ldots & x_{1n}\\
  x_{21} & x_{22} & \ldots & x_{2n}\\
  \vdots & \vdots & \ddots & \vdots\\
  x_{n1} & x_{n2} & \ldots & x_{nn}\\
  \end{array}
\right) \]
```

$$\mathbf{X} = \left(\begin{array}{cccc} x_{11} & x_{12} & \ldots & x_{1n} \\ x_{21} & x_{22} & \ldots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \ldots & x_{nn} \end{array} \right)$$

我们还可以利用空的定界符排版出这样的效果：

```
\[ |x| = \left\{
  \begin{array}{rl}
    -x & \text{if } x < 0,\\
    0 & \text{if } x = 0,\\
    x & \text{if } x > 0.
  \end{array} \right.
\]
```

$$|x| = \left\{ \begin{array}{rl} -x & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ x & \text{if } x > 0. \end{array} \right.$$

不过上述例子可以用 amsmath 提供的 cases 环境更轻松地完成：

```
\[ |x| =
  \begin{cases}
    -x & \text{if } x < 0,\\
    0 & \text{if } x = 0,\\
    x & \text{if } x > 0.
  \end{cases}
\]
```

$$|x| = \begin{cases} -x & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ x & \text{if } x > 0. \end{cases}$$

我们当然也可以用 array 环境排版各种矩阵。amsmath 宏包还直接提供了多种排版矩阵的环境，包括不带定界符的 matrix，以及带各种定界符的矩阵 pmatrix（$\left(\right.$）、bmatrix（$\left[\right.$）、Bmatrix（$\left\{\right.$）、vmatrix（$\left|\right.$）、Vmatrix（$\left\|\right.$）。使用这些环境时，无需给定列格式：

```
\[
\begin{matrix}
1 & 2 \\ 3 & 4
\end{matrix} \qquad
\begin{bmatrix}
x_{11} & x_{12} & \ldots & x_{1n}\\
x_{21} & x_{22} & \ldots & x_{2n}\\
\vdots & \vdots & \ddots & \vdots\\
x_{n1} & x_{n2} & \ldots & x_{nn}\\
\end{bmatrix}
\]
```

$$\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix} \quad \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nn} \end{bmatrix}$$

3.6 公式中的间距

绝大部分时候，数学公式中各元素的间距是根据符号类型自动生成的，需要我们手动调整的情况极少。我们已经认识了两个生成间距的命令 \quad 和 \qquad。在公式中我们还可能用到的间距包括 \,、\:、\; 以及负间距 \!，其中 \quad 、\qquad 和 \, 在文本和数学环境中可用，后三个命令只用于数学环境。文本中的 \␣ 也能使用在数学公式中。

无额外间距		aa	\,	␣	$a\,a$
\quad	␣␣	$a\quad a$	\:	␣	$a\:a$
\qquad	␣␣␣␣	$a\qquad a$	\;	␣	$a\;a$
\␣	␣	$a\ a$	\!	−␣	$a!a$

一个常见的用途是修正积分的被积函数 $f(x)$ 和微元 $\mathrm{d}x$ 之间的距离。注意微元里的 d 用的是直立体：

```
\[
\int_a^b f(x)\mathrm{d}x
\qquad
\int_a^b f(x)\,\mathrm{d}x
\]
```

$$\int_a^b f(x)\mathrm{d}x \qquad \int_a^b f(x)\,\mathrm{d}x$$

3.7 数学符号的字体控制

3.7.1 数学字母字体

LaTeX 允许一部分数学符号切换字体，主要是拉丁字母、数字、大写希腊字母以及重音符号等。表 3-14 给出了切换字体的命令。某一些命令需要字体宏包的支持。

表 3-14　数学字母字体

示例	命令	依赖的宏包
$ABCDEabcde1234$	\mathnormal{…}	
ABCDEabcde1234	\mathrm{…}	
ABCDEabcde1234	\mathit{…}	
ABCDEabcde1234	\mathbf{…}	
ABCDEabcde1234	\mathsf{…}	
ABCDEabcde1234	\mathtt{…}	
$\mathcal{ABCDE}$	\mathcal{…}	仅提供大写字母
$\mathcal{ABCDE}$	\mathcal{…}	eucal 仅提供大写字母
$\mathscr{ABCDE}$	\mathscr{…}	mathrsfs 仅提供大写字母
$\mathfrak{ABCDEabcde1234}$	\mathfrak{…}	amssymb 或 eufrak
$\mathbb{ABCDE}$	\mathbb{…}	amssymb 仅提供大写字母

3.7.2 数学符号的尺寸

数学符号按照符号排版的位置规定尺寸，从大到小包括行间公式尺寸、行内公式尺寸、上下标尺寸、次级上下标尺寸。除了字号有别之外，行间和行内公式尺寸下的巨算符也使用不一样的大小。LaTeX 为每个数学尺寸指定了一个切换的命令，见表 3-15。

表 3-15　数学符号尺寸

命令	尺寸	示例
\displaystyle	行间公式尺寸	$\sum a$
\textstyle	行内公式尺寸	$\sum a$
\scriptstyle	上下标尺寸	a
\scriptscriptstyle	次级上下标尺寸	a

我们通过以下示例对比行间公式和行内公式的区别。在分式中，分子分母默认为行内公式尺寸，示例中将分母切换到行间公式尺寸：

```
\[
  r = \frac
  {\sum_{i=1}^n (x_i- x)(y_i- y)}
  {\displaystyle \left[
  \sum_{i=1}^n (x_i-x)^2
  \sum_{i=1}^n (y_i-y)^2
  \right]^{1/2} }
\]
```

$$r = \frac{\sum_{i=1}^{n}(x_i - x)(y_i - y)}{\left[\sum_{i=1}^{n}(x_i - x)^2 \sum_{i=1}^{n}(y_i - y)^2\right]^{1/2}}$$

第四章　插入图片

4.1 基本使用

LaTeX 本身不支持插图功能，需要由 graphicx 宏包辅助支持。在调用了 graphicx 宏包以后，就可以使用 \includegraphics 命令加载图片了：

```
\includegraphics[<options>]{<filename>}
```

其中 ⟨*filename*⟩ 为图片文件名，与 \include 命令的用法类似，文件名可能需要用相对路径或绝对路径表示。图片文件的扩展名一般可不写。另外一定要注意，文件名里既不要有空格（类似 \include），也不要有多余的英文点号，否则宏包在解析文件名的过程中会出错。

另外 graphicx 宏包还提供了 \graphicspath 命令，用于声明一个或多个图片文件存放的目录，使用这些目录里的图片时可不用写路径：

```
% 假设主要的图片放在 figures 子目录下，标志放在 logo 子目录下
\graphicspath{{figures/}{logo/}}
```

\includegraphics 命令的可选参数 ⟨*options*⟩ 支持 ⟨*key*⟩=⟨*value*⟩ 形式赋值，常用的参数如下：

表 4-1　\includegraphics 命令的可选参数

参数	含义
width=⟨*width*⟩	将图片缩放到宽度为 ⟨*width*⟩
height=⟨*height*⟩	将图片缩放到高度为 ⟨*height*⟩
scale=⟨*scale*⟩	将图片相对于原尺寸缩放 ⟨*scale*⟩ 倍
angle=⟨*angle*⟩	令图片逆时针旋转 ⟨*angle*⟩ 度

4.2 排版多行多列图片

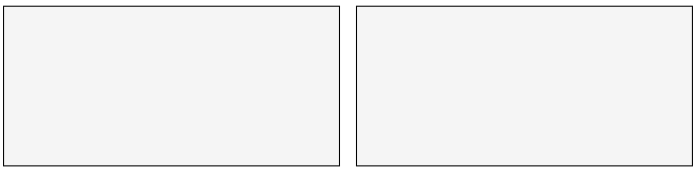
这里给个离职，模仿写就好：

```
\begin{figure}[htbp]
  \centering
  \includegraphics[width=...]{...}
  \qquad
  \includegraphics[width=...]{...} \\[..pt]
  \includegraphics[width=...]{...}
  \caption{...}
  \label{...}
\end{figure}
```

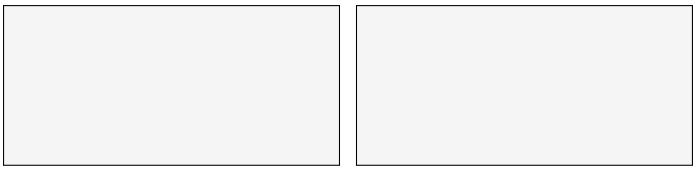
图 4-1　并排放置图片的示意。

或者使用这种方式，个人比较喜欢下面这种：

```
\begin{figure}[t]
  \centering
  \subfloat[图1]{
    \label{fig1}
    \includegraphics[width=0.5\textwidth]{图1}
  }
  \subfloat[图2]{
    \label{fig2}
    \includegraphics[width=0.5\textwidth]{图2}
  }\\
  \subfloat[图3]{
    \label{fig3}
    \includegraphics[width=0.5\textwidth]{图3}
  }
  \subfloat[图4]{
    \label{fig4}
    \includegraphics[width=0.5\textwidth]{图4}
  }
  \caption{多行多列子图}
  \label{fig}
\end{figure}
```



(a) 图 1　　(b) 图 2



(c) 图 3　　(d) 图 4

图 4-2　多行多列子图

第五章　插入表格

LaTeX 里排版表格不如 Word 等所见即所得的工具简便和自由，不过对于不太复杂的表格来讲，完全能够胜任。

排版表格最基本的 tabular 环境用法为：

```
\begin{tabular}[align]{column-spec}
  <item1> & <item2> & ... \\
  \hline
  <item1> & <item2> & ... \\
\end{tabular}
```

其中 ⟨*column-spec*⟩ 是列格式标记，在接下来的内容将仔细介绍；& 用来分隔单元格；\\ 用来换行；\hline 用来在行与行之间绘制横线。

直接使用 tabular 环境的话，会和周围的文字混排。此时可用一个可选参数 ⟨*align*⟩ 控制垂直对齐：t 和 b 分别表示按表格顶部、底部对齐，其他参数或省略不写（默认）表示居中对齐。

```
\begin{tabular}{|c|}
center-\\ aligned \\
\end{tabular},

\begin{tabular}[t]{|c|}
top-\\ aligned \\
\end{tabular},

\begin{tabular}[b]{|c|}
bottom-\\ aligned\\
\end{tabular} tabulars.
```

center- aligned	,	top- aligned	,	bottom- aligned	tabulars.

但是通常情况下 tabular 环境很少与文字直接混排，而是会放在 table 浮动体环境中，并用 \caption 命令加标题。

5.1 列格式

tabular 环境使用 ⟨*column-spec*⟩ 参数指定表格的列数以及每列的格式。

表 5-1 LaTeX 表格列格式

列格式	说明
l/c/r	单元格内容左对齐/居中/右对齐，不折行
p{⟨*width*⟩}	单元格宽度固定为 ⟨*width*⟩，可自动折行
|	绘制竖线
@{⟨*string*⟩}	自定义内容 ⟨*string*⟩

```
\begin{tabular}{lcr|p{6em}}
  \hline
  left & center & right
  & par box with fixed width\\
  L & C & R & P \\
  \hline
\end{tabular}
```

left	center	right	par box with fixed width
L	C	R	P

表格中每行的单元格数目不能多于列格式里 l/c/r/p 的总数（可以少于这个总数），否则出错。

@ 格式可在单元格前后插入任意的文本，但同时它也消除了单元格前后额外添加的间距。@ 格式可以适当使用以充当“竖线”。特别地，@{} 可直接用来消除单元格前后的间距：

```
\begin{tabular}{@{} r@{:}lr @{}}
  \hline
  1  & 1 & one \\
  11 & 3 & eleven \\
  \hline
\end{tabular}
```

1:1	one
11:3	eleven

另外 LaTeX 还提供了简便的将格式参数重复的写法 *{⟨*n*⟩}{⟨*column-spec*⟩}，比如以下两种写法是等效的：

\begin{tabular}{|c|c|c|c|c|p{4em}|p{4em}|}
\begin{tabular}{|*{5}{c|}*{2}{p{4em}|}}

有时需要为整列修饰格式，比如整列改变为粗体，如果每个单元格都加上 \bfseries 命令会比较麻烦。array 宏包提供了辅助格式 > 和 <，用于给列格式前后加上修饰命令：

```
% \usepackage{array}
\begin{tabular}{>{\itshape}r<{*}l}
  \hline
  italic & normal \\
  column & column \\
  \hline
\end{tabular}
```

*italic**	normal
*column**	column

辅助格式甚至支持插入 \centering 等命令改变 p 列格式的对齐方式，一般还要加额外的命令 \arraybackslash 以免出错。

```
% \usepackage{array}
\begin{tabular}{>{\centering\arraybackslash}p{9em}}
  \hline
  Some center-aligned long text. \\
  \hline
\end{tabular}
```

Some center-aligned long text.

array 宏包还提供了类似 p 格式的 m 格式和 b 格式，三者分别在垂直方向上靠顶端对齐、居中以及底端对齐。

```
% \usepackage{array}
\newcommand\txt{a b c d e f g h i}
\begin{tabular}{cp{2em}m{2em}b{2em}}
  \hline
  pos & \txt & \txt & \txt \\
  \hline
\end{tabular}
```

			a b c
		a b c	d e f
pos	a b c	d e f	g h i
	d e f	g h i	
	g h i		

5.2 列宽

在控制列宽方面，LaTeX 表格有着明显的不足：l/c/r 格式的列宽是由文字内容的自然宽度决定的，而 p 格式给定了列宽却不好控制对齐（可用 array 宏包的辅助格式），更何况列与列之间通常还有间距，所以直接生成给定总宽度的表格并不容易。

tabularx 宏包为我们提供了方便的解决方案。它引入了一个 X 列格式，类似 p 列格式，不过会根据表格宽度自动计算列宽，多个 X 列格式平均分配列宽。X 列格式也可以用 array 里的辅助格式修饰对齐方式：

```
% \usepackage{array,tabularx}
\begin{tabularx}{14em}{|*{4}{>{\centering\arraybackslash}X|}}
  \hline
  A & B & C & D \\ \hline
  a & b & c & d \\ \hline
\end{tabularx}
```

A	B	C	D
a	b	c	d

5.3 横线

我们已经在之前的例子见过许多次绘制表格线的 \hline 命令。另外 \cline{⟨*i*⟩-⟨*j*⟩} 用来绘制跨越部分单元格的横线：

```
\begin{tabular}{|c|c|c|}
  \hline
  4 & 9 & 2 \\ \cline{2-3}
  3 & 5 & 7 \\ \cline{1-1}
  8 & 1 & 6 \\ \hline
\end{tabular}
```

4	9	2
3	5	7
8	1	6

在科技论文排版中广泛应用的表格形式是三线表，形式干净简明。三线表由 booktabs 宏包支持，它提供了 \toprule、\midrule 和 \bottomrule 命令用以排版三线表的三条线，以及和 \cline 对应的 \cmidrule。除此之外，最好不要用其它横线以及竖线：

```
% \usepackage{booktabs}
\begin{tabular}{cccc}
  \toprule
  & \multicolumn{3}{c}{Numbers} \\
  \cmidrule{2-4}
  & 1 & 2 & 3 \\
  \midrule
  Alphabet & A & B & C \\
  Roman    & I & II& III \\
  \bottomrule
\end{tabular}
```

	Numbers		
	1	2	3
Alphabet	A	B	C
Roman	I	II	III

5.4 合并单元格

LaTeX 是一行一行排版表格的，横向合并单元格较为容易，由 \multicolumn 命令实现：

```
\multicolumn{<n>}{<column-spec>}{<item>}
```

其中 ⟨*n*⟩ 为要合并的列数，⟨*column-spec*⟩ 为合并单元格后的列格式，只允许出现一个 l/c/r 或 p 格式。如果合并前的单元格前后带表格线 |，合并后的列格式也要带 | 以使得表格的竖线一致。

```
\begin{tabular}{|c|c|c|}
  \hline
  1 & 2 & Center \\ \hline
  \multicolumn{2}{|c|}{3} & \multicolumn{1}{r|}{Right} \\
  \hline
  4 & \multicolumn{2}{c|}{C} \\
  \hline
\end{tabular}
```

1	2	Center
3		Right
4	C	

上面的例子还体现了，形如 \multicolumn{1}{⟨*column-spec*⟩}{⟨*item*⟩} 的命令**可以用来修改某一个单元格的列格式。**

纵向合并单元格需要用到 multirow 宏包提供的 \multirow 命令：

```
\multirow{<n>}{<width>}{<item>}
```

⟨*width*⟩ 为合并后单元格的宽度，可以填 * 以使用自然宽度。

我们看一个结合 \cline、\multicolumn 和 \multirow 命令的例子：

```
% \usepackage{multirow}
\begin{tabular}{ccc}
  \hline
  \multirow{2}{*}{Item} & \multicolumn{2}{c}{Value} \\ \cline{2-3}
  & First & Second  \\ \hline
  A & 1 & 2 \\  \hline
\end{tabular}
```

Item	Value	
	First	Second
A	1	2

5.5 行距控制

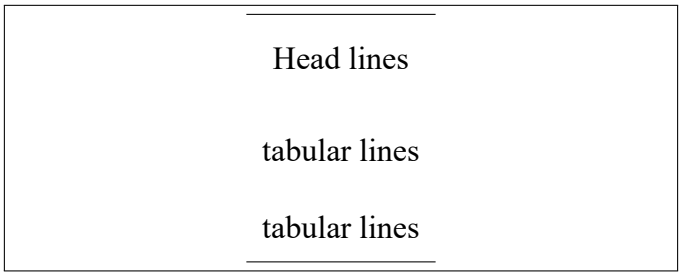
LaTeX 生成的表格看起来通常比较紧凑。修改参数 \arraystretch 可以得到行距更加宽松的表格：

```
\renewcommand\arraystretch{1.8}
\begin{tabular}{|c|}
  \hline
  Really loose \\ \hline
  tabular rows.\\ \hline
\end{tabular}
```

Really loose
tabular rows.

另一种增加间距的办法是给换行命令 \\ 添加可选参数，在这一行下面加额外的间距，适合用于在行间不加横线的表格：

```
\begin{tabular}{c}
  \hline
  Head lines \\[6pt]
  tabular lines \\
  tabular lines \\ \hline
\end{tabular}
```



但是这种换行方式的存在导致了一个缺陷——**表格的首个单元格不能直接使用中括号 [ ]**，否则 \\ 往往会将下一行的中括号当作自己的可选参数，因而出错。如果要使用中括号，应当放在花括号 {} 里面。或者也可以选择将换行命令写成 \\[0pt]。

致　谢

本模板编写参考了《LaTeX 入门》[1]，《简单高效 LaTeX》和《LaTeX 范例学习与试卷论文排版》等。

参考文献

[1] 刘海洋. Latex 入门[J]. 电子工业出版社, 北京, 2013.

作者简介

作者是计算机与软件学院软件工程专业的一名研究生，闲的蛋疼时候就爱瞎折腾了，要是能请我吃顿饭就好了。