

Gnuplot.py の使い方

高エネルギー加速器研究機構
加速器第一研究系
山本昇

November 5, 2008

Abstract

Linux, MacOSX, Windowsなどで手軽に数値グラフを作成するためのポピュラーなツールの一つに gnuplot がある。Gnuplot.py¹ は手軽にグラフを作成するためのプログラム gnuplot をプログラミング言語 Python から利用するためのパッケージである。Python と gnuplot を組み合わせることで、さまざまなデータ解析とその結果のグラフ表示が簡単に行える。この文書ではこの Gnuplot.py の利用法を説明する。

1 始めに

Gnuplot.py² は free なグラフ作成ソフトである gnuplot の機能を Python 言語から使えるようにする Python のモジュールである。Gnuplot.py は Konrad Hinsen 氏作成の Gnuplot.py に触発されて Michael Haggerty 氏が開発した Python module である。

Written by "Michael Haggerty", <mailto:mhagger@alum.mit.edu>. Inspired by and partly derived from an earlier version by "Konrad Hinsen", <mailto:hinsen@ibs.ibs.fr>. If you find a problem or have a suggestion, please "let me know", <mailto:mhagger@alum.mit.edu>. Other feedback would also be appreciated.

Gnuplot.py はシステムにインストールされた gnuplot コマンドを利用する形で動作する。Python および gnuplot の動作する環境、Linux, MacOSX, Windows など、であれば問題なく実行されるはずである。

2 簡単な例

Gnuplot.py は Python のモジュールとなっている。他の Python モジュールと同じように Gnuplot.py を使用するに先立ちこのモジュールを import する必要がある。

¹この文書は同じ著者による "gnuplot.py の使い方" を gnuplot.py モジュールの後継である Gnuplot.py モジュール向けに書き直したものである。

²この文書での説明は、Gnuplot.py version. 1.8/python 2.5 or 2.6/gnuplot 2.2.4 に基づいている。

```
import Gnuplot
```

あるいは

```
from Gnuplot import *
```

を実行する必要がある。以下の説明では、第一の形を使ったものとして説明を続ける。これは Gnuplot のモジュールをどこで使っているかを明確にするためで、実際の使用に際しては後者の形式で問題はない。python の起動毎に Gnuplot をインポートする事が面倒であれば、環境変数 PYTHONSTARTUP に設定されたファイルに上記の行を追加しておくとい。この環境変数によるスタートアップ・ファイルの読み込みは Python を対話的に使っただけ有効となるので、Python を非対話的に使うスクリプトの中で、このスタートアップ・ファイルを読み込ませるには、スクリプトの最初に、

```
import os; execfile(os.environ['PYTHONSTARTUP'])
```

を追加しておく。

2.1 一次元プロット : plot()

一次元のグラフを表示するには plot 関数を使う。例えば、サイン関数のグラフを表示するには、

```
from numpy import *
x=arange(256)*pi/128
y=sin(x)
gp=Gnuplot.Gnuplot()
gp.plot(y)
```

ここで作成した Gnuplot.Gnuplot クラスの実体 (インスタンスあるいはオブジェクト) gp のメソッドに適切なデータを渡すことで、gnuplot のグラフを作成する。

とする。これによって図 1 のグラフが X の画面に表示される。(MacOSX ではデフォルトの出力は AquaTerm になっている)

この画面をファイルに書き出すためには、

```
gp.plot(y)
gp.hardcopy('gpsample1.eps',eps=True,color=True,fontsize=24)
```

とファイル名を指定すればよい。

これらの例では横軸はデータの index となっていた。横軸を x にするためには、与えるデータを (x, y) で初期化した Gnuplot.Data オブジェクトとする [図 2]

```
d=Gnuplot.Data(x,y)
gp.plot(d)
```

デフォルトではプロットのデータ点は点毎にシンボルで表示される。同じデータを線分で表示するためには gnuplot の "set data style" オプションを利用する。

```
gp('set style data lines")
gp.plot(y)
gp.hardcopy('gpsample1.eps',eps=True,color=True,fontsize=24)
```

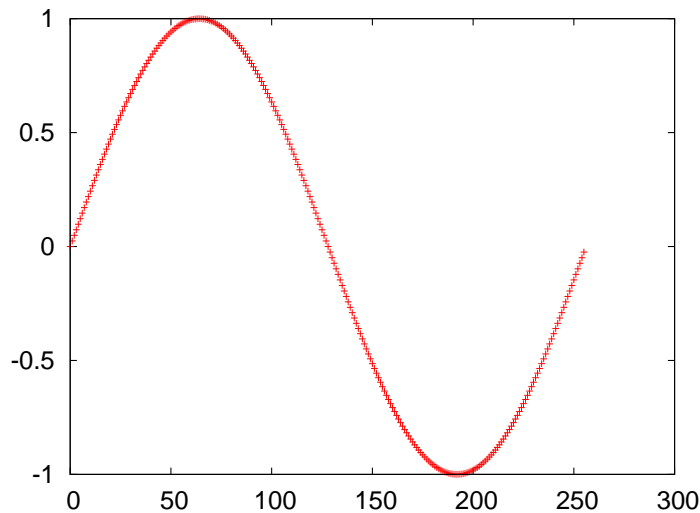


図 1: gnuplot.py による sin 関数の描画例

2.2 gnuplot オプションの変更

gnuplot にはさまざまなオプションが用意されている。これらのオプションを python から plot コマンドで指定するには例えば、

```
gp.plot(d,xrange='[0:6.3]',yrange='[-1.2:1.2]',
title='This is a Title', xlabel="x", ylabel="y")
```

とする [図 4] キーワード付きの引数 *xrange,yrange* 等を与えることで、gnuplot のオプション *xrange,yrange* を設定出来る。キーワードのオプションに与える値は文字列であることを注意しておく。keyword = 'val' を plot 関数の引数に与えることは gnuplot で、

```
set keyword val
```

を実行するのと同じである。val が空である場合には右辺として 空の文字列 (" または "") を指定する必要がある。xrange, yrange などはレンジを表す数値のペアをあたえることも可能である。

```
gp.plot(d,xrange=(0,6.3),yrange=(-1.2, 1.2),
title='This is a Title', xlabel="x", ylabel="y")
```

2.2.1 その他のオプションの指定

Gnuplot.Gnuplot オブジェクトの optiontypes メンバーは plot メソッドに与えることのできるオプションとその型を辞書型データとして保持している。

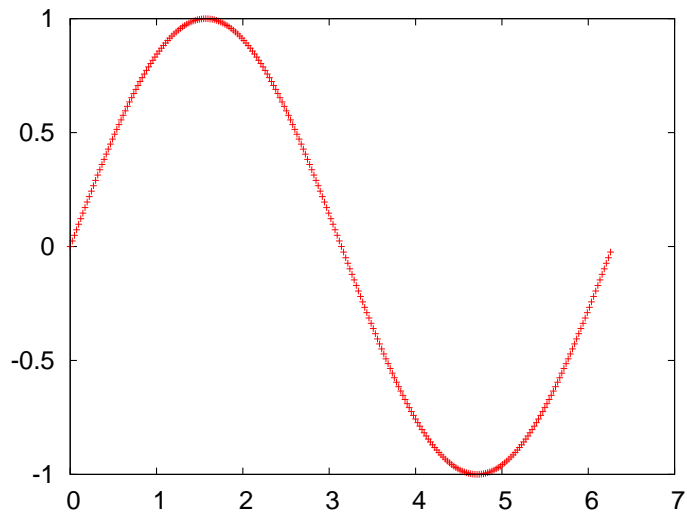


図 2: X 座標軸をデータとして与えて、sin 関数を描画

```
>>> print gp.optiontypes
{'polar': 'boolean',
 'xrange': 'range',
 'zlabel': 'string',
 'trange': 'range',
 'yrange': 'range',
 'title': 'string',
 'vrange': 'range',
 'xlabel': 'string',
 'ylabel': 'string',
 'output': 'string',
 'urange': 'range',
 'parametric': 'boolean',
 'zrange': 'range'}
```

これらのオプションは、

```
gp.xlabel("x-label")
gp.ylabel("y-label")
gp.title("Title string")
```

あるいは

```
gp.set(xlabel="x-label",
       ylabel="y-label",
       title="Title string")
```

としてセットすることができる。

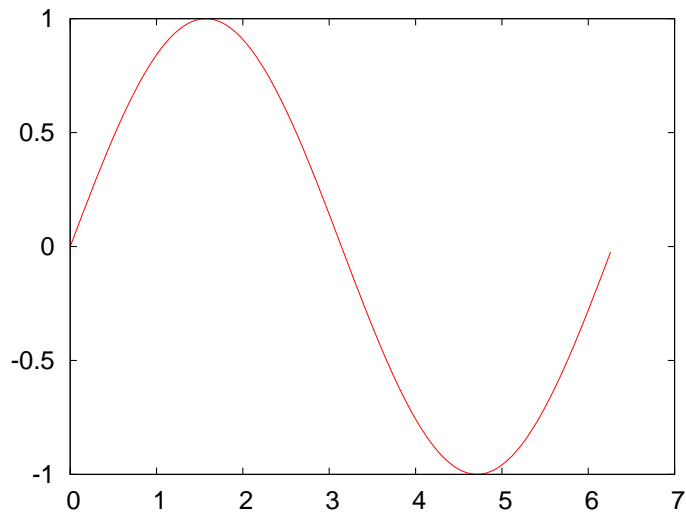


図 3: グラフを線表示

2.2.2 Gnuplot.Gnuplot のオブジェクトに直接 `gnuplot` のコマンドを渡す。

上記以外のオプションをセットするには、`Gnuplot.Gnuplot` オブジェクトの `gnuplot` コマンド実行機能を使う。`Gnuplot.Gnuplot` オブジェクトでは、`gp('set hidden')` などのように、`Gnuplot.Gnuplot` オブジェクトに直接 `gnuplot` のコマンドを渡して実行することができる。この機能をつかうことで、`gnuplot` の総ての機能を `Gnuplot.py` モジュールから利用できる。`gnuplot` のコマンドについては、`gnuplot` 付属のマニュアルを参考のこと。

3 複数のデータを一つのグラフに表示する。

複数のデータを一つのグラフに表示するには、単にデータを並べて、

```
y2=cos(x)
gp.plot(y,y2,yrange='[-1.2:1.2]',xrange="[*:*]")
```

とすればよい。`xrange="[*:*]"` は x 軸を autoscale にすることを意味している。
x - 軸の座標を指定するためには、

```
d1=Gnuplot.Data(x,y,title="sin(x)")
d2=Gnuplot.Data(x,y2,title="cos(x)")
gp.plot(d1,d2,xrange='[0:6.3]',yrange='[-1.2:1.2]')
```

とすればよい。このコマンドを実行すると図 6 に示されたグラフを生成する。

3.1 Data スタイル

`Data` オブジェクトを使うと、各データのスタイルを `with` オプションをつかって指定できる。

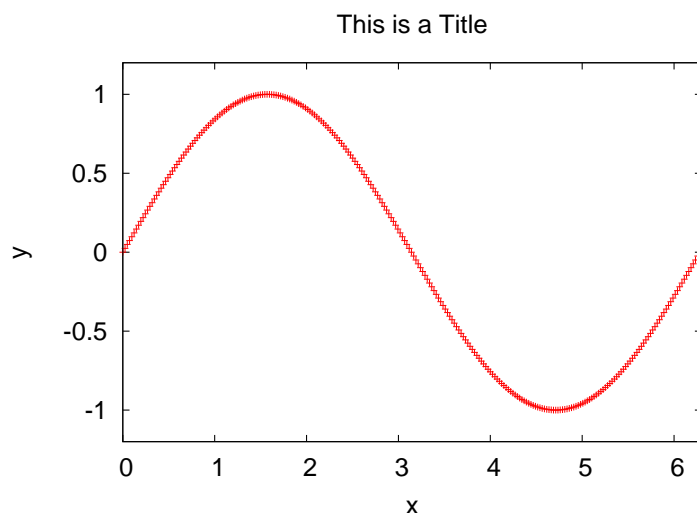


図 4: gnuplot オプション設定の例: `xrange='[0:6.3]',yrange='[-1.2:1.2]',title='\"This is a Title\"'` を設定した。

```
d1=Gnuplot.Data(x,y,title="sin(x)",with_"points pointtype 3")
d2=Gnuplot.Data(x,y2,title="cos(x)", with_"lines linestyle 4")
gp.plot(d1,d2,xrange=' [0:6.3] ',yrange=' [-1.2:1.2] ')
```

このオプションは、gnuplot の with コマンド

```
with <style> { {linestyle | ls <line_style>}
| {{linetype | lt <line_type>}
{linewidth | lw <line_width>}
{linecolor | lc <colorspec>}
{pointtype | pt <point_type>}
{pointsize | ps <point_size>}
{fill | fs <fillstyle>}
{nohidden3d}
{palette}}
}
```

を指定するものである。Python において with は予約語としてリザーブされているため、Data オブジェクトの指定では with_ を使うことに注意すること。<style>には、

lines, points, linespoints, impulses, dots, steps, fsteps, histeps, errorbars, labels, xerrorbars, yerrorbars, xyerrorbars, errorlines, xerrorlines, yerrorlines, xyboxes, histograms, filledcurves, boxerrorbars, boxxyerrorbars, financebars, candlestick, vectors, image, rgbimage, pm3d

のうちいずれかを指定する。

一般に plot コマンドで指定可能なオプションは Gnuplot.Data (や Gnuplot.GridData などの Gnuplot.PlotItem の子クラス) でオブジェクト生成時のオプションとして指定可能である。plot コマンドのオプションについては、gnuplot のマニュアルを参照のこと。

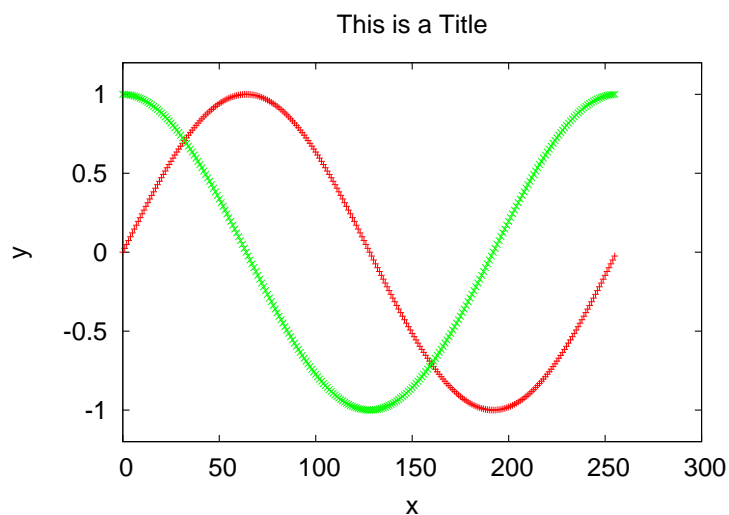


図 5: 複数データの一つのグラフへの表示

4 3次元プロット

splot メソッドを使うことで、三次元のグラフも作成できる。

```
d3=Gnuplot.Data(x,y,y2)
gp.splot(d3)
```

4.0.1 グリッドデータ

3次元のグラフを作成する際には、グリッドデータを使うこともできる。

```
m=[[sin(t**2+u**2) for u in x] for t in x]
gd=Gnuplot.GridData(m,x,x, binary=0)
gp('set parametric')
gp('set data style lines')
gp('set hidden')
gp('set contour base')
gp.splot(gd)
```

参考文献

- [1] N. Yamamoto “arr.py の使い方”, in preparation

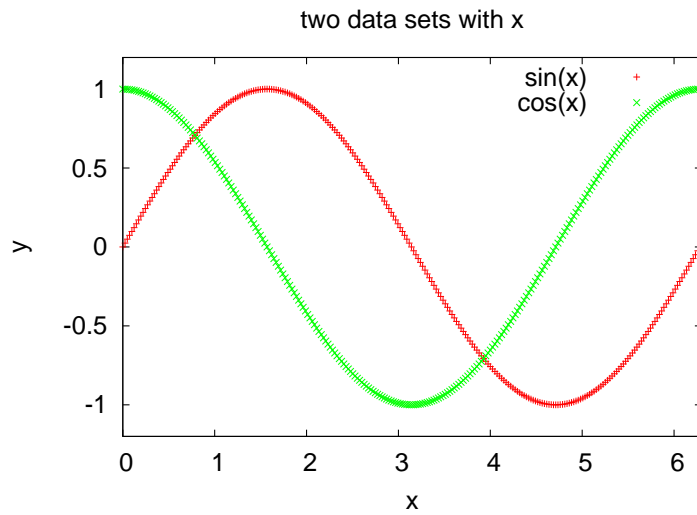


図 6: 複数データの表示 : x - 座標付きの場合。

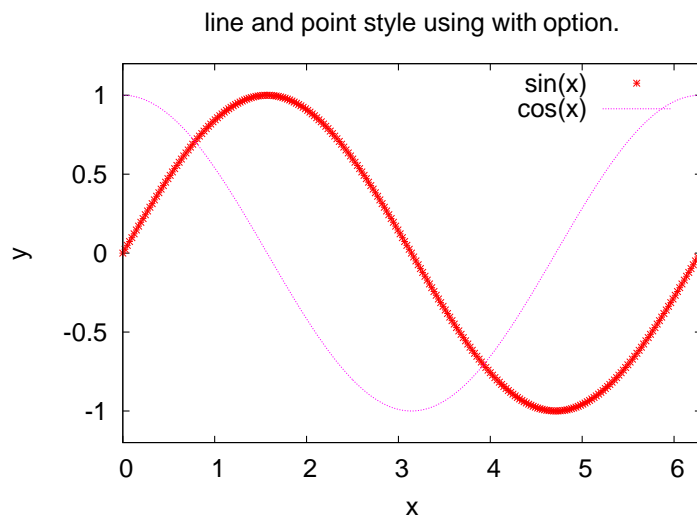


図 7: with_オプションによる plot シンボル、線種の指定。

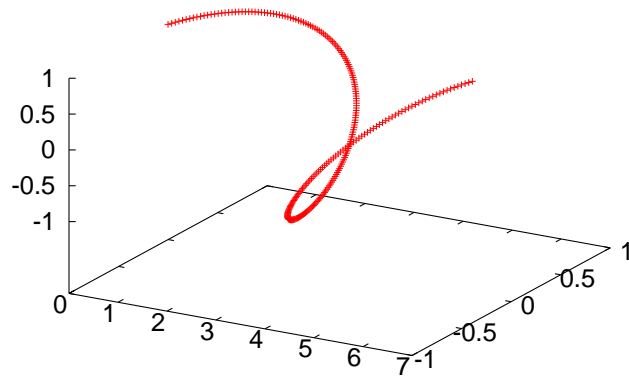


図 8: 3次元プロット

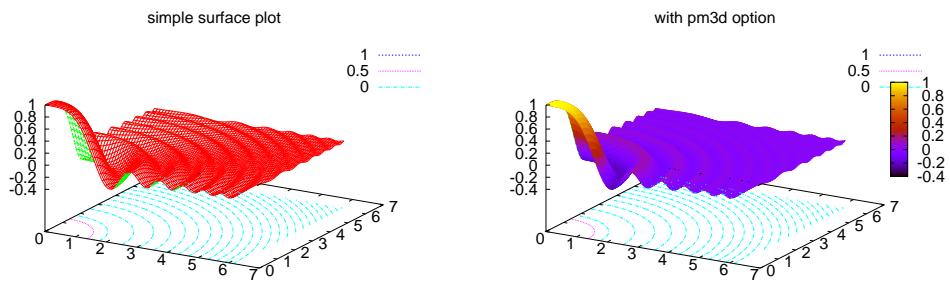


図 9: Surface plot without(Left) and with(Right) pm3d option