一、前言

接触python有近半年的时间了，和大多数接触过python的朋友一样，我被它的简单的表述和清晰的描写思路深深吸引，python的三方模块很是丰富，在不断的学习中，渐渐里就接触到了matplotlib，官方的英文文档很详细，虽然是英文的，我想对于只要过了四级的朋友，仔细看看明白不是什么难事，原始的官方英文文档有800多页，详细是不用说，但还是有点罗嗦，本书的目的是让你能快速知道怎么使用matplotlib，当然个人所总结的方法未必对于每个人都是容易接受的，但相信大多数人看完本书之后会对于matplotlib有一个比较清晰的理解，读者如果在文中发现错误还请麻烦反馈一下。本书是对于学习中需要用到matplotlib的人群编写的，你在阅读之前可能需要了解一些python的知识，如果你对于python还一点不懂，那么还是请你看下python相关的文档吧。

如在阅读本书中遇到任何问题请联系master sell <yishenggudou@gmail.com>

本文在线网址为：http://www.hblib.info/book/matplotlib.html

pdf在线网址为： http://www.hblib.info/book/matplotlib.pdf

作者个人博客网址为: http://www.hblib.info/

---master sell

---2010.2.14

二、matplotlib 简介

Matplotlib 是一个在 python 下实现的类 matlib 的纯 python 的三方库，旨在用 python 实现 matlab 的功能，是 python 下最出色的绘图库，功能很完善，其风格跟 matlib 很相似 ，

同时也继承了python的简单明了的风格,其可以很方便地设计和输出二维以及三维的数据，其提供了常规的笛卡尔坐标，极坐标，球坐标，三维坐标等。其输出的图片质量也达到了科技论文中的印刷质量，日常的基本绘图更不在话下。

其使用中要先安装numpy库（一个python下数组处理的三方库，可以很方便的处理矩阵，数组），对于做数据 图，其原理很简单，就是把函数变成关于X,Y,Z的坐标点的数组,如函数$Y=X^2$,我们画图是也是先找一组特征点(x,y),然后连接成线,matplotlib出图的过程和我们画图过程差不多,先生成X的一个取值数组,如要画区间[0,1]的图像,则先取[0,1]之内的一组数组(如:x=arange(0,1,0.01)表示x以0.01为步长取100个点),然后去对应x的对应y的值的一组数据 ,这样以坐标(x,y)画出的图就是一条曲线了。

matplotlib对于图像美化方面的比较完善，可以自定义线条的颜色和样式，可以在一张绘图纸上绘制多张小图，也可以在一张图上绘制多条线，可以很方便地将数据 可视化并对比分析。

三、Matplotlib 使用入门

Matplotlib的使用中有好几种输出风格，有matlab的风格，和官方文档的使用的as风格，各有所长，本文对比介绍matplotlib官方文档中的使用风格，
我们画图的目的是要将函数已图像显现出来，所以我们要用python处理的东西有两个，一个是函数，另一个是图像，函数部分在matplotlib的使用中是用了numpy这个三方库，这个库有着很大的科学计算功能，但是不用担心，在使用中你会发现他的好用，

3.1 函数处理部分（numpy 的使用）

具体的使用可以参见numpy的文档
我们使用matplotlib将数据 可视化时基本上我们只用到两种方法
1：arange函数类似于python的range函数，通过指定开始值、终值和步长来创建一维数组，注意数组不包括终值:

```
>>> import numpy as np
>>> np.arange(0,1,0.1)
array([ 0. ,   0.1,   0.2,   0.3,   0.4,   0.5,   0.6,   0.7,   0.8,   0.9])
```

此函数在区间[0,1]之间以0.1为步长生成一个数组。
如果第三个参数预设为1

```
>>> np.arange(0,10)
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
>>> np.arange(0,5.6)
array([ 0.,   1.,   2.,   3.,   4.,   5.])
>>> np.arange(0.3,4.2)
array([ 0.3,   1.3,   2.3,   3.3])
```

2：linspace函数通过指定开始值、终值和元素个数来创建一维数组，可以通过endpoint关键字指定是否包括终值，缺省设置是包括终值:

```
>>> np.linspace(0, 1, 12)
array([ 0.          ,  0.09090909,  0.18181818,  0.27272727,  0.36363636,
        0.45454545,  0.54545455,  0.63636364,  0.72727273,  0.81818182,
        0.90909091,  1.          ])
```

第三个参数 num 预设为 50

```
>>> np.linspace(0,10)
array([  0.        ,   0.20408163,   0.40816327,   0.6122449 ,
         0.81632653,   1.02040816,   1.2244898 ,   1.42857143,
         1.63265306,   1.83673469,   2.04081633,   2.24489796,
         2.44897959,   2.65306122,   2.85714286,   3.06122449,
         3.26530612,   3.46938776,   3.67346939,   3.87755102,
         4.08163265,   4.28571429,   4.48979592,   4.69387755,
         4.89795918,   5.10204082,   5.30612245,   5.51020408,
         5.71428571,   5.91836735,   6.12244898,   6.32653061,
         6.53061224,   6.73469388,   6.93877551,   7.14285714,
         7.34693878,   7.55102041,   7.75510204,   7.95918367,
         8.16326531,   8.36734694,   8.57142857,   8.7755102 ,
         8.97959184,   9.18367347,   9.3877551 ,   9.59183673,
         9.79591837,  10.        ])
```

Numpy 库有一般 math 库函数的数组实现；

如 sin，cos，log，

```
>>> x=np.arange(0,np.pi/2,0.1)
>>> y=sin(x)

Traceback (most recent call last):
  File "<pyshell#18>", line 1, in <module>
    y=sin(x)
NameError: name 'sin' is not defined
>>> y=np.sin(x)
>>> y
array([ 0.        ,  0.09983342,  0.19866933,  0.29552021,  0.38941834,
        0.47942554,  0.56464247,  0.64421769,  0.71735609,  0.78332691,
        0.84147098,  0.89120736,  0.93203909,  0.96355819,  0.98544973,
        0.99749499])
```

数组的最后一项不是 1 是因为数组的数据不是标准的浮点型的数据

```
>>> x
array([ 0. ,  0.1,  0.2,  0.3,  0.4,  0.5,  0.6,  0.7,  0.8,  0.9,  1. ,
        1.1,  1.2,  1.3,  1.4,  1.5])
```

如果要精确的浮点计算请参见 numpy 文档。

基本函数（三角，对数，平方，立方，等）的使用就是在函数前加上 np.这样就能实现数组的函数计算。

3.2 Matplotlib 使用风格介绍

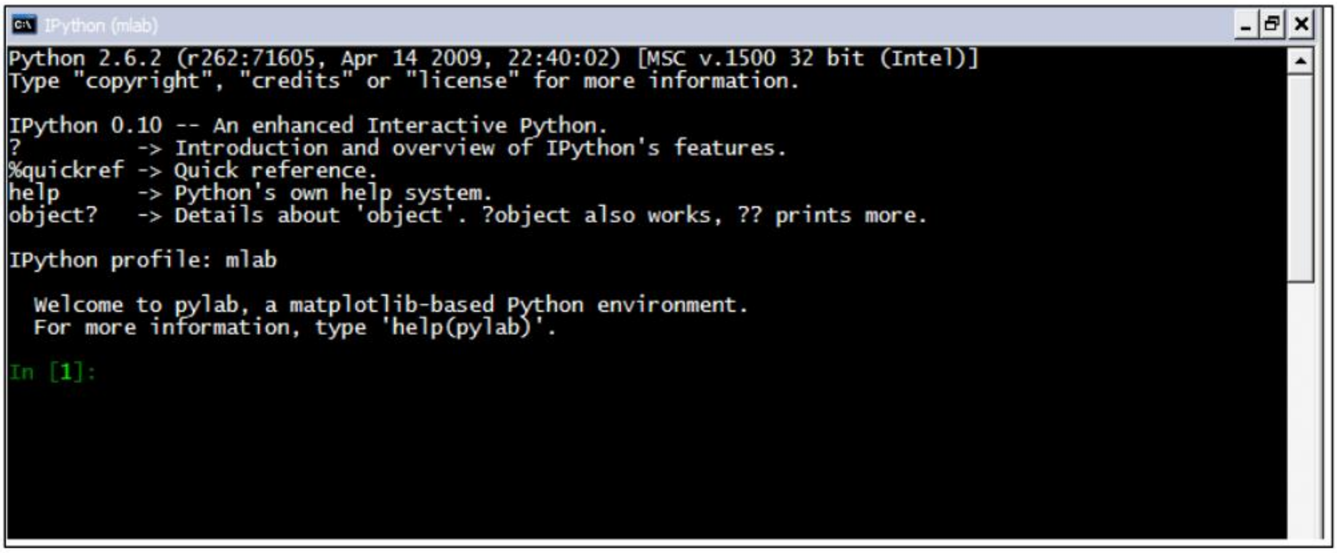
Matplotlib 的使用中有好几种输出风格，有类 matlab 的风格，和官方文档的使用的 as 风格，

各有所长，下面详细介绍这两种方法
先介绍 matlab 风格
下边的例子左边是 Matplotlib 的类 matlab 的风格，右边是原始的 matlab 实现，

```
from pylab import *

dt = 0.01
t = arange(0,10,dt)
nse = randn(len(t))
r = exp(-t/0.05)

cnse = conv(nse, r)*dt
cnse = cnse[:len(t)]
s = 0.1*sin(2*pi*t) + cnse

subplot(211)
plot(t,s)
subplot(212)
psd(s, 512, 1/dt)
```

```
% no import necessary

dt = 0.01;
t = [0:dt:10];
nse = randn(size(t));
r = exp(-t/0.05);

cnse = conv(nse, r)*dt;
cnse = cnse(1:length(t));
s = 0.1*sin(2*pi*t) + cnse;

subplot(211)
plot(t,s)
subplot(212)
psd(s, 512, 1/dt)
```

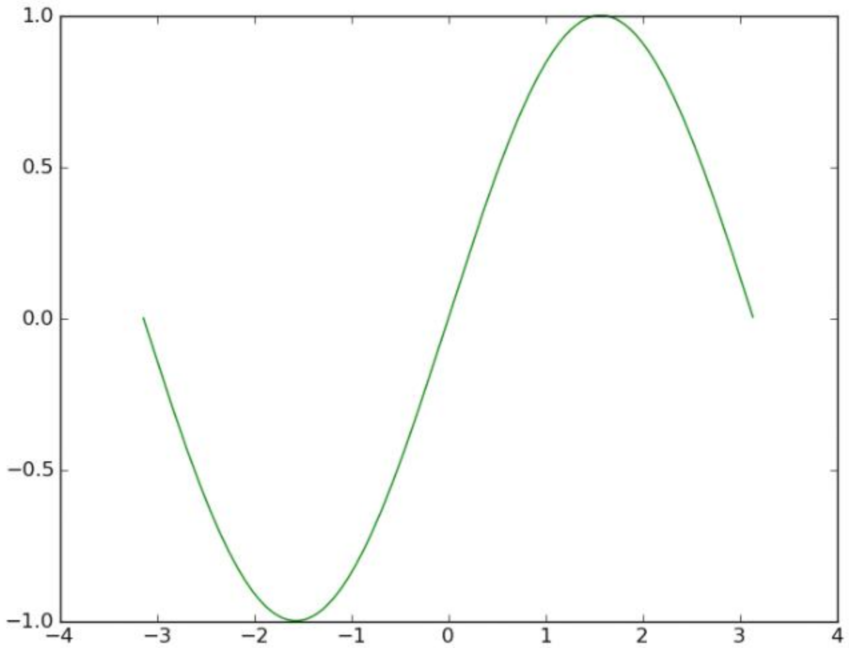
在 Ipython 下可以更好地使用 matlab 风格的输入，（Ipython 是一个优秀的 shell 工具，比自带的 IDEL 要好用）
下图是 Ipython 的使用界面：



```
Python 2.6.2 (r262:71605, Apr 14 2009, 22:40:02) [MSC v.1500 32 bit (Intel)]
Type "copyright", "credits" or "license" for more information.

IPython 0.10 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object'. ?object also works, ?? prints more.

IPython profile: mlab

  Welcome to pylab, a matplotlib-based Python environment.
  For more information, type 'help(pylab)'.

In [1]:
```

下面介绍官方文档中使用较多的 as 风格
如我要做一个正弦函数的在区间[-pi,pi]图像
>>> import numpy as np

```
>>> import matplotlib.pyplot as plt
```



```
>>> x=np.arange(-np.pi,np.pi,0.01)
>>> y=np.sin(x)
>>> plt.plot(x,y,'g')
[<matplotlib.lines.Line2D object at 0x01A9A3>]
>>>plt.show()
```

只需要四句代码，简单的正弦函数图就出来了，是不是很简单，至于图像的美化加标签等在后面的章节给大家介绍。

官方文档中给出了 104 个函数的简单介绍，我们的各种样式的图都可以用他们实现。

Function	Description
acorr	plot the autocorrelation function
annotate	annotate something in the figure
arrow	add an arrow to the axes
axes	create a new axes
axhline	draw a horizontal line across axes
axvline	draw a vertical line across axes
axhspan	draw a horizontal bar across axes
axvspan	draw a vertical bar across axes
axis	set or return the current axis limits
barbs	a (wind) barb plot

bar	make a bar chart
barh	a horizontal bar chart
broken_barh	a set of horizontal bars with gaps
box	set the axes frame on/off state
boxplot	make a box and whisker plot
cla	clear current axes
clabel	label a contour plot
clf	clear a figure window
clim	adjust the color limits of the current image
close	close a figure window
colorbar	add a colorbar to the current figure
cohere	make a plot of coherence
contour	make a contour plot
contourf	make a filled contour plot
csd	make a plot of cross spectral density
delaxes	delete an axes from the current figure
draw	Force a redraw of the current figure
errorbar	make an errorbar graph
figlegend	make legend on the figure rather than the axes
figimage	make a figure image
figtext	add text in figure coords
figure	create or change active figure
fill	make filled polygons
fill_between	make filled polygons between two curves
findobj	recursively find all objects matching some criteria
gca	return the current axes
gcf	return the current figure
gci	get the current image, or None
getp	get a graphics property
grid	set whether gridding is on

hexbin	make a 2D hexagonal binning plot
hist	make a histogram
hold	set the axes hold state
ioff	turn interaction mode off
ion	turn interaction mode on
isinteractive	return True if interaction mode is on
imread	load image file into array
imsave	save array as an image file
imshow	plot image data
ishold	return the hold state of the current axes
legend	make an axes legend
loglog	a log log plot
matshow	display a matrix in a new figure preserving aspect
pcolor	make a pseudocolor plot
pcolormesh	make a pseudocolor plot using a quadrilateral mesh
pie	make a pie chart
plot	make a line plot
plot_date	plot dates
plotfile	plot column data from an ASCII tab/space/comma delimited file
pie	pie charts
polar	make a polar plot on a PolarAxes
psd	make a plot of power spectral density
quiver	make a direction field (arrows) plot
rc	control the default params
rgrids	customize the radial grids and labels for polar
savefig	save the current figure
scatter	make a scatter plot
setp	set a graphics property
semilogx	log x axis

semilogy	log y axis
show	show the figures
specgram	a spectrogram plot
spy	plot sparsity pattern using markers or image
stem	make a stem plot
subplot	make a subplot (numrows, numcols, axesnum)
subplots_adjust	change the params controlling the subplot positions of current figure
subplot_tool	launch the subplot configuration tool
suptitle	add a figure title
table	add a table to the plot
text	add some text at location x,y to the current axes
thetagrids	customize the radial theta grids and labels for polar
title	add a title to the current axes
xcorr	plot the autocorrelation function of x and y
xlim	set/get the xlimits
ylim	set/get the ylimits
xticks	set/get the xticks
yticks	set/get the yticks
xlabel	add an xlabel to the current axes
ylabel	add a ylabel to the current axes
autumn	set the default colormap to autumn
bone	set the default colormap to bone
cool	set the default colormap to cool
copper	set the default colormap to copper
flag	set the default colormap to flag
gray	set the default colormap to gray
hot	set the default colormap to hot
hsv	set the default colormap to hsv

jet	set the default colormap to jet
pink	set the default colormap to pink
prism	set the default colormap to prism
spring	set the default colormap to spring
summer	set the default colormap to summer
winter	set the default colormap to winter
spectral	set the default colormap to spectral

四、2D 图的画法

我们在介绍之前先想想 2D 图的一些元素............
我在这里按我的思路写下一些：
坐标轴（尺度，区间...），线条（样式，颜色...），图和线的标签和注释，图像大小，图像里图片的排版（一张图像中多张图）.....
下面我们将分别介绍

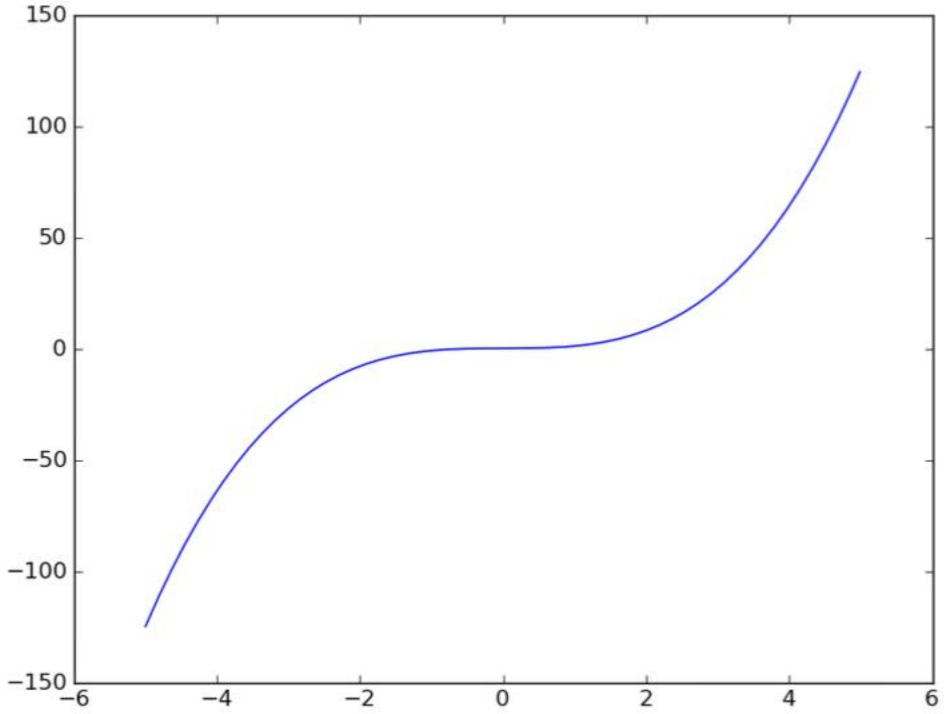
4.1 坐标轴

我们做出一个精确的 2D 图，其中不免要用数轴来显示坐标，下面介绍 matplotlab 里面的坐标轴使用：

1. 坐标区间设置

实例 1：设置坐标轴的区间

```
>>> import numpy as np
>>> import matplotlib.pyplot as plt
>>> x=np.arange(-5,5,0.01)
>>> y=x**3
>>> plt.axis([-6,6,-10,10])
[-6, 6, -10, 10]
>>> plt.plot(x,y)
[<matplotlib.lines.Line2D object at 0x01A86EB0>]
>>> plt.show()
```



细心的读者会发现 y 轴的区间和给定的区间并不一样，这里是系统自动调整了的，

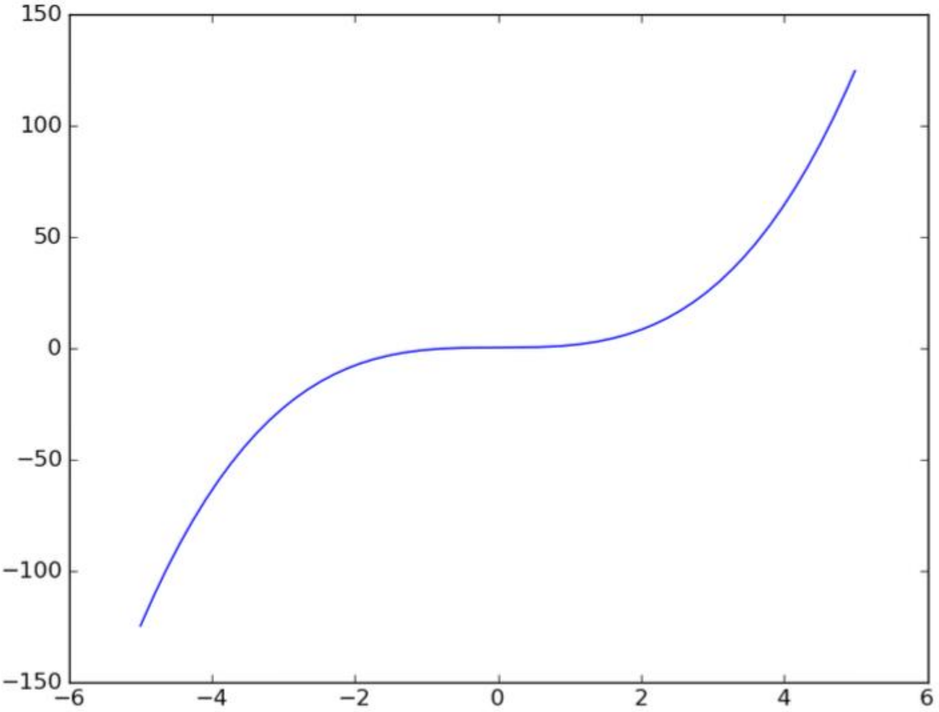
在 matplotlab 下的坐标轴设置还有另外一种方法

```
xlim( (xmin, xmax) )   #设置坐标轴的最大最小区间
xlim( xmin, xmax )      # 设置坐标轴的最大最小区间
ylim( (ymin, ymax) )   # 设置坐标轴的最大最小区间
ylim( ymin, ymax )      # 设置坐标轴的最大最小区间
```

请看下面实例：

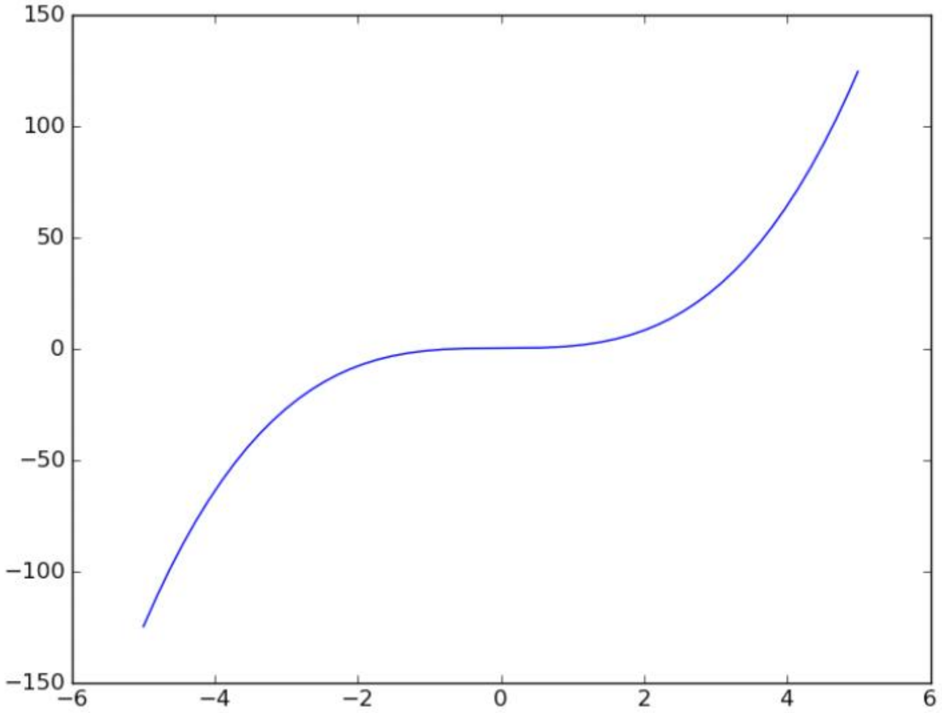
```
>>> import numpy as np
>>> import matplotlib.pyplot as plt
>>> x=np.arange(-5,5,0.01)
>>> y=x**3
>>> plt.xlim(-6,6)
(-6, 6)
```

```
>>> plt.ylim(-500,500)

(-500, 500)

>>> plt.plot(x,y)

[<matplotlib.lines.Line2D object at 0x01BB2FF0>]

>>> plt.show()
```



我们的程序根据我们的参数给出了图像，但是我们发现 y 轴区间仍然不是给定区间（甚至当设置 xlim（-2，2）时，x 轴的区间仍然和给定的参数不一致），下面我们将给予讨论：

```
>>> import numpy as np
>>> import matplotlib.pyplot as plt
>>> x=np.arange(-5,5,0.01)
>>> y=x**3
>>> plt.xlim(-2,2)
(-2, 2)
>>> plt.ylim(-500,500)
(-500, 500)
>>> plt.plot(x,y)
[<matplotlib.lines.Line2D object at 0x01AA6FF0>]
>>> plt.show()
```

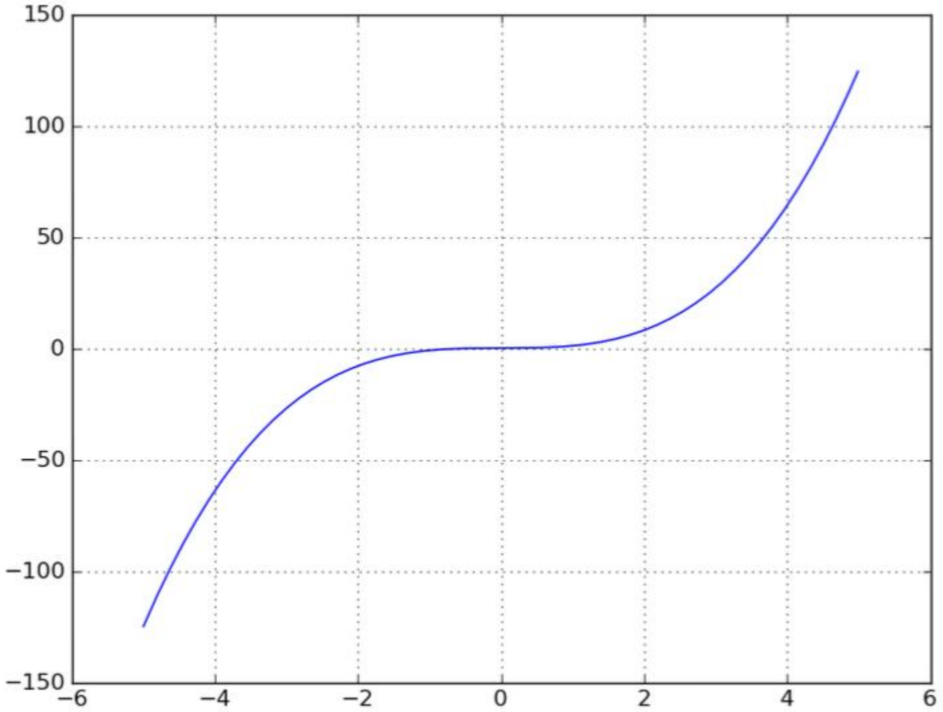


在 Ipython 下 show()之后使用参数 xlim(min,max),ylim(min,max)可以使 x 轴或者 y 轴显示到给定参数的区间，但是在 show()之前仍然是无效的。

2. 设置网格线

grid（）函数可以提供是否显示网格的选项，当你在 plot（）之后加上 grid（True）则可以在坐标上显示网格

```
>>> import numpy as np
>>> import matplotlib.pyplot as plt
>>> x=np.arange(-5,5,0.01)
>>> y=x**3
>>> plt.xlim(-2,2)
(-2, 2)
>>> plt.ylim(-500,500)
(-500, 500)
>>> plt.plot(x,y)
[<matplotlib.lines.Line2D object at 0x01AA6FF0>]
>>> plt.grid(True)
>>> plt.show()
```



3. 自定义坐标轴刻度

4. 自动标注坐标轴刻度

4.2 线条

线条颜色设置表

Alias	Color
‘b’	blue
‘g’	green
‘r’	red
‘c’	cyan
‘m’	magenta
‘y’	yellow
‘k’	black
‘w’	white

线条样式表

marker	description
‘.’	point
‘,’	pixel
‘o’	circle
‘v’	triangle_down
‘^’	triangle_up
‘<’	triangle_left
‘>’	triangle_right
‘1’	tri_down
‘2’	tri_up
‘3’	tri_left
‘4’	tri_right
‘s’	square
‘p’	pentago

	n
‘*’	star
‘h’	hexagon1
‘H’	hexagon2
‘+’	plus
‘x’	x
‘D’	diamond
‘d’	thin_diamond
‘|’	vline
‘_’	hline
TICKLEFT	tickleft
TICKRIGHT	tickright
TICKUP	tickup
TICKDOWN	tickdown
CARETLEFT	caretleft
CARETRIGHT	caretright
CARETUP	caretup
CARETDOWN	caretdown
‘None’	nothing
‘ ‘	nothing
‘’	nothing

4.3 标签注释

写作中……………………

4.4 图像排版

写作中……………………

五、3D 绘图方法

写作中……………………

六、后记

写作中……………………