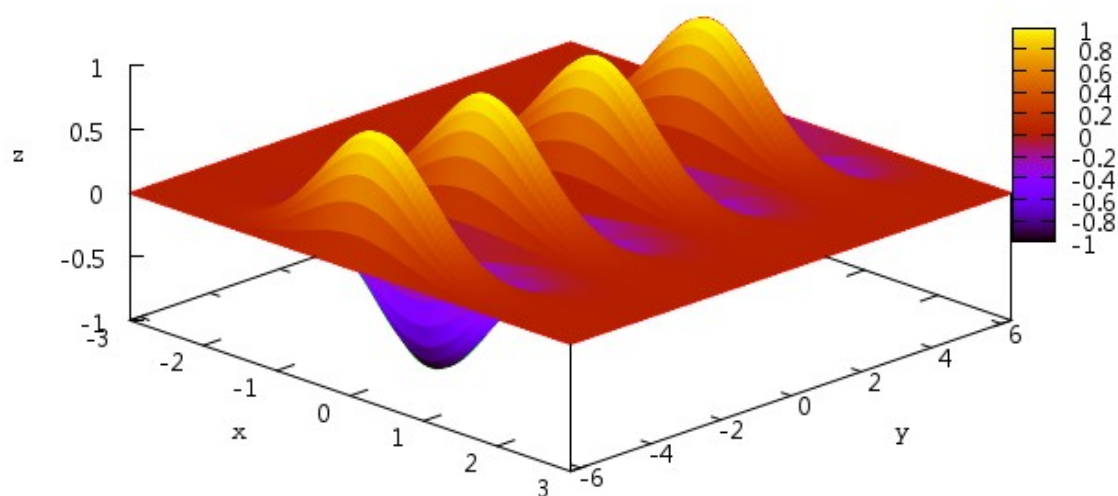


# Gnuplot

## ——科学绘图与数据可视化



数声风笛离亭晚，我想潇湘君想秦

## 说明

此文档根据个人学习使用 gnuplot 的经验写成。现公开之，希望对正在学习使用 gnuplot 者略有帮助。由于作者也是 gnuplot 的学习者，文中难免会有错误之处，欢迎指正 (qinjieli@126.com 或者 qinjieli@gmail.com)。如需转载此文，请务必确保文档的完整性，并注明“数声风笛离亭晚，我想潇湘君想秦”，且提供该说明。如需部分引用只需注明“数声风笛离亭晚，我想潇湘君想秦”。

## gnuplot 科学绘图与数据可视化

首先指出 gnuplot 是一个源代码受版权保护但是“自由”的软件（gnuplot 的主页上是这样说的“The source code is copyrighted but freely distributed (i.e., you don't have to pay for it).”）。该软件支持 linux, OS/2, MS Windows, OSX, VMS 等多种平台。软件主页 <http://www.gnuplot.info/> 提供软件的下载链接。该软件被广泛应用于科学绘图与数据的可视化。

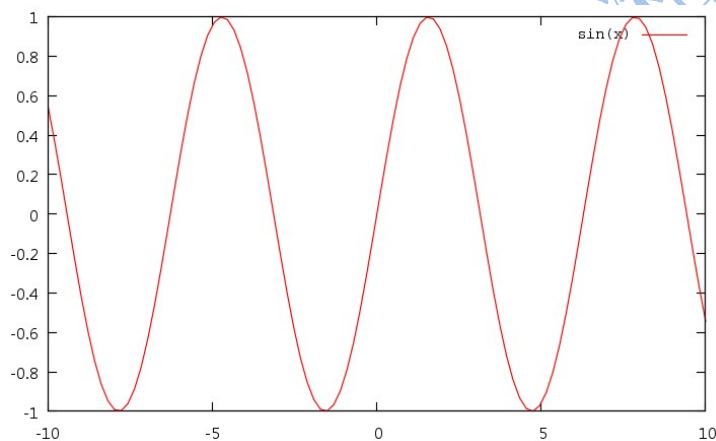
闲话少说，现在开始具体介绍该软件的使用。

### 一、简单的二维作图

Gnuplot 的作图是通过命令行实现的。运行该软件后，输入命令：

**plot sin(x)**

即可得到函数  $\sin(x)$  的图像。如图一所示。



图一 简单二维作图

这就是我们用 gnuplot 得到的第一张图片了，虽然粗糙了些，但是还算过得去，毕竟我们还没有做任何设置。

这里指出几点：

1. `plot` 是绘制二维曲线图像的命令。
2. 可以使用 `help plot` 获取关于 `plot` 帮助。（关于其他命令的帮助统一可以通过这种方式获得。）
3. `x` 是在笛卡尔坐标系下绘制二维曲线时的默认变量，默认取值范围为  $-10 \sim 10$ 。如果你用 `plot sin(y)` 来画正弦函数的图像那么会提示“undefined variable: y”。
4. 当弹出的窗口显示上述图像后，我们一定会习惯性地找“save”来保存图像了。但是不幸的是我们根本就没有发现一个菜单栏，更不用说“save”了。难道 gnuplot 不能保存文件？非也，如同我们开头讲的 gnuplot 是一个命令行式的软件，保存文件也需要使用命令的。事实上 gnuplot 的文件操作功能非常强大，可以将图像存储为格式众多的图像文件（jpeg、jif、ps、eps、pdf、latex、fig……）。具体如何保存下面介绍。

### 二、保存文件

Gnuplot 的默认输出终端是屏幕。我们只要改变默认的输出终端就可以将图像输出到文

件。

输入如下命令：

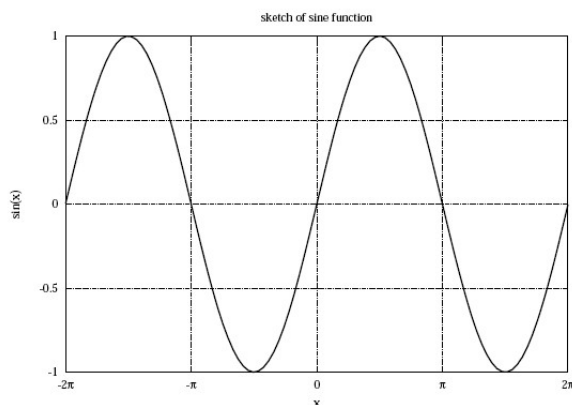
```
set term gif
set output "sin.gif"
plot sin(x)
```

这时候在当前目录（查看当前目录的方法是：命令 `pwd`）下面就可以找到一个文件名为“sin.gif”的文件，该文件就是我们所要的图像文件了。

下面指出这里的几个注意点：

1. 文件的输出位置也是可以改变的。例如，命令“`cd 'E:'`”可以将当前目录改为“E:”，这时文件就会输出到 E 盘根目录。
2. 事实上在科学作图中 gif 的图片是较少使用的，一般 eps 格式图片更常用（`latex` 与 `epslatex` 也比较常用）。所以可以设置终端为 `postscript`，即“`set term postscript`”，这时文件的扩展名用 `eps`。该软件还提供多种形式的输出终端，具体可以使用命令“`set term`”来查看。

### 三、较精细的二维作图



图三 较精细的二维作图

前面得到的图像实在是太粗糙了，粗糙到连 x 轴和 y 轴都没有进行任何标注。下面介绍如何将图像做精细一些。

1. 设置输出终端：

```
set term postscript enhanced font 'Time-roman, 22' #只有加上 enhanced 选项才能显示特殊符号，比如图三中的“π”
```

```
set output 'sin.eps'
```

2. 设置图例

```
set key off #设置不显示右上角的图例
```

3. 设置坐标轴

```
set xrange [-2*pi:2*pi] #设置 x 轴范围
```

```
set xtics ('-2{/Symbol p}' -2*pi, '-{/Symbol p}' -pi, '0' 0, '{/Symbol p}' pi, '2{/Symbol p}' 2*pi) #设置 x 轴坐标
```

```
set ytics -1, 0.5, 1 #设置 y 轴坐标
```

```
set xlabel 'x' #设置 x 轴标注
```

```
set ylabel 'sin(x)' #设置 y 轴标注
```

4. 设置网格

```
set grid xtics #设置 x 轴网格
set grid ytics #设置 y 轴网格
```

5. 设置标题

```
set title 'sketch of sine function' #设置图像标题
```

6. 画图

```
plot sin(x) linewidth 4 #以宽度为 4 的线画图
```

7. 退出 gnuplot

```
eixt #退出 gnuplot
```

图三 即是我们最后得到的图像。

这里我们介绍了二维作图中的一些基本设置，对这些设置更详细的介绍以及一些此处没有介绍的设置可以参考 gnuplot 的帮助文件。

## 四、二维数据可视化

下面是我们的一个数据文件，第一列为 x 坐标，第二列为 y 坐标，文件名为“data.dat”。

```
0.1000000    0.1380506
0.2000000    0.1699537
0.3000000    0.1955449
0.4000000    0.2077560
0.5000000    0.1987661
.....
2.600000    0.1397104
2.700000    0.1503764
2.800000    0.1431071
2.900000    0.1247329
3.000000    0.1012012
```

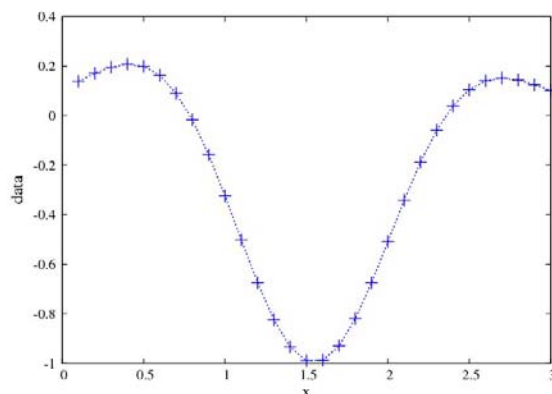
注：该数据文件可以用如下 Fortran 程序产生

```
program main
  open(11,file='data.dat')
  do j=1,30
    ry=0.1*dble(j)
    r=exp(-(ry-1.5)**2)*cos(2*d0*ry)
    write(11,*) ry,r
  end do
end program main
```

将上述文件拷贝到当前目录，然后用下面的命令画图：

```
set xlabel 'x' font 'Time-Roman,22' #设置 x 标注及其字体
set ylabel 'data' font 'Time-Roman,22' #设置 y 轴标注及其字体
set xtics font 'Time-Roman,18' #设置 x 轴坐标字体
set ytics font 'Time-Roman,18' #设置 y 轴坐标字体
set term postscript color enhanced
set output 'data.eps'
set pointsize 2.0
```

`plot 'data.dat' with linespoints linewidth 2.5 linetype 3 notitle #用  
linespoints 画图，线宽为 2.5，线的类型为 3，没有图例`  
我们得到的图像如图四所示。



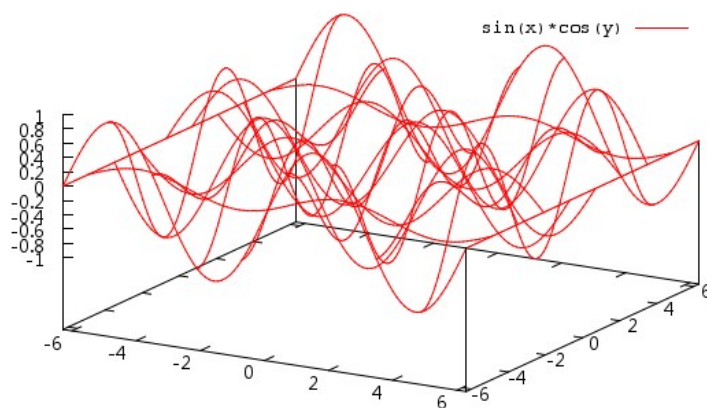
图四 二维数据可视化

## 五、简单三维作图

gnuplot 的三维作图命令是 `splot`。下面我们运用该命令画函数  $z = \sin(x) * \cos(y)$  的图像。

```
set xrange [-2*pi:2*pi] #设置 x,y 坐标范围，注意 gnuplot 可以识别常量 pi  
set yrange [-2*pi:2*pi]  
splot sin(x)*cos(y)
```

上述命令得到的图像如图五（一）所示。



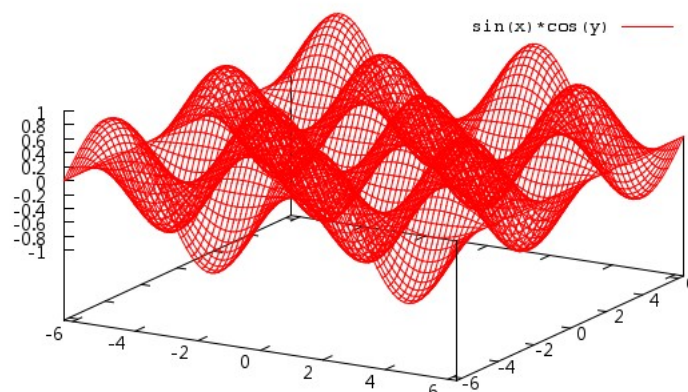
图五（一） 简单三维作图

这个图像实在有些“乱七八糟”了，我们是在无法接受。为什么会这样呢？主要的原因是等值线密度太低了（也就是网格太大了），我们通过下面的命令增加等值线密度，看看会不会有改观。

```
set isosamples 75, 75  
replot
```

此次作图得到图五（二）的图像，该图稍稍有些改观，但是依然很粗糙。我们可以用下面一些命令继续对图像进行更精细的修改。

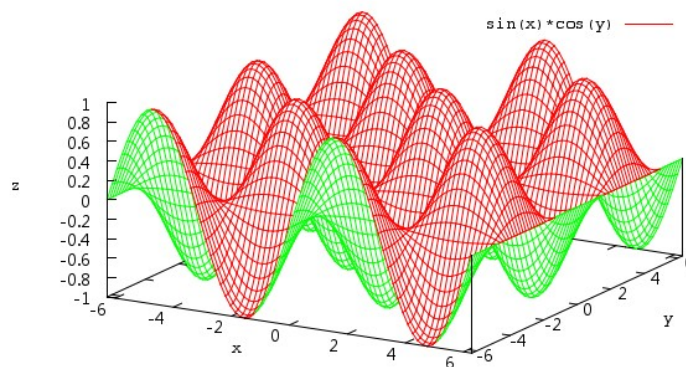




图五（二） 简单三维作图

```
set hidden3d #禁止三维情况下隐藏线的显示
set ticslevel 0 #将 z 轴零点平移到 xy 平面上
set xlabel 'x'
set ylabel 'y'
set zlabel 'z'
replot
```

replot 得到图五（三）的图像。这个三维图就相对漂亮些了。



图五（三） 简单三维作图

最后可以使用命令“set view \*,\*,\*,\*”改变三维图形的观察角度，后面四个参数的意义参照帮助文件。

## 六、等高线图

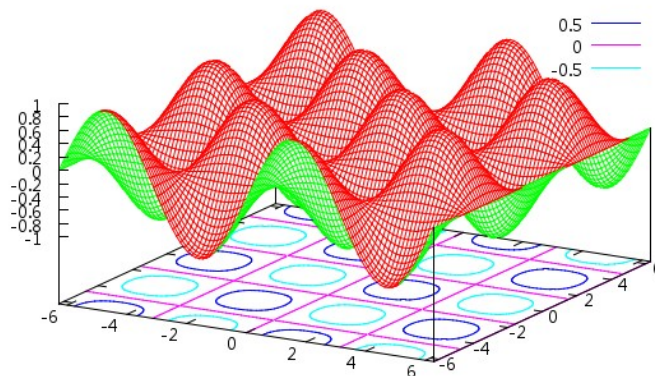
Gnuplot 中没有专门的等高线作图命令，等高线作图是通过 splot 实现的。具体实现方法如下：

仍然以函数  $z = \sin(x) * \cos(y)$  为例说明。

```
set xrange [-2*pi:2*pi]
set yrange [-2*pi:2*pi]
```

```
set isosamples 100,100
set hidden3d
set contour base #设置等高线模式，在底面显示等高线
splot sin(x)*cos(y) notitle
```

这时候得到的图像如图六（一）所示。

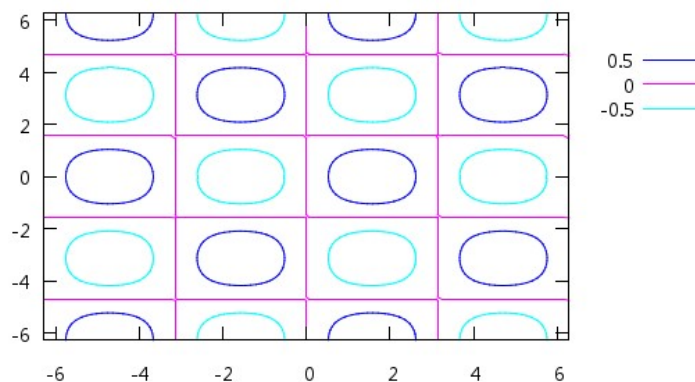


图六（一） 等高线图

此时图像不仅包括了我们要的等高线还包括有三维的曲面图像。如果这就是我们想要的，那么加上坐标轴标注，图像就基本完成了。如果我们只想要等高线，不想要三维的曲面，那么我们下面去掉三维曲面。

```
unset surface #设置不显示三维曲面
set view map #以 map 角度观察图像
set key out #在图像框外面显示图例
replot
```

得到如图六（二）的等高线图。



图六（二） 等高线图

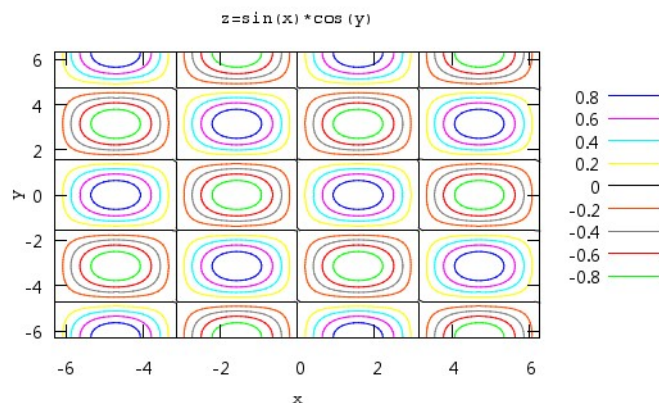
该等高线图只做了三条等高线，不太精细。我们需要对等高线做些设置。

```
set cntrparam levels discrete -0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8 #
设置等高线值
set xlabel 'x'
set ylabel 'y'
set title 'z=sin(x)*cos(y)'
```



## replot

图六（三）为我们最后得到的等高线图。



图六（三） 等高线图

## 七、三维数据可视化

下面是我们的一个数据文件，第一列为 x，第二列为 y，第三列为 z，文件名“data3d.dat”。

```

-1.000000    -1.000000    -3.3546262E-04
-1.000000    -0.900000    -5.9318740E-04
-1.000000    -0.800000    -9.8782987E-04
.....
-1.000000     0.800000    -9.8782987E-04
-1.000000     0.900000    -5.9318740E-04
-1.000000     1.000000    -3.3546262E-04
-0.900000    -1.000000    -8.6740870E-04
-0.900000    -0.900000    -1.5338099E-03
.....
.....
1.000000     0.900000    -5.9318740E-04
1.000000     1.000000    -3.3546262E-04

```

注：该数据文件可以用如下 fortran 程序产生：

```

program main
  open(11,file='data3d.dat')
  do i=1,21
    rx=0.1*dble(i-11)
    do j=1,21
      ry=0.1*dble(j-11)
      r=-exp(-5d0*rx**2-3d0*ry**2)
      write(11,*) rx,ry,r
    end do
  end do
end program main

```

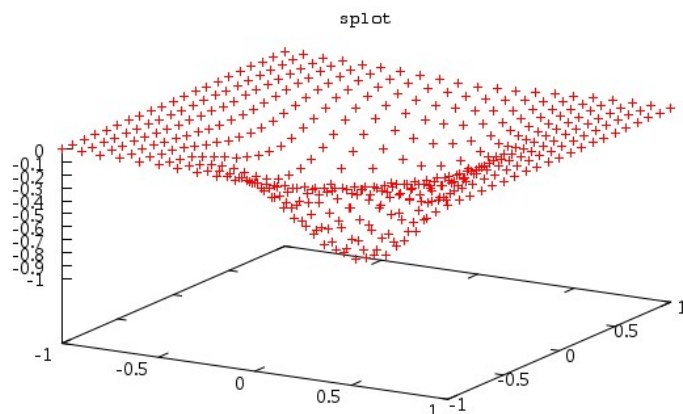
三维的数据可视化与二维相似。

将上述数据文件拷贝到当前目录，分别用下面命令作图。

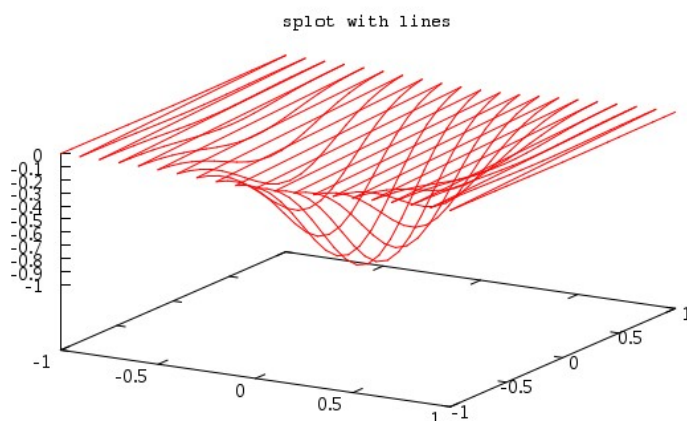
```
splot 'data3d.dat' notitle
```

```
splot 'data3d.dat' with lines notitle
```

结果如图七（一），七（二）所示。



图七（一） 三维数据可视化



图七（二） 三维数据可视化

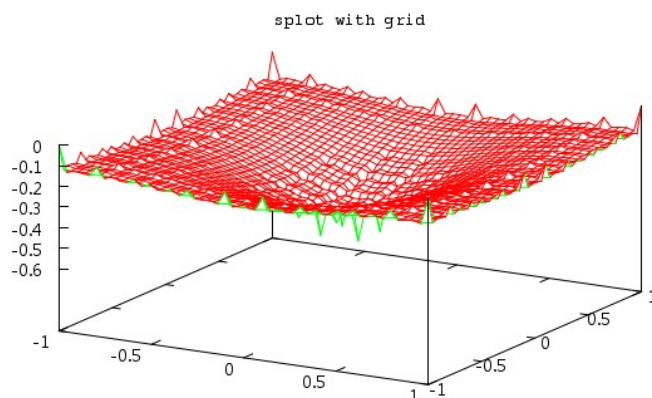
可见默认情况下 splot 画数据点。若用 lines 选项，则 gnuplot 将各个数据点顺次用线连接。

通常我们更喜欢画曲面图像。这可以通过设置网格实现。

```
set dgrid3d 50, 50
```

```
set hidden3d
```

```
replot
```



图七（三） 三维数据可视化

replot 的结果图七（三）所示。

细心观察会发现图像显示的数据与我们实际的数据点是不相同的。例如，我们的数据文件中存在点 (0.0, 0.0, -1.0)，这在图七（一）与图七（二）中也是很明显的，但是图七（三）中  $x=0.0, y=0.0$  时， $z$  的值看起来却大约是 0.6 左右。原因何在？这是因为在作图过程中 gnuplot 在网格内做了平均了。

那么我们如何禁止 gnuplot 做这样的平均呢？方法如下。

```
set dgrid3d splines #样条插值
```

再对图像做一些修饰：

```
set xlabel 'x'
```

```
set ylabel 'y'
```

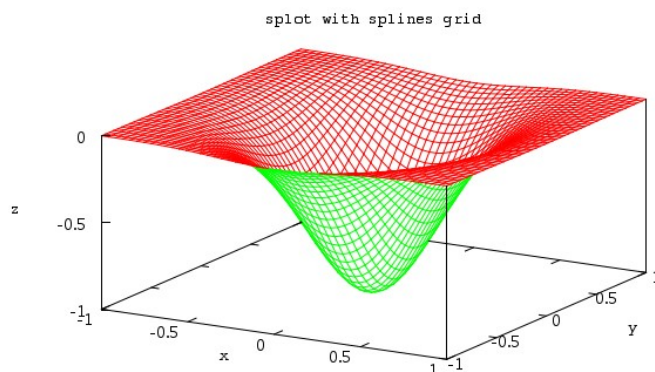
```
set zlabel 'z'
```

```
set title 'splot with splines grid'
```

```
set ticslevel 0
```

```
set ztics -1,0.5,0
```

```
replot
```



图七（四） 三维数据可视化

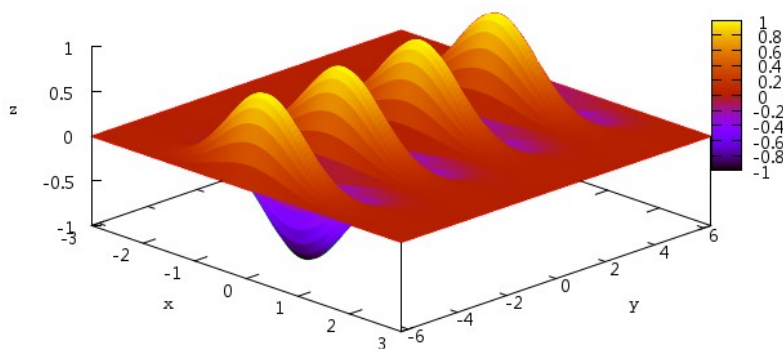
最后我们得到如图七（四）的数据图像。至此我们实现了三维数据的可视化。

## 八、漂亮的 pm3d 图像

pm3d 作图模式下可以画出漂亮的颜色（灰度）曲面及投影图。

```
set pm3d #设置 pm3d 模式
set xrange [-3:3]
set yrange [-2*pi:2*pi]
set isosamples 100,100
set hidden3d
set view 50,45,1,1 #改变视角
set xlabel 'x'
set ylabel 'y'
set zlabel 'z'
set ztics -1,0.5,1.0
splot exp(-x**2)*sin(2*y) notitle
```

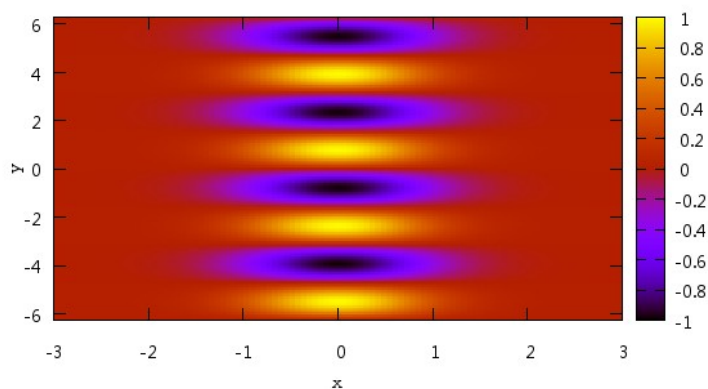
上述命令产生如图八（一）的图像。



图八（一） 漂亮的 pm3d 图像

改变视角就可以得到投影图，如图八（二）。

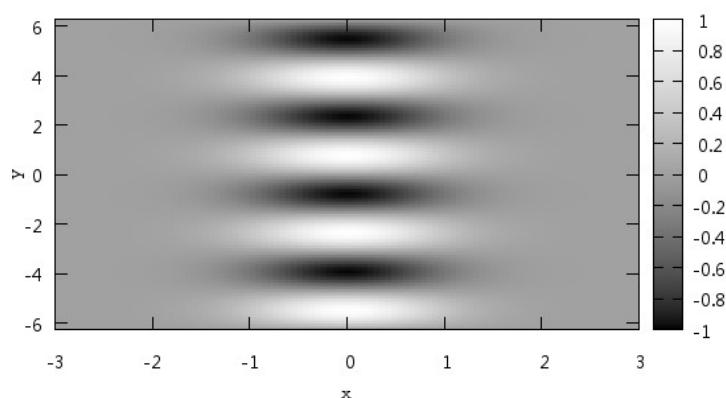
```
set view map
replot
```



图八（二） 漂亮的 pm3d 图像

图八（一）和（二）都使用了默认的颜色轴。Gnuplot 中通过命令“set palette gray”来画灰度图，如图八（三）所示。

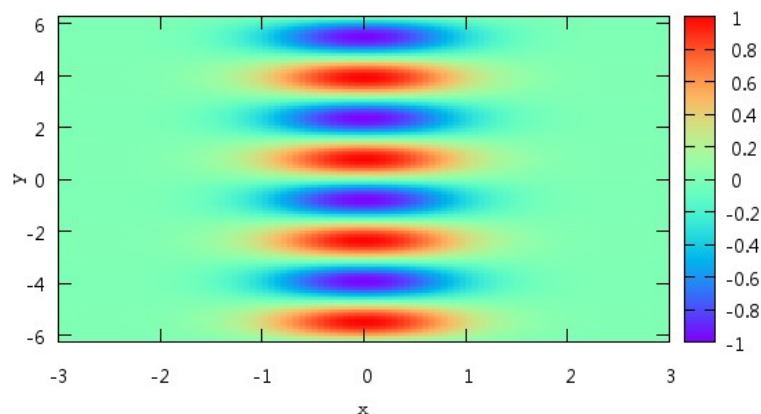
```
set palette gray
replot
```



图八（三） 漂亮的 pm3d 图像

使用彩色模式可以通过 “set palette rgbformulae \*,\*,\*” 改变颜色轴。

```
set palette rgbformulae 33,13,10
replot
```



图八（四） 漂亮的 pm3d 图像

上面的 rgbformulae 对于的颜色轴为“赤橙黄绿青蓝紫”的彩虹样式，如图八（四）所示。其他一些常用的 rgbformulae 可以通过 “help rgbformulae” 获得。

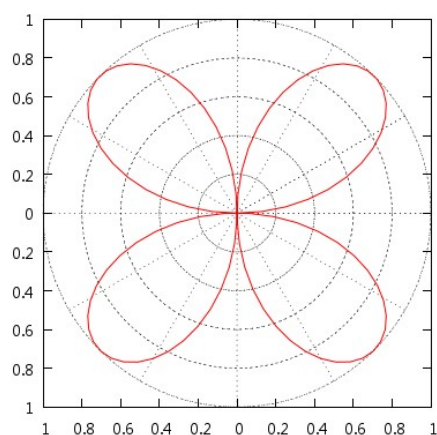
对数据文件同样可以画 pm3d 图像，方法基本相同，此处不再赘述。

## 九、极坐标系作图

gnuplot 支持极坐标作图，极坐标系下的默认变量为  $t$ ，默认取值范围为  $0 \sim 2\pi$ 。

```
set polar #设置极坐标模式
set xrange [-1:1]
set yrange [-1:1]
set grid polar
set size ratio 1 #设置图像宽高比
set xtics 0.2
set ytics 0.2
plot sin(2*t) notitle
```

上面的命令产生如图九（一）图像。



图九（一） 极坐标系作图

这样的极坐标图再加上一定的标注也就可以满足我们的需要了。但是有的时候我们希望将边框去掉，得到“完全的极坐标图像”。这可以通过如下方法实现。

```
set border linewidth 0 #设置边框的线宽为0，即不可见
set xtics textcolor rgbcolor 'white' #设置 x 轴标注的颜色为白色（背景色），即不可见
```

```
set ytics textcolor rgbcolor 'white' #同上
```

```
replot
```

```
set arrow from 0,0 to 1,0 #画一个从(0,0)到(1,0)的箭头
```

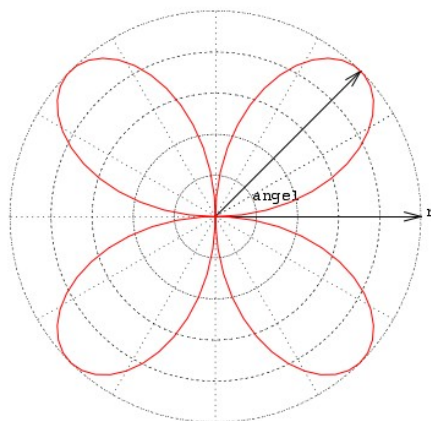
```
set arrow from 0,0 to 0.707,0.707 #画箭头
```

```
set label 'angel' at 0.18,0.1 #在(0.18, 0.1)处加一个标签
```

```
set label 'r' at 1.02,0.02 #加标签
```

```
replot
```

上面的命令去掉了边框与坐标轴的标注（实际并没有去掉，只是看不见了而已），并且加了两个箭头和两个标签，从而得到了一张“完全的极坐标图像”，如图九（二）所示。



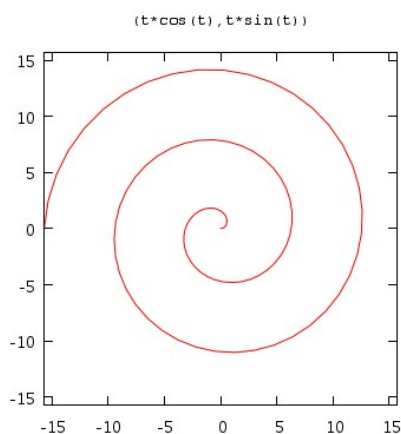
图九（二） 极坐标作图

## 十、参数方程作图

gnuplot 支持参数方程作图，参数方程模式下的默认变量为：曲线作图—— $t$ （默认取值范围 $-5 \sim 5$ ），曲面作图—— $u$  和  $v$ （默认取值范围 $-10 \sim 10$ ）。

首先我们看看参数曲线图。





图十（一）参数方程作图

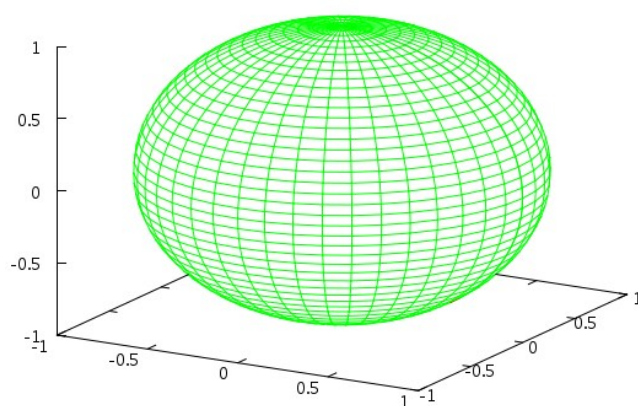
```
set parametric #设置为参数方程作图模式
set trange [0:5*pi] #设置参数范围
set xrange [-5*pi:5*pi] #设置作图范围
set yrange [-5*pi:5*pi]
set size ratio 1 #设置宽高比为 1
set title '(t*cos(t), t*sin(t))' #设置标题
plot t*cos(t), t*sin(t) notitle
```

我们得到如图十（一）所示的图像。

接下来我们看看参数曲面图。

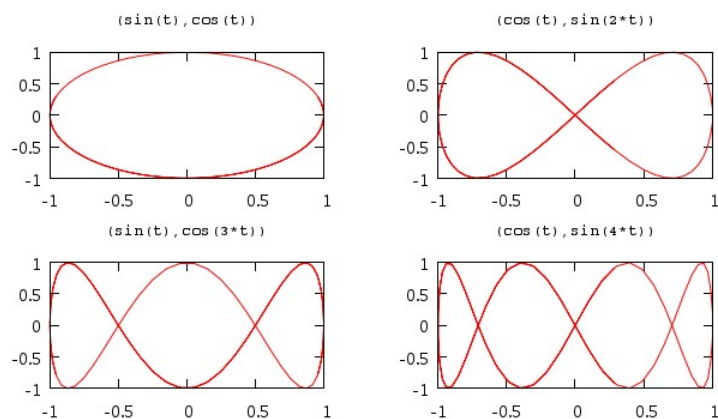
```
set parametric #设置参数模式
set urange [0:2*pi] #设置参数范围
set vrange [0:pi]
set xrange [-1:1] #设置作图范围
set yrange [-1:1]
set zrange [-1:1]
set isosamples 45, 45 #设置网格密度
set hidden3d #设置隐藏线
set ticslevel 0 #平移 z 轴
set view 60, 30, 1, 1.5 #设置视角
splot cos(u)*sin(v), sin(u)*sin(v), cos(v) notitle
```

上面的命令用来作单位球面，如图十（二）所示。



图十（二） 参数方程作图

## 十一、 图组——subplot



图十一 图组

subplot 是 matlab 中一个画一组图时的命令，gnuplot 中是没有该命令的，但是通过一些设置能够实现 subplot 的功能。这一部分我们绘制如图十一所示的一组图像。

下面介绍并分析完成上述作图所用的命令。

`set parametric` #设置参数模式

`set multiplot` #设置多图模式

`set size 0.45, 0.45` #设置图像的大小，我们要同时显示四幅图像，本来应该选宽高均为一半，即 0.5, 0.5，但是考虑到边界的空白我们稍稍设小一些

`unset key` #作图中不显示图例

`set xtics -1, 0.5, 1` #设置坐标轴

`set ytics -1, 0.5, 1`

`set origin 0.05, 0.5` #开始做左上角的图，设置幅图像的原点

`set title '(sin(t), cos(t))'` 设置左上角图的标题

`plot sin(t), cos(t)` #画左上角的图

`set origin 0.55, 0.5` #画右上角的图

`set title '(cos(t), sin(2*t))'`

`plot cos(t), sin(2*t)`

`set origin 0.05, 0.05` #画左下角的图

```

set title '(sin(t),cos(3*t))'
plot sin(t),cos(3*t)
set origin 0.55,0.05 #画右下角的图
set title '(cos(t),sin(4*t))'
plot cos(t),sin(4*t)

```

## 十二、 功能强大的 latex 以及 epslatex 终端——特殊符号的插入

gnuplot 提供两个非常重要的终端——latex 与 epslatex。这两个终端重要是因为借助 latex 强大的排版能力，我们可以实现许多仅仅靠 gnuplot 不容易或者根本就无法实现的功能。

当终端为 epslatex 时，图像的图形部分以 eps 文件输出，文字部分以 latex 文件输出；当终端为 latex 时，所有的图像信息都以 latex 文件输出。我们通过修改 latex 文件来增强 gnuplot 的功能。

此处我们主要介绍利用终端 epslatex 实现特殊符号的插入。

set term epslatex color standalone #设置终端为 epslatex，可以使用颜色，可以独立编译

```

set output 'initial_state.tex' #设置输出文件名
set xrange [-3:3] #设置作图范围
set xlabel 'x' #设置 x 轴标签
set ylabel '\Psi' #设置 y 轴标签，其中\Psi 是 latex 下面的一个希腊字母
set title 'The initial state of schrödinger equation
\frac{\partial \Psi}{\partial t} = \hat{H} \Psi' #设置标题，该标题用 latex 写成

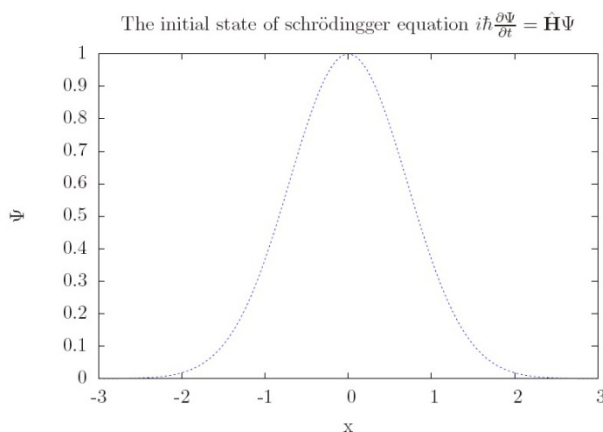
```

```

plot exp(-x**2) lt 3 notitle #画图，线型为 3，无图例，lt 是 linestyle 的简写
exit #退出 gnuplot

```

该段命令最后会在当前目录下面产生两个文件，分别为“initial\_state-inc.eps”和“initial\_state.tex”。编译文件 initial\_state.tex，最后生成 pdf 文件，得到的图像如图十二所示。



图十二 功能强大的 latex 以及 epslatex 终端

在该图中我们不仅插入了希腊字母与特殊符号‘ $\partial$ ’，还插入了一个数学公式，而且显示效果很是不错。

注：如果不能正常显示数学公式那么可以再 initial\_state.tex 中加入一行

```
\usepackage{amsmath}
```

即加上数学宏包。不过我的经验是不加也可以正常显示。

### 十三、 脚本——避免重复劳动

gnuplot 支持脚本，这可以节省很多重复劳动。

将作图命令写到一个文件，比如 plot.plt 中。启动 gnuplot 后通过命令

```
load 'plot.plt'
```

就可以完成作图了。这在要画好多幅相似的图形时是很有用的。

### 十四、 调用系统命令

gnuplot 可以调用系统命令，这在某些情况下是很有用处的。调用方法如下：

#### !系统命令

例如下面的命令分别在 Windows 和 Linux 系统下修改脚本文件 plot.plt。

Windows: !edit plot.plt

Linux : !vi plot.plt

### 十五、 中文支持

一些 gnuplot 的介绍文章称通过改变字体可以 gnuplot 的实现中文支持，但是“按图索骥”，生成图片中的中文依然是乱码（只有当输出终端为屏幕时显示正常）。如果这不能表明 gnuplot 不支持中文，那么这也表明至少 gnuplot 对中文的支持不太好。

这里介绍通过 latex 实现 gnuplot 对中文支持的方法。

set term epslatex color standalone #设置终端为 epslatex，可以使用颜色，可以独立编译

```
set output 'chinese_support.tex' #设置输出文件名
```

```
set xlabel 'x'
```

```
set ylabel 'y'
```

```
set title '正弦函数图像'
```

```
plot sin(x) lt 3 lw 3 notitle #lt 是 linestyle 的简写，lw 是 linewidth 的简写
```

```
exit #退出 gnuplot
```

此时在当前目录下即可找到文件“chinese\_support.tex”与“chinese\_support-inc.eps”。这时直接编译 tex 文件并不能得到中文显示。我们需要对 tex 文件做一点点改动。

打开 chinese\_support.tex，其中有一段是用来用来装载宏包的（该段的前面应该有一句注释“% Load packages”）。在这里我们加上一行

```
\usepackage{CJK}
```

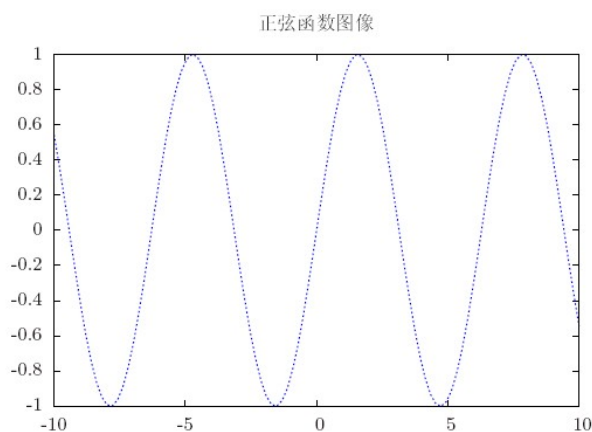
再找到\begin{document}，在后面加上一行

```
\begin{CJK*}{GBK}{song} #字体可以视个人喜好选择
```

最后找\end{document}，在其前面加上一行

```
\end{CJK*}
```

至此 latex 文件修改完毕，保存。编译修改后的 tex 文件，生产 pdf 文件，得到图像如图十五所示。



图十五 中文支持

我们得到的图像中正常显示中文了。

## 十六、 与其他软件的比较

现在科学作图及数据可视化工具以 `matlab`、`mathematica`、`origin` 以及 `gnuplot` 比较流行。此外还有一些开源的数学软件（比如 `scilab`）也可以进行科学绘图。这里简单的对这几个工具在绘图方面做一个比较。

**Matlab:** 绘图功能非常强大，绘制的图像很漂亮，既支持命令方式对图像修改也可以在图形模式下对图像修改。缺点在于该软件价格太高。

**Mathematica:** 绘图功能一般，绘制的图像漂亮程度一般，早期版本不支持图形模式下修改图像，**Mathematica7** 支持图形模式下修改图像，但是功能简单。此外该软件的价格也很高。

**Origin:** 绘图功能较强大，二维图与等高图很漂亮，三维图不漂亮，图形模式下作图，简单易学。价格昂贵。

**Gnuplot:** 绘图功能非常强大，绘制的图像很漂亮，只支持命令方式作图（`gnuplot4.4` 开始也在非常有限的范围支持图形模式下修改图像了，比如改变视角）。免费软件，无需考虑价格因素。

其他开源数学软件：绘图功能参差不齐，且多以 `gnuplot` 为绘图内核。多数免费。

以上比较表明可以与 `gnuplot` 在功能上匹敌的只有 `matlab`，但是 `matlab` 价格实在太昂贵了。所以 `gnuplot` 还是非常值得推荐的一款科学绘图与数据可视化工具软件。

——数声风笛离亭晚，我想潇湘君想秦于二零一零年八月