

L^AT_EX

贴心教程

BY 张子谋

献给

我接触 L^AT_EX 一年多的时光

前言

学习 \LaTeX 也已经一年多了，不能说用的很熟练，但无论是实验报告还是数模论文甚至还有马原课的论文，我都是使用 \LaTeX 完成的。这个软件给我的感觉就是用了它以后根本停不下来，以后写什么东西，哪怕是不怎么需要排版的一些文字，都会想用它来写。慢慢就觉得使用 Word 不太习惯了。当然，这两个软件都有各自的优势。

刚开始写这个教程时，一直在怀疑我这样做的意义。 \LaTeX 的教程网上一搜一大把，各路高人的书也是随处可见，我这样一个遇到个报错就要百度谷歌的人来凑什么热闹。所以在制作完本教程简易的封面和写了第一章以后就想放弃。我开始给自己找理由，让自己坚持下去。我刚开始学习 \LaTeX 时，在网上搜教程，或者是看书，每个人配置的方法，使用的编译命令都不一样，这让一无所知的我十分头大。而且有些书籍出版时间比较早，使用的中文配置方法并不是当下最优的。还有，有些书籍写的过于详细，这让想快速入门，马上用 \LaTeX 排出一篇论文的人望而生却。我在一年的学习中遇到了很多的问题，这些问题有的网上搜一下就解决了，而有的我在半年后才因为一个偶然的机缘将其解决。我不知道为什么我认为很重要的一个问题在网上很难搜到答案，当然了可能是我搜索能力的问题。

我觉得我找到了写这教程的理由。首先，我想给完全没有接触过 \LaTeX 的人提供一个快速入门的教程；其次，我想分享一些我积累的方法，虽然可能看起来很简单，但可能我是花了好几个小时查资料甚至是半年后才解决的；再次，能够写一本教程，用 \LaTeX 排出一个漂亮的排版，对我而言是一件很兴奋的事情。事实上，我在写这教程的过程中也在不断地学习，原先一些没能解决的问题居然就这样顺便解决了，原先不太理解的概念原理也慢慢清晰了起来。

虽然本教程力图提供一个快速入门的途径，但是讲的并不详细。本教程里讲的东西，可以分两种。一种是，网上一搜一大把，但是说法各不相同；一种是不太容易直接搜索到的东西，也是我一年多来积累到的一些方法。至于随便搜都能搜到而且答案非常唯一的那些东西，本教程基本没有涉及。所以我建议还是结合其他资料一起看。

对于文中的代码，尽量不要直接复制，因为可能会有多余的空格混杂在其中，导致编译报错。一个好的方法是复制完后手动将多余的一些空格删除。文中的所有代码都可以在这里<http://zhangzimou.com/file/latex.7z>下载到，所以读者也可以从那个网址中下载。本着分享的精神，数学建模国赛论文的模板和本教程的 \LaTeX 源码也可以在那里下载到。

由于个人水平有限，有些地方可能理解不到位，所使用的方法也可能不太合适，甚至是繁琐和错误的。再加上时间有限，写的比较仓促，错误和没有说明白的地方在所难免。希望读者能够提出批评指正。反馈邮箱为 zzm@zhangzimou.com。

张子谋

2014 年 10 月于西安交通大学

目录

前言	2
1 软件使用	1
1.1 基本使用方法	1
1.2 Texmaker 使用小技巧	2
2 L^AT_EX 文档	3
2.1 文档结构	3
2.2 中文支持	3
2.2.1 设置中文字体	4
2.2.2 字号	4
2.2.3 中文重定义	5
2.3 其他一些东西	5
3 表格	7
3.1 一般的表格	7
3.2 表格自定义	8
3.2.1 列宽	8
3.2.2 行宽	8
3.2.3 跨行跨列表格	9
3.2.4 宽表格	11
3.2.5 长表格	12
3.3 借助其他工具制作表格	12
4 图片	14
4.1 最简单的例子	14
4.2 稍微高级一点	14
4.2.1 并排摆放, 共享标题	14
4.2.2 题外话: 有用的 minipage	15
4.2.3 并排摆放, 各有标题	16
4.2.4 并排摆放, 共享标题, 各有子标题	16
4.3 灵活使用 minipage	17
4.4 再说点图片的事	20

5	数学公式	21
5.1	一些简单的数学公式	21
5.2	稍微高级的应用	22
5.2.1	括号	22
5.2.2	长公式分行	24
5.2.3	多行公式	25
5.3	一些需要注意的地方	26
6	扯点闲话	28
6.1	<code>\hfill</code> 和 <code>\vfill</code>	28
6.2	<code>\hspace</code> 和 <code>\vspace</code>	30
6.3	参考文献	31
6.4	数学建模国赛论文格式	31
6.4.1	承诺书	31
6.4.2	插入代码	33
	后记	35

第 1 章 软件使用

1.1 基本使用方法

在官网下载 texlive，或者直接在百度云搜索 texlive2014.iso（版本都差不多），然后针对自己的操作系统安装。具体怎么安装就不说了，非常简单。安装好后如果使用的是 Windows，在 CMD（命令行）中应该会有 latex、xelatex 等等这些命令。本教程使用的都是 xelatex 命令编译，这个命令的好处就是可以直接调用操作系统的字体，而不需要自己额外安装字体。

编辑器推荐使用 Texmaker，这也是我正在用的软件。安装好这个软件后需要设置编译选项，在选项（Options）下选择 Configure texmaker，得到如图1.1所示界面。可以看到有 latex、

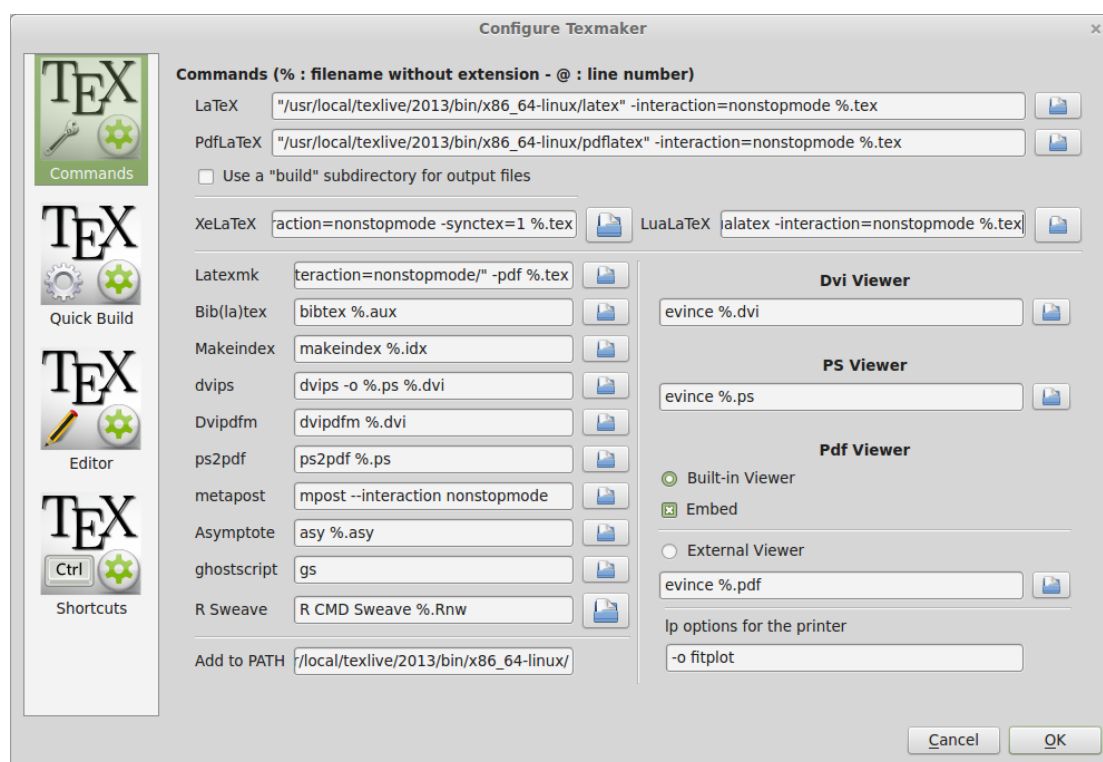


图 1.1: Configure texmaker 界面

pdflatex、xelatex 等编译选项，将其后的路径改为安装 texlive 时这些命令的安装目录，比如我的 xelatex 编译选项为

```
/usr/local/texlive/2013/bin/x86_64-linux/xelatex -interaction=nonstopmode -  
synctex=1 %.tex
```

其中.../xelatex 是 xelatex 这个命令的位置，后面的参数不加也可以，但是加了以后可以

起到一个交互界面的作用，比如在 Texmaker 中预览 pdf 时，按下 Ctrl 键并同时鼠标点击 pdf 中的某一部分内容，编辑框内的光标就会自动跳到所点击的那部分内容。读者可以自己试试。

接着选择 Quick Build，如图1.2所示。前面已经说了，本教程的编译命令都用的是 xelatex，

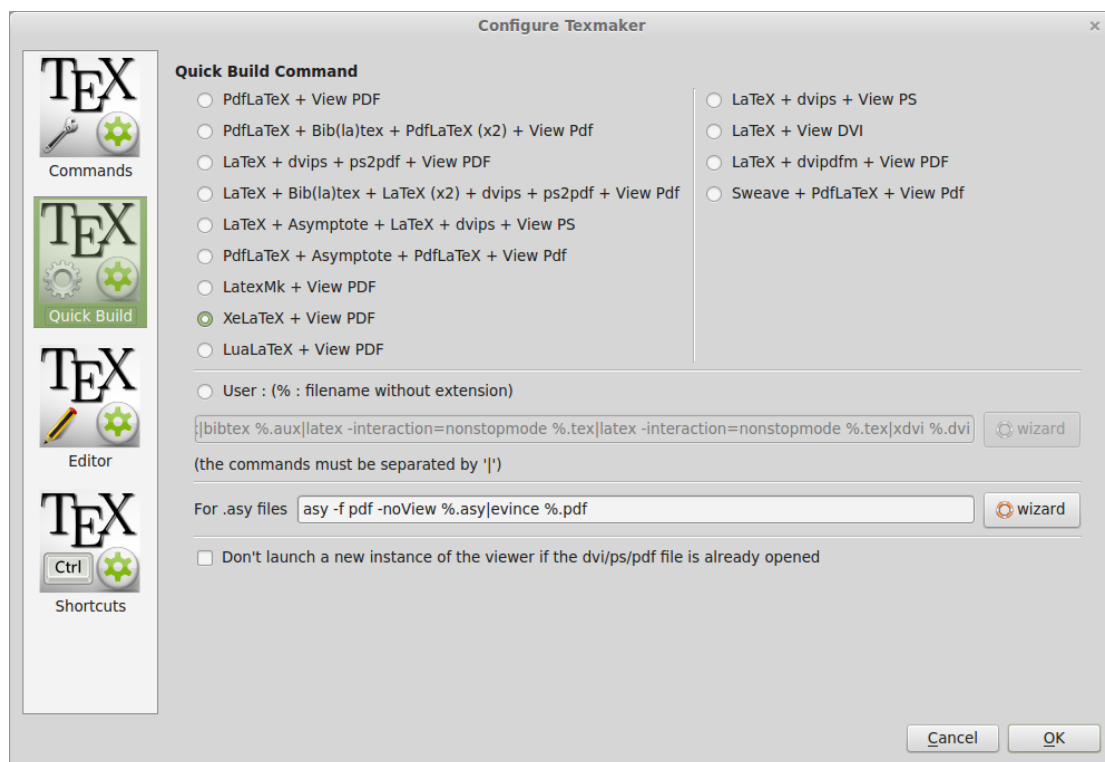


图 1.2: Quick Build 选项

所以选择了 xelatex+ViewPDF，意思就是按下 quick build 快捷键（F1）后，先对代码进行编译，然后在软件右边的 pdfViewer 中显示出编译出的 pdf 文件。

这个软件的其他设置可以自己摸索，按照以上设置后就可以开始使用 \LaTeX 了!!

1.2 Texmaker 使用小技巧

所谓的小技巧也都很简单，我一般也就用两个，一个是用户自定义的命令，一个是交互式界面。Texmaker 最左边有一个人形的选项，叫“user”，如图1.3所示。这里可以添加一些自己的命令。比如插入图片时总是要输入一大堆的代码，很麻烦，就把这些代码放到里面去，要插入图片的时候点击那个 figure，就自动在光标处添加了一段代码，只需要修改一些参数就能方便地插入图片了。

另外一个比较有用的小技巧就是交互界面了。在编译的时候加上这个选项：

```
-interaction=nonstopmode -syncTeX=1
```

这样点击文档的各种结构时 pdfViewer 中的预览也会自动跳到点击的位置。还有可以设置快捷键“Jump to pdf”，这样按下这个快捷后 pdfViewer 中的预览也会跳到当前光标所在处。这些东西自己试试就知道了。

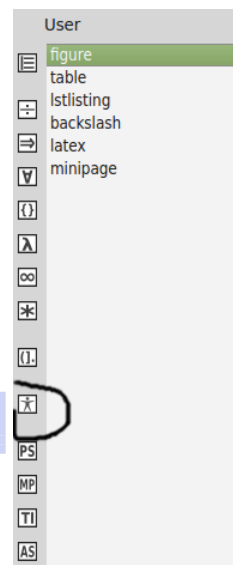


图 1.3: user 设置

第2章 L^AT_EX 文档

2.1 文档结构

这节来说说 L^AT_EX 的文档结构。首先来看一个简单的演示代码：

```
1 \documentclass[12pt,a4paper]{article}
2 \usepackage{xCJK}%使用 xCJK 宏包
3 \usepackage{fontspec}
4 \usepackage{titlesec}
5 %中文断行
6 \XeTeXlinebreaklocale "zh"
7 \XeTeXlinebreakskip = 0pt plus 1pt minus 0.1pt
8 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9 \begin{document}
10 \title{半导体材料的霍尔效应参数测试{ }}
11 \author{微电子{23 张子谋 2120503073}}
12 \date{}
13 \maketitle
14 \section{实验数据处理与分析}
15 \section{思考题}
16 \end{document}
17 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

第 9 行到第 16 行是正文，第 9 行以前称为导言区（具体的请参考书籍），第 17 行以后的所有内容都被忽略。第 1 行用于设置文档类，一般论文都用 `article` 文档类，这里设置正文 12pt 的字号，使用 A4 纸。以 `\` 开头的表示 L^AT_EX 中的命令。`\usepackage{}` 命令表示使用宏包，比如 `xCJK` 宏包可以支持中文字体，`titlesec` 宏包用于更改大标题的样式。每种宏包都提供各自的命令。`\usepackage{}` 命令就相当于 C 语言中的 `include` 命令，而宏包就相当于头文件。

各种设置基本都是放在导言区中，且有的设置只能放在导言区。由于我们编译时使用的是 `xelatex`，所以对于中文论文来说需要加上第 6 行和第 7 行，这两个命令可以对中文实现自动的断行。

2.2 中文支持

上一节使用了 `xCJK` 宏包，这个宏包可以支持中日韩三国的字体。

2.2.1 设置中文字体

首先要知道自己电脑上中文字体的名字。对于 Linux 用户，可以在命令行输入

```
fc-list :lang=zh
```

查看已安装的中文字体。Windows 用户嘛，我也不知道，不过这个其实不太重要，等下就会看到。

首先要设置论文的主字体，就是正文使用的字体。中文一般使用的是宋体，而英文一般用 Times New Roman，所以可以在导言区中加入：

```
1 \setCJKmainfont{宋体}
2 \setmainfont{Times New Roman}
```

这样设置的前提是电脑中安装了这两个字体，对于 Windows 用户来说应该是都安装了，Linux 用户就自己去找找如何安装 Windows 下的字体吧，或者安装 WPS，这样也会自动安装一些的中文字体。这样设置后正文中凡是中文，使用的都是宋体，凡是英文使用的都是 Times New Roman。

接着再添加其他的字体，同样在导言区中加入：

```
1 \setCJKfamilyfont{zh song}{SimSun} %宋体
2 \setCJKfamilyfont{zh hei}{SimHei} % 黑体
3 \setCJKfamilyfont{zh fs}{FangSong} %仿宋
4 \setCJKfamilyfont{zh kai}{KaiTi} %楷体
5 \newcommand*{\song}{\CJKfamily{zh song}} % 宋体
6 \newcommand*{\hei}{\CJKfamily{zh hei}} % 黑体
7 \newcommand*{\kai}{\CJKfamily{zh kai}} % 楷书
8 \newcommand*{\fangsong}{\CJKfamily{zh fs}} % 仿宋
```

不用太纠结这段代码的意思，可以用就行了。第 1 行到第 4 行第二个花括号里的必须是电脑中安装的字体名称，也可以直接写中文，比如 SimSun 就用宋体代替，SimHei 就用黑体代替。第一个花括号内的是自定的名称，要与后面第 5 到第 8 行相对应。newcommand 是定义一个新的命令，以后在文中如果想使用这些字体，那么就成这样：

```
{\hei 这是一段黑体的文字，}{\kai 这是一段楷体的文字，}{\fangsong 这是仿宋。}
```

花括号表示一个一个环境，字体的改变只会在花括号中起作用。得到的效果为：
这是一段黑体的文字，这是一段楷体的文字，这是仿宋。

2.2.2 字号

在导言区中加入以下代码，自定义字号的命令：

```
1 \newcommand{\erhao}{\fontsize{22pt}{\baselineskip}\selectfont}%二号
2 \newcommand{\xiaoer}{\fontsize{18pt}{\baselineskip}\selectfont}%小二
3 \newcommand{\sanhao}{\fontsize{16pt}{\baselineskip}\selectfont}%三号
4 \newcommand{\xiaosan}{\fontsize{15pt}{\baselineskip}\selectfont}%小三
5 \newcommand{\sihao}{\fontsize{14pt}{\baselineskip}\selectfont}%四号
6 \newcommand{\xiaosi}{\fontsize{12pt}{\baselineskip}\selectfont}%小四
```

```

7 \newcommand{\wuhao}{\fontsize{10.5pt}{\baselineskip}\selectfont}%五号
8 \newcommand{\xiaowu}{\fontsize{9pt}{\baselineskip}\selectfont}%小五

```

中文习惯用几号几号来表示字的大小，而英文习惯用磅数来表示字号的大小，之间的换算可以参照上面这个代码。如果想要更大或者更小的字号，可以自己设置。同样的，下面这段代码：

```
{\erhao 这是二号字，}{\sanhao 这是三号字，}{\wuhao 这是五号字。}
```

得到的效果为：

这是二号字，这是三号字，这是五号字。

当然了，字号和字体可以混合使用。

2.2.3 中文重定义

L^AT_EX 的一大特点就是在你自定义完模版后可以实现自动的排版。比如插入一个表格，表头是这样的：“表 1.1：标题”。在插入表格时实际只需要输入“标题”，“表 1.1：”是 L^AT_EX 自己加上去的。由于 L^AT_EX 默认是英文，所以如果不加设置的话，实际的效果是：“table1.1：标题”。这对中文论文来说显然是不行的。还有很多类似的地方需要重新设置。在导言区中加入如下代码：

```

1 \renewcommand{\contentsname}{目录}
2 \renewcommand{\listfigurename}{插图目录}
3 \renewcommand{\listtablename}{表格目录}
4 \renewcommand{\refname}{参考文献}
5 \renewcommand{\abstractname}{摘\ 要}
6 \renewcommand{\indexname}{索引}
7 \renewcommand{\tablename}{表}
8 \renewcommand{\figurename}{图}
9 \newtheorem{theorem}{定理}
10 \newtheorem{definition}{定义}
11 \newtheorem{property}{问题}
12 \newtheorem{proposition}{猜测}
13 \newtheorem{lemma}{引理}
14 \newtheorem{corollary}{推论}

```

这样就实现了中文重定义。

2.3 其他一些东西

至于怎么换行，怎么输入空格，本教程就不细说了，网上的资料很丰富。这里再说说其他的一些东西。

实际中会需要使用很多的宏包，所以干脆将那些会用到的和可能用到的宏包都加进去，这样也省事。如果编译的时候报错：

! Undefined control sequence.

如果你命令没有输错，那应该就是没有添加有关的宏包了。可以将下面这些加宏包的代码放

在导言区：

```
1 \usepackage{fancyhdr}%页眉页脚
2 \usepackage{xeCJK} %中文支持
3 \usepackage{fontspec}%字体
4 \usepackage{titlesec}%标题格式
5 \usepackage[top=2.5cm,bottom=2.5cm,left=2cm,right=2cm]{geometry}%页边距
6 \usepackage{amsmath} %数学公式
7 \usepackage{amssymb}
8 \usepackage{amscd}
9 \usepackage{listings}%插入代码
10 \usepackage{xcolor} %字体颜色
11 \usepackage{graphicx} %插入图片
12 \usepackage{subfig} %子图
13 \usepackage{tabularx}%插入表格
14 \usepackage{indentfirst} %首行缩进
15 \usepackage{array}
16 \usepackage{longtable}%长表格
17 \usepackage{multirow}%使用多栏宏包
18 \usepackage{wrapfig}%文字环绕
19 \usepackage{booktabs}
20 \usepackage{extrarrows}
21 \usepackage{ulem}
22 \usepackage{txfonts}
23 \usepackage{bm}
24 \usepackage{cite}%参考文献
25 \usepackage[super,square,comma,sort&compress]{natbib}
26 \usepackage{setspace}%设定行距
27 \usepackage[colorlinks,linkcolor=black,anchorcolor=black,citecolor=black,
    CJKbookmarks=True]{hyperref}
```

对于中文文章，格式要求首行缩进两个中文字符，所以首先添加 `indentfirst` 宏包，并且在导言区加入以下代码：

```
\setlength{\parindent}{2.45em}
```

这样首行就会缩进两个中文字符了。

第3章 表格

3.1 一般的表格

对于一个一般的表格如表3.1所示。

表 3.1: 这里是表头

	x1	x2	x3	x4
A	3	4	2	1
B	2	5	4	3
C	5	4	3	1
D	3	3	2	4

代码如下：

```
1 \begin{table}[htp] %浮动表格环境
2 \centering 表格居中放置 %
3 \caption{这里是表头}
4 \label{tab1} %用于引用
5 \begin{tabular}{c|c|c|c|c}%表格环境
6 \toprule[1.4pt] %1.4pt 的上划线
7 &x1&x2&x3&x4\\ \hline %\hline表示画一条横线
8 A&3&4&2&1\\ \hline
9 B&2&5&4&3\\ \hline
10 C&5&4&3&1\\ \hline
11 D&3&3&2&4\\
12 \bottomrule[1.4pt]%1.4pt 的下划线
13 \end{tabular}
14 \end{table}
```

通常来说 `\begin{tabular}` 就是一个表格环境，但是这个表格是紧接着前面的文字的，并且其后的文字也会紧跟在后面，无法像表3.1一样单独占据一个空行的位置。代码第一行 `\begin{table}` 表示浮动表格的环境，`[htp]` 是可选参数，也可以不加。`h(here)` 表示这个浮动体放在当前位置，`b(bottom)` 表示放在一页的底部，`t(top)` 表示放在一页的最顶上，`p` 表示放在一个浮动页（单独一页）。三个参数都加表示让 \LaTeX 来自动判断并选择最合适的情况。通常 \LaTeX 都会自己判断并选择最合适的情况，如果自己强制放在某个位置，可以加`!`，比如 `[!h]` 表示强制放在当前位置。

`tabular` 环境后的 `{c|c|c|c|c}`，其中一共有 5 个 `c` 表示表格有五列，`|` 表示画竖线，`c` 表示居中。`c` 也可以换成别的，比如 `r`(居右)，`f`(居左)。

对于表格的每一行，使用 `&` 将列隔开，使用 `\\` 换行到下一行。`\hline` 表示画一条横线。对于表格最上和最下的横线，加粗可能会更好看，所以可以用 `\toprule[1.4pt]` 将最上的横线设为 1.4pt 的宽度。对最下的横线使用 `\bottomrule[1.4pt]`。

3.2 表格自定义

3.2.1 列宽

表格3.1看起来有点窄，如果宽点应该会更好看，比如表3.2这样。

表 3.2: 这里是表头

	x1	x2	x3	x4
A	3	4	2	1
B	2	5	4	3
C	5	4	3	1
D	3	3	2	4

这个很好实现，只需要将第一行的每一列设置一个宽度就可以了。上文中代码第七行很简单，就是 `&x1&x2&x3&x4`。现在改为

```
\makebox[3em]{}&\makebox[6em]{x1}&\makebox[6em]{x2}&\makebox[6em]{x3}&\makebox[6em]{x4}\\hline
```

其中 `\makebox[宽度]{文本}` 表示生成一个 [宽度] 的盒子，这样第一列的宽度就是 3em，第二列到第五列的宽度都设为了 6em。em 是长度单位，表示当前字体的字符 "M" 的宽度。也可以用 cm、mm、pt 等等。

3.2.2 行宽

如果表3.2中某一元素换为一串很长的句子，如果不对表格进行修改，那么就如表3.3所示。

表 3.3: 这里是表头

	x1	x2	x3	x4
A	今天天气不错啊呵呵	4	2	1
B	2	5	4	3
C	5	4	3	1
D	3	3	2	4

可以看到，由于句子过长超出了原来设定的宽度，所以表格被自动拉宽了。现在我不想让表格被拉宽，我可以考虑将第二行变宽，如表3.4。

同样地可以使用盒子来实现。`\parbox[c][3em][c]6em{文本}` 表示生成一个竖直长度为 3em，水平长度为 6em 的盒子。c 表示本文居中放置，也可以使用 r(right)，l(left)。表3.4的代码如下：

表 3.4: 这里是表头

	x1	x2	x3	x4
A	今天天气不错 啊啊啊	4	2	1
B	2	5	4	3
C	5	4	3	1
D	3	3	2	4

```

1 \begin{table}[htp]
2 \centering\caption{这里是表头}\label{tab4}
3 \begin{tabular}{c|c|c|c|c}
4 \toprule[1.4pt]
5 \makebox[3em]{}&\makebox[6em]{x1}&\makebox[6em]{x2}&\makebox[6em]{x3}&\makebox[6
   em]{x4}\\ \hline
6 A&\parbox[c][3em][c]{6em}{今天天气不错啊啊啊}&4&2&1\\ \hline
7 B&2&5&4&3\\ \hline
8 C&5&4&3&1\\ \hline
9 D&3&3&2&4\\
10 \bottomrule[1.4pt]
11 \end{tabular}
12 \end{table}

```

3.2.3 跨行跨列表格

现在想实现一个如表4.6所示的表格。从表4.6中可以看出，出行方式其实占了两行，而出行目的其实占了三列。

图 3.1: 跨行跨列表格

出行方式	出行目的		
	工作商务出行	非工作出行	上学出行
公交车	28.16	6.35	3.18
出租车	38	8.57	4.3
社会小车	49	11	5.5
各出行所占比例	25.10%	71.30%	3.60%

要使用跨行跨列表格，首先要添加 `multirow` 宏包，在导言区加上 `\usepackage{multirow}`。对于跨行，可以使用 `\multirow` 命令，该命令格式为 `\multirow{要合并的行数}{该栏文本的宽度}{文本}`。如果想让 \LaTeX 自行决定文本的宽度，就在“该栏文本宽度”这个参数里使用 `*`。

对于如表3.5所示的表格，可以使用如下代码：

```

1 \begin{table}[htbp]
2 \begin{center}

```

表 3.5: 跨行表格

A	B	C	D
	E	F	G
0.5	-1.22	0.68	-1.06

```

3 \caption{跨行表格}
4 \begin{tabular}{cccc}
5 \toprule[1.4pt]
6 \multirow{ 2}{*}{A} & B & C & D \\
7 & E & F & G \\
8 0.5 & -1.22 & 0.68 & -1.06 \\
9 \bottomrule[1.4pt]
10 \end{tabular}
11 \label{tab6}
12 \end{center}
13 \end{table}

```

对于跨列，使用 `\multicolumn` 命令，格式为

```
\multicolumn{要合并的列数}{对齐方式}{文本}
```

其中对齐方式可以选择 c(居中), r(居右), l(居左)。对于表3.6所示的表格，代码如下：

表 3.6: 跨列表格

A	B		
C	E	F	G
0.5	-1.22	0.68	-1.06

```

1 \begin{table}[htbp]
2 \begin{center} 和 \是一样的，表格居中放置centering
3 \caption{跨列表格}
4 \begin{tabular}{c|c|c|c}
5 \toprule[1.4pt]
6 A&\multicolumn{3}{c}{B} \\
7 C&E&F&G \\
8 0.5 & -1.22 & 0.68 & -1.06 \\
9 \bottomrule[1.4pt]
10 \end{tabular}
11 \label{tab7}
12 \end{center}
13 \end{table}

```

这样，结合跨行和跨列，表4.6的代码为：

```
1 \begin{table}[h]
```

```

2 \begin{center}
3 \caption{跨行跨列表格}
4 \begin{tabular}{cccc}
5 \toprule[1.4pt]
6 \multirow{ 2}{*}{出行方式} & \multicolumn{ 3}{c}{出行目的} \\ \cline{ 2- 4}
7 & 工作商务出行 & 非工作出行 & 上学出行 \\ \hline
8 公交车 & 28.16 & 6.35 & 3.18 \\ \hline
9 出租车 & 38 & 8.57 & 4.3 \\ \hline
10 社会小车 & 49 & 11 & 5.5 \\ \hline
11 各出行所占比例 & 25.10\% & 71.30\% & 3.60\% \\ \hline
12 \bottomrule[1.4pt]
13 \end{tabular}
14 \label{tab5}
15 \end{center}
16 \end{table}

```

3.2.4 宽表格

有的时候一个表格太宽，如表3.7所示。可以发现，现在这个表格没有居中了。上文中为

表 3.7: 宽表格

这是一个很宽的表格	这真的是一个很宽的表格	真的非常非常非常的宽啊啊	不骗你，真的很宽
C	E	F	G

了使得表格居中，是将 `tabular` 放在 `center` 环境中，就像这样：

```

1 \begin{table}[h]
2 \begin{center}
3 \begin{tabular}{cc}
4 ...
5 \end{tabular}
6 \end{center}
7 \end{table}

```

或者是这样：

```

1 \begin{table}[h]
2 \centering
3 \begin{tabular}{cc}
4 ...
5 \end{tabular}
6 \end{table}

```

这种让表格居中的方法在表格过宽后失效，解决办法就是将 `tabular` 放在 `centerline` 环境中，代码如下：

```

1 \begin{table}[h]
2 \caption{宽表格}

```



```

3 \centerline{ %centerline环境开始
4 \begin{tabular}{c|c|c|c}
5 \toprule[1.4pt]
6 这是一个很宽的表格&这真的是一个很宽的表格&真的非常非常非常的宽啊啊&不骗你，真的很宽\\ \hline
7 C&E&F&G \\
8 \bottomrule[1.4pt]
9 \end{tabular}
10 } %centerling环境结束
11 \label{tab9}
12 \end{table}

```

效果如表3.8所示。注意，使用 `centerling` 时 `\caption{}` 不能放在里面，否则会报错。而使用 `centering` 时无所谓，放在里面还是外面都一样，但是使用 `center` 环境时应该放在里面，否则表头会距离表格太远，读者可以自己试试。

表 3.8: 宽表格

这是一个很宽的表格	这真的是一个很宽的表格	真的非常非常非常的宽啊啊	不骗你，真的很宽
C	E	F	G

3.2.5 长表格

如图3.2所示，这个是我们所要的效果，但是这个表格太长，无法在当前位置直接放下，如果用上文中的方法制作这个表格，那么 \LaTeX 会自动将表格后的文字接在“2 符号约定”后面，而把这个表格放于后一页的顶部。显然，这样不符合规范。解决的方法就是使用长表格。首先在导言区中添加宏包

```
\usepackage{longtable}
```

然后使用 `longtable` 环境使用长表格。简单的代码如下：

```

1 \begin{longtable}{c|c}
2 \toprule[1.4pt]
3 \makebox[3em]{符号}{ }&\makebox[6em]{符号说明} \\ \hline
4 $\theta$&最外木条与地面所成角度 \\ \hline
5 $x_i$&第$i$根木条的等效横坐标 \\
6 \bottomrule[1.4pt]
7 \end{longtable}

```

这里就不需要使用 `table` 和 `tabular` 环境了，直接用 `longtable` 就行。关于 `longtable` 更详细的用法请参阅有关资料。

3.3 借助其他工具制作表格

从上文可以看出 \LaTeX 制作表格是件很麻烦的事，如果表格中数据量大，那么这样一个一个输入要输入到什么时候？而且可以直接导入 Excel 中的表格吗？

1. 给定长方形平板尺寸为 $120\text{cm} \times 50\text{cm} \times 3\text{cm}$ ，每根木条宽 2.5cm ，连接桌腿木条的钢筋固定在桌腿最外侧木条的中心位置，折叠后桌子的高度为 53cm 。试建立模型描述此折叠桌的动态变化过程，在此基础上给出此折叠桌的设计加工参数和桌脚边缘线的数学描述。
2. 折叠桌的设计应做到产品稳固性好、加工方便、用材最少。对于任意给定的折叠桌高度和圆形桌面直径的设计要求，讨论长方形平板材料和折叠桌的最优设计加工参数，例如，平板尺寸、钢筋位置、开槽长度等。对于桌高 70cm ，桌面直径 80cm 的情形，确定最优设计加工参数。
3. 公司计划开发一种折叠桌设计软件，根据客户任意设定的折叠桌高度、桌面边缘线的形状大小和桌脚边缘线的大致形状，给出所需平板材料的形状尺寸和切实可行的最优设计加工参数，使得生产的折叠桌尽可能接近客户所期望的形状。给出这一软件设计的数学模型，并根据所建立的模型给出几个创意平板折叠桌。要求给出相应的设计加工参数，画出至少 8 张动态变化过程的示意图。

2 符号约定

符号	符号说明
θ	最外木条与地面所成角度
x_i	第 i 根木条的等效横坐标
a_i	第 i 根木条的长度
s_i	第 i 根木条的槽长
n	桌面一侧的木条数
l	平板长度的一半
r	平板宽度的一半，也是圆的半径
ϕ	桌子不倾倒的最大倾角

2

k	钢筋位置
θ_m	θ 的极限角
H	桌子高度
h	桌子重心的 z 坐标
f	桌子重心到最外桌脚沿 y 方向的距离

图 3.2: 长表格示意

针对数据量大的表格，可以先将数据输入到 Excel 中，这样方便快捷。接着可以借助于 Excel 转 LaTeX 插件，将 Excel 的表格转换成 LaTeX 代码。由于我是 Linux 系统，没有 Excel，所以就不演示了。这样的插件很好找，使用也很简单。对于 Linux 用户，使用系统自带的 Libreoffice Calc 软件，下载安装 Calc2LaTeX 插件，可以实现一样的效果。

第4章 图片

首先要说明一点， \LaTeX 中插入图片是一件比较头疼的事情，在这一点上比不过 Word。尽管如此，在掌握熟练以后还是能比较轻松地实现图片的插入的。

4.1 最简单的例子

最简单的例子如下，从本教程开始一直到这里的所有图片都是使用如下代码插入的。

```
1 \begin{figure}[htp] %图片浮动环境，类似表格中的 table [htp] 参数和表格的类似
2 \centering %图片居中
3 \includegraphics[height=10cm]{picture.png}%插入图片，设置图片的高度，图片的文件名
4 \caption{}%图片的标题
5 \label{} %引用
6 \end{figure}
```

[htp] 这个参数和表格中的一样，就不赘述了。`\includegraphics{}` 命令用于插入图片，[] 中是可选参数，不加默认为图片原始大小。`[height=10cm]` 表示图片的高度为 10cm，`[width=5cm]` 表示图片的宽度为 5cm。在只加这一个参数的情况下会保持图片的比例不变。如果是 `[height=10cm,width=5cm]`，则插入图片的高度为 10cm，宽度 5cm，比例可能就改变了。另一种方法是 `[scale=2]`，表示图片是原始大小的 2 倍。{} 中是图片的位置，如果图片在当前路径下输入图片的文件名就行。

4.2 稍微高级一点

4.2.1 并排摆放，共享标题

这里演示将两张图片并排摆放，并且共享标题。使用如下代码，效果如图4.1所示。

```
1 \begin{figure}[h]
2 \centering
3 \includegraphics[height=7cm]{figh_3.pdf}
4 \includegraphics[height=7cm]{figh_5.pdf}
5 \caption{并排摆放，共享标题}
6 \label{pic:fig1}
7 \end{figure}
```

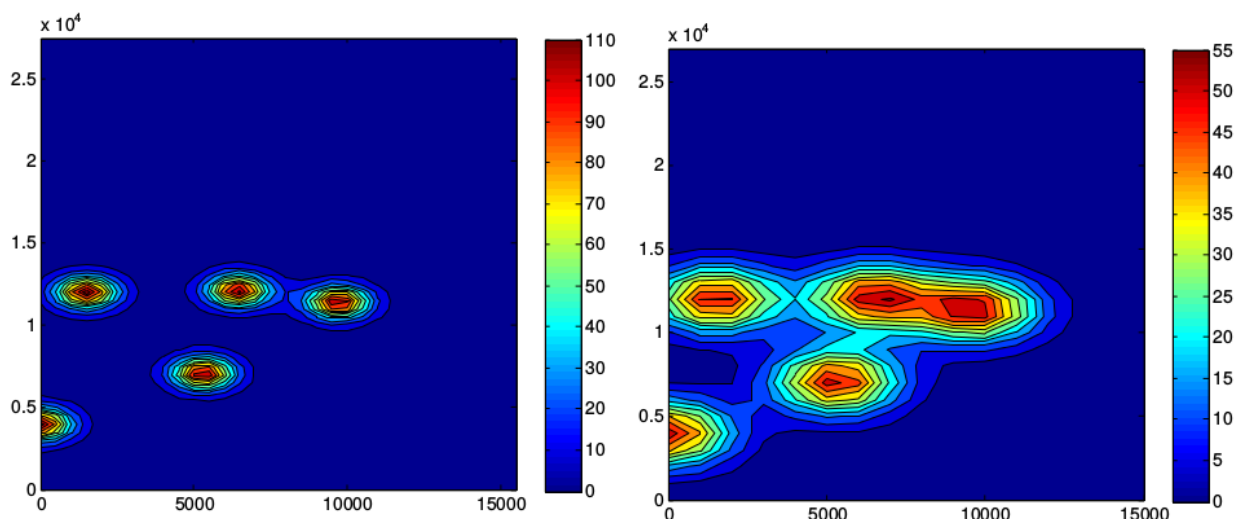


图 4.1: 并排摆放，共享标题

4.2.2 题外话：有用的 minipage

现在先转移一下话题，来看看一个非常有用的环境：`minipage`。按照字面意思来理解，就是“小页”。它可以将很难直接放在一起的两个部分放在一起，比如说图片，比如说表格。例如下面这个例子：

`minipage` 就是一个给定宽度的盒子。这是一个 `minipage` 的例子，将这一行分成三个部分，这个部分的底部和右边中线对齐，

这个部分的中线和左边的底部和右边的顶部对齐，字数太少了可能看的不明显，我再凑些字数呵呵

这个部分的顶部和左边的中线对齐。是不是很神奇？`minipage` 是一个很有用的环境，可以将图片、表格放在一起。

从上面这个例子可以看出有三个 `minipage` 并排摆放。`minipage` 的使用格式为：

```
\begin{minipage}[aligned]{width}...\end{minipage}
```

其中 `aligned` 为对齐方式，`c`、`t`、`b` 分别表示中线、顶行、低行与当前基线对齐。如果不加这个可选参数则默认是 `c`。参数 `width` 表示 `minipage` 的宽度。所以上面那三个 `minipage` 的代码如下：

```
1 \begin{minipage}[b]{0.33\textwidth} %\textwidth表示当前页面的宽度
2 minipage 就是一个给定宽度的盒子。这是一个 minipage 的例子，将这一行分成三个部分，这个部分的底部
   和右边中线对齐，
3 \end{minipage}
4 \hfill %\hfill表示水平填充，可看成是指示上下两个 minipage 的相对位置。
5 \begin{minipage}{0.33\textwidth}
6 这个部分的中线和左边的底部和右边的顶部对齐，字数太少了可能看的不明显，我再凑些字数呵呵
7 \end{minipage}
```

```

8 \hfill
9 \begin{minipage}[t]{0.32\textwidth}
10 这个部分的顶部和左边的中线对齐。是不是很神奇？minipage 是一个很有用的环境，可以将图片、表格放在
    一起。
11 \end{minipage}

```

4.2.3 并排摆放，各有标题

为了实现并排摆放，各有标题，需要使用 minipage 环境。先看效果，如下图所示。显然

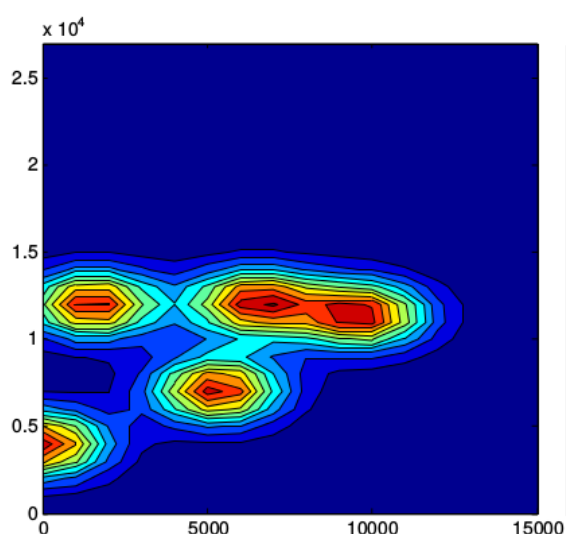


图 4.2: 扩散 5 小时

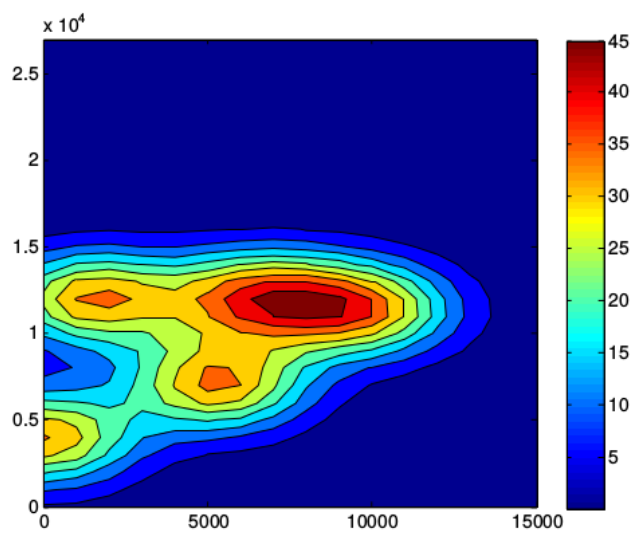


图 4.3: 扩散 7 小时

我们需要两个 minipage。代码如下：

```

1 \begin{figure}[h]
2 \begin{minipage}{0.5\textwidth}
3   \centering
4   \includegraphics[width=\textwidth]{figh_5.pdf}
5   \caption{扩散 5 小时}
6 \end{minipage}
7 \begin{minipage}{0.5\textwidth}
8   \centering
9   \includegraphics[width=\textwidth]{figh_7.pdf}
10  \caption{扩散 7 小时}
11 \end{minipage}
12 \end{figure}

```

4.2.4 并排摆放，共享标题，各有子标题

如果想要两幅并排的图片共享标题，并各有子标题，可以使用 subfig 宏包提供的 \subfloat 命令。效果如下图所示。

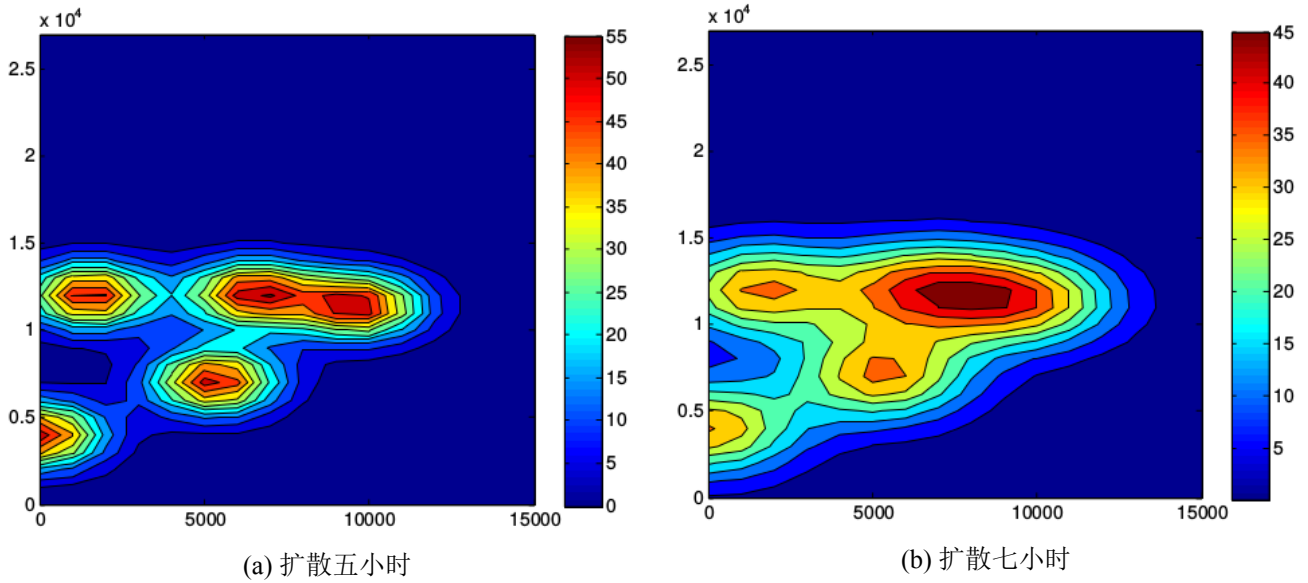


表 4.1: PM2.5 浓度随扩散时间的变化

可以直接使用 `\subfloat` 命令实现这个功能，但是如果图片的标题超出了图片的宽度，就会出现折行，所以最好的办法是在 `\subfloat` 命令里面嵌套一个 `minipage`，因为 `minipage` 是有宽度的。代码如下：

```

1 \begin{figure}[htbp]
2 \subfloat[扩散五小时]{
3 \begin{minipage}{0.5\textwidth}
4 \centering
5 \label{pic:figh_5}
6 \includegraphics[width=\textwidth]{figh_5.pdf}
7 \end{minipage}
8 }
9 \subfloat[扩散七小时]{
10 \begin{minipage}{0.5\textwidth}
11 \centering
12 \includegraphics[width=\textwidth]{figh_7.pdf}
13 \label{pic:figh_7}
14 \end{minipage}
15 }
16 \caption{PM2.5浓度随扩散时间的变化}
17 \end{figure}

```

4.3 灵活使用 minipage

学习了以上内容，摆放图片就变得轻松了起来。比如想要实现下图：
很简单，代码如下：

```

1 \begin{figure}[h]
2 \subfloat[扩散三小时]{

```

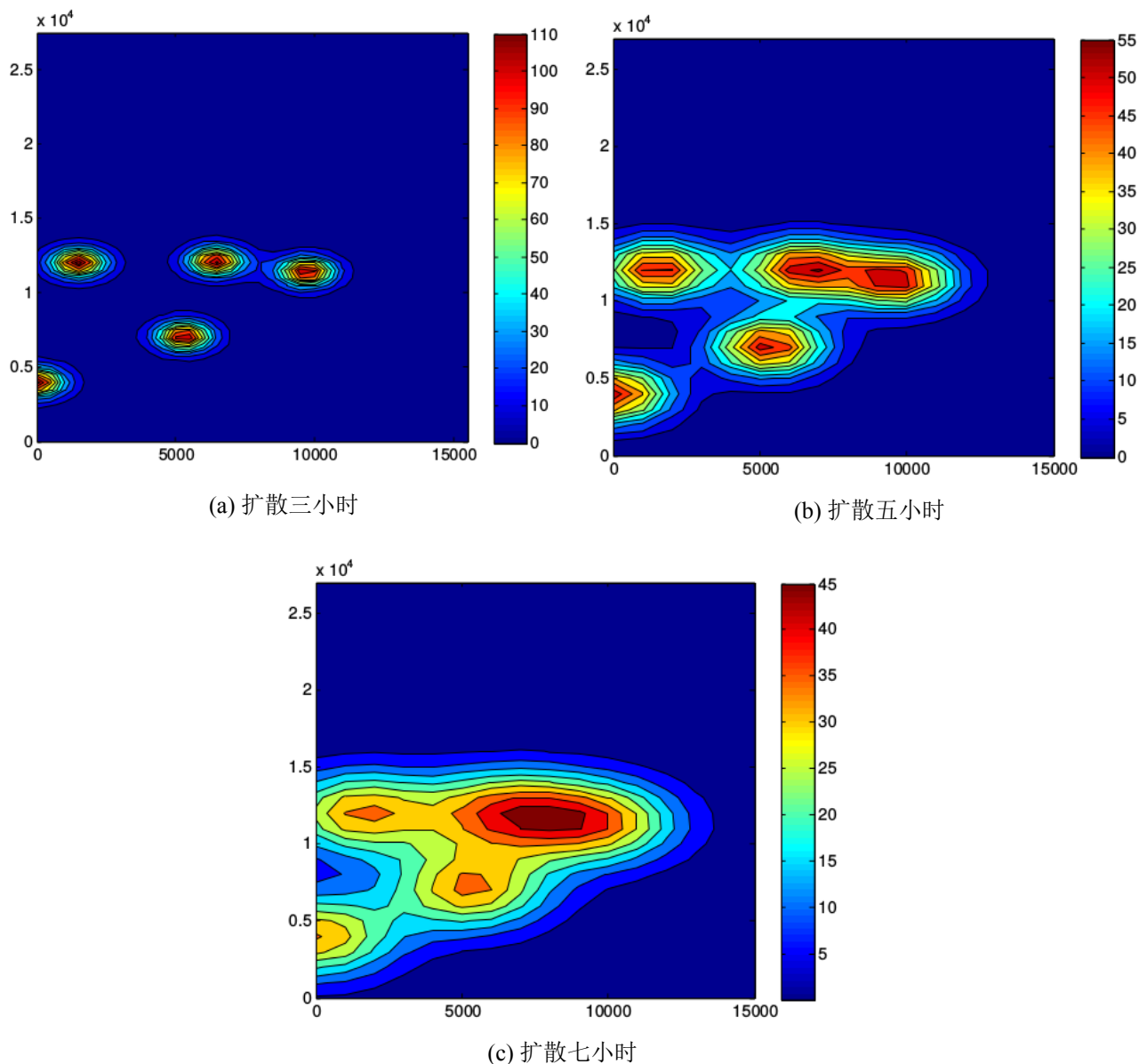


图 4.4: PM2.5 浓度随扩散时间的变化

```

3 \begin{minipage}{0.5\textwidth}
4   \centering
5   \includegraphics[width=\textwidth]{figh_3.pdf}
6   \end{minipage}
7 }
8 \subfloat[扩散五小时]{
9 \begin{minipage}{0.5\textwidth}
10  \centering
11  \includegraphics[width=\textwidth]{figh_5.pdf}
12  \end{minipage}
13 }
14 \vfill    %数值填充，理解为指示上下部分的相对位置
15 \centering %居中

```

```

16 \subfloat[扩散七小时]{
17 \begin{minipage}{0.5\textwidth}
18   \centering
19   \includegraphics[width=\textwidth]{figh_7.pdf}
20   \end{minipage}
21 }
22 \caption{PM2浓度随扩散时间的变化.5}
23 \end{figure}

```

还可以实现如下图表混排：

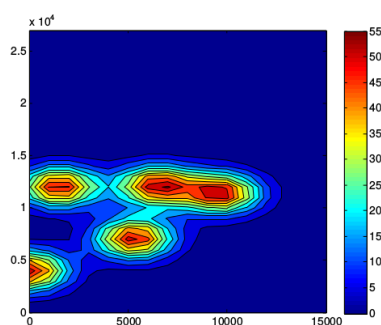


图 4.5: 这是图片

图 4.6: 这是表格

出行方式	出行目的		
	工作商务出行	非工作出行	上学出行
公交车	28.16	6.35	3.18
出租车	38	8.57	4.3
社会小车	49	11	5.5
各出行所占比例	25.10%	71.30%	3.60%

这个有一点难度，关键是搞清楚插入图片和表格的机理。上一章讲插入表格，用到了 `table` 环境，这一章讲插入图片，用到了 `figure` 环境。这两个环境没有什么区别（好吧可能有细微的差别），你可以试试把上面的 `figure` 全部换成 `table`，或者把 `table` 全部换成 `figure`，编译出来的效果都是一样的。这两个环境其实就是提供一个浮动环境，让这环境里面的内容可以脱离上下文的联系，浮动到任何一个地方。既然这样，图片混排可以用如下方法实现：

```

1 \begin{figure}[h] %浮动环境，这个环境放在 h 当前位置
2 \begin{minipage}{0.3\textwidth} %这是一个 minipage，宽度为页面宽度的 0.3 倍
3 \includegraphics[width=\textwidth]{figh_5.pdf}%在这个 minipage 中插入图片
4 \caption{这是图片}
5 \label{}
6 \end{minipage}
7 \begin{minipage}{0.7\textwidth}%另一个 minipage
8 \begin{center}
9 \caption{这是表格}
10 \begin{tabular}{cccc} %在这个 minipage 中插入表格
11 \toprule[1.4pt]
12 \multirow{ 2}{*}{出行方式} & \multicolumn{ 3}{c}{出行目的} \\ \cline{ 2- 4}
13 & 工作商务出行 & 非工作出行 & 上学出行 \\ \hline
14 公交车 & 28.16 & 6.35 & 3.18 \\ \hline
15 出租车 & 38 & 8.57 & 4.3 \\ \hline
16 社会小车 & 49 & 11 & 5.5 \\ \hline
17 各出行所占比例 & 25.10\% & 71.30\% & 3.60\% \\
18 \bottomrule[1.4pt]
19 \end{tabular}
20 \label{tab5}

```



```
21 \end{center}  
22 \end{minipage}  
23 \end{figure} %结束浮动环境
```

4.4 再说点图片的事

\LaTeX 可以支持很多种的图片格式，有矢量图格式，也有位图格式。个人建议最好使用矢量图（某些特殊情况除外），因为矢量图在放大缩小后不会变得模糊，最清晰，而且也可以让读者复制图片中的文字。

这里再说说我用的制图软件。首先我是 Linux 用户，由于 Windows 下软件众多，凡是 Linux 下有的，Windows 下都有，而且 Windows 还拥有大量的商业级软件，这些软件的强大功能是那些小型开源软件所无法比拟的。所以这部分适用于 Linux 用户，Windows 用户可以略过，或是参考一下。

画示意图我一般用的是 GeoGebra，流程图用的是 Dia，矢量绘图软件用的是 Inkscape。有很多 \LaTeX 或 Linux 高手使用的绘图软件都是基于命令行的，像 MetaPost。这些软件虽然功能很强大，但与那些用鼠标点来点去的软件相比毕竟门槛太高了，反正我不会用。我感觉 GeoGebra 是个非常好用的软件，功能也很强大。这里提供一个小窍门。导出图片时如果直接导出 pdf 或 eps 格式，实际的效果跟原图相比相差较大，而导出 png 格式虽然和原图很像，但不是矢量图，所以最好的方法是先导出 svg 格式，然后用 Inkscape 打开，保存为 pdf 格式。实际中发现这样的效果和原图完全相同，应该是最好的方法了。

同样的，使用 Dia 画流程图时也是先保存为 svg，然后用 Inkscape 保存为 pdf，这样的效果和原图基本相同。

第5章 数学公式

终于到了最激动人心也是 L^AT_EX 最擅长的一部分了!!

5.1 一些简单的数学公式

废话少说。输入行间公式，比如这样 $f(x) = x \sin x^3$ ，使用 `$...$`。输入行内公式，比如

$$f(x) = x \sin x^3$$

使用 `$$...$$`。如果想要让公式编号，则使用 `equation` 环境：

`\begin{equation}...\end{equation}`

一些最简单的公式输入这里就不介绍了，网上一找一大堆。但是为了充实下本部分的内容，还是扯一些。

希腊字母：

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

一些小帽子:

\acute{x}	<code>\acute{x}</code>	\tilde{x}	<code>\tilde{x}</code>	\mathring{x}	<code>\mathring{x}</code>
\grave{x}	<code>\grave{x}</code>	\breve{x}	<code>\breve{x}</code>	\dot{x}	<code>\dot{x}</code>
\bar{x}	<code>\bar{x}</code>	\check{x}	<code>\check{x}</code>	\ddot{x}	<code>\ddot{x}</code>
\vec{x}	<code>\vec{x}</code>	\hat{x}	<code>\hat{x}</code>	\dddot{x}	<code>\dddot{x}</code>

如果使用的是 texmaker 编辑器, 很多数学公式可以直接从工具栏里面找到, 也没有必要去背这些东西。Windows 用户可以下载安装 MathType, 这个软件可以用鼠标点击输入数学公式, 然后导出 L^AT_EX 的代码, 也很方便。Linux 用户可以下载安装 EqualX, 也有类似功能。

5.2 稍微高级的应用

输入数学公式有的时候没有看起来那么简单, 可能会对数学公式有一些要求, 也可能会遇到许多的问题。

5.2.1 括号

括号很简单, 但是要做到美观还是需要注意一些的。比如这个

$$\left(\frac{a}{b}\right)$$

显然就不美观, 括号太小。可以让 L^AT_EX 自动调整括号的大小:

`\left(\frac{a}{b}\right)`

得到的效果为

$$\left(\frac{a}{b}\right)$$

对小括号和中括号, 左括号分别是 `\left(` 和 `\right[`, 但是对大括号, 左括号就变为 `\left\{`。注意其中的区别, 大括号多了一个 `\`。

对于中括号和大括号代码如下

`\left[\frac{a}{b}\right]`

`\left\{\frac{a}{b}\right\}`

显示效果如下

$$\left[\frac{a}{b}\right] \quad \left\{\frac{a}{b}\right\}$$

经常需要使用大括号, 比如下面这种情况

$$f(x) = \begin{cases} x = \cos(t) \\ y = \sin(t) \\ z = \frac{x}{y} \end{cases} \quad (5.1)$$

使用如下代码:

```
1 \begin{equation}
2 f(x)=\left\{
3 \begin{aligned}
4 &x= \cos(t) \\
5 &y= \sin(t) \\
6 &z=\frac{x}{y}
7 \end{aligned}
8 \right.
9 \end{equation}
```

其中 & 表示对齐的位置。还可以使用 array 环境, 和 tabular 完全类似:

```
1 \begin{equation}
2 F=\left\{
3 \begin{array}{rcl}
4 A&& \{0< S_L\} \\
5 B&& \{S_L \leq 0 < S_M\} \\
6 C&& \{S_M \leq 0 < S_R\} \\
7 D&& \{S_R \leq 0\}
8 \end{array} \right.
9 \end{equation}
```

效果如下:

$$F = \begin{cases} A & 0 < S_L \\ B & S_L \leq 0 < S_M \\ C & S_M \leq 0 < S_R \\ D & S_R \leq 0 \end{cases} \quad (5.2)$$

还有 cases 环境:

```
1 \begin{equation}
2 f(x)=
3 \begin{cases}
4 0& x=0 \\
5 1& x!=0
6 \end{cases}
7 \end{equation}
```

效果如下:

$$f(x) = \begin{cases} 0 & x = 0 \\ 1 & x \neq 0 \end{cases} \quad (5.3)$$

注意, cases 环境不需要自己输入大括号, 但是前两种情况需要自己输入大括号。显然, 上面的公式只有左大括号, 而没有右大括号, 这个时候也不能直接忽略右大括号, 因为左和右是配对的, 所以上面的公式中在 \left\{ 左大括号的最后需要 \right. 与之配对。注意, right 后面有一个点。

5.2.2 长公式分行

有一个比较长的公式

$$c(x, y, z) = \sin x \sin y \sin z \frac{Q}{4\pi \sqrt{(D_y + D_{y'} - D_{my})(D_z + D_{z'} - D_{mz})}} \times \exp \left[-\frac{u}{4x} \left(\frac{y^2}{D_y + D_{y'} - D_{my}} + \frac{z^2}{D_z + D_{z'} - D_{mz}} \right) \right] \quad (5.4)$$

这个公式太长了，一行放不下，所以要考虑分行放。有几种方法。

1. 对齐

在 equation 环境中嵌套 split 环境，代码如下：

```

1 \begin{equation}
2 \begin{split}
3 c(x, y, z) = & \sin\{x\} \sin\{y\} \sin\{z\} \frac{Q}{4\pi \sqrt{(D_y + D_{y'} - D_{my})(D_z + D_{z'} - D_{mz})}} \times \\
4 & \exp \left[ -\frac{u}{4x} \left( \frac{y^2}{D_y + D_{y'} - D_{my}} + \frac{z^2}{D_z + D_{z'} - D_{mz}} \right) \right] \\
5 \end{split}
6 \end{equation}

```

其中的 & 表示对齐的位置，\\ 表示换行。

$$c(x, y, z) = \sin x \sin y \sin z \frac{Q}{4\pi \sqrt{(D_y + D_{y'} - D_{my})(D_z + D_{z'} - D_{mz})}} \times \exp \left[-\frac{u}{4x} \left(\frac{y^2}{D_y + D_{y'} - D_{my}} + \frac{z^2}{D_z + D_{z'} - D_{mz}} \right) \right] \quad (5.5)$$

2. 不对齐

如果觉得还要自己去指定哪边对齐太麻烦，那就不对齐了，使用 multiline 环境：

```

1 \begin{multiline}
2 c(x, y, z) = \sin\{x\} \sin\{y\} \sin\{z\} \frac{Q}{4\pi \sqrt{(D_y + D_{y'} - D_{my})(D_z + D_{z'} - D_{mz})}} \times \\
3 \exp \left[ -\frac{u}{4x} \left( \frac{y^2}{D_y + D_{y'} - D_{my}} + \frac{z^2}{D_z + D_{z'} - D_{mz}} \right) \right] \\
4 \end{multiline}

```

这样第一行的公式居左，而第二行的公式居右。效果如下：

$$c(x, y, z) = \sin x \sin y \sin z \frac{Q}{4\pi \sqrt{(D_y + D_{y'} - D_{my})(D_z + D_{z'} - D_{mz})}} \times \exp \left[-\frac{u}{4x} \left(\frac{y^2}{D_y + D_{y'} - D_{my}} + \frac{z^2}{D_z + D_{z'} - D_{mz}} \right) \right] \quad (5.6)$$

以上的分行都没有把括号打断，如果要从左右括号间分行，比如这样

$$a = (a + b + c + c + e) \quad (5.7)$$

直接用上面的方法就会报错。前面讲括号时已经说过了，左和右要成对出现，所以遇到从括号中间分开的情况复杂一些，本例的代码如下：

```
1 \begin{equation}
2 \begin{split}
3 a=&\left(a+b+c\right. \\
4 &\left.+c+e\right) \\
5 \end{split}
6 \end{equation}
```

5.2.3 多行公式

有的时候经常需要输入多行公式，这又分好几种情况：

1. 每个子公式都有编号

比如这样

$$f(x) = x^3 \quad (5.8)$$

$$g(x) = \sin x \quad (5.9)$$

这里使用 `align` 环境，看起来和连续使用两个 `equation` 环境一样，但是连续使用两个 `equation` 换进会使得这两个公式之间的距离过大。如果不想编号，在 `align` 后面加上 `*` 就行了。代码如下：

```
1 \begin{align}
2 &f(x)=x^3 \quad \label{eq:rel1} \\
3 &g(x)=\sin{x} \quad \label{eq:rel2} \\
4 \end{align}
```

2. 只有一个编号在中间位置

比如这样

$$\begin{aligned} f(x) &= x^3 \\ g(x) &= \sin x \end{aligned} \quad (5.10)$$

代码如下：

```
1 \begin{equation}\label{eq:12}
2 \begin{split}
3 &f(x)=x^3 \\
4 &g(x)=\sin{x} \end{split} \end{equation}
```

```

5 \end{split}
6 \end{equation}

```

同样的，如果不想要编号，就在 `equation` 后面加 `*`。

3. 多行公式居中

上面的多行公式都是左对齐的，如果想居中，那么可以使用 `gather` 环境：

```

1 \begin{gather}
2 A_0 = \sin{x_0} \\
3 A_1 = x_1 \cos{x_0} \\
4 A_2 = x_2 \cos{x_0} - \frac{1}{2!} x_1^2 \sin{x_0}
5 \end{gather}

```

效果如下：

$$A_0 = \sin x_0 \quad (5.11)$$

$$A_1 = x_1 \cos x_0 \quad (5.12)$$

$$A_2 = x_2 \cos x_0 - \frac{1}{2!} x_1^2 \sin x_0 \quad (5.13)$$

如果想要只有一个编号，可以使用 `equation` 环境，再嵌套上面介绍过的 `array` 环境：

```

1 \begin{equation}
2 \begin{array}{c}
3 A_0 = \sin{x_0} \\
4 A_1 = x_1 \cos{x_0} \\
5 A_2 = x_2 \cos{x_0} - \frac{1}{2!} x_1^2 \sin{x_0}
6 \end{array}
7 \end{equation}

```

得到效果如下：

$$A_0 = \sin x_0 \quad (5.14)$$

$$A_1 = x_1 \cos x_0$$

$$A_2 = x_2 \cos x_0 - \frac{1}{2!} x_1^2 \sin x_0$$

总之，多思考，多总结，就能够解决很多的问题。

5.3 一些需要注意的地方

我使用 \LaTeX 的一个感受就是，这东西会让我变成一个完美主义者，最起码我在使用这个软件的时候会变成完美主义者。我觉得这其中的原因就是，你已经投入了这么多时间，还做的这么烂，真的好不甘心，于是就花更多的时间来让排版变得更加美观。唉，都是坑啊。

关于数学公式的输入，有一些需要注意的地方。数学公式中如果是单个字母就要用斜体，比如 x, y, z ，而多个字母代表一个整体时就不用斜体了，比如 \sin, \cos, \tan 。所以输入 \sin 时就

不能这样输入 \sin ，而应该这样输入 \sin 。还有应该注意，微分符号 d 也不是斜体，这样写

$$\frac{dy}{dx}$$

是不对的，应该这样写 $\frac{\mathrm{d}y}{\mathrm{d}x}$ ，效果为：

$$\frac{\mathrm{d}y}{\mathrm{d}x}$$

所以， $\mathrm{}$ 命令表示在数学环境中输入非数学斜体的符号。如果觉得写微分符号时每次都用这个命令太麻烦，可以在导言区自定义一个命令：

```
\newcommand{\ud}{\mathrm{d}}
```

这样每次只需要输入 \ud 就可以了。

第6章 扯点闲话

6.1 \hfill 和 \vfill

个人感觉这两个是很重要的命令，在摆放比较复杂的图片和表格结构时必不可少。

`\hfill` 就是水平填充，它将位于它前面的内容推到最左边排版，而将其后的内容推到最右边排版，在中间留下空位。对两个放在一起的 `minipage`：

```
1 \begin{minipage}{0.3\textwidth}
2 这是一段文字，很长很长的文字呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵
3 \end{minipage}
4 \begin{minipage}{0.3\textwidth}
5 \textcolor{red}{这是另外一段文字，也挺长的呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵}
6 \end{minipage}
```

效果为：

这是一段文字，很长很长的 这是另外一段文字，也挺长
文字呵呵呵呵呵呵呵呵呵呵 的呵呵呵呵呵呵呵呵呵呵
呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵 啊啊啊啊啊啊啊啊

如果使用 `\hfill`，如下：

```
1 \begin{minipage}{0.3\textwidth}
2 这是一段文字，很长很长的文字呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵
3 \end{minipage}
4 \hfill
5 \begin{minipage}{0.3\textwidth}
6 \textcolor{red}{这是另外一段文字，也挺长的呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵}
7 \end{minipage}
```

那么效果为：

这是一段文字，很长很长的 这是另外一段文字，也挺长
文字呵呵呵呵呵呵呵呵呵呵 的呵呵呵呵呵呵呵呵呵呵
呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵 啊啊啊啊啊啊啊啊

如果想要实现这两个 `minipage` 紧贴在一起呢？就像下面这样：

这是一段文字，很长很长的 这是另外一段文字，也挺长
文字呵呵呵呵呵呵呵呵呵呵 的呵呵呵呵呵呵呵呵呵呵
呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵呵 啊啊啊啊啊啊啊啊

同样可以使用 \hfill 填充。首先在第一个 minipage 左边加一个空的盒子，然后使用 \hfill 填充，接着第二个 minipage 紧跟在第一个 minipage 后面，并且在第二个 minipage 后再加一个空盒子，再加入一个 \hfill 填充进去。代码如下：

```

1 \mbox{}%第一个空盒子
2 \hfill %水平填充
3 \begin{minipage}{0.3\textwidth} %第一个 minipage
4 这是一段文字，很长很长的文字啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊
5 \end{minipage}
6 \begin{minipage}{0.3\textwidth} %第二个 minipage
7 \textcolor{red}{这是另外一段文字，也挺长的啊啊啊啊啊啊啊啊啊啊啊啊}
8 \end{minipage}
9 \hfill %水平填充
10 \mbox{}%第二个空盒子

```

\vfill 的效果与 \hfill 类似，不过就是水平填充变成了竖直填充。现在想要实现下面的效果：

这是一段文字，很长很长的 这是另外一段文字，也挺长
文字啊啊啊啊啊啊啊啊啊啊 的啊啊啊啊啊啊啊啊啊啊
啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊 啊啊啊啊啊啊啊啊

现在是竖直填充，是不是很
神奇啊啊啊啊啊啊啊啊啊啊

现在需要先将这三个 minipage 加入一个浮动环境中，代码如下：

```

1 \begin{figure}[h] %浮动环境
2 \mbox{}
3 \hfill
4 \begin{minipage}{0.3\textwidth}
5 这是一段文字，很长很长的文字啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊
6 \end{minipage}
7 \begin{minipage}{0.3\textwidth}
8 \textcolor{red}{这是另外一段文字，也挺长的啊啊啊啊啊啊啊啊啊啊啊啊}
9 \end{minipage}
10 \hfill
11 \mbox{}
12 \vfill %竖直填充
13 \begin{center} %居中放置
14 \begin{minipage}{0.3\textwidth}
15 \textcolor{blue}{现在是竖直填充，是不是很神奇啊啊啊啊啊啊啊啊啊啊}
16 \end{minipage}
17 \end{center}
18 \end{figure}

```

如果不加 figure 这个浮动环境，那么根据 \vfill 命令的定义，蓝色那段将会和其后的文本排到页面的最底部，使得蓝色那段和黑色红色那段之间有很多的空白。读者可以自己试试。

说白了，这两个命令可以用于表示前后两个部分的相对关系，`\hfill` 表示前后两个是放在同一水平上的，而 `\vfill` 表明前后两个在位置上是上下的关系。第四章讲插入图片时就用到这两个命令，读者可自己体会下。

6.2 \hspace 和 \vspace

`\hspace{width}` 命令产生一个 `width` 长的水平空格，`\vspace{width}` 命令产生一个 `width` 长的垂直空格。当起新的一行时，`\hspace{width}` 产生的空格会被忽略，这时要使用 `\hspace*{width}` 强制产生空格。同样，`\vspace*{width}` 也是强制产生垂直空格。`width` 也可以为负。

这两个命令在调整两个部分之间的相对位置时很有用，比如图6.1是正常的位罝，现在我想让接下来的图片的位置变得不太正常，先左移动到边上，然后往上移动 4cm，变成图6.2。

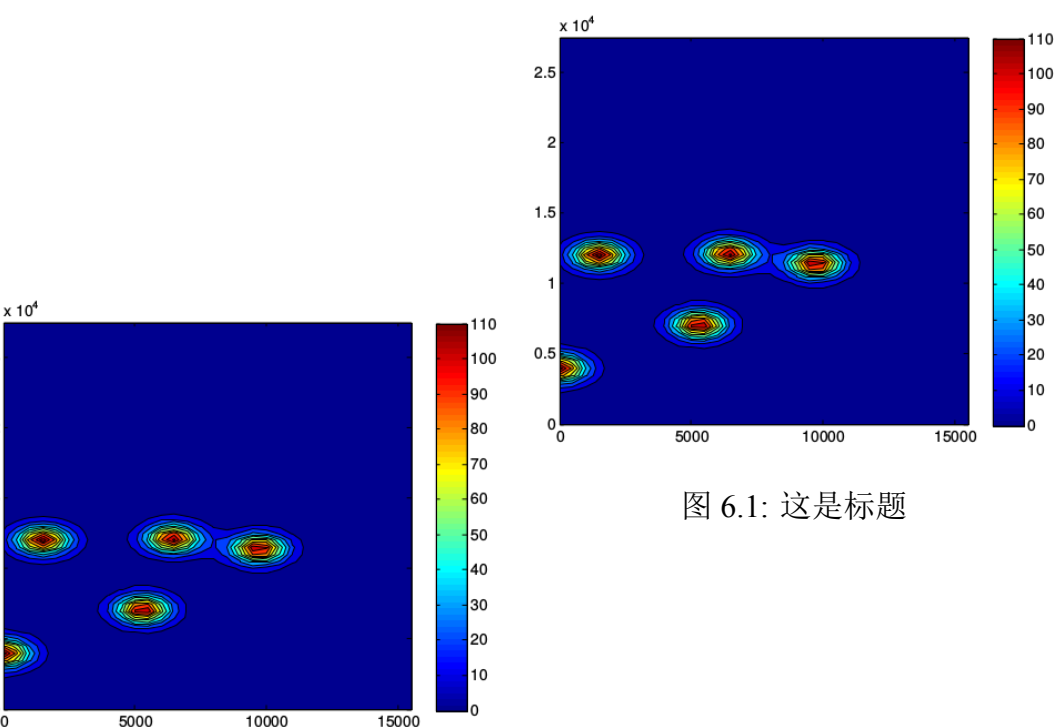


图 6.1: 这是标题

图 6.2: 这是标题

实现的代码如下：

```

1 \begin{figure}[htp]
2 \vspace*{-4cm}\hspace*{-25.4mm}
3 \begin{minipage}{6cm}
4 \centering
5 \includegraphics[height=6cm]{figh_3.pdf}
6 \caption{这是标题}
7 \label{pic:float2}
8 \end{minipage}
9 \end{figure}

```

这里先是一个 figure 浮动环境，然后嵌套了一个 minipage，通过移动 minipage 来实现图片的移动。

虽然 L^AT_EX 无法像 Word 那样鼠标拖动移动图片，但是通过使用这两个命令实现位置的调整还是挺好用的。

6.3 参考文献

这节介绍如何加入参考文献。首先添加宏包 cite。在正文的最后（注意要是 `\end{document}` 之前）使用如下代码：

```
1 \begin{thebibliography}{99} %99是文献字符的最大数量
2 \bibitem{label} {文献的名称}
3 \end{thebibliography}
```

然后在需要引用的地方使用 `\cite{label}` 就可以实现文献的引用了。比如

```
1 这是一个引用参考文献的例子\cite{asd。}
```

就得到这样的效果：这是一个引用参考文献的例子^[1]

6.4 数学建模国赛论文格式

我用 L^AT_EX 写过几篇国赛论文，在这点上还是有些经验的。其实数学建模论文的格式和一般论文没什么不同，不过就是增加了一些小要求。那些字体字号，插图表格什么的就不说了，完全可以使用前面说过的方法。我就说说几个其实也算不上重要的方面，就当这部分是进阶吧。我的数学建模国赛论文模板可以在这里下载到<http://zhangzimou.com/file/latex.7z>。

6.4.1 承诺书

网上有很多用 L^AT_EX 写的国赛论文模版，我去年刚学 L^AT_EX 时有下载一些看看，感觉没怎么看懂。很蛋疼的一点是，那些看起来很高端的模版都在承诺书上下了一番功夫，他们将数学建模委员会提供的那两页承诺书用 L^AT_EX 给编了出来。我一直都觉得这样做虽然能够体现他们强大的 L^AT_EX 使用能力，但是意义不大。承诺书本来就已经提供，而且原则上是不允许修改的，你再怎么照着做也不可能完全一样。我向来的做法是用 Word 在承诺书上写个人信息，然后生成 pdf 文件，再用 pdf 拼接软件将其与论文正文拼接起来。这样做确实有些麻烦，特别是最后的时候，修改一下论文就要重新生成 pdf 文件，又要重新来合并，会浪费不少时间，甚至还可能出现忘记合并的情况。还有一个很大的缺陷，如果论文中有超链接，那么拼接后这些超链接就消失了（至少我用的 pdf 拼接软件是这样）。虽然论文最后都是打印出来，有没有超链接完全无所谓，但随着使用 L^AT_EX 的时间越来越长，我对生成的电子版的要求也越来越高，也就越来越不能忍受这种方法。

我的想法很简单。还是一样，用 Word 填个人信息，然后生成 pdf 文件。不同的是，我将生成的两页 pdf 直接用 L^AT_EX 插入到论文中，省去了额外拼接的麻烦，也保留住了超链接。

可以使用插入图片的命令插入 pdf 文件，但是当图片过大时会出现一个问题。每个页面都有一个叫页边距的东西，它直接限制了所插入图片的左上角的位置。比如我现在要居中插

入一个比较大的图片，如图6.3所示。显然它没有居中，因为它太长了，默认不能超出左边的

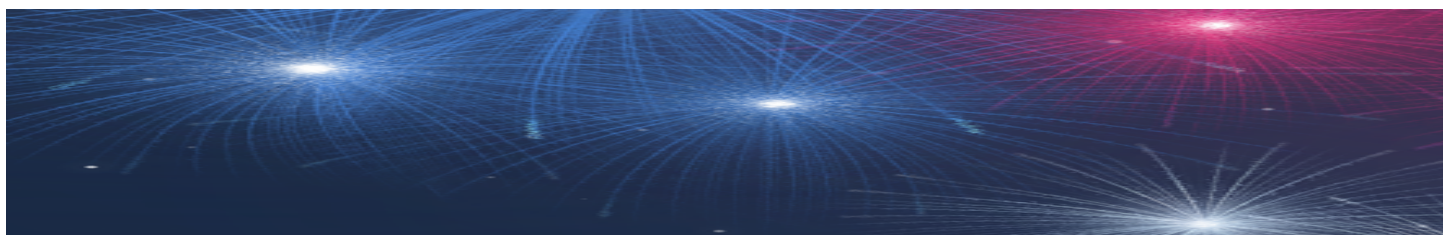


图 6.3: 居中插入一个比较大的图片

竖直基线。同样的，它也不能超出上面的水平基线。如果直接插入承诺书，得到的结果也是类似的，这里就不演示了，读者可自行实验一下。

那方法就有了，改变页边距嘛。添加 `geometry` 宏包，在要改变页边距的那一页使用如下命令改变页边距：

```
\newgeometry{margin=0cm}
```

这将该页四周的页边距都变为 `0cm`。现在使用如下命令插入承诺书：

```
1 \newgeometry{margin=0cm}
2 \begin{figure}[h]
3 \centering
4 \includegraphics[scale=1]{Page1.pdf}%scale=1表示 pdf 文件的原始大小
5 \end{figure}
```

这样得到的效果就好像是将那一页 `pdf` 加了进来，读者可自己实验一下。这样好像就搞定了！

所以就使用以下的代码：

```
1 \begin{document}    %文档刚开始
2 \newgeometry{margin=0cm} %设定从此页以后四周的页边距都为 0cm
3 \begin{figure}[h]    %开始插入第一页
4 \centering
5 \includegraphics[scale=1]{Page1.pdf}
6 \end{figure}
7 \newpage            %起新的一页
8 \begin{figure}[h]    %开始插入第二页
9 \centering
10 \includegraphics[scale=1]{Page2.pdf}
11 \end{figure}
12 \newpage            %起新的一页
13 \restoregeometry    %将页边距还原为原来值，不加不行
14 ...
```

但还是有问题。两页的承诺书是要放在论文的最开始两页，这没错，我们放在了文档刚开始的地方。如果还使用 `\title{}` 命令来添加论文的标题的话，由于这个命令会默认将标题放在最前面，也就是放在了承诺书的前面，所以得到的不是想要的效果。既然这样，我们就只好自己定义一个添加标题的命令了。在导言区中加入如下代码：

```

1 \newcommand{\mytitle}[1]{
2 \noindent %不要缩进
3 \begin{minipage}{\textwidth}%一个宽为版心宽度的 minipage
4 \begin{center}%居中
5 {#1} %这是自己加的参数，这里就是标题
6 \end{center}
7 \end{minipage}
8 }

```

这样我们定义了一个命令 `\mytitle{}`，花括号里面加的是标题，还可以使用 `\hei\sanhao` 等命令改变字体字号。现在接着在插入承诺书的代码后加上：

```

1 ... %这上面是承诺书插入的代码
2 \newpage %开始新的一页
3 \restoregeometry %还原页边距
4 \mytitle{\hei\sanhao 基于优化模型的创意拼版折叠桌设计} %这里是标题
5 ...

```

这样就基本搞定了。我之所以说基本搞定了，是因为还需要做点工作。这样生成的标题可能位置不太让人满意，所以可以通过前文介绍过的 `\vspace{}` 命令来对标题进行竖直移动（已经是居中了，所以不用水平移动）。还有，既然前面加了两张承诺页，那么标题这一页的页码就变成了 3，所以可以在 `\mytitle{}` 后使用命令：

```
\setcounter{page}{1}
```

在当前页重新从 1 开始编号。如果这页不想编号，可以使用：

```
\thispagestyle{empty}
```

然后在新开始的一页使用 `\setcounter{page}{1}` 重新从 1 开始编号。

看起来好像有点麻烦，其实理解了以后非常简单，再说了这些代码都是可以重复利用的，这次搞定了下次就能够直接用了。

6.4.2 插入代码

数学建模论文是需要将程序代码附在附录后的，所以这里讲讲怎么插入代码。首先添加 `listings` 宏包。在导言区中可以对插入的代码进行全局设置：

```

1 \lstset{
2     numbers=none, %设置行号位置
3     numberstyle=\wuhao, %设置行号大小
4     escapeinside=``, %逃逸字符 (1 左面的键)，用于显示中文
5     breaklines, %自动折行
6     extendedchars=false, %解决代码跨页时，章节标题，页眉等汉字不显示的问题
7     xleftmargin=0em, xrightmargin=0em, aboveskip=0em, %设置边距
8     showspaces=false %不显示空格
9 }

```


以上是全局设置，对所有插入的代码都适用。具体参数表示的含义请查阅网上的资料。如果要插入代码，使用如下方法：

```
1 \begin{lstlisting}[numbers=left]
2 这里是一段代码
3 \end{lstlisting}
```

中括号里是可选参数，也是局部设置，代码显示时会优先使用局部设置，在不与全局参数冲突的情况下再使用全局设置。比如我想局部设置行号居左，显示空格，那么可以这样

```
1 \begin{lstlisting}[numbers=left,showspaces=true]
2 这里是一段代码
3 \end{lstlisting}
```

有时代码过长，占的页数太多，所以更改代码行距非常重要。添加宏包 `setspace`，使用如下命令：

```
1 \begin{spacing}{行距}
2 这里是需要更改行距的内容
3 \end{spacing}
```

在“行距”中填“1”就表示 1 倍行距，填 0.5 就表示 0.5 倍行距。这样就可以局部地更改行距了。

后记

花了大概 25 小时写完了这篇教程，不管写的多烂，多难看懂，我都觉得很有成就感。虽然只有区区 40 页，但这也算是我人生中写的第一本书了。从来都是看别人写的教程，终于写完后，我都有点不敢相信自己居然也写了这样的一篇教程。这并不是多难的事情，但是从 0 到 1 的突变总是质的飞越。我很高兴我坚持了下来。

虽然写作这教程花了我大量的时间，但我觉得这是值得的。人这一辈子，有几次是真的热血沸腾地想做一件事。人生的意义在于尝试，只有勇于尝试不同的新鲜事物，才能够感受到这个世界的丰富多彩，也才能体会到其实自己能够做到的事情还有很多。我不在乎这篇教程能够被多少人看到并有多少人真的认真地学习了，我只知道我做完了一件我一拍脑袋突发奇想并真的想做的事。

在写作的这段时间里我也在不断地学习，也收获了很多。同时，我也强烈地感受到自己多方面的不足。其实无论是写博客也好，写教程也罢，都是一个反省的过程，因为在这段时间里你会强迫自己把原先消化完全和未消化完全的东西都一股脑地呕出来，然后重新嚼烂，再重新吞下。原谅我如此恶心的比喻。真的是这样，很多时候你觉得你知道，其实你根本不知道。当然了说这句话本身在逻辑上是有矛盾的。

如果真的有人能够从我这篇教程中得到哪怕一点的收获，我都很满足了。希望我没有误导那些因为缘分偶然间看到这篇教程的读者。由于水平有限，错误在所难免，如果你能够指出本教程的错误或者提出改进的意见那就太好了，欢迎反馈到我的邮箱:zzm@zhangzimou.com.

最后愿各读者使用 \LaTeX 排版时讲究精益求精。当然了，生活和学习中的精益求精更重要。

张子谋

2014 年 10 月于西安交通大学