

# L<sup>A</sup>T<sub>E</sub>X 从入门到魔怔

2022 年 4 月 6 日

## 目录

<b>1</b>	<b>L<sup>A</sup>T<sub>E</sub>X 简介</b>	<b>1</b>
1.1	文档类别 . . . . .	2
1.2	基础环境 . . . . .	3
1.2.1	图片 . . . . .	3
1.2.2	表格 . . . . .	5
<b>2</b>	<b>进阶操作</b>	<b>5</b>
2.1	参考管理系统 . . . . .	5
2.2	页面设置 . . . . .	6
2.2.1	留边与页面尺寸 . . . . .	6
2.2.2	颜色 . . . . .	6
2.2.3	页眉和页脚 . . . . .	6
2.2.4	使用 listings 进行高亮代码 . . . . .	7
2.2.5	画图 . . . . .	8
<b>3</b>	<b>浮动对象</b>	<b>8</b>

## 1 L<sup>A</sup>T<sub>E</sub>X 简介

L<sup>A</sup>T<sub>E</sub>X 是一个使用高级标记语言的排版系统。L<sup>A</sup>T<sub>E</sub>X 使用一组丰富的命令来指定文档的组成部分（标题、章节、粗体字、斜体字、图表等等），并让 L<sup>A</sup>T<sub>E</sub>X 来处理格式问题。当然，你还可以对文件进行个性化处理，进一步指定文件的格式和功能。

一个 L<sup>A</sup>T<sub>E</sub>X 文档 (\*.tex 文件) 被分成两个主要部分：导言区 (preamble) 和文件的主体 (main body)。导言区包含了所有的全局规格和格式设置，如纸张大小、字体大小、是什么类型的文件（文章、书籍、报告、信件...）。它也是我们包含”包”的地方，它允许访问宏库以增加 L<sup>A</sup>T<sub>E</sub>X 的功能。然后是文件的主体，通常被分为不同的部分、章节或小节。

导言区从这个命令开始：

```
\documentclass[ ]{ }
```

大括号允许你指定文件类型（或”类别”）–标准的类型是 article、book、report 和 letter。写完了导言区，主文件的开头和结尾是：

```
\begin{document} ... \end{document}
```

在 \begin{document} 以前的命令，将被视为处于导言区中的命令。

## 1.1 文档类别

L<sup>A</sup>T<sub>E</sub>X 的标准分段命令是：`\part{}`、`\section{}`、`\subsection{}` 和 `\paragraph{}`。这些都是三个标准文档类别所共有的，而 `report` 和 `book` 还有一个更高的分段命令 `\chapter{}`，`article` 还有一个更低的分段命令 `\subsection{}`。

还有一些其他区别，诸如：

- 1 `book` 和 `report` 中的每一章都有图表和其他图表以及方程式的编号，而文章中的编号则是全面的；
- 2 书籍和报告类的文件以标题页（如果指定的话）作为独立的一页开始，而文章则没有；
- 3 `\abstract{}` 在 `book` 类中是不可用的

`\documentclass[ ]{ }` 中的方括号为可选参数，有如下几类：

纸张尺寸 标准选项包括：`a4` 纸、`a5` 纸、`b5` 纸、`letterpaper`、`executivepaper` 和 `legalpaper`。默认的页面大小是 US letter，类似于 A4。

字体大小 如 12pt。对于标准文档类别，可以接受 10pt（默认）、11pt 和 12pt。进一步的字体大小规范可以在文档正文中实现；

单面双面 默认情况下，书籍是双面的，而文章和报告是单面的；

草稿模式 不编排图片，允许更快的编排速度。

这些只是设置了全局的格式，在文件的主体部分，你可以在需要时对某些选项进行局部修改。在指定了文档类别和所需的选项后，导言部分还允许我们调用“包”，这样我们就可以访问 L<sup>A</sup>T<sub>E</sub>X 中特定功能的宏。包含这些的命令是：

```
\usepackage[ ]{ }
```

流行的软件包包括 `geometry`、`graphicx`、`amsmath`、`xcolor`，以及其他许多软件包。包的资源库及其文档可以在 [www.ctan.org](http://www.ctan.org)。表 1 给出了最流行的软件包和它们的用途的概述。

包名称	作用
<code>graphicx</code>	在文档中包含图片的必要条件。提供 <code>includegraphics</code> 命令
<code>babel</code>	改变文件的默认语言（对正确的连字符和换行符有用）
<code>amsmath</code> , <code>amssymb</code>	提供了一套全面的命令来格式化数学语句以及符号
<code>hyperref</code>	L <sup>A</sup> T <sub>E</sub> X 中的超链接管理，允许链接到文本中的图、表和其他项目。
<code>geometry</code>	允许根据需要定义页面设置，如控制边距和文本宽度等功能
<code>microtype</code>	控制字距和其他字体设置
<code>booktabs</code> , <code>multicol</code> , <code>tabularx</code> , <code>tabulary</code>	高级表格格式化
<code>tikz</code> , <code>pgfplots</code>	绘图和制图软件包（第三章）
<code>fancyhdr</code>	构建页眉和其他高级页面格式化
<code>biblatex</code> , <code>natbib</code>	参考文献管理（第二章）

表 1: L<sup>A</sup>T<sub>E</sub>X 中常用的包

通常需要进行的第一项修改是调整页边距和行距。要做到这一点，可以使用 `geometry` 包。它不仅可以用各种单位来规定页边距，还可以规定许多其他参数。边距可以用：

```
\usepackage[margin=1.5cm]{geometry}
```

来设置，也可以单独指定每个边距，用

```
\usepackage[left=2cm, right=2cm, top=1.54cm, bottom=1cm]{geometry}
```

改变行距的一个简单方法是使用 `setspace` 包。通过在导言区中设置 `\usepackage[...]{setspace}`，我们可以设置 `[doublespacing]` 来获得一个双倍行距的文件，或使用：

```
\usepackage[onehalfspacing]{setspace}
```

## 1.2 基础环境

### 1.2.1 图片

在 L<sup>A</sup>T<sub>E</sub>X 中导入 `graphicx` 包，我们可以使用 `\includegraphics[ ]{ }` 命令引入一个图片，并且可以在方括号位置直接定义图片的尺寸，如：`[scale=1.3]`；同时也可以间接设置图片的大小：

```
\includegraphics[width = 0.8 \textwidth]{figure.png}
```

上面的例子是将图片调整一个为：页宽 80% 的图片并引入。如果想同时指定只需要设置为：`[width = 2cm, height = 5 cm]`，想要旋转一定角度也可以设置：`[angle=90]`。

在大括号中，我们必须传递图片的名称（路径名）。文件扩展名可以省略。L<sup>A</sup>T<sub>E</sub>X 搜索图片的默认位置可以在导言中设置为全局变量：

```
\graphicspath{{./figures}{./additionalFigures}}
```

通过将 `\includegraphics[ ]{ }` 命令放在 `\begin{figure}...\end{figure}` 环境中，我们可以指定其相对于页面的位置，并给该图一个标题和标签。我们可以通过使用 `\begin{figure}[h]` 来告诉 L<sup>A</sup>T<sub>E</sub>X，让它在文本中的位置与源代码中的位置大致相同。如果我们想把图表放在页面的顶部或底部，我们可以用 `t` 或 `b` 代替 `h`；使用 `p` 可以把它放在自己的页面上，而添加一个 `!` 可以覆盖 L<sup>A</sup>T<sub>E</sub>X 默认的定位参数。L<sup>A</sup>T<sub>E</sub>X 中图表的默认对齐方式是在页面的左边。由于图像经常需要居中，在图的环境中放置命令 `\centering` 可以实现这一目的。

标题的位置是在图的上方还是下方，只取决于 `\caption{...}` 命令是在 `\includegraphics[ ]{ }` 命令的上方还是下方。标签很有用，因为我们可以用它们来指代文本中的一个图（或表格、方程式等），而 L<sup>A</sup>T<sub>E</sub>X 会自动用正确的序号来代替它。有时候我们可能希望我们的文字包裹着图片，我们则需要



图 1: 我的微信头像

引入 `wrapfig` 包。同时，我们通过使用 `\begin{wrapfigure}{alignment}{width} ... \end{wrapfigure}` 环境。对于对齐方式，`r` 对应右对齐、`l` 对应左对齐，而对于宽度则和常规图形环境相同。实际的图形本身也像以前一样被插入。

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget,

consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

(测试文字) Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.



图 2: 右对齐

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.



图 3: 左对齐

请注意, 图的宽度不是相对于图框而言的, 而是相对于文本宽度而言的, 因此我们将它们的宽度设置为相同, 这样图就可以填满图框的空间。

(测试文字) Sed mattis, erat sit amet gravida malesuada, elit augue egestas diam, tempus scelerisque nunc nisl vitae libero. Sed consequat feugiat massa. Nunc porta, eros in eleifend varius, erat leo rutrum dui, non convallis lectus orci ut nibh. Sed lorem massa, nonummy quis, egestas id, condimentum at, nisl. Maecenas at nibh. Aliquam et augue at nunc pellentesque ullamcorper. Duis nisl nibh, laoreet suscipit, convallis ut, rutrum id, enim. Phasellus odio. Nulla nulla elit, molestie non, scelerisque at, vestibulum eu, nulla. Ut odio nisl, facilisis id, mollis et, scelerisque nec, enim. Aenean sem leo, pellentesque sit

amet, scelerisque sit amet, vehicula pellentesque, sapien.

### 1.2.2 表格

L<sup>A</sup>T<sub>E</sub>X 提供了许多软件包和工具来生成表格。两个常用的环境即 `table` 和 `tabular`，前者提供了表格的定位、对齐、标签和标题，后者则是表格本身。需要注意的是，`table` 环境不是必需的，我们可以简单地用 `\begin{tabular}` 和 `\end{tabular}` 插入一个表格，但是这将让我们能够调整的参数变少。

cell 11	cell 12	cell 13
cell 21	cell 22	cell 23

表 2: Table caption

如同在 `Figure` 环境中一样，方括号中的位置变量允许我们将表格放置在各种地方。`\centering` 使得表格居中放置。在 `Tabular` 环境中，右侧的 `{}` 表示表格内部元素的对齐方式：

<code>l</code>	<code>r</code>	<code>c</code>	<code>p{3 cm}</code>	<code> </code>
左对齐	右对齐	居中放置	列宽 3 cm	边界

表 3: `tabular` 命令的参数列表

在每一行下部画一个横线可以使用，`\hline` 命令。

## 2 进阶操作

本章涉及高级格式化选项，如参考文献管理、页面设置和字体管理。此外，还介绍了新命令的创建以及与其他格式和编程语言的整合。本章最后对交叉引用、超链接和其他工具进行了一些介绍。

### 2.1 参考管理系统

对于任何需要大量参考文献的大型作品和文件来说，最好先生成一个包含所有参考文献的 `.bib` 文件。这个文件必须按照特定模式进行格式化，然后可以用来在 L<sup>A</sup>T<sub>E</sub>X 中引用。

要生成这个 `.bib` 文件，有两种选择：

- 使用允许导出 `.bib` 文件的参考文献管理程序（由于 L<sup>A</sup>T<sub>E</sub>X 的广泛采用，流行的参考文献管理程序如 Zotero、Mendeley 或 EndNote 都有这种功能）；
- 也可以手动创建一个 `.bib` 文件，这只需要一个简单的文本文件，其中包含一个所需的 `.bib` 格式的参考文献列表。

在 `.bib` 文件中，类型用 `@` 表示（例如：`@book`），然后在大括号内给出条目，字段之间用逗号分隔；

- 第一个字段是关键词（在引用时使用）；
- 作者姓名用 `< 姓, 名 >` 来表示，并用 `"and"` 来分隔；
- 当作者字段中出现机构或组织时，最好用一组额外的大括号将其括起来，以避免自动缩短第一个“名字”。

BIBTeX 用样式 (style) 来管理参考文献的写法。BIBTeX 提供了几个预定义的样式, 如 plain, unsrt, alpha 等。如果使用期刊模板的话, 可能会提供自用的样式。样式文件以 .bst 为扩展名。使用样式文件的方法是在源代码内 (一般在导言区) 使用 `\bibliographystyle` 命令:

```
\bibliographystyle{ < < bib-name > > }
```

这里 `< bst-name >` 为 .bst 样式文件的名称, 不要带 .bst 扩展名。当准备好了 .bib 数据库并且设置好样式文件后, 在正文中使用 `\cite` 命令去引用一个参考文献 [1], 若希望没在正文中被引用的文献也出现在结尾的列表中, 请使用 `\nocite{*}` 命令。最后, 在需要列出参考文献的位置, 使用 `\bibliography` 命令:

```
\bibliography{ < bib-name > }
```

## 2.2 页面设置

在 L<sup>A</sup>T<sub>E</sub>X 中, 页面的大小是在一开始就用 `\documentclass` 命令设置的 (例如, 用 `\documentclass[a4paper]{article}`), 而更精细的调整。有两个包在这方面特别有用, 即 geometry 包和 fancyhdr 包。

参数	作用
<code>\pageheight, \pagewidth</code>	文档页面高度、宽度
<code>\textwidth, \textheight</code>	文本区域的宽度、高度
<code>\linewidth</code>	某一行宽度
<code>\baselineskip</code>	段内行间距
<code>\parskip</code>	自然段间距

表 4: L<sup>A</sup>T<sub>E</sub>X 可用空间参数

有了这些参数作为参考系, 我们可以很容易地定义一些对象的尺寸, 比如我们之前做过的, 我们可以使用 `\includegraphics[width=0.5\linewidth]{}` 来定义一个 50% 行宽度的图片。另一方面, 我们也可以使用确切的数值以及单位: cm, mm, pt, sp, in, a 和 em/ex (分别代表‘M’和‘x’).

### 2.2.1 留边与页面尺寸

### 2.2.2 颜色

### 2.2.3 页眉和页脚

包 fancyhdr 可以帮助我们实现精美的页眉和页脚, 它允许我们对页眉和页脚进行完全定制。页眉和页脚的具体参数要在导言区进行设置, 最常见的是使用命令: 一些特殊的字段, 例如, `\thepage`

参数	作用
<code>\rhead{}</code>	右上文字
<code>\lhead{}</code>	左上文字
<code>\chead{}</code>	中上文字
<code>\rfoot{}</code>	右下角
<code>\lfoot{}</code>	左下角
<code>\cfoot{}</code>	中下部

表 5: fancyhdr 常用字段

将生成页码，类似的还有`\thechapter`和`\thesection`，而`\leftmark`和`\rightmark`则分别生成当前章节的名称和编号（报告或书籍/文章），以及章节/分节（报告或书籍/文章）。

```
\renewcommand{\headrulewidth}{... pt}
```

可以用上述的命令来规定页眉（foot，页脚）与正文之间的分隔线的粗细。为任何一行设置 0pt，就可以删除该行。

在下面的例子中，我们将尝试为我们页面的顶部和底部生成一个横幅，以帮助示范其中的一些命令。因此，在序言中，我们将首先消除页眉、页脚和文本之间的线条，并使页眉也覆盖我们的左右页边距。要做到这一点，我们使用命令`\fancyhfoffset[]{}{}`，并将其设置为等于我们的左右页边距，在本例中为 2.3 厘米，从而将页眉和页脚延伸到文档的边缘这里，还调用了软件包 `xcolor` 和 `graphicx`，以便进行颜色选择和图表支持。

要想在页面顶部加入一个全彩的矩形横幅，我们可以使用 `\fancyhead{}{}` 命令，并在其中心设置一个方框。要获得 L<sup>A</sup>T<sub>E</sub>X 的方框，可以使用命令`\fbox{}{}`，但如果我们想要一个彩色的方框，我们可以使用 `\fcolorbox{frame color}{box background color}{}{}`。

### 2.2.4 使用 listings 进行高亮代码

```
#include<stdio.h>
int main(){
    hello, world!;
}
return 0;
/* \begin{verbatim} ... */
```

```
import numpy as np

def incmatrix(genl1,genl2):
    m = len(genl1)
    n = len(genl2)
    M = None #to become the incidence matrix
    VT = np.zeros((n*m,1), int) #dummy variable
    return M
```

Listing 1: 这是 python 代码

```
#include<stdio.h>
int main(){
    printf("hello, world!");//测试一下中文注释
    printf("hello, world!");//测试一下中文注释
}
return 0;
```

Listing 2: C

### 2.2.5 画图

Tikzit 是一个简单的 GUI 工具，它允许绘制基本的图形和图表。它生成的 Tikz/PGF 输出代码可以直接嵌入在 L<sup>A</sup>T<sub>E</sub>X 代码中，可以在 <https://tikzit.github.io/> 下载。

对于特别是数学绘图、草图、几何图形和其他图表，<https://www.mathcha.io/> 提供了一个易于使用的图形环境，可以很容易地导出到 L<sup>A</sup>T<sub>E</sub>X 代码。

为了绘制更复杂的图形，Inkscape 是一个开放源码的矢量图形编辑器，它允许导出到 PDF，可以让 figures 中的任何文本与主文件共享相同的字体。它可以在 <https://inkscape.org/> 下载。

## 3 浮动对象

### 参考文献

- [1] Austin Benson and Dan Frank. Cme 193: Introduction to scientific python lecture 4: Numpy and scipy. 2013.
- [2] Dilip Datta. *LaTeX in 24 hours: a practical guide for scientific writing*. Springer, 2017.
- [3] Marco Öchsner and Andreas Öchsner. *Advanced LaTeX in Academia: Applications in Research and Education*. Springer Nature, 2021.