

# Gnuplot入门

王勇

version 1.4

## 1 Gnuplot简介

Gnuplot是一个基于命令行的免费作图软件，支持多种平台。Gnuplot最初是为了方便学者与学生绘制数学函数和数据；但现在已能够支持多种用途，例如用作网页脚本，以及作为Octave等第三方软件的绘图引擎。Gnuplot支持二维三维图形的绘制，能够绘制如点，直线，几何体，等高线，向量场，曲面等图形，也能在图形上书写说明文字。Gnuplot的输出也多种多样，支持交互图形界面，多种文件格式以及多种打印机。

本gnuplot入门，将通过实例介绍gnuplot最基本的操作与应用。

## 2 Gnuplot基本作图

Gnuplot支持两种模式：交互模式（interactive mode）和脚本模式（batch mode）。如果需要对图形进行各种调节，则可以使用交互模式，通过命令行实时观察图形的变化。在终端中输入gnuplot可开启交互模式。而在脚本模式中，可以将需要使用的命令存在一个文件中，通过调用“gnuplot 文件名”来运行。

Gnuplot中作图的命令主要是plot（二维）和splot（三维），二者有很多共同点。作图命令的语法是

```
plot {<ranges>}
    {<function> | {"<datafile>" {datafile-modifiers}} }
    {axes <axes>} {<title-spec>} {with <style>}
    {, {definitions,} <function> ...}
```

下面将通过例子来讲解gnuplot的基本作图。

### 2.1 函数曲线

Gnuplot绘制函数曲线非常简单，开启gnuplot交互模式后，输入

```
plot (exp(-(x-4)**2/4)+exp(-(x+4)**2/4))*sin(x)
```

即可把函数 $(e^{-(x-4)^2/4} + e^{-(x+4)^2/4}) \times \sin(x)$ 绘制出来，如图1所示。在本例中，exp(...) 是指数函数 $e^{\dots}$ ，(x-4)\*\*2表示 $(x-4)^2$ 。在gnuplot中，加减乘除由+ - \*/表示；幂由\*\*表示， $x**a = x^a$ 。

Gnuplot支持的数学函数有

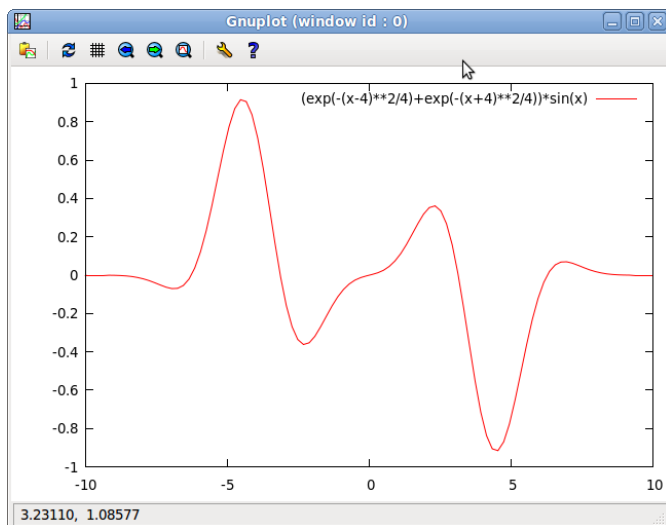


Figure 1: 绘制函数曲线

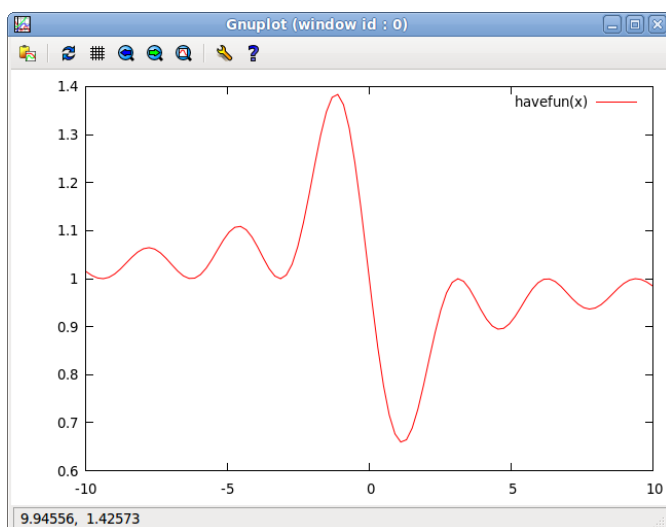


Figure 2: 绘制自定义函数

|              |           |             |          |            |          |             |
|--------------|-----------|-------------|----------|------------|----------|-------------|
| abs(x)       | acos(x)   | acosh(x)    | arg(x)   | asin(x)    | asinh(x) | atan(x)     |
| atan2(y,x)   | atanh(x)  | besj0(x)    | besj1(x) | besy0(x)   | besy1(x) | ceil(x)     |
| cos(x)       | cosh(x)   | erf(x)      | erfc(x)  | exp(x)     | floor(x) | gamma(x)    |
| ibeta(p,q,x) | inverf(x) | igamma(a,x) | imag(x)  | invnorm(x) | int(x)   | lambertw(x) |
| lgamma(x)    | log(x)    | log10(x)    | norm(x)  | rand(x)    | real(x)  | sgn(x)      |
| sin(x)       | sinh(x)   | sqrt(x)     | tan(x)   | tanh(x)    |          |             |

当然，gnuplot也允许用户自定义函数，例如，自定义函数 $havefun(x) = \cos\left(\frac{\sin(x)}{\sqrt{x}}\right)$ 的图像也可以通过plot画出（图2）。

```

havefun(x)=cos(sin(x)/sqrt(x))
plot havefun(x)

```

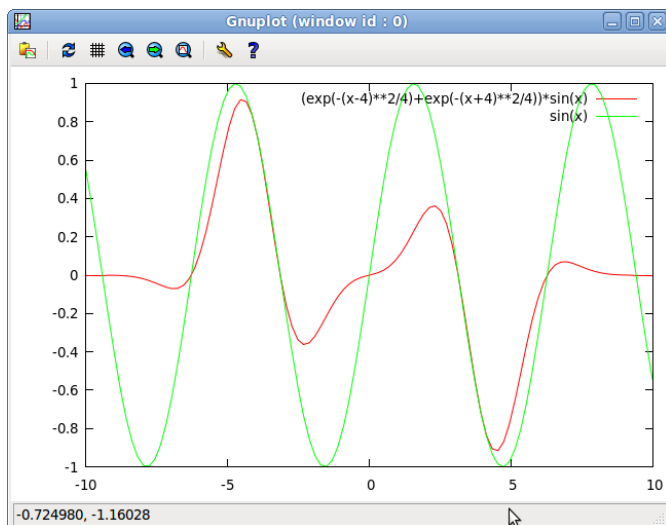


Figure 3: 多个函数的绘制

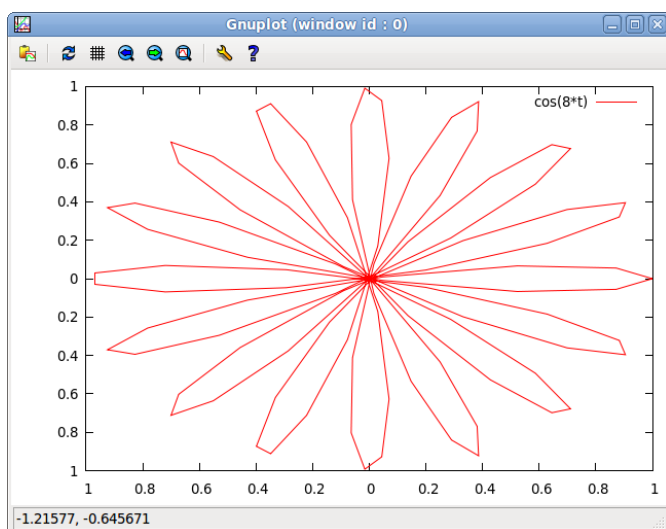


Figure 4: 极坐标曲线

Plot命令也可以同时画多个图，各个函数（数据文件）之间用逗号隔开。有时为了美观，人们把两个函数（或数据文件）分别写在不同行，这时则需要用“\”来表明这些行属于同一命令。例如，（图）

```
plot (exp(-(x-4)**2/4)+exp(-(x+4)**2/4))*sin(x), \
sin(x)
```

另外，gnuplot也可以绘制极坐标曲线。以玫瑰图 $r = \cos(n\theta)$ 为例，取 $n = 8$ （图4）。注意，在普通的函数绘制中，变量是 $x, y$ ；但是极坐标中，变量是 $t$ 。

```
set polar
plot cos(8*t)
```

对于二元函数，可以通过splot绘制曲面。splot用法与plot基本一致。例如，通过splot绘制二元函数 $x^2 + y^2$ 如图5所示。

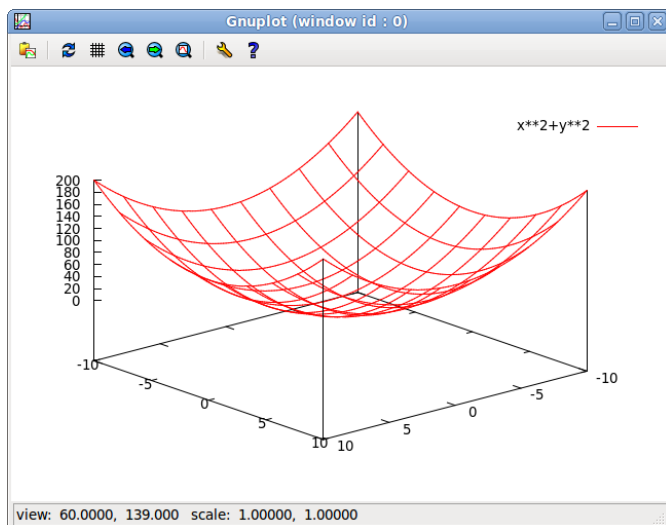


Figure 5: 二元函数曲面图

```
plot x**2+y**2
```

## 2.2 数据图像

将数据可视化是gnuplot的核心，也是最常用的功能。首先需要说明一下文件格式：最基本的文件包含两列： $x$ 和 $y$ ，中间由空格或tab分开。Gnuplot默认以#开头的行为注释行，注释行的数据将不会在图上画出。为了示例的完整性，在本例中，我们增加一列：误差。请将下面的数据保存在文件amp.dat中。该数据描述了阻尼谐振的振幅。

| #Time(s) | Amplitude(cm) | Error-Amplitude(cm) |
|----------|---------------|---------------------|
| 0.0000   | 5.5000        | 0.9322              |
| 0.2500   | 3.5781        | 0.3983              |
| 0.5000   | 0.0000        | 0.2844              |
| 0.7500   | -3.0288       | 0.2348              |
| 1.0000   | -3.9409       | 0.6686              |
| 1.2500   | -2.5639       | 0.3601              |
| 1.5000   | 0.0000        | 0.2369              |
| 1.7500   | 2.1702        | 0.0262              |
| 2.0000   | 2.8238        | 0.3629              |
| 2.2500   | 1.8371        | 0.5087              |
| 2.5000   | 0.0000        | 0.4358              |
| 2.7500   | -1.5550       | 0.3669              |

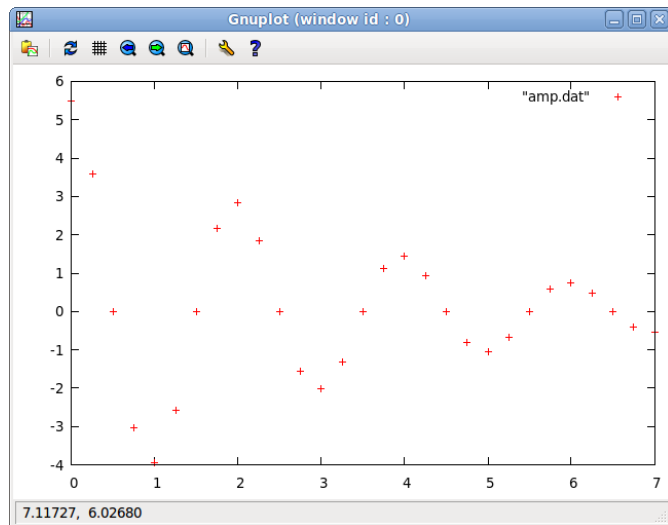


Figure 6: 数据作图

|        |         |        |
|--------|---------|--------|
| 3.0000 | -2.0233 | 0.2310 |
| 3.2500 | -1.3163 | 0.3458 |
| 3.5000 | 0.0000  | 0.2075 |
| 3.7500 | 1.1142  | 0.2695 |
| 4.0000 | 1.4498  | 0.0653 |
| 4.2500 | 0.9432  | 0.0927 |
| 4.5000 | 0.0000  | 0.0888 |
| 4.7500 | -0.7984 | 0.0975 |
| 5.0000 | -1.0388 | 0.1573 |
| 5.2500 | -0.6758 | 0.0735 |
| 5.5000 | 0.0000  | 0.1147 |
| 5.7500 | 0.5721  | 0.0316 |
| 6.0000 | 0.7443  | 0.0349 |
| 6.2500 | 0.4843  | 0.0511 |
| 6.5000 | 0.0000  | 0.0717 |
| 6.7500 | -0.4099 | 0.0164 |
| 7.0000 | -0.5333 | 0.0376 |

在gnuplot中输入

```
plot "amp.dat"
```

即可以将数据绘成图像，如图6所示。

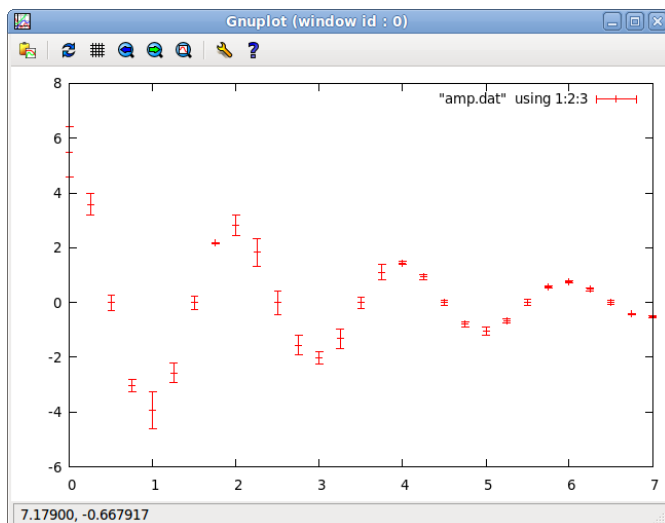


Figure 7: 误差棒

Gnuplot默认绘出数据文件的前两列。为了将误差也同时绘出，我们需要使用文件修饰字using和with。修饰字using就是“使用”的意思。我们需要同时使用1列、2列和3列，则可以在plot “amp.dat”后面添加using 1:2:3。With主要用来调整绘图格式，例如可以用with yerrorbars来告诉gnuplot我们要画y的误差棒。完整的命令及其输出（图）如下所示：

```
plot "amp.dat" using 1:2:3 with yerrorbars
```

## 2.3 数据拟合

数据拟合是在学习工作中经常需要用到的。Gnuplot提供了fit命令来实现拟合；fit命令与plot也存在着很多共同点。

```
fit {[xrange] {[yrange]}} <function> '<datafile>',
    {datafile-modifiers}
    via '<parameter file>' | <var1>{,<var2>,...}
```

在拟合之前，需要先对要拟合的参数赋值（一般也先定义拟合函数）。以阻尼谐振为例，拟合函数是

$$A(x) = A_0 e^{-x/\tau} \times \cos(\omega x + \phi)$$

参数是 $A_0, \tau, \omega, \phi$ 。

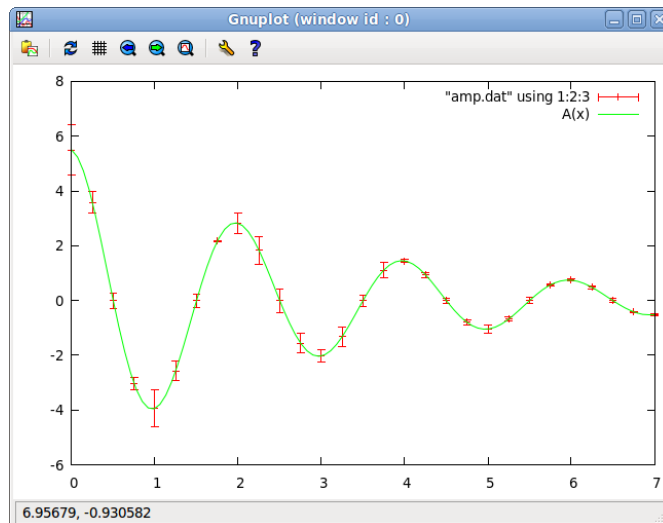


Figure 8: 参数拟合

```
A0=1
tau=1
omega=1
phi=0
A(x)=A0*exp(-x/tau)*cos(omega*x+phi)
fit A(x) "amp.dat" using 1:2 via A0, tau, omega, phi
```

Gnuplot使用ML进行拟合，并会把中间过程及结果打印在屏幕上，而且保存在fit.log 文件中以备后用。拟合后拟合函数的参数自动赋值。可以将拟合函数和数据画在一起，进行比较，如图所示。

```
plot "amp.dat" using 1:2:3 with yerrorbars, A(x)
```

## 2.4 图像的保存

Gnuplot提供多种文件格式以保存绘制出的图像，如eps, png, jpeg, gif等。用户可以通过设置终端及文件名来保存绘制的图像，其中XXXX是终端类型，yyy是文件后缀名（文件格式）。

```
set terminal XXXX
set output "filename.yyy"
replot
```

几种基本的终端类型与文件格式的对应关系如下表所示：

| 终端类型                | 文件格式 |
|---------------------|------|
| postscript          | ps   |
| postscript enhanced | eps  |
| jpeg                | jpeg |
| gif                 | gif  |
| png                 | png  |

## 2.5 脚本的保存与导入

Gnuplot提供了保存和导入脚本的功能，这在命令比较多或需要重复使用时非常有用。保存命令的语法是

```
save {<option>} '<filename>'
```

注意，文件名必须用引号（双引号或者单引号）引起来。其中选项（option）可以省略。省略选项时，所有的函数，变量，定制（set，将在下节讲解）以及最后的plot命令将会保存在文本文件中。该文本文件可以通过load命令导入。

```
load "<filename>"
```

## 3 Gnuplot中图形的美化

Gnuplot提供了强大的功能，使用户可以几乎随心所欲地修改并美化绘制的图形。主要使用的命令是set。这里值得注意的是，使用set命令不会立即改变现有图形，而需要使用plot，splot或replot将所做的修改可视化。美化图形还可以使用plot/splot中的with选项。下面通过实例来讲解。

### 3.1 坐标轴

在大多数情况下，作图时需要标定坐标轴以及横纵坐标的范围。这些可以通过set完成，如下及图9所示。

```
set xlabel 'This is x axis' #标定横坐标
set ylabel 'This is YYYYY' #标定纵坐标
set xrange [10:20] #设定横轴范围
set yrange [0.2:2.0] #设定纵轴范围
plot sin(x)+cos(x) #绘图
```

注意，在gnuplot中， $[min : max]$ 表示范围。

### 3.2 标题

图的标题可以通过set title 'this is a title'设定，如图10所示。



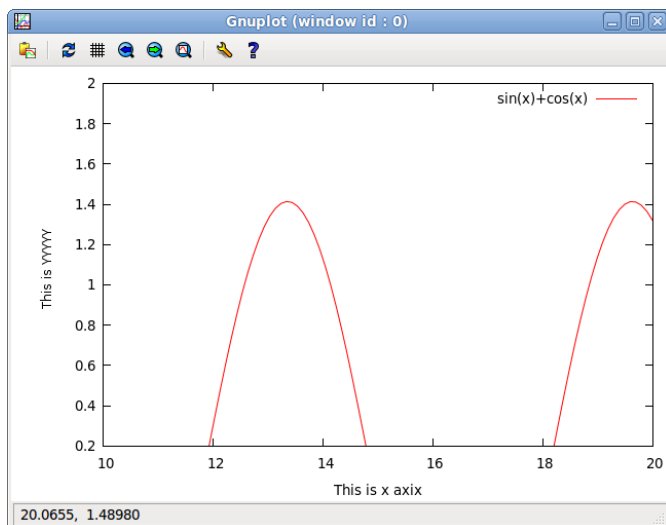


Figure 9: 设定坐标轴

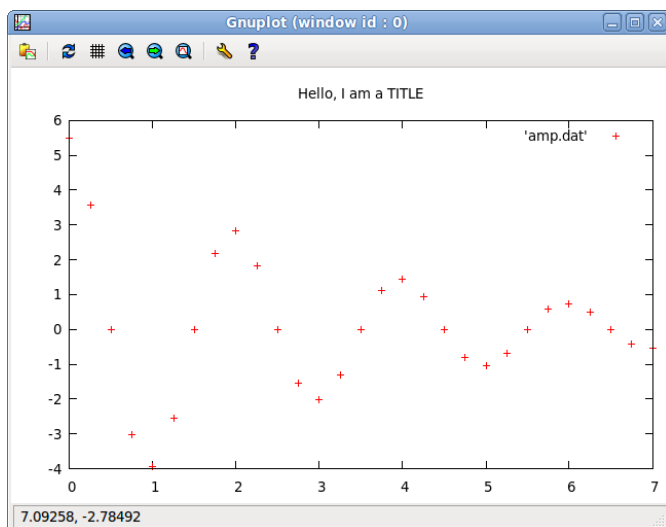


Figure 10: 设定标题

```
set title 'Hello, I am a TITLE'
plot 'amp.dat'
```

### 3.3 图例

图例 (legend) 在gnuplot中其实叫做title，不过它与图的标题是不同的。图例可以在plot命令中使用title选项。如果不需要图例，可以使用notitle选项。效果如图11所示。

```
set multiplot layout 1,2 #绘制多图
plot 'amp.dat' title 'Experimental Data'
plot 'amp.dat' notitle
```

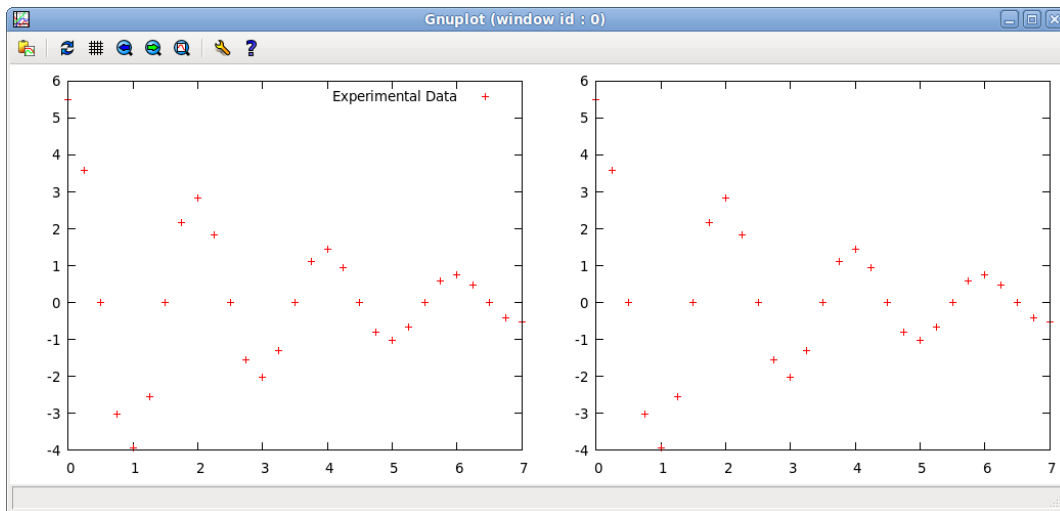


Figure 11: 显示与隐藏图例

### 3.4 绘制类型（点，线，盒子，阶跃，脉冲等）

在gnuplot的二维图形绘制中，能够使用点（points）、线（lines）、盒子（boxes）、阶跃（steps）、脉冲（impulses）等；通过with选项使用。同时，可以通过给定数值设定点的类型（pointtype）和大小（pointsize）以及线的类型（linetype），颜色（linecolor）和宽度（linewidth）。例如（图12），

```
set multiplot layout 3,2

plot 'amp.dat' with points pointtype 6 pointsize 2
plot 'amp.dat' with lines linetype 3 linecolor 2 linewidth 2
plot 'amp.dat' with boxes
plot 'amp.dat' with steps
plot 'amp.dat' with impulses
plot 'amp.dat' with filledcurves
```

值得注意的是，点和线的类型、颜色是与选取的终端（terminal）有关的，不同的终端有不同的显示效果。

### 3.5 光滑函数曲线

绘制函数曲线时，有时会因为取点太少以致所得到的曲线不够光滑（如图4）；此时可通过set samples number（默认取点数是100）来光滑曲线。如图13所示。

```
set samples 10000

set polar

plot cos(8*t)
```

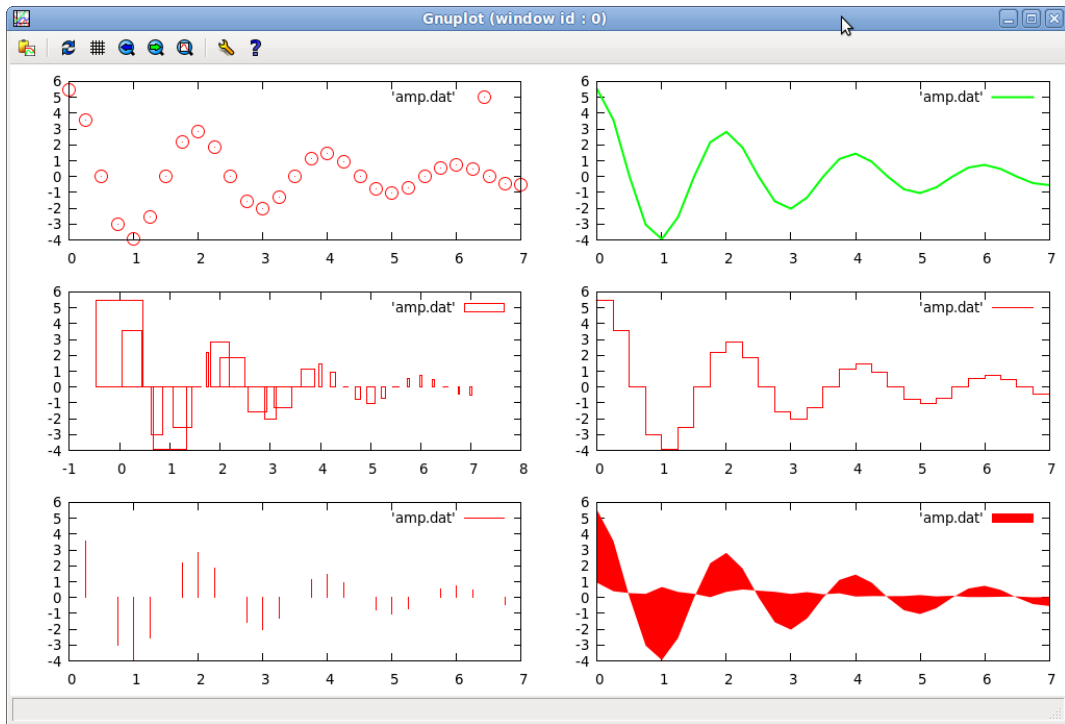


Figure 12: 设定类型

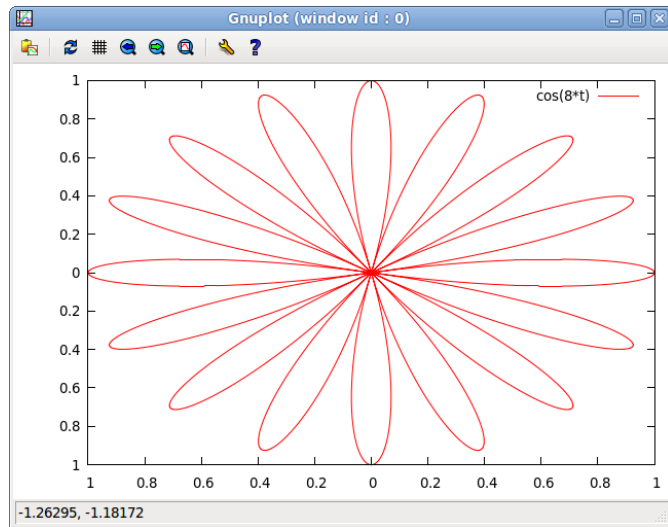


Figure 13: 光滑曲线

## 4 更多帮助

1. 官方文档 : <http://www.gnuplot.info/documentation.html>
2. FAQ : <http://www.gnuplot.info/faq/index.html>
3. Not So FAQ : <http://t16web.lanl.gov/Kawano/gnuplot/index-e.html>