

# $\LaTeX$ 插图与绘图简介

China $\TeX$  在线培训课程

邓东升 ( ddswhu )

Fudan University  
School of Economics

# 目录安排

- 1 引言
- 2 一个简单的例子
- 3 位图与矢量图
- 4 编译方式与图形格式
- 5 图形格式转化与灰度图
- 6 插入图形
- 7 绘图工具简介

# 引言

西方学者常说：A picture is worth a thousand words。译作：一图胜千言，意思是说，一幅图形可以简单明确地表达很多错综复杂，千言万语都难以描述的信息。所以说一篇优秀的论文应该是图文并茂。

$\LaTeX$  的绘图功能比较简单，如使用  $\LaTeX$  相关的各种绘图宏包或者绘图工具比如：METAPOST、PSTricks、PGF (TikZ) 等，借助这些工具我们可以画图非常复杂的图形，但缺点是不直观，命令繁琐，不易熟练掌握。

现在，通常是用 Matlab、R、Visio 等功能强大的绘图工具先把图形画好，然后插入到  $\LaTeX$  源文件中。

## 一个简单的例子

需要插入的图片（logo.png）在我们的主文件夹内，我们使用 PDFLaTeX 编译，图片的宽度为  $0.3 * \text{文档宽度}$ （`\textwidth`），标题为“这是一个 LOGO”，为方便引用，我们设置标签为“fig:elegantlatex\_logo”。使用到的命令如下：

```
1 \begin{figure}[htbp]
2   \centering
3   \includegraphics[width=0.3\textwidth]{logo.png}
4   \caption{这是一个 LOGO}\label{fig:elegantlatex_logo}}
5 \end{figure}
```

The logo for ElegantLaTeX, featuring the text "ElegantLaTeX" in a blue serif font, followed by a stylized orange and blue symbol that resembles a calligraphic flourish or a partial circle.

Figure: 这是一个 LOGO

# 位图

图形的存储格式很多，一般分为两大类：位图与矢量图，都是以数字形式存储，解释方法各不相同。

**位图 ( bitmap )**：也称点阵图，栅格图象，像素图。

- 最小单位由像素 ( Pixel ) 构成的图，缩放会失真；
- 由像素阵列的排列来实现其显示效果的；
- 每个像素有自己的颜色信息，可操作对象为像素 ( HSB )。
- 每英寸所拥有的像素数目用 PPI ( 分辨率的单位 ) 表示。

# 矢量图

矢量图 ( **vector** ): 也叫做向量图。

- 由数学公式定义的线段和曲线组成；
- 纪录了元素形状及颜色的算法；
- 可以将其任意缩放和旋转，都不会失真；
- 矢量图与分辨率无关；
- 矢量图文件尺寸一般比较小

## 编译方式与图形格式支持

我们通常使用 **LaTeX**、**PDFLaTeX**、**XeLaTeX** 编译源文件。各种编译方式下图形格式支持如下：

- 1 **LaTeX** 直接支持 EPS、PS 图形文件，间接支持 JPEG、PNG 等格式；
- 2 **PDFLaTeX** 直接支持 PNG、PDF、JPEG 格式图形文件，间接支持 EPS；
- 3 **XeLaTeX** 直接支持 BMP、JPEG、PNG、EPS 和 PDF 图形格式。

### 注意事项：

在使用 **PDFLaTeX** 时，如果要插入 EPS，可以先把 EPS 转化为其他格式（比如 PDF、JPG），或者在导言区加载 `epstopdf`。

## 批量得到 eps 图档

虽然「 $\text{\LaTeX}$  只能识别 EPS 格式的图档」是多年的误传，但是仍有许多杂志和期刊只接受 EPS 格式的图档。所以，尽管在日常使用中，我们很少会用到 EPS 格式，但是有时候不得不用。

在现在的 TeX 发行版中，一般都带有一个 `bmeps` 的小程序，它能够将 PNG，JPEG 和 BMP 等格式的位图转换成 EPS 格式的图档。以 JPEG 格式为例，批处理命令（来源于小 L）如下：

```
1 for /f %%i in ('dir /b *.jpg') do (  
2     @echo %%i  
3     bmeps -c %%i %%~ni.eps  
4     @echo Finished  
5 )
```

[点击此处有惊喜！](#)



## 使用 ImageMagick 得到灰度图

在实际写作中，我们更可能需要用到灰度图，这里推荐使用 ImageMagick ( 需要自行安装 )，这个是一个非常强大的图形处理软件。单个图片的转化使用 `convert 1.jpg -colorspace Gray 1_out.jpg`。

送福利时刻：

```
1 mkdir out
2 for %%B in (*.jpg) do convert "%%B" -colorspace Gray "out/%%B"
```

戳这里有奖！

## BoundingBox 问题

由于历史原因，**LaTeX** 编译程序不能提取 JPEG、PNG 等点阵图形的尺寸信息，所以它在处理这些图形文件时需要范围框。**PDFLaTeX** 和 **XeLaTeX** 的用户可以跳过本节内容。

EPS 的范围框如下，其中前两个参数是图形左下角的坐标 (通常就是原点)，后两个参数是右上角的坐标，缺省长度单位是 **bp**。

```
1 %!PS-Adobe-2.0 EPSF-2.0
2 %%BoundingBox: 0 0 510 284
```

有了范围框，**LaTeX** 在编译源文件时就可以为插图预留空间；它输出的 DVI 只记录图形尺寸和文件名，因为具体的图形处理由后面的驱动负责。找不到范围框时，**LaTeX** 就会报错，

```
1 ! LaTeX Error: Cannot determine size of graphic in fig.png
2 (no BoundingBox).
```

## 解决 BoundingBox 方案一

在使用 LaTeX 编译的时候，比较好解决 BoundingBox 缺失问题的方法是使用 `bmpsize` 宏包（谢谢小 L），使用方法如下：

```
1 \documentclass{article}
2 \usepackage[dvipdfm]{graphicx}
3 \usepackage{bmpsize}
4 \begin{document}
5
6 \includegraphics{logo.png}
7
8 \end{document}
```

## 解决 BoundingBox 方案二

有两种方法可以为点阵图形提供范围框：

- 1 准备一个单独的范围框文件；
- 2 插入图形时加范围框参数。

使用 TeX 发行版中的 `ebb` 小工具可以获取 BoundingBox 信息，比如下面的命令会生成一个 `1.bb` 文件。

```
1 ebb 1.jpg
```

使用范围框文件：

```
1 \DeclareGraphicsRule{.jpg}{eps}{.bb}{}  
2 \includegraphics[width=0.6\textwidth]{1.jpg}
```

使用范围框参数：

```
1 \includegraphics[bb=0 0 300 200,width=0.6\textwidth]{1.jpg}
```

## 插图基本命令

我们插图一般使用到的宏包是 `graphicx`，插图的基本命令如下：

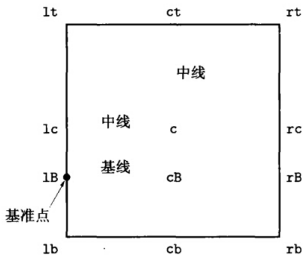
```
1 \usepackage{graphicx}
2 \includegraphics[width=0.5\textwidth]{fig.png}
```

使用 `LaTeX` 时，如果事先没有生成 `.bb` 文件的话，需要或加范围框参数。`PDFLaTeX` 和 `XeLaTeX` 不需要该参数。

## 图形操作参数说明

`\includegraphics` 命令有一些参数选项可以用于缩放、旋转、裁剪等图形操作，简要说明如下：

- `width=x,height=y`: 宽度和高度，绝对尺寸，可用任意长度单位；
- `scale=s`: 缩放比，相对尺寸，使用上面参数与缩放时，绝对尺寸起作用；
- `keepaspectratio`: 保持长宽比；
- `angle=a`: 旋转角度；
- `origin=hv`: 旋转中心，( 参考右图 )；
- `trim=l b r t`: 左下右上裁剪值；



## 文件名与路径

若想省略文件后缀或路径名，可以使用下面的命令：

```
1 \DeclareGraphicsExtensions{.eps,.mps,.pdf,.jpg,.png}
2 \DeclareGraphicsRule{*}{eps}{}{}
3 \graphicspath{{C:/secret_garden/}}
4 \graphicspath{{./img/}}
5 \graphicspath{{first_dir/}{second_dir/}{third_dir/}}
```

说明如下：

- 1 第一行指定后缀列表让编译程序自行查找；
- 2 第二行指出未知后缀的都是 EPS；
- 3 后三行设置缺省搜索路径，分别使用了绝对路径、相对路径、多个路径。

## figure 环境

插图通常需要占据大块空间，所以在文字处理软件中用户经常需要调整插图的位置。figure 环境可以自动完成这样的任务；这种自动调整位置的环境称作浮动环境 (float)，还有一个常用到的浮动环境是 table。

```
1 \begin{figure}[htbp]
2 \centering
3 \includegraphics{myphoto.jpg}
4 \caption{有图有真相}
5 \label{fig:myphoto}
6 \end{figure}
```



Figure: 有图有真相



## 并排摆放，共享标题

当我们需要两幅图片并排摆放，并共享标题时，可以在 `figure` 环境中使用两个 `\includegraphics` 命令：

```
1 \begin{figure}[htbp]
2   \centering
3   \includegraphics{leftfoot.png}
4   \includegraphics{rightfoot.png}
5   \caption{向左走向右走}
6 \end{figure}
```



Figure: 向左走向右走

## 并排摆放，各有标题

如果想要两幅并排的插图各有自己的标题，可以在 `figure` 环境中使用两个 `minipage` 环境，每个里面插入一幅图。不用 `minipage` 的话，因为插图标题的缺省宽度是整个行宽；两幅插图就会上下排列。



Figure: 向左走



Figure: 向右走

## 并排摆放，各有标题

```
1 \begin{figure}[htbp]
2   \centering
3   \begin{minipage}{60pt}
4     \centering
5     \includegraphics[scale=0.4]{leftfoot.png}
6     \caption{向左走}
7   \end{minipage}
8   \hspace{10pt}%
9   \begin{minipage}{60pt}
10    \centering
11    \includegraphics[scale=0.4]{rightfoot.png}
12    \caption{向右走}
13  \end{minipage}
14 \end{figure}
```

`\caption` 命令会把环绕它的 `minipage` 环境“变成” `figure` 环境。

## 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一个标题，并且各有自己的子标题，可以使用 Steven D. Cochran 开发的 `subfig` 宏包。它提供的 `\subfloat` 命令，并且总图和子图可以分别有标题和引用。



(a) 向左走      (b) 向右走

Figure: 向左走向右走

**注意：**对于子图的支持，还有另外一个更新的宏包：`subcaption`，推荐大家使用。不过，两者都存在各自的问题，关于 `subfig` 和 `subcaption` 的讨论与比较，请参考 <http://tex.stackexchange.com>。

## 并排摆放，共享标题，各有子标题

```
1 \begin{figure}[htbp]
2 \centering
3 \subfloat[向左走]{
4   \label{fig:subfig_a}
5   \includegraphics[scale=0.4]{leftfoot.png}
6 }
7 \hspace{10pt}%
8 \subfloat[向右走]{
9   \label{fig:subfig_b}
10  \includegraphics[scale=0.4]{rightfoot.png}
11 }
12 \caption{向左走向右走}
13 \label{fig:subfig}
14 \end{figure}
```

## METAPOST 绘图

1980 年代末 John D. Hobby 设计了一种绘图语言以及编译器，这就是 METAPOST，METAPOST 灵感来源于 Knuth 的 METAFONT。它的输出是 EPS，支持彩色，可以在图形中做文字标注，并且插入 TeX 源码，不过也继承了 METAFONT 的一些缺点：数值变量精度较低，且绝对值不能超过 4096；只支持部分 PostScript 功能。

- ① 一个 METAPOST 中可以包含多个图形，注意开始与结束声明；
- ② 使用 mpost 生成的文件是 MPS（特殊 EPS）；
- ③ 借助 EMP 宏包可以在  $\text{\LaTeX}$  中直接用 METAPOST 绘图。

## PSTricks 绘图

PSTricks 是一个基于 PostScript 的绘图包，有了它用户就可以直接在  $\text{\LaTeX}$  文档中插入绘图命令。PSTricks 作者是 van Zandt, 1997 年之后由 Herbert Voß 以及 Denis Girou 等人在维护。

- PSTricks 中缺省长度单位是 1cm；
- 绘图命令放在 `pspicture` 环境里；
- 需要指定画布的大小，即作图左下右上的坐标；
- 注意这个矩形要能容纳所有图形对象；
- 支持 [LaTeX](#)，[XeLaTeX](#) 在线编译。

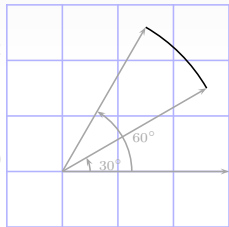
更多内容参考：*Graphics with PSTricks*。

# PSTricks 示例

```

1 \begin{pspicture}(-1,-1)(3,3)
2   \psgrid[gridcolor=blue!30,
3     gridlabelcolor=black!40,
4     subgriddiv=1]
5   \psarc(0,0){3}{30}{60}
6   \SpecialCoor
7   \psline[linecolor=mygray]{->}(0,0)(3,0)
8   \psline[linecolor=mygray]{->}(0;0)(3;30)
9   \psline[linecolor=mygray]{->}(0;0)(3;60)
10  \psarc[linecolor=mygray]{->}(0,0){0.5}{0}{30}
11  \psarc[linecolor=mygray]{->}(0,0){1.25}{0}{60}
12  \uput[r](0.5;15){
13    \color{mygray}$\scriptstyle 30^\circ$}
14  \uput[r](1.25;30){
15    \color{mygray}$\scriptstyle 60^\circ$}
16 \end{pspicture}

```





# TikZ 绘图

PGF 和 Beamer 的作者都是 Till Tantau。Tantau 当初开发 Beamer 是为了准备 2003 年他的博士学位论文答辩，之后它在 CTAN 上流行开来。2005 年 PGF 从 Beamer 项目中分离出来，成为一个独立的宏包。而 TikZ 是 PGF 的前端，我们一般都是用 TikZ。

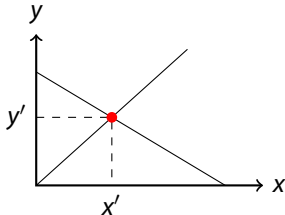
- ① 配合恰当算法可以得到非常精确的结果；
- ② 支持 PDFLaTeX 与 XeLaTeX 等；
- ③ 编译速度快，非常舒服的体验；
- ④ 学习难度大，绘图代码不直观更加复杂。

## TikZ 示例

```

1 \begin{tikzpicture}
2 \draw [<->,thick] (0,2) node (yaxis) [above] {$y$}
3   |- (3,0) node (xaxis) [right] {$x$};
4 \draw (0,0) coordinate (a_1) -- (2,1.8) coordinate (a_2);
5 \draw (0,1.5) coordinate (b_1) -- (2.5,0) coordinate (b_2);
6 \coordinate (c) at (intersection of a_1--a_2 and b_1--b_2);
7 \draw[dashed] (yaxis |- c) node[left] {$y'$}
8   |- (xaxis -| c) node[below] {$x'$};
9 \fill[red] (c) circle (2pt);
10 \end{tikzpicture}

```



## 关于我



邓东升，就读于复旦大学经济学院（硕士），2010 接触  $\text{\LaTeX}$ ， $\text{\LaTeX}$  用户以及爱好者，比较热衷于 beamer、绘图（TikZ）、模板设计等。2013 年与黄晨成共同建立 Elegant $\text{\LaTeX}$ ，发布 ElegantBook 等模板。

主页：<http://ddswhu.com/>

链接：<http://elegantlatex.org/>

邮箱：[ddswhu@outlook.com](mailto:ddswhu@outlook.com)