

宏包 xjtlumath 文档

贺官羽铭

2021 年 6 月 13 日

Copyright

Copyright © Guanyuming He 2021-2021. The document is licensed under the MIT license.

目录

第一部分 基本信息	ii
1 这宏包是干嘛的？	ii
2 历史	ii
3 中文版历史	iii
第二部分 教程	1
4 教程 1: Ashley 的第一份资料	1
4.1 安装和配置	1
4.2 基本书写	1
4.3 调整字体	3
4.4 键入数学公式	5
4.5 空白管理	7
4.6 列表和其他环境	7
5 教程 2: Delilah 和复杂的数学公式	11
5.1 多行公式	11
5.2 矩阵	13
5.3 文字和运算符	15
5.4 符号	16
6 教程 3: 月处理 floats	18
6.1 插入图像	18
6.2 表	19
6.3 放置 floats	21
6.4 Floats 的目录	22
6.5 关于图的一些建议	22

第一部分 基本信息

1 这宏包是干嘛的？

我最初这个 \LaTeX 包最初是为西浦数学社的资料部使用的。然而我发现我的大学缺乏学生的 \LaTeX 模板，因此我决定将其扩展为其它学生也可以使用的模板。

该宏包包含一些用于数学文档的有用命令和环境，并重新定义了一些现有样式，使其适合我们大学的样式。

2 历史

2020 年，我加入了西浦数学社的资料部。该部门制作有关数学的资料并将其分发给学生以帮助他们。在我加入时人们正在用 Microsoft Word 和 MathType 准备资料。我看到这是一个用 \LaTeX 的力量来增强资料的好机会，所以我建议使用 \LaTeX 来准备文档。

建议通过了，我们的团队开始转移我们的工作环境。在这个过程中，我写了一个小包，即是这个包的前身，在我们的资料中使用。我们部门很多人对 \LaTeX 几乎一无所知，我承诺新的准备过程不会很困难，我设法定义了整体程序，分享了一些关于 \LaTeX 的必要知识，并教他们如何使用此宏包。

我承认我当时有点不负责任。我确实描述了工作程序是如何以及我的包应该如何使用，但我较少分享关于 \LaTeX 的内容。我的包和工作流简化了一些概念，因此团队不需要关心 \LaTeX 的某些部分，例如 documentclass 和 preamble。然而 \LaTeX 本身与 Word 仍有很大不同。我低估了我的同事学习基本的 \LaTeX 技巧的难度，所以这对他们中的一些人来说是一种折磨。后来我有些意识到这个问题，但我被自己的业务所占据，并没有给予足够的重视。

最终的结果基本是成功的，而且大多已经足够好了。但我为我的同事感到难过，因为这件事对他们来说实际上并不那么容易。另外，我最近参加了西浦数学俱乐部 15 周年纪念活动，这在某种程度上增强了我对数学俱乐部的归属感。出于这些原因，我决定将来为西浦的学生留下一些东西。我重写了整个包，并添加了一系列教程，涵盖了 \LaTeX 的基础知识以及制作材料，以便资料部门的人在 \LaTeX 中制作资料的工作量会大大减少。教程采

用故事的形式，因为我认为这种方法为学习 \LaTeX 增添了更多乐趣。

除此之外，西浦的学生甚至其他 \LaTeX 用户可能会发现这个包和文档有用。所以它的最终形式被决定为一个带有文档的模板，在我的个人帐户下作为 GitHub repository 发布。这是因为我们的数学社目前没有 GitHub 帐户。一旦它建立了一个账户，我将把这个存储库转移到它的官方帐户。

3 中文版历史

由于 XJTLU 的 University Communication Language 是英语，所以此文档最开始被用英语写作。此中文版来源于对英文版的翻译。

第二部分 教程

4 教程 1: Ashley 的第一份资料

Ashley 最近加入了资料部门。现在他被指派写一部分涵盖基本微积分的资料。他非常兴奋，因为这将是他的第一个资料，也是他对 \LaTeX 的第一次尝试。首先，他需要知道如何在他的 PC 上使用 \LaTeX 和别的宏包。在本教程中，我们将跟随 Ashley，看看他在资料准备中学到了什么。

4.1 安装和配置

要使用 \LaTeX ，Ashley 需要在他的计算机上安装一个 \TeX 发行版（可能还有一个编辑器）。 \LaTeX 官方网站上列出了几种流行的发行版：<https://www.latex-project.org/get/>。这些发行版的安装和配置非常简单，Ashley 在几分钟内完成了它们。

然后，Ashley 想要一个编辑器来编写 \LaTeX 文档。他了解到 \TeX studio 是一个很好的工作室，所以他下载并安装了它。

Ashley 现在需要做的是获取该宏包的副本并安装它。(TBD)

4.2 基本书写

Ashley，迫不及待地等待开始他的第一项工作，单击打开 Templates 文件夹并导航到 material-book 文件夹。他打开 encapsulation.tex，发现它是这样的：

```
... Some code ...

\input{chapters.tex}

... Some code ...
```

正如文件名所暗示的，这是用于制作将作为书籍出版的资料。一本书包含一些章节，在资料写作中，每一章都是资料主题的一个特定部分。例如，Ashley 的工作是微积分资料的一部分，因此他需要为他的工作开始一章。他打开 chapter.tex 并发现它是一个空白文档。然后他添加以下命令开始他的第一章并在 encapsulation.tex 上运行 \LaTeX 以检查输出（左侧是他的命令的结果，出现在不同的字体系列中，右侧是他的命令，背景为灰色）：

1 Key points in calculus

```
\chapter{Key points in calculus}
```

`\chapter{...}` 是一个 \LaTeX *command*，它以 `\.` 用大括号 `{}` 包裹的单词形成命令的参数。在这个地方，参数是章节的标题。Ashley 注意到 \LaTeX 会自动放大标题字体并使它们加粗。与许多其他排版软件不同的是，用 \LaTeX 只需要知道要做什么的逻辑思路（例如在某个地方会有一个名为 xxx 的章节），它将为作者控制外观。

Ashley 对这个结果很满意。然后他键入一些段落。在 \LaTeX 中，段落由一个空行分隔。

1 Key points in calculus

```
\chapter{Key points in calculus}
```

What does Ashley write in these paragraphs?

```
What does Ashley write in these
paragraphs? Well, in fact, I don't
know. You may find him and ask him
yourself.
```

Well, in fact, I don't know. You may find him and ask him yourself.

But, wait a minute, how do I find Ashley when he doesn't really exist? Well, this is a good question.

```
But, wait a minute, how do I find
Ashley when he doesn't really exist?
Well, this is a good question.
```

Ashley 注意到段落由 \LaTeX 自动缩进。然而，他也注意到该章正下方的段落没有缩进（在中文环境中有所不同）。

除了章节之外， \LaTeX 还提供了以下分节命令：

- section
- subsection
- subsubsection
- paragraph
- subparagraph

您可能会注意到此处包含该 `paragraph`。事实上，分节命令 `\paragraph` 像其他分节命令一样为段落生成标题。

当 Ashley 转向整个输出时，他看到他的章节出现在目录中：

Contents

1	Key points in calculus	1
---	-------------------------------------	---

```
\chapter{Key points in
calculus}
```

L^AT_EX 为所有分节命令生成目录¹。出于这个原因，Ashley 必须在 `encapsulation.tex` 上运行 L^AT_EX 两次以获得正确的目录。

当 Ashley 为某个部分写了很长的标题时，目录变得非常难看。为了解决这个问题，Ashley 可以指定目录中使用的节的标题的简短形式，如下所示：

Contents

1	A Short Name	1
---	---------------------------	---

```
\chapter[A Short Name]
{A very very long
Caption}
```

4.3 调整字体

到目前为止，Ashley 知道如何指示 L^AT_EX 做一些基本的事情。虽然他感觉精力充沛，正在全速写作，但很快就遇到了一些问题。Ashley 想强调一些关键字，例如“limit”。他后来了解到命令 `\emph` 指示 L^AT_EX 强调传递给它的文本，如下所示。

Calculus is the study of *limits*.

```
Calculus is the study of \emph{limits}.
```

这东西挺 L^AT_EX，因为 Ashley 只告诉 L^AT_EX 强调它，而无法控制 L^AT_EX 如何做到这一点。虽然大多数时候这就足够了，但 Ashley 想要更多。他想知道如何明确控制文本的外观，因为 L^AT_EX 无法满足所有情况下的所有需求。L^AT_EX 确实提供了对字体的某些默认操作，Ashley 可以使用它们来控制文本的大小、系列和样式。

¹ 其深度低于目录 (toc) 深度。toc 的实际生成过程有点复杂，这里就不赘述了

Ashley can tell \LaTeX to adjust the font size like: very very small the size of scripts the size of foot notes small font just being normal a bit bigger large text very big huge damn huge

Ashley can tell \LaTeX to adjust the font size like:
`{\tiny very very small}` `{\scriptsize the size of scripts}` `{\footnotesize the size of foot notes}` `{\small small font}` `{\normalsize just being normal}` `{\large a bit bigger}` `{\Large large text}` `{\LARGE very big}` `{\huge huge}` `{\HUGE damn huge}`

这一次 Ashley 看到了与命令 `\emph{}` 不同的东西。此处的文本在大括号内，命令与文本一起在括号内给出。被一对大括号括起来的东西被称为在组内。在组内调用的命令会影响整个组。

除了大小，Ashley 还能够控制字体的样式和系列。如表 1 所示， \LaTeX 提供了以下命令来控制字体样式和系列：

Command	Used in a group	Action
<code>\textrm{...}</code>	<code>{\rmfamily...}</code>	Text in Roman family
<code>\textsf{...}</code>	<code>{\sffamily...}</code>	Text in sans serif family
<code>\texttt{...}</code>	<code>{\ttfamily...}</code>	Text in typewriter family
<code>\textmd{...}</code>	<code>{\mdseries...}</code>	Text in medium series
<code>\textbf{...}</code>	<code>{\bfseries...}</code>	Text in bold series
<code>\textup{...}</code>	<code>{\upshape...}</code>	Text in upright shape
<code>\textit{...}</code>	<code>{\itshape...}</code>	Text in <i>italic</i> shape
<code>\textsl{...}</code>	<code>{\slshape...}</code>	Text in <i>slanted</i> shape
<code>\textsc{...}</code>	<code>{\scshape...}</code>	Text in SMALL CAPS shape
<code>\emph{...}</code>	<code>{\em...}</code>	Text <i>emphasized</i>
<code>\textnormal{...}</code>	<code>{\normalfont...}</code>	Text in default font

表 1: 标准字体更改命令和声明

这些命令中的许多命令都提供了组内使用版本和普通版本。Ashley 可以根据自己的需要选择使用哪个版本。

Ashley 还想学习如何更改字体颜色。他很惊讶这个包文档没有提供这样的描述。与包作者联系后得知，由于资料是黑白打印的，所以没有涉及这个主题，因此更改颜色几乎没有用。

4.4 键入数学公式

这是 Ashley 工作中最激动人心的部分 — 输入公式！尽管他之前在 L^AT_EX 方面的经验很少，但他已经了解到 L^AT_EX 可以生成高质量的数学公式，正如他之前在 Math Stack Exchange (<https://math.stackexchange.com/>) 和知乎 (<https://www.zhihu.com/>)。

由于他之前在这些网站上的经验，他对如何在 L^AT_EX 中编写公式有所了解。

通常，L^AT_EX 中的公式分为两种类型：*inline* 和 *displayed*。前一种类型的公式被一对美元符号包围： $\$...\$$ ，而后一种类型的公式被一对双美元符号包围： $\$...\$$ 。这些分隔符是原始的 T_EX 分隔符。L^AT_EX 分别为内联和显示数学提供了两对分隔符： $\langle...\rangle$ 和 $\llbracket...\rrbracket$ 。事实上，应该避免使用 T_EX 简写 $\$...\$$ 来显示数学，因为它可能会导致 L^AT_EX 中出现奇怪的问题。

顾名思义，*inline* 公式位于一行文本中，而 *displayed* 公式则显示在正文之外。

The derivative of a function f can be written as f' ,
or as

$$\frac{df}{dx}$$

```
The derivative of a function $f$  
can be written as $f'$, or as  
\[  
\frdt{x}  
\]
```

在这里，Ashley 使用了 xjtlumath 提供的命令：`\frdt`。此命令有两个参数，其中第一个参数是可选的，默认值为 f 。它们分别代表函数和变量。当 Ashley 想要将 g 对 x 的导数排版时，他会写 `\frdt[g]{x}`，给出 $\frac{dg}{dx}$ 。可选参数被包含在两个方括号内传递给命令。像 Ashley 一样小心，您可能会注意到，在 *inline* 模式下，公式会在一定程度上缩小，以便它可以放在一行中。

Ashley 对渲染效果相当满意。然后他写了一些方程式和文本。但是当他想引用一个方程时，问题就出现了。他如何处理他想要引用的等式？在这里，Ashley 被介绍了另一种给出显示数学的方法：环境 `equation`。

The fundamental theorem of calculus can be expressed in the form of Equation 1

$$\int_a^b f(x) \, dx = F(b) - F(a) \quad (1)$$

The fundamental theorem of calculus can be expressed in the form of Equation \ref{eq:fundthmcal}

$$\int_a^b f(x) \, dx = F(b) - F(a)$$

label{eq:fundthmcal}

Ashley 通过他之前的知识理解了这个等式：下划线 (`_`) 将下标 (a) 引入积分符号 \int (`\int`)，而插入符号 (`^`) 将上标 (b) 引入其中。命令 `\dx` 由 `xjtlumath` 提供。如果 Ashley 直接输入 `dx`，结果会是这样的 $\int f(x)dx$ ，难看。`\dx` 优化结果。注意 `\dx` 只能用来表示积分变量，因为它在它前面加了一点空格。如果 Ashley 需要使用其他变量，他需要使用命令 `\dd`。`xjtlumath` 也为多重积分提供了相同的功能。

$$\int f(t) \, dt, \quad \iint f(x, y) \, dx \, dy, \quad \iiint f(x, y, z) \, dx \, dy \, dz$$

$$\iint f \, dr \, d\theta, \quad \iiint f \, dz \, dr \, d\theta, \quad \iiint f \, d\rho \, d\theta \, d\phi$$

```

\[
\int \int \int f(t) \, dt, \quad \text{\textbf{\textit{quad}}}
\int \int f(x,y) \, dx dy, \quad \text{\textbf{\textit{quad}}}
\int \int \int f(x,y,z) \, dx dy dz
\]

\[
\int \int f \, dr dt, \quad \text{\textbf{\textit{quad}}}
\int \int \int f \, dz dr dt, \quad \text{\textbf{\textit{quad}}}
\int \int \int f \, dr dt dp
\]

```

然而, Ashley 对 L^AT_EX 中的交叉引用以及环境 `equation` 一无所知。让我们向 Ashley 解释它们。在 L^AT_EX 中, 一个环境由 `\begin` 命令开始并以 `\end` 命令结束。环境的名称被传递给这对命令。环境 `equation` 给出了一个显示的数学方程, 它是被计数的。`equation` 末尾的 `\label` 命令捕获计数器以及其他一些信息, 例如其位置, 并将其存储在由赋予 `\label` 的名称表示的标签中。要使用标签, Ashley 需要 `\ref` 命令, 该命令打印计数器²。

equation 环境给出了一个计数器，而 \[N] 没有。在后者使用标签会导致标签被定向到另一个计数器，因此仅当该事物被计数时才使用标签。

²并另外生成一个可点击的超链接, 点击时导航到方程的位置, 这是 hyperref, 此模板加载的包的效果。

4.5 空白管理

\LaTeX 中的空间管理比仅仅输入空格要复杂一些。Ashley 是个细心的人，他很快发现句子后面的空格比单词之间的空格大一点（你可以放大.pdf 文件看到这个）。 \LaTeX 决定一个空格作为句子末尾的空格，如果

1. 终止句子的标点符号后紧跟此空格，并且
2. 如果终止句子的标点符号是句号，则句号前的字母要是小写。

大多数句子都按照上述规则结束，但也有一些例外。例如， \LaTeX 可能将 Mr. 作为句子结尾的符号，从而产生错误的间距。在这种情况下，Ashley 需要通过 `~` 和 `\@` 手动配置 \LaTeX 。

In another world, Mr. Ashley was once loved by Miss Scarlett. In this world, Mr. Ashley has a PC. He loves programming on his PC.

In another world, Mr.~Ashley was once loved by Miss Scarlett. In this world, Mr.~Ashley has a PC\@. He loves programming on his PC.

Ashley 很高兴学会了如何管理空间。然而很快他又发现了另一个问题。

Some commands like \LaTeX seems to eat the space after it.

Some commands like `\LaTeX` seems to eat the space after it.

为了解决这个问题，Ashley 需要在命令的末尾添加一个空组（`{ }`）。

至于数学公式，事情就变得不一样了。 \LaTeX 忽略数学模式下的所有空格，无论是内联的还是显示的。要添加额外的空格，Ashley 必须使用表 2 中显示的命令。

4.6 列表和其他环境

现在 Ashley 已经学会了处理文本，他继续他的写作。很快，他不得不再次停止，因为他正在制定一份列表。起初，他对列表进行了硬编码，如下所示：

1. Something
2. Something
3. Something

1. Something
2. Something
3. Something

Command	Effect (approximately)
<code>\,</code>	$\frac{3}{18}$ quad (\mathbb{I})
<code>\:</code>	$\frac{4}{18}$ quad (\mathbb{I})
<code>\;</code>	$\frac{5}{18}$ quad (\mathbb{I})
<code>\</code> (<code>\</code> followed by a space)	a space
<code>\quad</code>	Width of ‘M’ in current font (\mathbb{I})
<code>\qquad</code>	2 quad (\mathbb{I})

表 2: 空格在数学模式下

然而，这个解决方案看起来相当愚蠢。另外，如果 Ashley 想要更改号码，他需要手动完成。他想知道 \LaTeX 是否有一些更方便的方法来做到这一点。

幸运的是有。 \LaTeX 提供了几种处理列表的环境。一个例子如下所示。

1. Something	<code>\begin{enumerate}</code>
	<code>\item Something</code>
2. Something	<code>\item Something</code>
	<code>\item Something</code>
3. Something	<code>\end{enumerate}</code>

现在 Ashley 几乎拥有他需要知道的一切。他对自己写的东西很满意，感到很高兴。现在对他来说重要的是他想以更花哨的方式展示定理、定义和其他东西，以便他的读者可以专注于这些。

为此，`xjtlumath` 提供了一些花哨的环境。

Definition 4.1: Absolute Convergence	<code>\begin{definition}[Absolute Convergence]</code>
An infinite series $\sum_{n=0}^{\infty} a_n$ is said to be absolutely convergent iff	An infinite series $\sum_{n=0}^{\infty} a_n$ is said to be absolutely convergent iff
$\sum_{n=0}^{\infty} a_n $	<code>\[</code>
converges	<code>\sum_{n=0}^{\infty} a_n </code>
	<code>\]</code>
	converges
	<code>\end{definition}</code>

xjtlumath 加载 amsthm, 它提供了 proof 环境。当 Ashley 使用这个环境写证明时, 他发现在环境的末尾出现了一个 q.e.d. 符号。

Now we prove the mean value theorem for definite integrals. That is, for a continuous function f that is bounded on $[a, b]$, the definite integral $\int_a^b f(x) dx = f(c)(b - a)$, where $c \in [a, b]$.

证明. Let m, M be the infimum and supremum of $f([a, b])$, respectively. Therefore, $m \leq f \leq M$, and

$$\int_a^b m dx \leq \int_a^b f(x) dx \leq \int_a^b M dx$$

, which gives

$$\begin{aligned} m(b-a) &\leq \int_a^b f(x) dx \leq M(b-a) \\ m &\leq \frac{\int_a^b f(x) dx}{b-a} \leq M \end{aligned} \quad (2)$$

Since that f is continuous, it can reach every value between the infimum and supremum of its range. That is, $\exists c \in [a, b], f(c) = \frac{\int_a^b f(x) dx}{b-a}$. Substitute $f(c)$ back to equation 2 gives what the theorem states. \square

```
Now we prove the mean value theorem for
definite integrals. That is, for a continuous
function  $f$  that is bounded on  $[a,b]$ , the
definite integral  $\int_a^b f(x) dx = f(c)(b-a)$ , where  $c \in [a,b]$ .
\begin{proof}
Let  $m,M$  be the infimum and supremum of  $f([a,b])$ , respectively.
Therefore,  $m \leq f \leq M$ , and
\[
\int_a^b m dx \leq \int_a^b f(x) dx \leq \int_a^b M dx
\]
, which gives
\begin{align}
m(b-a) &\leq \int_a^b f(x) dx \leq M(b-a) \\
m &\leq \frac{\int_a^b f(x) dx}{b-a} \leq M
\end{align}
\label{eq:meanvalint}
\end{proof}
```

Ashley 能够通过使用 proof 提供的 \qedhere 命令来控制 q.e.d. 符号出现的位置。如果这个命令在之前被给出, 那么它就不会出现在最后。

证明. Some words...

$$a + b = c$$

Som words...

```
\begin{proof}
Some words...
\[
a+b=c \quad \square \quad \text{\qedhere}
\]
Som words...
\end{proof}
```

除了 definition 和 proof, Ashley 还能够使用 theorem, proposition,

corollary, lemma, axiom, and example. 除了 proof 之外, 这些环境拥有各自的计数器, Ashley 可以简单地使用标签来引用它们。

不过, Ashley 的同事超抱怨到当他尝试写中文资料时, 这些环境的标题还是英文。为了支持中文, xjtlumath 给这些环境 (不幸的是, 在中文环境下, proof 的标题也变成了中文, 就没有英文版了的) 增加了中文版本, 只要在最后加一个 c 即可使用中文版本。

这些设施极大地帮助了 Ashley 的资料准备, 他很快就会完成他的工作
.....

5 教程 2: Delilah 和复杂的数学公式

Delilah 正在研究有关线性代数的材料的一部分。随着工作的进行，她将获得处理 \LaTeX 中复杂数学公式的能力，尤其是 `xjtlumath` 中加载的 `ams` 包提供的那些方法。

5.1 多行公式

线性方程组是线性代数的基本部分。当 Delilah 尝试输入一组方程时，她遇到了一个问题。在预定义的 `\[\]` 和环境 `equation` 中，她找不到开始新行的选项。甚至 \LaTeX 的换行选项，如 `\\` 和 `\newline` 在那里也不起作用。一个方程组当然不应该放在一行，那她现在该怎么办？后来她了解到环境 `aligned` 旨在允许方程组在多行中对齐：

$$\begin{aligned} x + y &= 1 \\ x - y &= 2 \end{aligned}$$

```
\[
\begin{aligned}
x+y &= 1\\
x-y &= 2
\end{aligned}
\]
```

这里，与符号 `&` 用在方程要对齐的符号之前。换行符 `\\` 开始一行新的等式。请注意，此处不能使用其他换行操作。

Delilah 喜欢这个结果，但她觉得方程式太孤单了。她认为为他们添加一个大花括号会安慰他们。 \LaTeX 支持将分隔符放在一组事物之前和之后的语法。

$$\left\{ \begin{aligned} x + y &= 1 \\ x - y &= 2 \end{aligned} \right.$$

```
\[
\left\{
\begin{aligned}
x+y &= 1\\
x-y &= 2
\end{aligned}
\right.
\]
```

`\left` 命令定义要放在左边的东西，而 `\right` 命令定义要放在右边的东西。Delilah 不想把任何东西放在右边，所以写了 `.` 表示空。

对于一组不需要对齐的方程, 或者对于一个太长而不能放在一行中的单个方程, 没有对齐的环境 *gathered* 是更好的选择:

$$\begin{aligned}\cos z &= 1 - \frac{z^2}{2!} + \frac{z^4}{4!} - \frac{z^6}{6!} + \cdots \\ &= \sum_{n=0}^{\infty} \frac{(-1)^n z^{2n}}{(2n!)}\end{aligned}$$

```
\[
\begin{gathered}
\cos {z} = 1 - \frac{z^2}{2!} + \frac{z
^4}{4!} - \cdots \\
= \sum_{n=0}^{\infty} {\frac{(-1)^n z^{2n}
}}{(2n!)}}
\end{gathered}
\]
```

写了几组方程后, Delilah 想引用其中的一组。她使用 `equation` 环境而不是 `\[\]`, 但发现方程是作为一个整体编号的。

$$\begin{aligned}x + y &= 1 \\ x - y &= 2\end{aligned}\tag{3}$$

```
\begin{equation}
\begin{aligned}
x+y &= 1\\
x-y &= 2
\end{aligned}
\end{equation}
```

所以她很难在一个组中引用单个方程。amsmath 为此提供环境 `align`:

$$\begin{aligned}x + y &= 1 \\ x - y &= 2\end{aligned}\tag{4}\tag{5}$$

```
\begin{align}
x+y &= 1\\
x-y &= 2
\end{align}
```

如果她不想为单个方程编号, 则需要在该行的末尾附加 `\nonumber`。

$$\begin{aligned}x + y &= 1 \\ z &= 10 \\ x - y &= 2\end{aligned}\tag{6}\tag{7}$$

```
\begin{align}
x+y &= 1\\
z &= 10 \nonumber\\
x-y &= 2
\end{align}
```

如果没有 `'ed'` 后缀, *gather* 也是一个独立的环境, 可以完成 *gather* 的工作。但是普通版本和 `"ed"` 版本之间有一个主要区别。Delilah 发现不可能在对齐或聚集之前再次放置括号, 因为它们不需要但被数学环境包围。此

外，它们的宽度固定为文本的宽度，而它们的“ed”版本可以是任何宽度。如果没有‘ed’后缀，*gather* 也是一个独立的环境，可以完成聚集的工作。但是普通版本和“ed”版本之间有一个主要区别。Delilah 发现不可能在对齐或聚集之前再次放置括号，因为它们不需要但被数学环境包围。此外，它们的宽度固定为文本的宽度，而它们的“ed”版本可以是任何宽度。

与 *equation* 环境一样，它们的星号版本默认不给出数字。

$$\begin{aligned} x + y &= 1 \\ x - y &= 2 \end{aligned}$$

```
\begin{align*}
x+y &= 1\\
x-y &= 2 \\
\end{align*}
```

Delilah 能够将多组方程放在一个列中，只需在组之间添加&号即可。

$$\begin{aligned} x + y &= 1 & a + b &= 3 \\ x - y &= 2 & a - b &= 4 \end{aligned}$$

```
\begin{align*}
x+y &= 1 & a+b &= 3\\
x-y &= 2 & a-b &= 4 \\
\end{align*}
```

align 自动调整方程组之间的空间。

5.2 矩阵

矩阵对于线性代数至关重要，因为它们表示从一个向量空间到另一个特定基底的线性映射。此外，系数矩阵和增广矩阵便于操作线性方程。

amsmath 提供了多种输入矩阵的环境。

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$$

```
\[
\begin{bmatrix}
1&2&3&4\\
5&6&7&8\\
9&10&11&12\\
13&14&15&16
\end{bmatrix}
\]
```

环境 *pmatrix*、*Bmatrix*、*vmatrix* 和 *Vmatrix* 分别产生分隔符 $()$ 、 $\{\}$ 、 $|$ 和 $||$ 。

为了在 inline 模式下使用矩阵, Delilah 使用环境 `smallmatrix`, 它在 `amsmath` 中没有 `p,b,B,v,V` 版本, 因为决定分隔符是作者的责任。

The matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ is so small and cute!

```
The matrix  $\left(\begin{smallmatrix} a&b\\c&d \end{smallmatrix}\right)$  is so small and
cute!
```

当 Delilah 尝试将分数放入矩阵中时, 她发现有些烦人事。

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ 1 & \frac{1}{4} & \frac{1}{5} \end{bmatrix}$$

```
\[
\begin{bmatrix}
1&\frac{1}{2}&\frac{1}{3}\\
1&\frac{1}{4}&\frac{1}{5}
\end{bmatrix}
```

上面和下面的分数非常接近, 以至于它们相互接触! 这不是 Delilah 想要的, 她很惊讶 `LaTeX` 没有检测到这一点并做一些事情。幸运的是, 在 `amsmath` 环境中, 允许将可选参数传递给 `\\` 以定义行之间的实际垂直空间。对于分数, `2ex` 是一个不错的选择。此外, 分数处于 inline 模式。`\dfrac` 命令给出 `displayed` 模式的分数。

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ 1 & \frac{1}{4} & \frac{1}{5} \end{bmatrix}$$

```
\[
\begin{bmatrix}
1&\dfrac{1}{2}&\dfrac{1}{3}\\
1&\dfrac{1}{4}&\dfrac{1}{5}
\end{bmatrix}
```

有时矩阵太大而无法完全显示。在这些时候, 使用省略号很重要。当 Delilah 写出矩阵的逆时, 她使用省略号。

$$A^{-1} = \frac{1}{\det A} \begin{bmatrix} C_{11} & C_{21} & \cdots & C_{n1} \\ C_{12} & C_{22} & \cdots & C_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ C_{n2} & C_{n2} & \cdots & C_{nn} \end{bmatrix}$$

```
\[
A^{-1} = \frac{1}{\det A}
\begin{bmatrix}
C_{11} & C_{21} & \cdots & C_{n1} \\
C_{12} & C_{22} & \cdots & C_{n2} \\
\vdots & \vdots & \ddots & \vdots \\
C_{n2} & C_{n2} & \cdots & C_{nn}
\end{bmatrix}
\]
```

5.3 文字和运算符

要将文本放入数学环境中，Delilah 使用 `amsmath` 提供的 `\text` 命令。

Definition 5.1: Null Space

The null space of an $m \times n$ matrix A , written as $\text{Nul } A$, is the set of all solutions of the homogeneous equation $Ax = \mathbf{0}$. In set notation,

$$\text{Nul } A = \{x : x \text{ is in } \mathbb{R}^n \text{ and } Ax = \mathbf{0}\}$$

```
\begin{definition}[Null Space]
The null space of an  $m \times n$ 
matrix  $A$ , written as  $\text{Nul } A$ ,
is the set of all solutions
of the homogeneous equation  $A\vec{x} = \vec{0}$ . In set
notation,
\[
\text{Nul } A = \{\vec{x} : \vec{x} \text{ is in } \mathbb{R}^n \text{ and } A\vec{x} = \vec{0}\}
\]
\end{definition}
```

`\Nul`, `\sin`, ... 等命令是数学运算符。 \LaTeX 中的部分预定义数学运算符显示在表 3 中。

实际上，运算符 `\Nul` 和 `\Span` 是由 `xjtlumath` 定义的，如西浦一年级线性代数教科书中的形式。此外，`xjtlumath` 更改了 \LaTeX 中的默认 `\vec` 命令，以便矢量以粗体形式出现，而不是在其上方带有箭头。

一些运算符，如 `\lim`，旨在支持对它取极限。也就是说，在 `displayed` 模式下，当试图使用 `_` 给这样的运算符一下标时，下标将出现在运算符的底部。

•

Result	Command	Result	Command	Result	Command
\arccos	<code>\arccos</code>	\arcsin	<code>\arcsin</code>	\arctan	<code>\arctan</code>
\cos	<code>\cos</code>	\sin	<code>\sin</code>	\tan	<code>\tan</code>
\cot	<code>\cot</code>	\sec	<code>\sec</code>	\csc	<code>\csc</code>
\cosh	<code>\cosh</code>	\sinh	<code>\sinh</code>	\tanh	<code>\tanh</code>
\lim	<code>\lim</code>	\liminf	<code>\liminf</code>	\limsup	<code>\limsup</code>
\ln	<code>\ln</code>	\log	<code>\log</code>	\lg	<code>\lg</code>
\max	<code>\max</code>	\min	<code>\min</code>	\sup	<code>\sup</code>
\inf	<code>\inf</code>				
\ker	<code>\ker</code>	\det	<code>\det</code>	\exp	<code>\exp</code>

表 3: 一些预定义的数学运算符

$$\lim_{x \rightarrow 0} f(x)$$

```
\[
\lim_{x \to 0} f(x)
\]
```

Delilah 能够通过使用 `\limits` 和 `\nolimits` 来明确控制极限样式。请注意，这两个命令只能在支持采取极限的操作后使用。

$$\lim_{x \rightarrow 0} f(x)$$

$$\lim_{x \rightarrow 0} f(x)$$

```
$_{\lim \limits_{x \to 0} f(x)}$
\[
\lim_{x \to 0} f(x)
\]
```

5.4 符号

数学环境中的标准 L^AT_EX 字体整洁干净。然而在一些特殊场合，Delilah 想改变一些符号的字体。例如，为了表示一些常规集合，她使用黑板字体。

$$\mathbb{R} \quad \mathbb{N} \quad \mathbb{Q} \quad \mathbb{Z}$$

```
\[
\mathbb{R} \quad \mathbb{N} \quad \mathbb{Q} \quad \mathbb{Z}
\]
```

每次都写 `\mathbb` 有点烦人。为此，`xjtlumath` 为它们定义了简写。

$\mathbb{R} \quad \mathbb{Q} \quad \mathbb{Z} \quad \mathbb{N} \quad \mathbb{N}^+$
$$\begin{aligned} & \backslash[\\ & \backslash\text{setr} \quad \backslash\text{quad} \quad \backslash\text{setq} \quad \backslash\text{quad} \quad \backslash\text{setz} \quad \backslash\text{quad} \quad \backslash\text{setn} \\ & \backslash\text{quad} \quad \backslash\text{setnp} \\ & \backslash] \end{aligned}$$

Likenormaltext **bold**

$$\backslash[\backslashmathrm{Like\ normal\ text}\quad\backslash\mathbf{quad}\quad\backslash\mathbf{bold}\backslash]$$

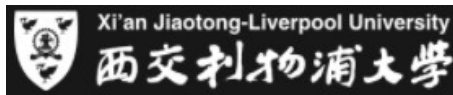
6 教程 3: 月处理 floats

月正在为数学部的月刊准备材料。她想让她的材料有趣且易于理解，因此她使用了许多图形和表格。

图形、表格和许多其他占据随机宽度和高度（通常很大）区域的东西在 \LaTeX 中被视为 *floats*。Floats 在今天的文档中很常见，但它们在排版时会造成很大的麻烦。在本节中，我们将与月一起了解如何在 \LaTeX 中处理 floats。

6.1 插入图像

要在 \LaTeX 中插入图像，包 `graphicsx`（由模板文件加载）是一个不错的选择。它提供了接受输入图像文件名和一些可选说明符的命令 `\includegraphics`。



```
\includegraphics[width=\textwidth]{assets/
examplelogo.jpg}
```

月很快发现，简单地使用这个命令并不是一个好的选择，因为如果图像没有足够的垂直空间，它会被放置在下一页，留下一个很大的空白区域，非常难看。此外，她无法为图像提供标题或引用它。

所以，月用使图像成为 *figure* 的 `figure` 环境包裹了图像。



Figure 1: Example Logo

Figure 1 shows the figure Yue uses.

```
\begin{figure}
\includegraphics[width=\textwidth]{assets
/examplelogo.jpg}
\caption{Example Logo}
\label{fig:example}
\end{figure}
Figure \ref{fig:example} shows the figure
Yue uses.
```

\LaTeX 会自动给它一个数字，以便月能够引用它。请注意，由于内部实现原因，`\label` 只能紧跟在 `\caption` 之后，以免引用错误。

6.2 表

即使使用数字确实需要额外的环境，但它仍然很简单，月很快就熟悉了。然而，在 \LaTeX 中处理表格更为复杂。

要在 \LaTeX 中生成类似表格的内容，月必须使用特殊环境。tabular 和 array 是其中的两个具体例子。事实上，这两种环境在大多数方面是相似的，一个主要区别是 array 经常用于数学模式。

array 和 tabular 的语法类似于 Delilah 使用的矩阵环境之一，尽管在这里月必须明确指定列行为。

Entry 1	Entry 2
a	b

```
\begin{tabular}{|c|c|}
\hline
Entry 1 & Entry 2\\
\hline
a & b\\
\hline
\end{tabular}
```

月不太明白 |c|c| 是什么意思，所以她在网上搜索了这个。这告诉她传递给 tabular 的这个参数指定了每一列。字母 c 告诉表格该列中的内容应该居中。另外两个对齐说明符 l 和 r 分别可用于“左”和“右”。竖线表示在此位置应插入一条垂直线（比较指示在当前行顶部插入水平线的 \LaTeX 命令 \hline ）

当给定 c、l 和 r 时，列的宽度由内容的宽度决定。通过使用另一个说明符“p”，Yue 能够控制列的宽度，其中内容是左对齐的。

Entry 1	Entry 2
a	b

```
\begin{tabular}{|c|p{2cm}|}
\hline
Entry 1 & Entry 2\\
\hline
a & b\\
\hline
\end{tabular}
```

当有许多列具有相同的说明符时，月可以使用这种语法 $\text{*{num}{spe}}$ 来重复说明符，其中 num 是重复的次数，spe 是说明符。

a	a	a	a	a	a	a
a	a	a	a	a	a	a

```
\begin{tabular}{|*{7}{c|}}
\hline
a&a&a&a&a&a&a\\
\hline
a&a&a&a&a&a&a\\
\hline
\end{tabular}
```

月不喜欢用线来分行列的表格，因为她认为它们不整洁。她想用空间来分隔内容。她可以在列说明符之间使用 `@{\hspace{}}` 来指定列间空间，并在 `\\` 之前使用 `\vspace{}` 在下一行之前添加额外的空间。

a	b	b
a	b	b
c	d	d

```
\begin{tabular}{c@{\hspace{1cm}}cc}
a & b & b\\
a & b & b\vspace{.5cm}\\
c & d & d\vspace{.5cm}\\
\end{tabular}
```

实际上，在`@{}`里面，月不仅可以使⤿空⤿，还可以使⤿其他任何内容。

crime scene <POLICE, stay away> people
crime scene <POLICE, stay away> people

```
\begin{tabular}{c@{ <POLICE, stay away> }c}
crime scene & people \\
crime scene & people \\
\end{tabular}
```

与环境 figure 一样，环境 table 被设计为接受类似表的内容。

a	b
c	d

Table 1: Example Table

Table 1 shows a table.

```
\begin{table}
\begin{tabular}{cc}
a & b \\
c & d
\end{tabular}
\caption{Example Table}
\label{tab:example}
\end{table}
Table \ref{tab:example} shows a table.
```


既然月了解了如何在 L^AT_EX 中操作类似表格的内容，她很快就会继续工作。在某些时候，她必须创建一个包含至少 20 列的超大表，她发现它超出了页面。当她尝试使用 `\small` 来减小文本的大小时，她发现她必须为表格的每个条目复制它，这意味着要重复数百次。为了控制整体风格，Yue 需要在 `table` 的开头和 `tabular` 的开头之间给出控制命令。

[illegible][illegible]

6.3 放置 floats

月曾经使用 Microsoft Word，它可以将 floats 放在用户想要放置的任何位置。自从她转向 L^AT_EX 一段时间以来，一切都很好，但现在月出现了问题。一个图从输出中“消失了”。一次次检查她的代码和输出后，她意外地发现下一页出现了这个图。这真的让她很困惑。由于 T_EX 的内部算法，技术上不可能将每个 float 排列在用户想要放置它们的位置。根据 *the L^AT_EX Companion*,

“Floats are often problematic in the present version of L^AT_EX, because the system was developed at a time when documents contained considerably less graphical material than they do today.”

然而 L^AT_EX 确实提供了一些选项，允许 Yue 在某种程度上控制 floats 的位置。对于 figure 或 table 环境，Yue 能够向它传递一个可选参数，指定所需的位置。有五个放置说明符，它们可以按任何顺序组合在一起。

! 忽略一些 L^AT_EX 限制³。

³`LATEX` 在尝试放置 floats 时有一些限制。例如，如果 float 的高度大于页面高度的某种程度，则在尝试放置此 float 时无法将其放置在页面底部。

h 尝试将 float 准确地放置在环境被给定时位置。如果尝试失败并且除此之外没有其他说明符除了！被给定，说明符将更改为 t。

t 尝试将 float 放置在页面顶部。

b 尝试将 float 放置在页面底部。

p 尝试将浮动放置在 float 页面（由 L^AT_EX 生成的用于放置 floats 的页面）

L^AT_EX 尝试按照上述列表从上到下的顺序根据说明符放置一个 float。通常，文档的所有 floats 都可以被正确处理。但是如果一个 float 被证明无法处理，作者应该调整（大概率减少）它的宽度和高度。

6.4 Floats 的目录

正如开头提到的，在月的材料中，有很多 floats。她想知道是否有办法为他们提供快速参考。

与 `\tableofcontents` 一样，L^AT_EX 提供了以下两个命令，分别打印文档中使用的所有图形列表和所有表格列表。

`\listoffigures` and `\listoftables`

出现在列表中的 float 名称由 float 的 caption 定义。如果 caption 看起来太长，Yue 可以向 caption 传递一个可选参数，该参数将显示在列表中。另外，不要忘记编译文件至少两次以使列表正确显示。

6.5 关于图的一些建议

月被建议对图像使用矢量图，因为当图像与输出文件一起缩放时矢量图是无损的。

有几个软件包可以直接在 L^AT_EX 中绘制图像，但使用它们都需要付出很大的努力。建议使用现代工具生成适当的图像（例如 Mathematica 能够导出绘制的数学图。）。)