

第七讲 插图与绘图

— PGF/TikZ 宏包

潘建瑜

华东师范大学数学系

jypan@math.ecnu.edu.cn

2014

本讲内容

- ① 图形概述
- ② PGF/TikZ 宏包介绍
- ③ 路径 (path)
- ④ 简单作图
- ⑤ 填充
- ⑥ 作图选项
- ⑦ 阴影

本讲内容 (续)

⑧ 坐标

⑨ 箭头

⑩ 参数的作用域

⑪ 坐标变换

⑫ 重复

⑬ 添加标注

⑭ 函数作图与 plot 命令

图形概述

图形的分类：矢量图与点阵图

- **矢量图**：用数学表示，保存图的几何特征，如直线的端点、样条曲线的控制结点等
点阵图：通常以矩阵形式表示，存贮图像的每个像素点的颜色，可通过复杂的算法进行压缩
- **矢量图**：图形的精确数学刻划，具有很高的精度，经放缩、坐标变换而不丢失信息，一般文件较小
点阵图：图像的直接采样，可表现复杂的内容，但放缩、坐标变换后可能会失真，通常文件较大
- **矢量图**：可用来表示具有一定逻辑的示意图，如流程图、数学曲线等
点阵图：可用来表示不能用数学逻辑关系描述的照片等，或特别复杂的数据图、3D 图等
- **矢量图**输出时需要先转换为一定精度的点阵图

- ① 图形概述
- ② PGF/TikZ 宏包介绍
- ③ 路径 (path)
- ④ 简单作图
- ⑤ 填充
- ⑥ 作图选项
- ⑦ 阴影
- ⑧ 坐标
- ⑨ 箭头
- ⑩ 参数的作用域
- ⑪ 坐标变换
- ⑫ 重复
- ⑬ 添加标注
- ⑭ 函数作图与 plot 命令

PGF/TikZ 宏包

PGF/TikZ 宏包: 功能强大的绘图宏包

- PGF: Portable Graphics Format
- 开发者: Till Tantau (也是幻灯片宏包 beamer 的作者)
- 主页: <http://sourceforge.net/projects/pgf/>
- 支持 \LaTeX 和 pdf\LaTeX 编译
- PGF 可精确绘制复杂的几何图形及各种曲线
- 可以与其它数学(作图)软件配合使用: 如 gnuplot, Mathematica
- 丰富的网络资源 (见课程主页上的网络链接)

<http://www.texample.net/tikz/>

PGF/TikZ 宏包

pgf 的使用

- 使用时只需调用 `tikz` 宏包 (pgf 的前端)

```
\usepackage{tikz}
```

- 可以根据需要调用 `tikz` 扩展

```
\usetikzlibrary{arrows,backgrounds,scopes,plotmarks, ...}
```


- `tikz` 绘图是基于坐标系的，原点在当前位置 (左下角)
- 每条绘图命令以分号结束
- 默认长度单位是 1cm
- 使用长度或坐标时，可以带单位，也可以不带单位

PGF/TikZ 宏包

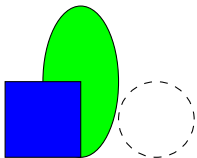
PGF/TikZ 绘图的两种使用方式：命令方式和环境方式

- 命令方式：`\tikz`

你  好 你 `\tikz \draw (0pt,0pt) -- (30pt,6pt);` 好

你  好 你 `\tikz{\draw (0pt,0pt) -- (30pt,6pt);}` 好

- 绘图环境：`tikzpicture`



```
\begin{tikzpicture}
  \draw[dashed] (2,.5) circle (0.5);
  \draw[fill=green] (1,1) ellipse (.5 and 1);
  \draw[fill=blue] (0,0) rectangle (1,1);
\end{tikzpicture}
```

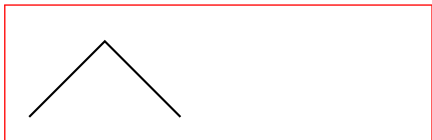

- ① 图形概述
- ② PGF/TikZ 宏包介绍
- ③ 路径 (path)**
- ④ 简单作图
- ⑤ 填充
- ⑥ 作图选项
- ⑦ 阴影
- ⑧ 坐标
- ⑨ 箭头
- ⑩ 参数的作用域
- ⑪ 坐标变换
- ⑫ 重复
- ⑬ 添加标注
- ⑭ 函数作图与 plot 命令

Path 路径

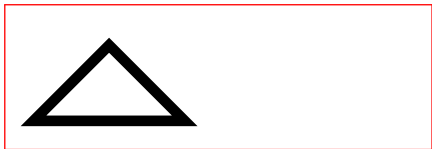
PGF/TikZ 绘图基本单元: 路径 (path)

- **路径**: The basic building block of all pictures. A path is a series of straight lines and curves that are connected.
- **路径的基本元素**: **点**, **连接方式**
 - **点**: 通过**坐标**或其它方式给出
 - **连接方式**: 直线, 曲线, 弧线, ...
- **路径可以被画, 填充, 裁剪, ...**

```
\tikz\path[draw,thick]%
(1,1) -- (2,2) -- (3,1);
```

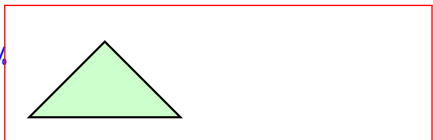


```
\tikz\path[draw,line width=4pt]%
(1,1) -- (2,2)--(3,1)--cycle;
```

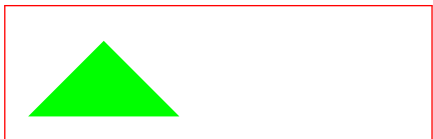


Path 路径

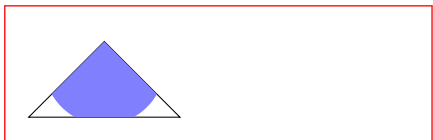
```
\tikz\path[draw,thick,fill=green!20]%
  (1,1)--(2,2)--(3,1)--cycle;
```



```
\tikz\path[fill=green]%
  (1,1) -- (2,2) -- (3,1) -- cycle;
```



```
\begin{tikzpicture}
  \path[clip,draw]%
    (1,1)--(2,2)--(3,1)--cycle;
  \path[fill=blue!50]%
    (2, 1.7) circle (.8);
\end{tikzpicture}
```



Path 路径

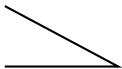
- 通常可使用缩写形式, 如

```
\draw = \path[draw]
\fill = \path[fill]
\clip = \path[clip]
\filldraw = \path[fill,draw]
\shade = \path[shade]
. . .
```

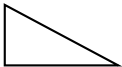
- ① 图形概述
- ② PGF/TikZ 宏包介绍
- ③ 路径 (path)
- ④ 简单作图**
- ⑤ 填充
- ⑥ 作图选项
- ⑦ 阴影
- ⑧ 坐标
- ⑨ 箭头
- ⑩ 参数的作用域
- ⑪ 坐标变换
- ⑫ 重复
- ⑬ 添加标注
- ⑭ 函数作图与 plot 命令

简单作图

- 直线



```
\draw (0,0) -- (2,0) -- (0,1);
```



```
\draw (0,0) -- (2,0) -- (0,1) -- cycle;
```

- 圆 (圆心, 半径), 椭圆 (中心, 长半轴, 短半轴)



```
\draw (0,0) circle (10pt);
```



```
\draw (0,0) ellipse (20pt and 10pt);
```

简单作图

- 矩形 (对角线上的两个点)



```
\draw (0,0) rectangle (2,1);
```

- 网格



```
\draw[step=5pt] (0,0) grid (30pt,20pt);
```

→ 步长选项: `step`, `xstep`, `ystep`



```
\draw[xstep=10pt,ystep=5pt]%  
(0,0) grid (30pt,20pt);
```

- 圆弧 (起始点, 角度范围, 半径)



```
\draw (0,0) arc (0:135:1);
```

- 椭圆弧 (起始点, 角度范围, 长半轴, 短半轴)



```
\draw (0,0) arc (0:270:1 and 0.6);
```

- 圆角



```
\draw[rounded corners] (0,0) -- (0,0.5) -- (1,0.5);
```



```
\draw[rounded corners=10pt]%  
(0,0) -- (0,0.5) -- (1,0.5);
```


简单作图

- 抛物线 (顶点, 终点)



```
\tikz\draw[thick] (0,0) parabola (-1,2);
```

- 可以使用 `bend` 选项另外指定顶点



```
\draw (0,0) parabola bend (1,1) (2,0);
```

简单作图

- 一般曲线：三次 Bézier 曲线 (两个控制点)



```
\draw (0,0) .. controls (1,1)%
      and (2,1) .. (2,0);
```

- 若只给一个控制点，则第二个点与第一个点相同



```
\draw[line width=10pt] (0,0) .. controls (1,1) %
  .. (4,0) .. controls (5,0) and (5,1) .. (4,1);
\draw[color=gray] (0,0) -- (1,1) -- (4,0) %
  -- (5,0) -- (5,1) -- (4,1);
```

简单作图

- \sin, \cos 函数图形: $[0, \pi/2]$



```
\tikz\draw[thick] (0,0) sin (1,1);
```



```
\tikz\draw[thick] (0,0) sin (2,1);
```

- 只能画 $[0, \pi/2]$ 之间的图形



```
\tikz\draw (0,0) sin (1.57,1) cos (3.14,0) %
sin (4.71,-1) cos (6.28,0);
```

- ① 图形概述
- ② PGF/TikZ 宏包介绍
- ③ 路径 (path)
- ④ 简单作图
- ⑤ 填充**
- ⑥ 作图选项
- ⑦ 阴影
- ⑧ 坐标
- ⑨ 箭头
- ⑩ 参数的作用域
- ⑪ 坐标变换
- ⑫ 重复
- ⑬ 添加标注
- ⑭ 函数作图与 plot 命令

填充

- 填充: `\fill`, `\filldraw`



```
\fill[red] (0,0) rectangle (2,1);
```

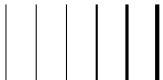


```
\filldraw[fill=red,draw=blue]%  
(0,0) rectangle (2,1);
```

- ① 图形概述
- ② PGF/TikZ 宏包介绍
- ③ 路径 (path)
- ④ 简单作图
- ⑤ 填充
- ⑥ 作图选项**
- ⑦ 阴影
- ⑧ 坐标
- ⑨ 箭头
- ⑩ 参数的作用域
- ⑪ 坐标变换
- ⑫ 重复
- ⑬ 添加标注
- ⑭ 函数作图与 plot 命令

作图选项

- 线条的粗细: thin, very thin, ultra thin, thick, very thick, ...
- line width=长度: 手工设置线条的粗细



```

\begin{tikzpicture}
  \draw[very thin] (0,0) -- (0,1);
  \draw[thin] (0,0) -- (0,1);
  \draw (0,0) -- (0,1);
  \draw[thick] (0,0) -- (0,1);
  \draw[very thick] (0,0) -- (0,1);
  \draw[ultra thick] (0,0) -- (0,1);
\end{tikzpicture}

```

作图选项

- 线的形状: `solid` (缺省值), `dashed`, `densely dashed`, `loosely dashed`, `dotted`, `densely dotted`, `loosely dotted`
- `dash pattern=值`: 手工设置线条形状

```

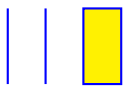
\begin{tikzpicture}[thick]
  \draw (0,0) -- (0,2);
  \draw[dotted] (0,0) -- (0,2);
  \draw[loosely dotted] (0,0) -- (0,2);
  \draw[densely dotted] (0,0) -- (0,2);
  \draw[dashed] (0,0) -- (0,2);
  \draw[dash pattern=on 2pt off 3pt on 4pt %
    off 4pt] (0,0) -- (0,2);
  \draw[dash pattern=on 10pt off 10pt, %
    dash phase=5pt] (0,0) -- (0,2);
\end{tikzpicture}

```

- `pattern`: 设置虚线的格式, 如 “on 2pt off 3pt on 4pt off 4pt” 表示 “画 2pt 实线, 空 3pt, 再画 4pt, 然后再空 4pt, 不断重复这个过程”
- `phase`: 第一根虚线的缩短长度

作图选项

- 颜色: `color=颜色`, `draw=颜色`, `fill=颜色`, . . .




```

\begin{tikzpicture}[thick]
  \draw[color=blue] (0,0) -- (0,1);
  \path[draw=blue] (0.5,0) -- (0.5,1);
  \draw[blue,fill=yellow] (1,0) rectangle (1.5,1);
\end{tikzpicture}

```

- 透明度: `opacity=值`



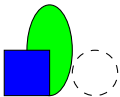
```

\begin{tikzpicture}
  \fill[blue] (0,0) rectangle (0.5,1);
  \fill[blue,opacity=0.5] (1,0) rectangle (1.5,1);
\end{tikzpicture}

```

作图选项

- 缩小或放大图形: `scale=倍数`



```
\begin{tikzpicture}[scale=0.6]
  \draw[style=dashed] (2,.5) circle (0.5);
  \draw[fill=green] (1,1) ellipse (.5 and 1);
  \draw[fill=blue] (0,0) rectangle (1,1);
\end{tikzpicture}
```

作图选项

- 双线: `double`

→ 双线之间的距离: `distance=距离`

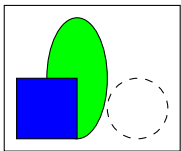


```
\begin{tikzpicture}[very thick]
  \draw[double] (0,0) arc (180:90:1cm);
  \draw[double distance=2pt] %
    (1,0) arc (180:90:1cm);
  \draw[thin, double distance=2pt] %
    (2,0) arc (180:90:1cm);
\end{tikzpicture}
```

→ `double distance` 之间不能加逗号

- 显示绘图区域边界: `backgrounds` 扩展

```
\usetikzlibrary{backgrounds}
```



```
\begin{tikzpicture}%
    [scale=0.8, show background rectangle]
    \draw[style=dashed] (2,.5) circle (0.5);
    \draw[fill=green] (1,1) ellipse (.5 and 1);
    \draw[fill=blue] (0,0) rectangle (1,1);
\end{tikzpicture}
```

- 可使用 `minipage` 环境或 `\hspace*`, `\vspace` 将图像放置到指定的地方

- ① 图形概述
- ② PGF/TikZ 宏包介绍
- ③ 路径 (path)
- ④ 简单作图
- ⑤ 填充
- ⑥ 作图选项
- ⑦ 阴影**
- ⑧ 坐标
- ⑨ 箭头
- ⑩ 参数的作用域
- ⑪ 坐标变换
- ⑫ 重复
- ⑬ 添加标注
- ⑭ 函数作图与 plot 命令

阴影

- 阴影: `\shade`, `\shadedraw`



```
\shade (0,0) rectangle (2,1)
```

- 默认是从上到下，从灰到白渐变
- 可使用下面的选项指定颜色的变化方式
`left`, `right`, `top`, `bottom`, `inner`, `outer`, `ball`

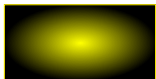
阴影



```
\tikz\shade[top color=yellow,%
             bottom color=black]%
(0,0) rectangle (2,1);
```



```
\tikz\shade[left color=yellow,%
             right color=black]%
(0,0) rectangle (2,1);
```




```
\tikz\shadedraw[inner color=yellow,%
                 outer color=black,draw=yellow]%
(0,0) rectangle (2,1);
```



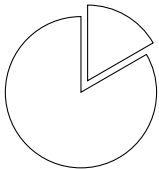
```
\tikz\shade[ball color=green]%
(0,0) circle (.5cm);
```

- ① 图形概述
- ② PGF/TikZ 宏包介绍
- ③ 路径 (path)
- ④ 简单作图
- ⑤ 填充
- ⑥ 作图选项
- ⑦ 阴影
- ⑧ 坐标
- ⑨ 箭头
- ⑩ 参数的作用域
- ⑪ 坐标变换
- ⑫ 重复
- ⑬ 添加标注
- ⑭ 函数作图与 plot 命令

坐标

 **坐标:** 可通过以下几种方式来指定

- 使用直角坐标 (x, y) , 如: $(0,1)$, $(0.4\text{cm},5\text{pt})$
- 使用极坐标 $(\theta : r)$, 如: $(30:1\text{cm})$
- 使用相对位置:
 - 一个加号: $+(0,5\text{pt})$ (从指定的点向上移 5pt)
 - 两个加号: $++(5\text{pt},0\text{pt})$ (从当前点向右移 5pt, 并把此点作为新的当前点)
- 相对位置的使用



```

\draw (0,0) -- (90:1cm) arc (90:360:1cm) %
      arc (0:30:1cm) -- cycle;
\draw (60:5pt) -- +(30:1cm) arc %
      (30:90:1cm) -- cycle;

```

坐标



```
\draw (0,0) -- ++(1cm,0cm) -- ++(0cm,1cm)
      -- ++(-1cm,0cm) -- cycle;
```



```
\draw (0,0) -- +(1cm,0cm) -- +(1cm,1cm)
      -- +(0cm,1cm) -- cycle;
```

→ 一个加号：不更新当前点的位置；两个加号：更新当前点的位置

- 使用交点



```
\draw (0,0) -- (1,1);
\draw (0,1) -- (1,0);
\draw[blue] (0,0.5) -- %
(intersection of 0,0--1,1 and 0,1--1,0);
```

→ 两条线的起点和终点不能加括号

- ① 图形概述
- ② PGF/TikZ 宏包介绍
- ③ 路径 (path)
- ④ 简单作图
- ⑤ 填充
- ⑥ 作图选项
- ⑦ 阴影
- ⑧ 坐标
- ⑨ 箭头**
- ⑩ 参数的作用域
- ⑪ 坐标变换
- ⑫ 重复
- ⑬ 添加标注
- ⑭ 函数作图与 plot 命令

箭头

- 箭头



```
\draw[->] (0,0) -- (1,1);
```

```
\draw[<->] (0,0) -- (1,1);
```

→ 使用各种类型的箭头：需要调用 `arrows` 扩展














```
\usetikzlibrary{arrows}
```













```
\draw[thick,->,>=stealth] (0,0) -- (1,1);
```

```
\draw[o-stealth] (0,0) -- (1,1);
```

箭头

	<code>\draw[->,>=latex] (0,0) -- (1,0);</code>
	<code>\draw[->,>=triangle 90] (0,0) -- (1,0);</code>
	<code>\draw[->,>=triangle 60] (0,0) -- (1,0);</code>
	<code>\draw[->,>=triangle 45] (0,0) -- (1,0);</code>
	<code>\draw[->,>=open triangle 90] (0,0)--(1,0);</code>
	<code>\draw[->,>=angle 90] (0,0) -- (1,0);</code>
	<code>\draw[->,>=angle 60] (0,0) -- (1,0);</code>
	<code>\draw[->,>=angle 45] (0,0) -- (1,0);</code>
	<code>\draw[->,>=hooks] (0,0) -- (1,0);</code>
	<code>\draw[(-)] (0,0) -- (1,0);</code>
	<code>\draw[o-*] (0,0) -- (1,0);</code>
	<code>\draw[diamond-open diamond] (0,0)--(1,0);</code>
	<code>\draw[->,>=serif cm] (0,0) -- (1,0);</code>

箭头

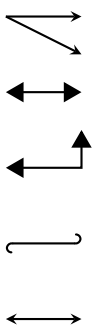
	<code>\draw[->,>=left to] (0,0) -- (1,0);</code>
	<code>\draw[->,>=right to] (0,0) -- (1,0);</code>
	<code>\draw[->,>=left hook] (0,0) -- (1,0);</code>
	<code>\draw[->,>=right hook] (0,0) -- (1,0);</code>
	<code>\draw[->,>=angle 60 reversed] (0,0)--(1,0);</code>
	<code>\draw[<->,>=angle 60] (0,0)--(1,0);</code>
	<code>\draw[->,>=round cap] (0,0) -- (1,0);</code>
	<code>\draw[->,>=butt cap] (0,0) -- (1,0);</code>
	<code>\draw[->,>=triangle 90 cap] (0,0)--(1,0);</code>
	<code>\draw[->,>=fast cap] (0,0) -- (1,0);</code>

- `reversed` 参数几乎对所以箭头都适用
- 折线、弧线也可以加箭头
- 最后四个命令中加了参数: `line width=1ex`

- ① 图形概述
- ② PGF/TikZ 宏包介绍
- ③ 路径 (path)
- ④ 简单作图
- ⑤ 填充
- ⑥ 作图选项
- ⑦ 阴影
- ⑧ 坐标
- ⑨ 箭头
- ⑩ 参数的作用域**
- ⑪ 坐标变换
- ⑫ 重复
- ⑬ 添加标注
- ⑭ 函数作图与 plot 命令

参数的作用域

- `\begin{tikzpicture}` [参数]: 对整个绘图起作用
- `draw` [参数]: 对所绘的图形起作用
- `scope` 环境: 可用来控制局部环境中的参数作用范围



```






\begin{tikzpicture}[>=stealth]
  \draw[->] (0,4) -- (1,4);
  \draw[->] (0,4) -- (1,3.5);
  \begin{scope}[>=triangle 60]
    \draw[<->] (0,3) -- (1,3);
    \draw[<->] (0,2) -- (1,2) -- (1,2.5);
    \draw[<->,>=left hook] (0,1) -- (1,1);
  \end{scope}
  \draw[<->] (0,0) -- (1,0);
\end{tikzpicture}

```


参数的作用域

- 调用 `scopes` 扩展后，可以使用**大括号**替代 `scope` 环境

```
\usetikzlibrary{scopes}
```

	<code>\begin{tikzpicture}[>=stealth]</code>
	<code>\draw[->] (0,4) -- (1,4);</code>
	<code>\draw[->] (0,4) -- (1,3.5);</code>
	<code>{ [>=triangle 60]</code>
	<code>\draw[<->] (0,3) -- (1,3);</code>
	<code>\draw[<->] (0,2) -- (1,2) -- (1,2.5);</code>
	<code>\draw[<->,>=left hook] (0,1) -- (1,1);</code>
	<code>}</code>
	<code>\draw[<->] (0,0) -- (1,0);</code>
	<code>\end{tikzpicture}</code>

参数的作用域

- 可以在一个路径内部使用 `scope`



```
\tikz\draw (0,0) -- (1,1) %
           -- (2,0) -- (3,1)
           -- (3,0) -- (2,1); \\\[10pt]
```



```
\tikz\draw (0,0) -- (1,1) %
           { [rounded corners]--(2,0)--(3,1) }%
           -- (3,0) -- (2,1);
```

→ 注: 有许多选项不支持这种用法, 如线的颜色等

- ① 图形概述
- ② PGF/TikZ 宏包介绍
- ③ 路径 (path)
- ④ 简单作图
- ⑤ 填充
- ⑥ 作图选项
- ⑦ 阴影
- ⑧ 坐标
- ⑨ 箭头
- ⑩ 参数的作用域
- ⑪ 坐标变换**
- ⑫ 重复
- ⑬ 添加标注
- ⑭ 函数作图与 plot 命令

坐标变换


- `xshift`, `yshift`, `shift`: 移动一定的距离 (要带单位)

如: `xshift=2pt`, `shift={{(2pt,3pt)}}`, ...

```

\begin{tikzpicture}
  \draw (0,0) -- (1,0) %
  [yshift=3pt] (0,0) -- (1,0);
\end{tikzpicture}


```



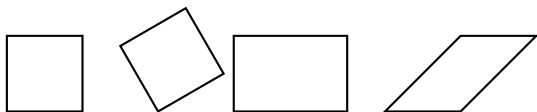
```

\begin{tikzpicture}
  \fill[blue] (0,0) circle (2pt) %
  [shift={{(5pt,5pt)}}] (0,0) circle (2pt) %
  [shift={{(5pt,5pt)}}] (0,0) circle (2pt);
\end{tikzpicture}

```



- rotate: 旋转一定角度, 如: rotate=30
- scale, xscale, yscale: 缩放, 如:
scale=2, xscale=-1 (翻转)
- xslant, yslant: 倾斜



```

\begin{tikzpicture}
  \draw (0,0) rectangle (1,1);
  \draw[xshift=2 cm,rotate=30] (0,0) rectangle (1,1);
  \draw[xshift=3 cm,xscale=1.5] (0,0) rectangle (1,1);
  \draw[xshift=5 cm,xslant=1] (0,0) rectangle (1,1);
\end{tikzpicture}

```

- ① 图形概述
- ② PGF/TikZ 宏包介绍
- ③ 路径 (path)
- ④ 简单作图
- ⑤ 填充
- ⑥ 作图选项
- ⑦ 阴影
- ⑧ 坐标
- ⑨ 箭头
- ⑩ 参数的作用域
- ⑪ 坐标变换
- ⑫ 重复**
- ⑬ 添加标注
- ⑭ 函数作图与 plot 命令

重复动作

- 重复动作：for 循环

`\foreach` 变量 in {值表} 命令

- 命令可以使用大括号括起来，或以分号结束

$y = 1, y = 2, y = 3,$ `\foreach \x in {1,2,3} {$y=\x,$};`



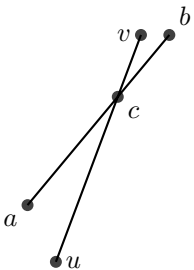
`\foreach \x in {1,...,10}`
`\draw (\x,0) circle (3pt);`

- 变量名前要加反斜线
- `\foreach` 可是使用多个变量

- ① 图形概述
- ② PGF/TikZ 宏包介绍
- ③ 路径 (path)
- ④ 简单作图
- ⑤ 填充
- ⑥ 作图选项
- ⑦ 阴影
- ⑧ 坐标
- ⑨ 箭头
- ⑩ 参数的作用域
- ⑪ 坐标变换
- ⑫ 重复
- ⑬ 添加标注**
- ⑭ 函数作图与 plot 命令

添加标注

- `\coordinate [label=角度:标注] (标记) at (x,y)`



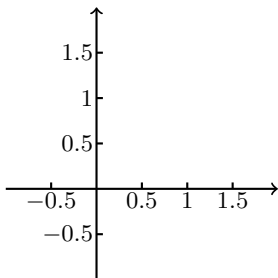
```

\begin{tikzpicture}[thick]
  \coordinate [label=-135:$a$] (a) at (0,0);
  \coordinate [label=45:$b$] (b) at (2.5,3);
  \coordinate [label=0:$u$] (u) at (0.5,-1);
  \coordinate [label=180:$v$] (v) at (2,3);
  \draw (a) -- (b) (u) -- (v);
  \coordinate [label=-45:$c$] (c) at %
    (intersection of a--b and u--v);
  \foreach \p in {a,b,c,u,v} \fill %
    [opacity=0.75] (\p) circle (2pt);
\end{tikzpicture}

```

添加标注

- 添加标注: `node`



```

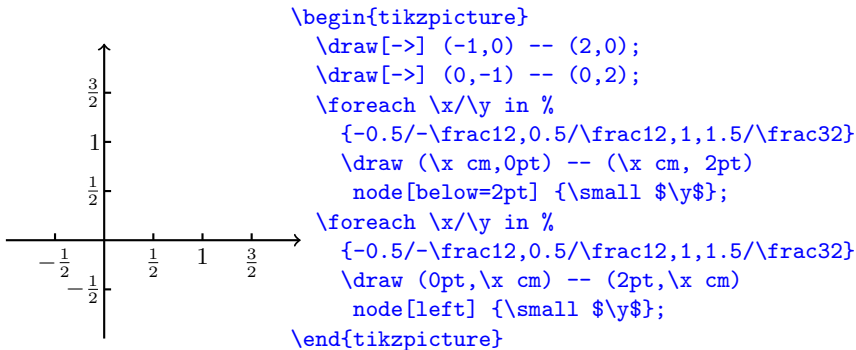
\begin{tikzpicture}
  \draw[>-] (-1,0) -- (2,0);
  \draw[>-] (0,-1) -- (0,2);
  \foreach \x in {-0.5,0.5,1,1.5}
    \draw (\x cm,0pt) -- (\x cm, 2pt)
      node[anchor=north] {\small $\x$};
  \foreach \y in {-0.5,0.5,1,1.5}
    \draw (0pt,\y cm) -- (2pt,\y cm)
      node[anchor=east] {\small $\y$};
\end{tikzpicture}

```

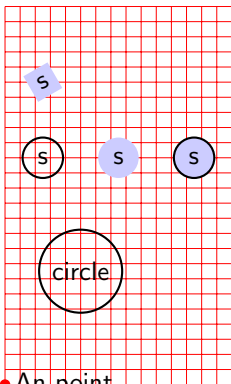
→ `anchor` 选项: 指定文本排放的相对位置, 取值有:
`north`, `south`, `west`, `east` 以及它们的组合

\foreach 多变量举例

→ 也可以直接使用 `below`, `above`, `left`, `right` 等，并可以设置额外的平移距离



- `\node[选项] at (x,y) [选项] {text}`



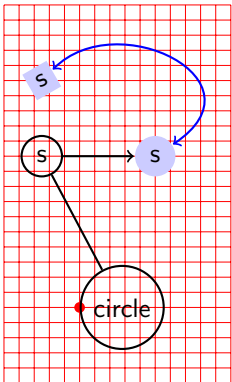
```

\begin{tikzpicture}[thick,fill=blue!20]
  \draw[step=0.2cm,red,very thin]%
    (0,0) grid (3,5);
  \fill[red] (0,0) circle (2pt);
  \node[right=0] at (0,0) {An point};
  \node at (1,1.5) [circle,draw] {circle};
  \node at (0.5,3) [circle,draw] {s};
  \node at (1.5,3) [circle,fill] {s};
  \node at (2.5,3) [circle,draw,fill] {s};
  \node at (0.5,4) [fill,rotate=30] {s};
\end{tikzpicture}

```

- An point

- `\node(标记) [选项] at (x,y) [选项] {text}`



```

\begin{tikzpicture}[thick,fill=blue!20]
  \draw[step=0.2cm,red,very thin]%
    (0,0) grid (3,5);
  \fill[red] (1,1) circle (2pt);
  \node(a)[right=0] at (1,1) %
    [circle,draw] {circle};
  \node(b) at (0.5,3) [circle,draw] {s};
  \node(c) at (2,3) [circle,draw,fill] {s};
  \node(d) at (0.5,4) [fill,rotate=30] {s};
  \draw (a) -- (b) [->]-- (c);
  \draw[blue,<->] (d) .. controls +(1,1) %
    and +(1.5,1) .. (c);
\end{tikzpicture}

```

- ① 图形概述
- ② PGF/TikZ 宏包介绍
- ③ 路径 (path)
- ④ 简单作图
- ⑤ 填充
- ⑥ 作图选项
- ⑦ 阴影
- ⑧ 坐标
- ⑨ 箭头
- ⑩ 参数的作用域
- ⑪ 坐标变换
- ⑫ 重复
- ⑬ 添加标注
- ⑭ 函数作图与 plot 命令

函数作图与 plot 命令

- 函数作图: `plot`

```
plot[选项] coordinates {(x1, y1) (x2, y2) ... (xn, yn)}
```

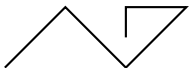
```
plot[选项] file {文件名}
```

```
plot[选项] 坐标表达式
```

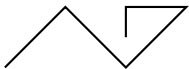
```
plot[选项] function {函数 (gnuplot)}
```

plot

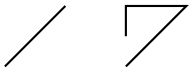
```
plot[选项] coordinates  $\{(x_1, y_1) (x_2, y_2) \dots (x_n, y_n)\}$ 
```



```
\begin{tikzpicture}
  \draw plot coordinates %
    {(0,0) (1,1) (2,0) (3,1) (2,1) (2,0.5)};
\end{tikzpicture}
```



```
\begin{tikzpicture}
  \draw (0,0) -- (1,1) plot coordinates %
    {(2,0) (3,1) (2,1) (2,0.5)};
  \draw (0,0) -- (1,1) -- plot coordinates %
    {(2,0) (3,1) (2,1) (2,0.5)};
\end{tikzpicture}
```



plot

→ 光滑选项: `smooth`



```
\begin{tikzpicture}[thick,scale=0.8]
  \draw[yshift=2cm] plot coordinates %
    {(0,0) (1,1) (2,0) (3,1) (2,1) (2,0.5)};
```



```
\draw plot[smooth] coordinates %
  {(0,0) (1,1) (2,0) (3,1) (2,1) (2,0.5)};
\end{tikzpicture}
```

→ `smooth cycle`: 光滑封闭曲线

plot

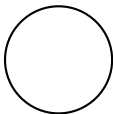
→ 松紧度选项: `tension`, 缺省值为: 0.55



```
\begin{tikzpicture}[thick,smooth cycle]
\draw[yshift=4cm] plot coordinates %
{(0,0) (1,0) (1,1) (0,1)};
```



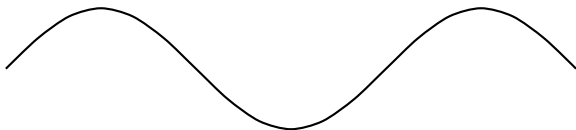
```
\draw[yshift=2cm] plot[tension=0.2] %
coordinates {(0,0) (1,0) (1,1) (0,1)};
```



```
\draw plot[tension=1] coordinates %
{(0,0) (1,0) (1,1) (0,1)};
\end{tikzpicture}
```

利用文件中的数据绘图

```
plot[选项] file {文件名}
```



```
\begin{tikzpicture}[thick,smooth,scale=0.8]  
  \draw plot file {pgf-file.table};  
\end{tikzpicture}
```

- 数据文件要求：每行只能有两个数字，用空格隔开；
井号 “#” 或百分号 “%” 开始的行为注释行

坐标表达式绘图

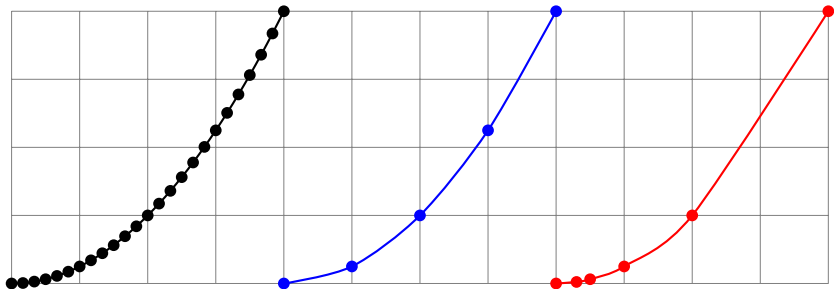
```
plot[选项] {坐标表达式}
```



```
\begin{tikzpicture}[thick,smooth,scale=0.6]
  \draw plot[domain=0:2] (\x,{\x * \x/2});
\end{tikzpicture}
```

● 相关选项

- `variable=变量名`: 指定变量, 默认为 x
- `samples=数`: 样本点的个数, 默认为 25
- `domain=a:b`: 绘图区间, 默认为 $-5:5$
- `samples at={ x_1, x_2, \dots, x_n }`: 指定样本点



```

\begin{tikzpicture}[thick,smooth,domain=0:4,mark=*]
  \draw[very thin,gray] (0,0) grid (12,4);
  \draw plot (\x,{\x * \x/4});
  \draw[blue,xshift=4cm] %
    plot[samples=5,variable=\t] (\t,{\t * \t/4});
  \draw[red,xshift=8cm] %
    plot[samples at={0,0.3,0.5,1,2,4}] (\x,{\x * \x/4});
\end{tikzpicture}

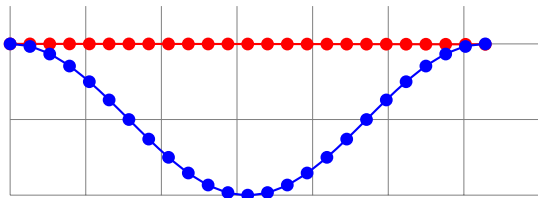
```

● 支持的数学运算和函数

→ +, -, *, /, ^

→ abs, exp, ln, pow, sqrt, sin, cos, tan, sec, pi, ...

→ x r → 将弧度转化成度数



```
\begin{tikzpicture}[thick,smooth,domain=0:2*pi,mark=*]
  \draw[very thin,gray] (0,-1) grid (12,2);
  \draw[red] plot (\x,{cos(\x)});
  \draw[blue] plot (\x,{cos(\x r)});
\end{tikzpicture}
```

Gnuplot 绘图

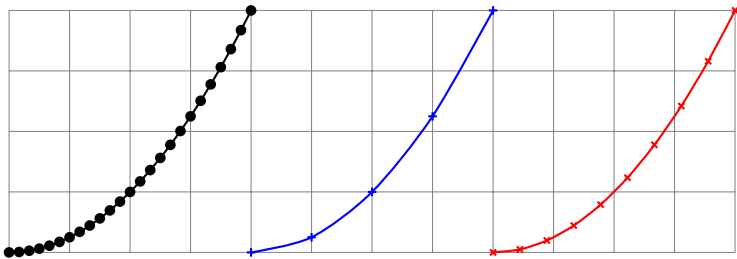
- 与 Gnuplot 结合绘图 (需要安装 gnuplot)

```
plot[选项] function {函数 (gnuplot)}
```

```
\begin{tikzpicture}[thick,smooth,domain=0:4]
  \draw[very thin,gray] (0,0) grid (4,4);
  \draw plot function{x*x/4};
\end{tikzpicture}
```

标记选项

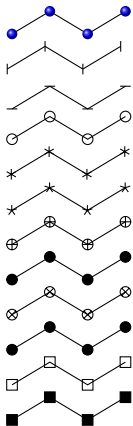
- **mark** 选项：给绘图节点做标记，取值有： $*$ (点), $+$ (加号), x (叉号)



```
\begin{tikzpicture}[thick,smooth,domain=0:4,scale=0.9]
  \draw[very thin,gray] (0,0) grid (12,4);
  \draw plot[mark=*] (\x,{\x * \x/4});
  \draw[blue,xshift=4cm] plot[samples=5,mark=+] (\x,{\x * \x/4});
  \draw[red,xshift=8cm] plot[samples=10,mark=x] (\x,{\x * \x/4});
\end{tikzpicture}
```


标记选项

- 更多标记选项: `\usetikzlibrary{plotmarks}`



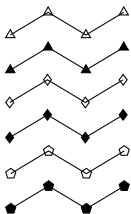
```

\tikz\draw plot[mark=ball] ...
\tikz\draw plot[mark=|] ...
\tikz\draw plot[mark=-] ...
\tikz\draw plot[mark=o] ...
\tikz\draw plot[mark=asterisk] ...
\tikz\draw plot[mark=star] ...
\tikz\draw plot[mark=oplus] ...
\tikz\draw plot[mark=oplus*] ...
\tikz\draw plot[mark=otimes] ...
\tikz\draw plot[mark=otimes*] ...
\tikz\draw plot[mark=square] ...
\tikz\draw plot[mark=square*] ...

```

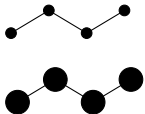
标记选项

- 更多标记选项: `\usetikzlibrary{plotmarks}`



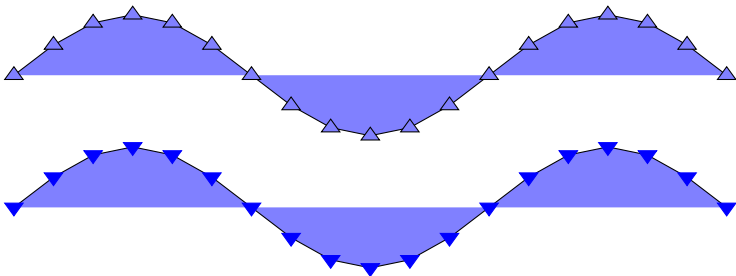
```
\tikz\draw plot[mark=triangle] ...
\tikz\draw plot[mark=triangle*] ...
\tikz\draw plot[mark=diamond] ...
\tikz\draw plot[mark=diamond*] ...
\tikz\draw plot[mark=pentagon] ...
\tikz\draw plot[mark=pentagon*] ...
```

- 设置标记的大小: `mark size`



```
\tikz\draw plot[mark=*] coordinates ...
\tikz\draw plot[mark=*,mark size=1ex] ...
```

- 设置标记的属性: `mark options={. . .}`

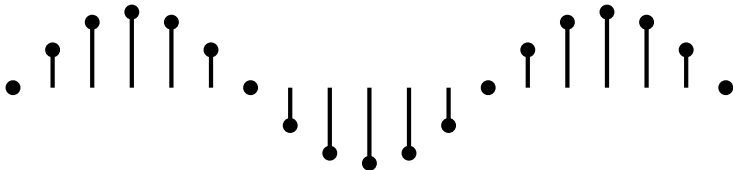


```
\tikz\draw plot[mark=triangle*] file {pgf-file.table};
\tikz\draw plot[mark=triangle*, %
  mark options={color=red,rotate=180}] file {pgf-file.table};
```

其它选项

- 其它选项

→ `ycomb`, `xcomb`, `polar comb`

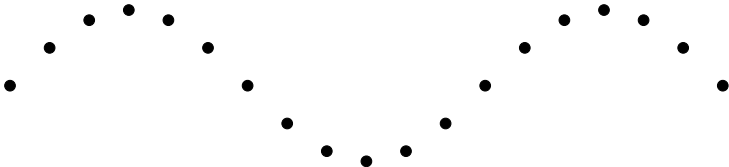


```
\tikz\draw[ultra thick] plot[ycomb,mark=*] %  
file {pgf-file.table};
```

其它选项

- 其它选项

→ `only marks`



```
\tikz\draw plot[only marks,mark=*] %  
file {pgf-file.table};
```

其它选项

- 图形的 **Bounding Box**: 绘图结束后, pgf 会计算出所画图形的边界, 并改变图形盒子的大小, 使之与实际图形的边界相吻合
- **baseline**: 设置图形盒子的基线 (与周边文本的对齐方式)
 - 没使用这个选项时, 图形盒子的底线为基线

$A \rightarrow B$ `$A \mathbin{\tikz %`
 `\draw[->>] (0pt,0.5ex) -- (3ex,0.5ex);} B$`

$A \rightarrow B$ `$A \mathbin{\tikz[baseline=0pt] %`
 `\draw[->>] (0pt,0.5ex) -- (3ex,0.5ex);} B$`

$A \rightarrow B$ `$A \mathbin{\tikz[baseline=2pt] %`
 `\draw[->>] (0pt,0.5ex) -- (3ex,0.5ex);} B$`

- PGF/TikZ 的更多用法: 参见 [pgfmanual.pdf](#)