

# 1. Gnuplot 简易教程

(第二次增添修改)

张晓军 吉林大学物理学院 博士生。

[mr-zhangxj@hotmail.com](mailto:mr-zhangxj@hotmail.com)

## 写在前面的话

本人的 Gnuplot 的学习笔记，资料来源主要来自 Google 和 Gnuplot 自身。不少资料上声称 Gnuplot 要比 Origin 跟强大。本人不置可否。显然，如果要快速得到一个图表，Origin 甚至不需要学习。鼠标点点即可搞定。相比之下 Gnuplot 需要记住一些命令，如果对打字比较慢的话还会大大的降低效率。如果你的目的很简单，就是为了文章中得一个数据图，仅此而已，选择 Origin, It's a best choice. 如果你喜欢 Linux，喜欢在命令行下做事情。学一学 Gnuplot 还是不错的。Gnuplot 提供给你大量个性化的选择。还有一点就是结合了 bash 脚本后，可以进行批处理工作：比如本人需要处理上百个数据文件的画图问题。如果用 Origin，要点太多次的鼠标。（只是个人看法，Origin 能不能实现批处理的问题，我不会。）

之说以称这个教程是简易教程，主要有两点：1、内容少，与 gnuplot 的官方文件相比，这个教程实在是精简了许多。但是，我在初学 gnuplot 的时候，一直想要找到这样的教程。2、这个教程语言不专业。甚至有的专业名词是错误的。因为本人不专业。见谅。

多说几句。学习 Linux 就是用它来做点事情。比如，纯文本编辑可以用 emacs，它的教程是自带的，而且写的非常好。自动办公软件一般 gnome 自带。生成 PDF 用 latex，现在叫 texlive 了。随便大一点儿的书店就能买到教程了。科学绘图使用 gnuplot。普通的画图软件 gnome 自带了。程序语言的话，gcc 是 C 语言的编译器。g++ 编译 C++， gfortran 编译 fortran 语言。Octave 是类似 matlab 的东西，很容易学习。至于 mathematica，这个东西有 Unix 版。是可以在 Linux 下安装使用的。不过得花钱，你懂的，不说了。查看 pdf 文件的问题，gnome 自带的文件查看器可以搞定了，xpdf 也可以查看 PDF 文件。所以，如果你想做科研，足够了。至于 bash 脚本的使用。你可以百度“Linux 视频教程”可以找到一个叫做“尚观”网站。这里的 Linux 的初级教程是免费的，而且很好。如果网速太慢就去网吧看吧。

我的意思是。不要把精力放在字体美化，3D 桌面什么的上面。花了很多精力，不值，已经有很多东西要学习了，不是吗？记得当年，我想把 gnome 的字体搞得好看点，花费了很多时间。后来才发现。好多东西都是在命令行下作的。呵呵。白扯了。

还有啊，你可别相信这样的话“学习 Linux 就把 Windows 卸了”。应该在 Windows 在安装虚拟机，在虚拟机上安装 Linux。这样不耽误事儿。有些言论不可轻信。比如，经常看到别人说 Windows 很烂，不如 Linux。你想想 Windows 那么烂，为什么那多数的电脑是安装 Windows 呢？以前 Ubuntu 的宣传口号只是说“你有另外一种选择”。

学习 Linux 最快的途径是找到一个 Linux 高手，问他。你还在大学里的话。肯定有个师兄师姐在用 Linux，抓住他（她）不放。当然很多事情还是的自己解决。当然官方资料英文居多。不难，看着看着你就能看明白了。而且你会发现，官方资料考虑到了这一点。写的很啰嗦的。呵呵。跑题了。

在此感谢一下曾经帮助我学习 Linux 的人们。

PS: 本人水平一般, 难免出错, 多多指正

## 文中的符号

\$           普通用户在终端输入的命令  
#           root 用户在命令行下如入的命令 (明确指出是脚本文件的内容的情况除外)  
gnuplot>   进入 gunplot 后输入的命令

## 文档内容

获得 gnuplot .....	2
获得帮助.....	3
1 函数绘图 .....	4
1.1 数据类型 .....	4
1.2 自定义函数 .....	5
1.3 操作符 .....	6
2 导出图片 .....	6
2.1 图片格式 .....	6
2.2 图片大小 .....	7
3 样式 .....	8
3.1 标题, 横轴和纵轴标示.....	8
3.2 特殊字符和希腊字母.....	10
3.3 标度 .....	10
3.3 画图的样式 .....	12
3.4 key.....	13
3.5 边界 .....	14
4 脚本文件 .....	15
5 数据绘图 .....	15

# 获得Gnuplot和帮助

## 获得 gnuplot

对于 Debian 和 Ubuntu 使用命名

```
# apt-get install gnuplot  
$ sudo apt-get install install gnuplot
```

即可获得 Gnuplot, 当然这样安装的一般不是最新版本, 如果想要见识一下最新版可以到 GNU 的官方网站

<http://www.gnuplot.info/>

也可以到 GNU 的官方网站

<http://www.gnu.org/>

这里可以找到所有的 GNU 软件。

## 获得帮助

Gnuplot官网 (<http://www.gnu.org/>) 上有大量帮助。特别是其中的例子 (Demos)，这是可以快速开始使用Gnuplot的有效方法。Gnuplot最牛叉的帮助在这里：

<http://www.gnuplot.info/documentation.html>

找到最新版的 official document，那里有个 PDF 版本的东西，下载下来看看吧 224 页，而且只有英文和日文两种。呵呵，不过这里的资料的确是最权威，最齐全的了。这本书的名字就叫做 <Gnuplot 4.4>。我们后面会提到它的。

还有一个地方，Baidu -> gnuplot 就会找到

[http://dsec.pku.edu.cn/dsectest/dsec\\_cn/gnuplot/](http://dsec.pku.edu.cn/dsectest/dsec_cn/gnuplot/)

这里的内容也不错。如果想要最便捷的信息，使用 Gnuplot 自带的帮助系统，当然你就要看英文了。下面就说说这个自带帮助：

安装好 Gnuplot 后，在终端输入 gnuplot。Gnuplot 顺便给出了版本信息。我的电脑上给出的信息是这样的。

```
G N U P L O T
Version 4.4 patchlevel 0
last modified March 2010
system: Linux 2.6.32-686
...
```

也就是说，这里谈到的都是 4.4 的版本，当然，和别的版本比较起来，区别不大。如果要退出 Gnuplot，输入 q 回车即可

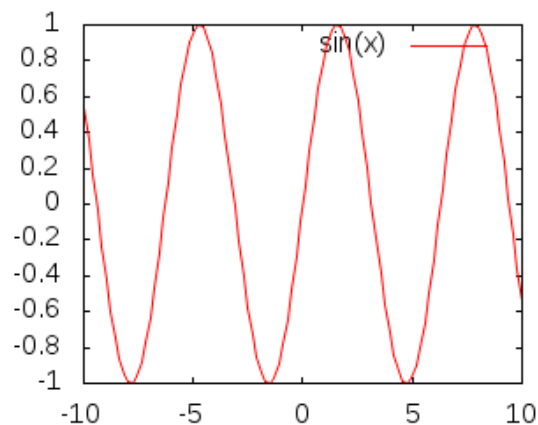
```
gnuplot> q
```

上面给出的命令，只需要输入 q，“gnuplot>”是 gnuplot 给出的提示符。在开始提到 gnuplot 自带的帮助之前，我们先画第一个图：启动 Gnuplot，输入：

```
gnuplot> plot sin(x)
```

这时就会看到一个图图跳出来了 (Fig 01)。你得到图图的长宽比例可能和我的不同，因为我调整过。Anyway，大致的图片就应该是这样的了。

Fig 01



如果你没有在桌面系统下，比如 Gnome。那么当然就不会有图蹦出来。使用命名 `help` 启动帮助，比如，我们看看 `plot` 命令的帮助是什么：

```
gnuplot> help plot
```

系统给出了一大堆帮助信息，当然我们最关心的 `plot` 函数的格式：

Syntax:

```
plot {<ranges>}  
{<iteration>}  
{<function> | {"<datafile>" {fatafile-modifiers}}}  
...
```

我只是列出了函数形式一部分。`{}`的意思是可选，`|`的意思是“或者”（这是我猜的）比如 `plot sin(x)`，只是用到了`{<function>}`部分。注意到终端的最下面写着：

```
Press return for more
```

那就按 `enter` 键吧，后面的信息就显示出来的了，你还可以看到一些例子。所有的信息显示完毕后，就推出帮助了，回到了

```
gnuplot>
```

如果你已经推出了帮助。再进入 `plot` 的帮助：

```
gnuplot> help plot
```

还有东西没说完。下面按 `q`（可能是 `question` 的简称），终端上显示出了：

```
Subtopics available for plot:
```

```
acsplines axes bezier
```

```
....(太多了，我省略了)
```

```
Subtopic of plot:
```

它给出了 `plot` 相关的下一层帮助，输入 `axes`，你就可以看到关于 `axes` 的帮助了，然后你还可以看 `axes` 的下一层帮助（如果 `axes` 头下一层的帮助的话，呵呵）。如何回到上一层？按回车，就会不停向上一层进军了。如果想要中途推出帮助系统，按 `Ctrl+c`

## 1 函数绘图

实际上面我们已经使用了函数绘图了，我们画了 `sin(x)` 的图像，Gnuplot 中还集成了很多函数，查看这些函数还是使用 `help`

```
gnuplot> help functions
```

是 `functions`，不要少了 `s`。这里的函数已经很多了，但是这里的也不是最全的，如果你知道最全的，就要到前面提到的那个 `<Gnuplot 4.4>` 的第 25 页上看看，哇，好多。如果看其中的一个函数的具体帮助，你会发现函数帮助不在 `functions` 的下一层。需要使用 `help sin` 来查看 `sin` 的帮助。如果你要画 `x` 从 50 到 100 区间的 `sin` 函数，那么

```
gnuplot> plot [50:100] sin(x)
```

画两条曲线，并使用 `t` 作为自变量

```
gnuplot> plot [t=50:100] sin(t), cos(t)
```

### 1.1 数据类型

Gnuplot 里的数据类型，其实这个软件比较智能，能够自动识别数据类型。比如 1、4、48 这都是整数，1.2、3.14、3e8 这都是实数。也支持复述类型{1,2}，试一试

```
gnuplot> a={2,3}
gnuplot> print a
gnuplot> {2.0, 3.0}
```

实部和虚部都变成实数了。这是 Gnuplot 自己干的，知道这些数据类型很重要，因为有的时候会影响结果，输入

```
gnuplot> print 2/3
gnuplot> 0
```

还有一个很特别的数 1/0，这个数会被忽略，这还是有用的。当然

```
gnuplot> print 1/0
```

会报错的。

## 1.2 自定义函数

Gnuplot 允许用户自定义函数。很简单，

```
gnuplot> homer(x)=sin(x)+rand(x)
gnuplot> plot homer(x)
```

得到了图片：(Fig 02)。你是不是发现没有画出来啊？使用命令

```
gnuplot> replot
```

这里有一个比较有趣的函数形式，需要提一下：

a?b:c

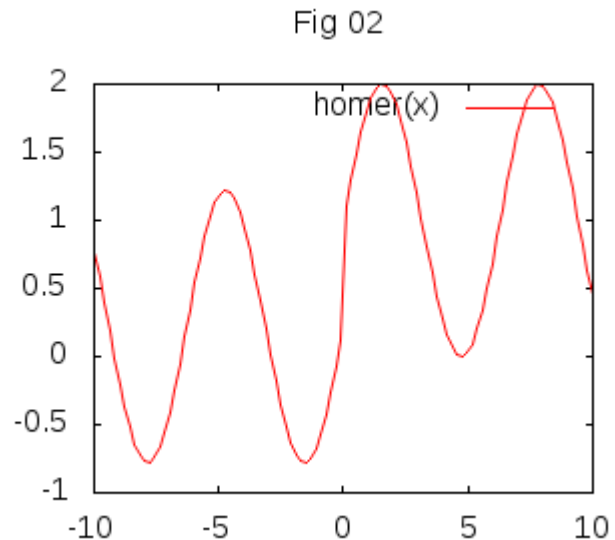
如果 a 为真，就计算 b，不然，计算 c

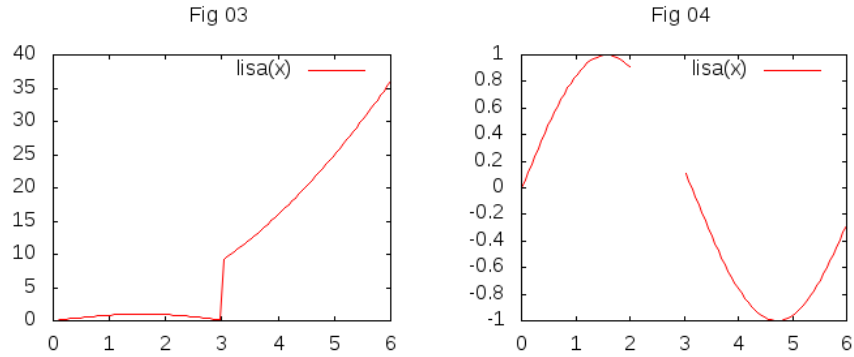
```
gnuplot> bart(x)=(x<3)? sin(x) : x ** 2
gnuplot> plot [0:6] bart(x)
gnuplot> replot
(Fig 03)
```

看一会儿就明白了，呵呵，下面在看一看 1/0 的威力。

```
gnuplot> lisa(x) = (x>2 && x<3)? 1/0 : sin(x)
gnuplot> plot [0:6] lisa(x)
gnuplot> replot
(Fig 04)
```

看到了吧，没了一段，呵呵。lisa(x)的意思是如果 x 在 2 和 3 之间，函数值忽略，其他的是 sin(x)。





### 1.3 操作符

这个我就从别的地方粘贴做了：呵呵

#### Binary Operators

Symbol	Example	Explanation
<code>**</code>	<code>a**b</code>	exponentiation
<code>*</code>	<code>a*b</code>	multiplication
<code>/</code>	<code>a/b</code>	division
<code>%</code>	<code>a%b</code>	* modulo
<code>+</code>	<code>a+b</code>	addition
<code>-</code>	<code>a-b</code>	subtraction
<code>==</code>	<code>a==b</code>	equality
<code>!=</code>	<code>a!=b</code>	inequality
<code>&amp;</code>	<code>a&amp;b</code>	* bitwise AND
<code>^</code>	<code>a^b</code>	* bitwise exclusive OR
<code> </code>	<code>a b</code>	* bitwise inclusive OR
<code>&amp;&amp;</code>	<code>a&amp;&amp;b</code>	* logical AND
<code>  </code>	<code>a  b</code>	* logical OR
<code>?:</code>	<code>a?b:c</code>	* ternary operation

\* 的意思是，只能是整数之间的操作，还有一个是！表示逻辑非。

## 2 导出图片

### 2.1 图片格式

导出图片需要设置两个东西，一个是你想要导出图片的类型，Gnuplot 里叫做 terminal（简称 term，它涉及的不只是文件类型）。还有就是图片的名称。

比如你要保存成 eps，写文章用 latex 时经常用吧。

```
gnuplot> set terminal postscript eps
```

可以简写为

```
gnuplot> set term post eps
```

然后设置保存文件的名称

```
gnuplot> set output "sin.eps"
```

名称中可以使用路径，如

```
gnuplot> set output "../pic/sin.eps"
```

但是这样是不能保存图片的。必须要

```
gnuplot> replot
```

才能保存成图片。如果需要带颜色。就这样

```
set term term post eps color
```

还有其他的图片格式，查看帮助

```
gnuplot> help terminal
```

比如 jpeg 格式

```
gnuplot> set term jpeg
```

```
gnuplot> set output "sin.jpg"
```

```
gnuplot> replot
```

本文档中使用的都是 png 图片，它的设置是：

```
gnuplot> set term png truecolor size 350, 300
```

```
gnuplot> set output "4.png"
```

```
gnuplot> replot
```

truecolor: png 图片的色彩就这么要求，看帮助就知道了。

size 是我们下面要提到的东西

## 2.2 图片大小

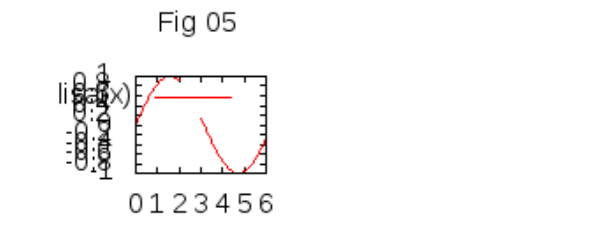
图片大小有两个设置方法，这个很不同，一个是在 term 设置

```
gnuplot> set term png truecolor size 350, 300
```

这指的是画布的大小，数字的单位是像素。如果是照片的话，画布就是整个照片的大小。还有就是

```
gnuplot> set size 0.5, 0.5
```

这设置的画布上的图像的大小，就是照片上，有颜色的景物的大小。这个数字 0.5 是比例，指原来的一半。使用了这个命令产生的效果是这样的。(Fig 05)



PS: 这里谈到的区别不适合 eps 图片, 而且说的是 `gnuplot 4.4`

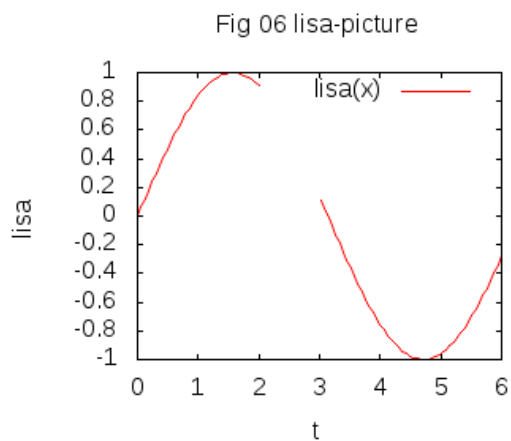
### 3 样式

#### 3.1 标题, 横轴和纵轴标示

加入标题, 和横轴、纵轴标示, 输入

```
gnuplot> lisa(x) = (x>2 && x<3)? 1/0 : sin(x)
gnuplot> plot [0:6] lisa(x)
gnuplot> set title "Fig 06 lisa-picture"
gnuplot> set xlabel "t"
gnuplot> set ylabel "lisa"
gnuplot> replot
```

(Fig 06)



下面就存在这样一个问题。在提交文章的时候, 编辑可能说你的图片的 `xlabel` 的字太小了, 那么你就需要变大字号。这个问题我们可以通过 `help` 来解决:



```
gnuplot> help xlabel
```

看到了 xlabel 的函数形式了吧:

Syntax:

```
set xlabel {"<label>"}{offset <offset>}{font "<font>{,<size>}" }
      {{textcolor | tc} {lt <line-type> | default}} {rotate by
<degrees>}...
```

这些单词肯定认识吧。font 就是关于字体和字号的设置。按照 help 给出的提示设置一下? 呵呵, 这回就没有那么容易了, Gnuplot 里没有自己的字体。对于 PNG (本文档中用到的图片格式), JPG, GIF 这三个格式的图片来说, 如果只是想改变字体, 那么 terminal 里自带了几个关键词来给你使用: tiny, small, medium, large, giant.

```
set term png tiny
```

这样图片上的字都是 tiny 规定的那么大。如果这个并不满足你的要求, 你想要改变默认字体, 和字体的大小。那么你要做的事情会多一点了。

首先, 要告诉 Gnuplot 字体放在了什么地方, 这是 bash 的一个环境变量 GDFONTPASH 传递的消息。那么字体在什么地方呢? 我的系统的字体文件在这里

```
/usr/share/fonts/truetype/ttf-dejavu/
```

这个目录里有很多 ttf 的文件。这就是字体文件。我们选用 DejaVuSans.ttf 这个字体文件。你的系统肯定也差不多。那么我们需要在

在终端输入

```
$ export GDFONTPASH="/usr/share/fonts/truetype/ttf-dejavu/"
```

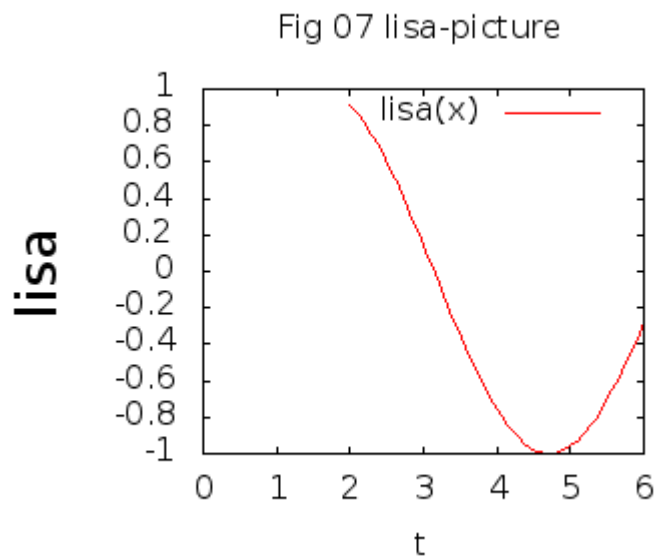
然后进入 gnuplot

```
Gnuplot> set term png truecolor font "DejaVuSans.ttf" size 350, 300
```

我选择了 DejaVuSans.ttf 这个字体文件。然后就好办了

```
gnuplot> set title "Fig 07 lisa-picture"
gnuplot> set ylabel "lisa"
gnuplot> set xlabel "t" font "DejaVuSans.ttf, 20"
gnuplot> lisa(x) = (x>2 && x<3)? 1/0 : sin(x)
gnuplot> plot [0:6] lisa(x)
gnuplot> replot
```

(Fig 07)



如果要处理 eps 图片，就没那么麻烦了：

```
gnuplot> set term post eps color
gnuplot> set xlabel "t" font "Courier, 20"
gnuplot> set size 0.45, 0.45
gnuplot> replot
```

就可以了。

### 3.2 特殊字符和希腊字母

使用希腊字母也很简单就是 enhanced 选项。

```
gnuplot> set term png enhanced
```

但是在我的电脑上不支持，我想是因为我的系统中的字体太少了。在官方的文件上说，png 图片是支持 enhanced 的。但是保存成 eps 图片的话，我的终端是支持的。

```
gnuplot> set term post eps enhanced
gnuplot> set xlabel "{/Symbol a}"
```

这样 x 轴的标示就是希腊字母了。

你可以在<gnuplot 4.4>的第 21 页找到其他的特殊字符。这里不说了。

ps： 你可以在终端用命令吧 eps 图片转化为 png 图片

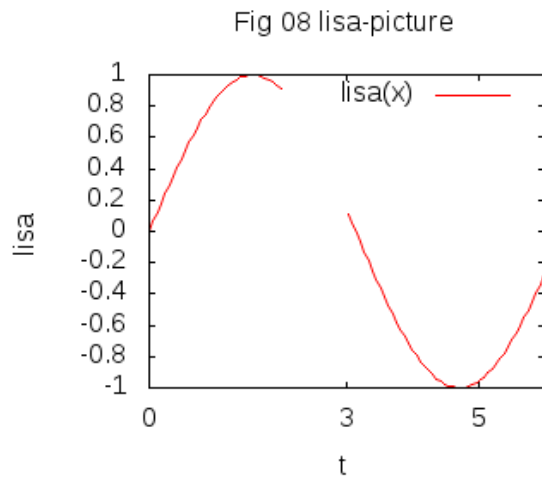
```
$ convert lisa.eps lisa.png
```

### 3.3 标度

这个是由 xtics 和 ytics 来控制的。你不喜欢默认生成的，你可以自己来定

```
Gnuplot> set xtics (0,3,5)
Gnuplot> replot
```

(Fig 08)

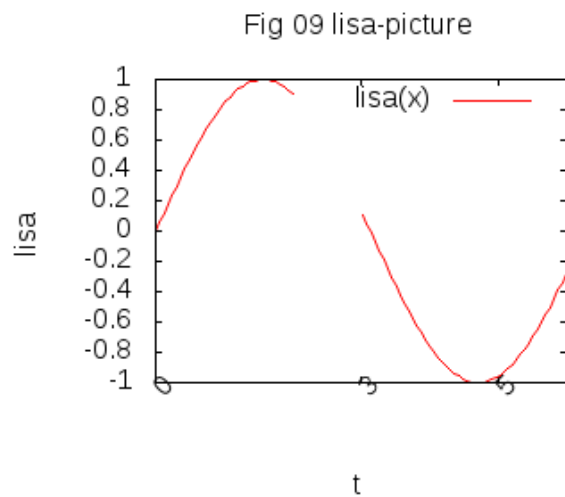


或者

```
Gnuplot> set xtics 0,2,6
Gnuplot> replot
```

就是 0 到 6，步长为 2。 查看一下 `help xtics`，你会发现 `xtics` 是可以旋转的。

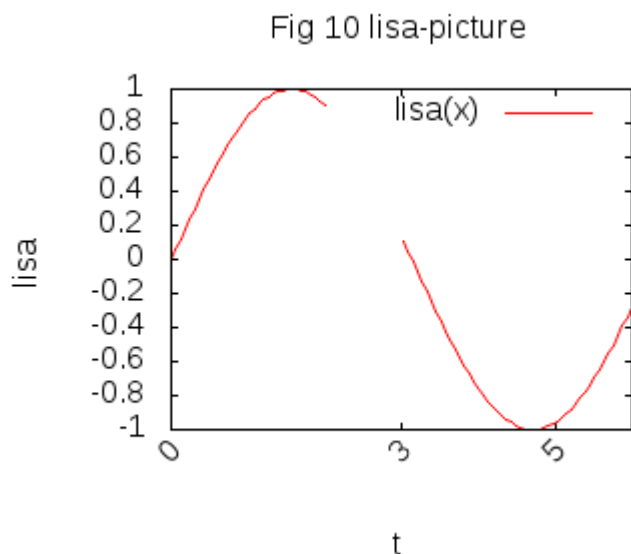
```
Gnuplot> set xtics rotate by 45
Gnuplot> replot (Fig 09)
```



可是这时候标度跑到轴上去了，我们在 `offset` 一下。这个帮助里也有。

```
Gnuplot> set xtics rotate by 45 offset -0.5, -0.5
Gnuplot> replot
```

(Fig 10)



当然其他的对象也是有 offset 的。

有的时候，需要 `xtics` 显示成科学计数法。设置这个东东有点儿像 C 语言

```
gnuplot> set format x "0.1e"
```

也可是同时设置 x 轴和 y 轴

```
gnuplot> set format xy "0.1e"
```

### 3.3 画图样式

下面就是用“线”画图，还是“点”画图的问题。很简单

```
gnuplot> plot lisa(x) with lines
gnuplot> plot lisa(x) with points
```

分别是用“线”和用“点”画图。还有其他的模式。比如 `dots`，这些东西叫做 `style`。你可以使用 `help style` 找到他们。我们就集中在 `lines` 和 `points`

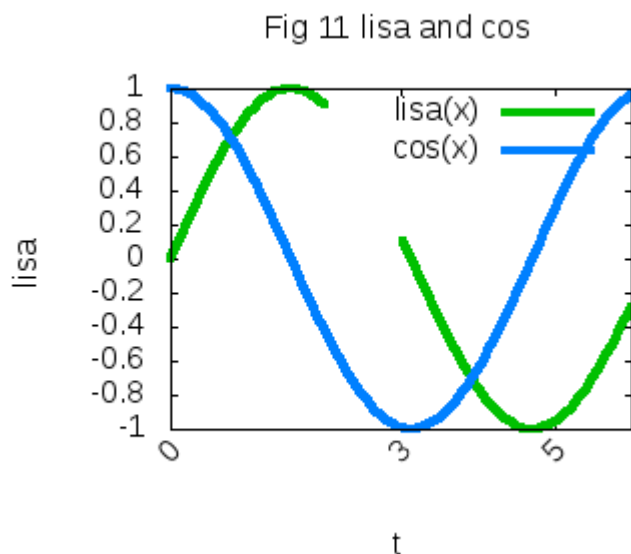
`gnuplot` 中可以使用简称，这个很方便，`with` 简称 `w`，`lines` 简称 `l`，`point` 简称 `p`

```
gnuplot> plot lisa(x) w p (用点画图)
```

还可以设置线的样式(`linetype`，简称 `lt`)，宽度(`linewidth`，简称 `lw`)，颜色：  
(`linecolor`，简称 `lc`)

```
gnuplot> plot lisa(x) w l lt 2 lw 4, cos(x) w l lt 2 lw 4
```

(Fig 11)



使用点画图几乎是一样的，就是指定 `pointtype(pt)`, `pointsizes(ps)`

### 3.4 key

key，这个东西中文的名字是图例吧？如果你不知道什么 key。输入

```
gnuplot> set key off
gnuplot> replot
```

或者

```
gnuplot> unset key
gnuplot> replot
```

什么东西没了，什么东西就是 key。然后输入命令

```
gnuplot> set key on
gnuplot> replot
```

或者

```
gnuplot> set key
gnuplot> replot
```

让它显示出来。关于 key 我想最关心的就是 key 的位置和 key 的字符的大小。审稿的编辑很在乎字符的大小。还是 `help key` 查看帮助：

```
gnuplot> help key
```

这样你就找到了 font，其实和设置 xlabel 一样。

改变位置有两种做法，都在 `help key` 的下面的帮助里：

```
set key at 1, 5
set key left
```

不止可以 left 哦，还有 right center bottom top。

还有一个问题，就是这个自动生成的 key 可能不是你想要的。那么就需要禁

止它自动生成 key

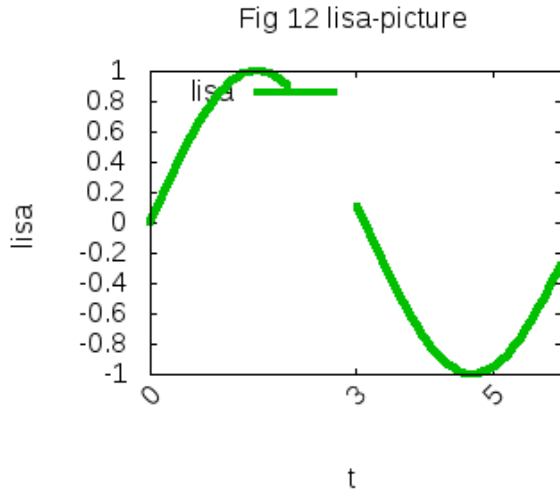
```
gnuplot> set key noautotitle
```

重新设置的方法是在 plot 里加入命令, 如

```
gnuplot> plot lisa(x) title "lisa"
```

这样 key 里是 lisa 了.

(Fig 12)



这个 title 不是图片的 title。图片的 title 已经在上面提到过。你可以在 `help key` 中看到很多其他的设置, 其实都是一个模式的设置方法 而且帮助信息里还有好多例子。自己研究一下吧。

### 3.5 边界

有时候那个 xlabel 跑到了图片的外面去了, 因为它太大了 (一般出现在 eps 图片中)。这时你需要设置一下边界。一共有上、下、左、右四个边界: `tmargin`, `bmarg`, `lmargin`, `rmargin`。我们设置下边界 `bmarg`, 看看怎么设置

```
gnuplot> help margin
```

(`help bmarg` 没什么有用的信息), `help margin` 给出了所有的 margin 的设置方法:

Syntax:

```
set bmarg {{at screen} <margin>}
```

...

我们这样设置

```
gnuplot> set bmarg at screen 0.2
```

```
gnuplot> replot
```

基本上搞定了, 这个 0.2 值的是 0.2 倍的图片高度, 同样的方法可以设置 `tmargin`, `lmargin`, `rmargin`。

## 4 脚本文件

下面的问题就是脚本文件，在命令行里输入这么多行，从编辑的角度上来看是不方便的，可以把所有的这些命令写在一个文件中。Linux 不强调后缀名，所以可以随便取名字啦

比如我们建立一个文件 `sin.gpt`，内容如下

```
# test gnuplot
set term post eps color
set output "sin.eps"
set xlabel "x"
set ylabel "y"
set key noautotitle
plot [0:3.14] sin(x) title "zhang" w l ls 1 lw 5
```

在终端运行

```
$ gnuplot sin.gpt
```

就搞定了。（上面的东西可能有错，我可能敲错字母）

可能只这样还是觉得麻烦，要不停地输入 `gnuplot sin.gpt`，还要到图形界面下去点击图片。你也可以使用一个 `bash` 脚本来搞定：比如文件 `plot.sh`

```
#!/bin/bash
# just for gnuplot
gnuplot << EOF
set term post eps color
set output "sin.eps"
set xlabel "x"
set ylabel "y"
set key noautotitle
plot [0:3.14] sin(x) title "zhang" w l ls 1 lw 5
EOF

evince sin.eps

# end of file plot.sh
```

其中 `evince` 是 Gnome 的图片查看工具。不要忘了改可执行属性

```
$ chmod a+x plot.sh
```

然后：

```
$ ./plot.sh
```

这样就开始画图了，简单多了。

## 5 数据绘图

数据画图可能是最有用的，比如我们有一个数据文件：`data.dat`，内容如下

```
1 1
2 4
```

```
3  9
4  12
...
```

可以这样直接画图

```
gnuplot> plot "data.dat" w l ls 1 lw 5
```

第一列就是 x 轴，第二列是 y 轴。当然也可以反着来，第二列做 x 轴，第一列是 y 轴。

```
gnuplot> plot "data.dat" using 2:1
```

如果数据有很多列，比如 data2.dat 文件是

```
0  0  1
1  1  1
2  4  0.6
.....
```

同样的可以用第二列和第三列画图

```
gnuplot> plot "data2.dat" using 2:3
```

还可以只画出某一个区间的图

```
gnuplot> set xrange [2:4]
gnuplot> replot
```

这样 x 轴就只取 2 到 4 范围内的值。

还有就是你可能需要拟合一下：这个命令是 fit，例如

```
gnuplot> zhang(x)=a*x**2 + b*x + c
gnuplot> fit zhang(x) "data.dat" via a, b, c
```

(\*\*是乘方)

Gnuplot 会生成一个 fit.log 文件。里面有 a, b, c 的拟合结果。马上就可画图了

```
gnuplot "data.dat", f(x)
```

看看拟合的是不是很好呢？

画三维图的命令式 splot，和 plot 几乎是一样的。不过他不同于 matlab，splot 你需要提供三列数据。就像 origin 那样。你可以 help splot。此内容以后补齐。

这里只是一个小小的总结。我相信很适合初学者：更多东西就靠你不停地发现了。

---

ps:

```
set origin 0,0 #设置零点的位置为 0, 0
```

unset 命令可以把对象弄没了

```
unset border
```

```
unset xtics
```

```
unset key
```

可以自己弄着看