

GNUPLOT の易しい使い方

1. 画面上に図を出す

```
gnuplot> plot [x-range] [y-range] "filename" using x:y:z:w  
               title "titlename" with lines n
```

x-range : 横軸の範囲を設定する（省略可）。
書式は [x-low:x-high]（一方のみでも可）。
ただし、y-range を設定する時は省略不可（[] とする）。

y-range : 縦軸の範囲を設定する（省略可）。
書式は [y-low:y-high]（一方のみでも可）。

x-range, y-range 共に、[t=low:high] という
ように変数名を指定する（例 : t）と、以後の計
算式中等においてその名前が有効になる。

filename : データファイル名を指定する（計算式等も可）。
なお、'<' でファイル名を始めると、awk 等のコマ
ンドを通過させることが可能。

using x:y:z:w : データファイルの何列目から値を得るかを指定する。
x --- 横軸
y --- 縦軸
z, w --- errorbars 指定時のみ使用
z のみの時は y-delta
z:w の時は y-low:y-high
Default は 0:1 であり、また using n は
using 0:n と同じである。

title : 絵の右上に出すタイトルを指定する（省略可）。
なお、省略すると filename になる。

with lines n : with lines で細線による描画を指定する。n は省

略可であるが、値が大きい程点線の間隔が広くなる。
with 以降全てが省略可。また、with は w と、
lines は l と略記が可能である。

他に with の後に来るものとして、points,
dots, errorbars, linespoints, impulses が
あり、lines と同様に n によって種類を指定
する。

"file-name" 以降は、', ' で区切ることによって、複数書くことが可能であり、
それらは1つの図中におさめられる。また同様に、using で指定した x の値
を用いて計算式 (例 : $\cos(x)$) を書くことも可能である。

gnuplot > replot --- 直前の plot コマンドを繰り返す。

2. 図を postscript プリンターに出す

```
gnuplot > set terminal postscript
gnuplot > set output "|lpr"
gnuplot > plot -----(1 . と同様に入力する)
gnuplot > set terminal x11 (環境によって変わる)
gnuplot > set output
```

output の指定をファイル名にする (例 : "filename") と、postscript 形式の
データがセーブされるので、プリンター出力を ghostview 等で確認すること
が可能である。

postscript 形式の図は Default で Landscape, 横 10inch、縦 7 inch であり、
それ以外のサイズの図が必要な場合は、size コマンドを用いて図の大きさを
指定する必要がある。

```
gnuplot > set size 0.5, 0.5 --- half-size 指定
```

size コマンドの引数は Default の値に対する相対値であり、計算式を書くこ
とも可能である。

3. データファイルの書式

```
# comment
data1-1 {data2-1 data3-1 ...}
```

```
data1-2 {data2-2 data3-2 ...}
data1-3 {data2-3 data3-3 ...}
...
```

各行の # 以降はコメントとみなされる。

data#-n の # は、1 . の using で指定する値を示している。

4. ラベル & タイトル

```
gnuplot > set label tag "label-text" at x, y justification
gnuplot > show label --- ラベル確認
gnuplot > set nlabel tag --- ラベル消去
```

tag : ラベルの番号 (0 以上)。省略すると最も古く、最も使われていなかったものが指定される。
(消去の時は、省略すると全消去)

at x, y : ラベルを表示する位置。

justification : left --- 左寄せ (Default)
center --- 中央
right --- 右寄せ

```
gnuplot > set title "title-text" xoff, yoff --- タイトルを設定
gnuplot > show title --- 確認
```

```
gnuplot > set xlabel "label" xoff, yoff --- x 軸のラベルを設定
gnuplot > show xlabel --- 確認
```

```
gnuplot > set ylabel "label" xoff, yoff --- y 軸のラベルを設定
gnuplot > show ylabel --- 確認
```

Default の設定では、タイトルは図の上部中央、ラベルは各軸の中心に表示されるが、offset を指定することで、任意の位置に表示させることが可能である。当然、offset は省略可である (xoff のみ省略する場合は、', 'が必要)。また、title-text や label を省略すると、タイトルや各軸のラベルは clear される。

5. 矢印

```
gnuplot > set arrow tag from sx, sy to ex, ey nohead
```

```
gnuplot > show arrow          --- 矢印確認
gnuplot > set noarrow tag     --- 矢印消去
```

from *sx, sy* : 矢印の起点。省略すると原点になる。
to *ex, ey* : 矢印の終点。定義済のものの修正時には省略可。
nohead : 指定すると矢印の head がなくなる (Default は head あり)。nohead 指定を取り消すには head を指定する。

tag については、4. を参照のこと。

6. スケール & レンジ

```
gnuplot > set autoscale axes  --- auto-scaling を設定
gnuplot > set noautoscale axes ---                解除

gnuplot > set logscale axes   --- log スケールを設定
gnuplot > set nologscale axes ---                解除
gnuplot > show logscale      ---                確認
```

axes よって設定、解除する軸を指定する。一度に複数の軸を指定することが可能であり、また省略すると全ての軸を指定したことになる。

```
gnuplot > set xrange [x-low:x-high] --- x 軸のレンジを設定
gnuplot > set yrange [y-low:y-high] --- y 軸
```

各軸の表示範囲を [low <= (x, y) <= high] に限定する。レンジ設定を解除するには 'set autoscale' を用いる。

7. PLOT STYLE

```
gnuplot > set data style stylename --- スタイルの指定
gnuplot > show data style          ---                確認
```

stylename には lines, points, dots, errorbars, linespoints, impulse のいずれかを指定する。

8. 変数の利用

```
gnuplot > (変数) = (定数)
```

とすると、その後（変数）を（定数）の値をもつものとして参照できる。

```
( 例 ) gnuplot > n=2
      gnuplot > plot "mydata" using 1:n --- 'using 1:2' 指定になる。
```

```
gnuplot > show variables    --- 設定済み変数の確認
```

9. バッチ処理

gnuplot command を羅列したファイル（例：myplot.dat）を用意し、
gnuplot > load "myplot.dat"
とすると、gnuplot はファイル中のコマンドを一括して実行した後、プロンプトに戻る。

10. 図をL a T e Xに取り込む

gnuplot を用いて画かれた図は、一度ファイル（例：plot.tex）に落とすことでL a T e Xの文章中に取り込むことが可能である。

```
gnuplot > set terminal latex
gnuplot > set output "plot.tex"
gnuplot > plot -----(1 . と同様に入力する)
gnuplot > set terminal x11 (環境によって変わる)
gnuplot > set output
```

文章中に取り込まれる図のサイズは Default で横5 inch、縦3 inchであり、それ以外のサイズの図が必要な場合は、size コマンドで図の大きさを指定する必要がある。（2 . を参照のこと）

```
gnuplot > set size 0.6, 0.6    (A 4 縦に2つの図を並べる時の大きさの例)
```

(参考) L a T e X文章中での取り込み例

```
\begin{figure}
\begin{center}
\input{plot} --- 上で指定したファイル名（'.tex' は不要）
\end{center}
\end{figure}
```

(詳細はL a T e Xのマニュアルを参照のこと)

11. 図をT g i fに取り込む

10. と同様に (terminal 設定は `tgif`) することで、`gnuplot` を用いて画かれた図は `T g i f` に取り込むことが可能である。

Gnuplotの使い方補足

データファイルからデータを読み込み、グラフを描く

<読み込むデータファイルの形式>

右のファイル(a.dat)の場合、列指定をしない限り、plot コマンドでは、1列目をX軸、2列目をY軸にとって、グラフを描きます。

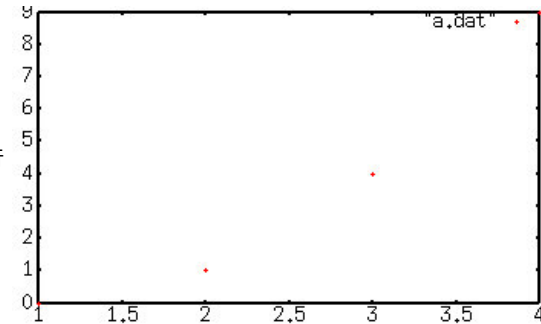
つまり、

```
gnuplot> plot "a.dat"
```

とした場合は、1列目がX軸、2列目がY軸になります。
#ではじまる行はコメント行とみなされ無視されます。
列と列の間の区切りは、空白(スペースもしくはタブ)で行います。

例(a.dat)

#1列目	2列目	3列目
1	0	2.3
2	1	3.5
3	4	4.8
4	9	6.2



<使用する行を指定する>

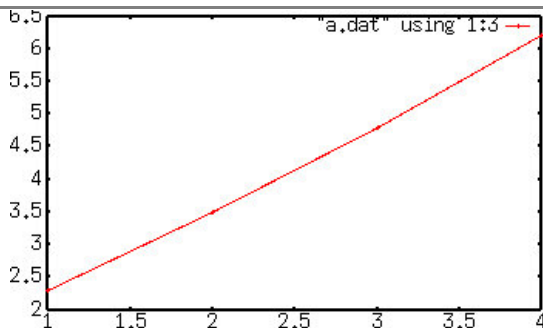
a.datの場合、

```
gnuplot> plot "a.dat" using 1:3
```

とすることで、1列目をX軸に2列目をY軸にすることができます。

少し見にくいですので、点の間に線を引きましょう。

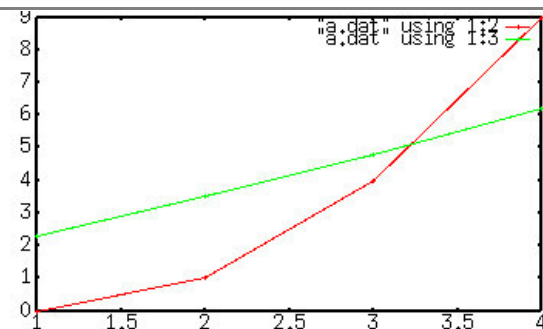
```
gnuplot> plot "a.dat" using 1:3 with  
linespoints
```



<2つ以上のグラフを同時に表示する>

```
gnuplot> plot "a.dat" using 1:2 with  
linespoints, "a.dat" using 1:3 with linespoints
```

のように、カンマで区切ることで、2つ以上のグラフが同時表示可能です。

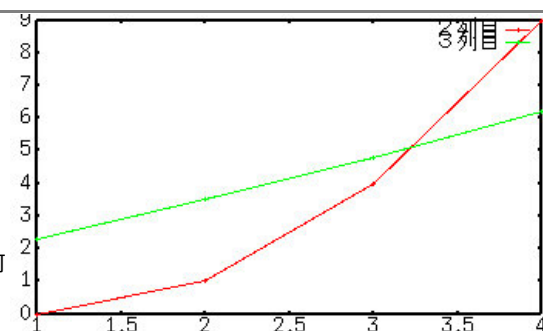


<凡例をつける>

デフォルトでは、凡例に使用したデータファイル名等が表示されます。これでは、かっこ悪いので、データごとに凡例を付けましょう。

```
gnuplot> plot "a.dat" using 1:2 ti "2列目"  
with linespoints, "a.dat" using 1:3 ti "3列目"  
with linespoints
```

のように、各グラフ毎に凡例(タイトル)をつけることが可能になります。

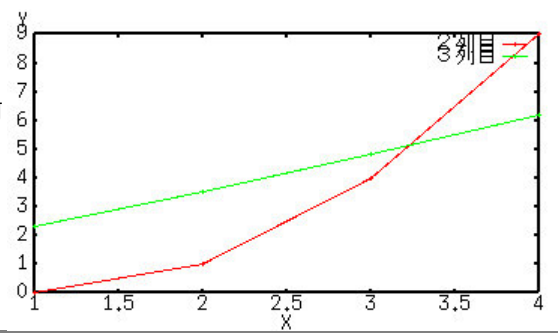


<軸名をいれる>

レポートを提出する際には、当然軸名を記入しなければいけなくなりますので、グラフにいれておきましょう。

```
gnuplot> set xlabel "X"  
gnuplot> set ylabel "Y"  
gnuplot> replot
```

のように、各グラフ毎に凡例(タイトル)をつけることが可能になります。



Last modified: May 6, 2005