

GNUPLOT による数値データのグラフ化
2009年版 v.0.0.1

鳥取大学工学部応用数理工学科

平成 21 年 3 月 24 日

目次

第1章 起動と終了	2
第2章 グラフの描画	4
第3章 グラフの装飾	6
第4章 様々なグラフの書き方	9
4.1 エラーバーが付いたグラフを描く	9
4.2 2つのY座標をもつグラフを描く	9

第1章 起動と終了

まずは gnuplot を起動してみましょう。すると、以下のような文章が表示されます (下記は一例で、バージョンなどによって多少違う可能性があります)。

```
G N U P L O T
Version 4.0 patchlevel 0
last modified Thu Apr 15 14:44:22 CEST 2004
System: MS-Windows 32 bit
```

```
Copyright (C) 1986 - 1993, 1998, 2004
Thomas Williams, Colin Kelley and many others
```

```
This is gnuplot version 4.0. Please refer to the documentation
for command syntax changes. The old syntax will be accepted
throughout the 4.0 series, but all save files use the new syntax.
```

```
Type 'help' to access the on-line reference manual.
The gnuplot FAQ is available from
http://www.gnuplot.info/faq/
```

```
Send comments and requests for help to
<gnuplot-info@lists.sourceforge.net>
Send bugs, suggestions and mods to
<gnuplot-bugs@lists.sourceforge.net>
```

```
Terminal type set to 'windows'
gnuplot>
```

これで gnuplot が起動しました。この画面を「コマンドラインのウィンドウ」と呼びます。

では実際に使ってみましょう。試しに「plot sin(x)」とキーボードをうってください。

```
gnuplot> plot sin(x)
```

と表示されているはずですが。さらにリターンキーを押すと、別ウィンドウが表示され、三角関数が描画されているはずですが。このようにコマンド（命令）を入力していくことで、操作します。

gnuplot を終了するには、コマンドラインのウィンドウに戻って、

```
gnuplot> exit
```

と入力すると、終了させることができます。

第2章 グラフの描画

さっそく、グラフを描いてみましょう。ここでは、数値データを xy プロット形式のグラフにする、という作業を考えます。以下、数値データは data.txt というファイルに入っているものとします。data.txt は、以下のようなテキストファイルとします。

```
1.2 1.5 2.3
1.3 1.6 2.2
1.4 1.7 2.2
1.5 1.75 2.5
```

ただし、データの間には、空白 (スペース) を入れておく必要があります。

「1 列目のデータを x 軸に、2 列目のデータを y 軸にとったグラフ」を描くには、gnuplot のコマンドラインで以下のように入力します。

```
gnuplot> plot "data.txt" using 1:2 title "graph1" with linespoints
```

「graph1」というタイトルのグラフが描けました (タイトル表示は省略できます)。ここで「with linespoints」は、「点と線で描画する」という意味です。なお、「using」「title」「with」は、この順番で書かないとエラーになるようです。

線や点の種類を設定するときは、例えば以下のように、末尾に 2 つの数字を付加します。

```
gnuplot> plot "data.txt" using 1:2 title "graph1" with linespoints 5 3
```

最後の 2 つの数字のうち、1 つ目 (上では「5」) が線種を、2 つ目が (上では「3」) が点種を表します。線種は 1 から 9 まで指定でき、「1=赤」「2=緑」といった具合に色を表します。ただし、白黒プリンタに出力するときは、「1=実線」になり、2 以降は種々の破線や点線になります。一方、点種は 1 から 6 まで指定でき、「1=菱形」「2=十字形」といったように記号を表します。

「with」部分を以下のように変えることで、多彩なグラフが描けます (省略すると「with points」が仮定されます)。

```
with points   点で描画する
with lines    線で描画する
with boxes    棒グラフに描画する
```

with steps 階段状に描画する

with impulses インパルス(垂線)で描画する

with dots 小さな点で描画する(データ点が極めて多数あるときに有用)

なお、「with points」で描く際も、点種を指定するには、

```
gnuplot> plot "data.txt" using 1:2 title "graph1" with points 5 3
```

と、書かねばなりません。この場合、線種に相当する値(上では「5」)は無視されます。

今度は、「1列目のデータをx軸に、2列目のデータと3列目のデータをy軸にとったグラフ」を描きましょう。以下のようなコマンドで実行できます。

```
gnuplot> plot "data.txt" using 1:2 title "graph1" with lines,"data.txt"
using 1:3 title "graph2" with lines
```

2本のグラフが描けたでしょう。さらに、データファイルをちょっと書き換えて、

```
1.2  1.5    2.3
```

```
1.3  1.6    2.2
```

```
1.4  1.7    2.2
```

```
1.5  1.75   2.5
```

としてみましょ。2行目と3行目の間に改行をいれただけです。このデータを「linespoints」で同様にグラフ化してみましょ。2番目のデータと3番目のデータの間には、線が引かれていません。「lines」のときも同様です。

慣れてくると、長いコマンドをいちいち入力するのが面倒になるかもしれません。そんな場合には、省略形を用いると良いでしょう。例えば、

```
gnuplot> pl "data.txt" u 1:2 w lp
```

は、

```
gnuplot> plot "data.txt" using 1:2 with linespoints
```

と等価です。

描いたグラフを消去するときは、以下のコマンドで実行できます。

```
gnuplot> clear
```

第3章 グラフの装飾

最低限のグラフの描き方は上記で終わりです。ここでは、装飾の設定を手動で行います。

まず、描画範囲を手動で設定しておきましょう。グラフが描かれている状態で、以下のように入力します。

```
gnuplot> set xrange [1:1.6]
gnuplot> set yrange [1:3]
gnuplot> replot
```

これで、x軸の範囲が [1,1.6] に、y軸の範囲が [1,3] に設定されたグラフが描けます。最後の「replot」は、「直前に行った plot 命令を再実行する」というコマンドです。さらに、

```
gnuplot> set grid
gnuplot> replot
```

としてみましょ。目盛の位置に線が引かれます。線を消したければ、「grid」の代わりに「nogrid」として同様に設定します。

目盛の位置も手動で設定しましょう。

```
gnuplot> set xtics 1.1, 0.1, 1.3
gnuplot> replot
```

と実行してみましょ。これは、「x軸上で、1.1から1.3までを、0.1刻みで目盛を指定せよ」という意味です。結果として、「x=1.1,1.2,1.3」の3か所に目盛が刻んであるはずですが、y軸についても「ytics」として、同様の設定ができます。

これとは別に、目盛を一つ一つ設定する方法もあります。

```
gnuplot> set xtics (1.1, pi/2, 1.3)
gnuplot> replot
```

何がプロットされたかは、お分かりでしょう。ただしこれでは、円周率が有限桁の小数で表示されてしまいます。その場合、

```
gnuplot> set xtics (1.1, "PI/2" pi/2, 1.3)
gnuplot> replot
```

とすれば、良いでしょう。なお、ここでの円周率(「pi」)は、適当に打ち切られた値が入力されるようなので、気をつけてください。

数値の表示桁数まで設定したい場合は、以下のように設定します。

```
gnuplot> set format x "%7.3f"  
gnuplot> set xtics (1.1, pi/2, 1.3)  
gnuplot> replot
```

ここで、「"%7.3f"」は、「全体では7桁、小数点以下3桁」という形式です。「format x ""」は、目盛だけ表示され、数値は表示されません。デフォルト値(自動設定された値)に戻りたいときは、以下のように、指定無しで「xtics」を行います。

```
gnuplot> set xtics  
gnuplot> replot
```

グラフの装飾について、他のコマンドをまとめておきます。「x」を「y」に変えれば、y座標の設定になるのは、言うまでもないでしょう。

コマンドリスト

`set noxtics` 目盛を表示しない。

`set logscale x` 対数軸に設定する。

`set nologscale x` 上記コマンドのキャンセル。

`set xzeroaxis 1` $y=0$ の位置に、線を引く。

最後の数字(例では「1」)は、線の種類(第2章参照、例では実線になる)で、省略可。

`set title "example"` グラフにタイトルをつける。例では「example」。

`set xlabel "data 1"` x軸にタイトルをつける。例では「data1」。

`set size 0.721,1.0` グラフのサイズを変更する。例では、x軸方向に0.721倍、y軸方向に等倍する。

この例の場合、グラフがちょうど正方形になる。

`set size square` グラフがちょうど正方形になる。(古いバージョンのgnuplotでは、未サポート)

設定した環境パラメータを見たい場合は、「set」の代わりに「show」を使います。例えば、「show title」とすれば、現在設定されているタイトル(上の例では「example」)が表示されます。「show all」で全ての環境パラメータを表示できます。この場合、自動設定されている環境パラメータも全て表示されますので、(ここで説明していない環境パラメータも含めた)長いリストが表示されます。

第4章 様々なグラフの書き方

第2章で取り上げた以外にも、様々なグラフを描くことができます。ここでは、いくつかのグラフを紹介します。

4.1 エラーバーが付いたグラフを描く

エラーバーが付いたグラフを描くには、データファイルの形式を、行ごとに以下のように書かなければなりません。

(x座標) (主データ) (エラーバーの下限值) (エラーバーの上限値) 例えば、data2.txt として、以下のようなファイルを作成します。

```
1.0 2.0 1.9 2.1
1.1 2.2 1.5 2.3
1.2 2.1 2.0 2.2
1.3 1.9 1.8 2.2
```

gnuplot コマンドラインで、

```
gnuplot> plot "data2.txt" with errorbars
```

と入力すれば、エラーバー付きのグラフが描かれます。

4.2 2つのY座標をもつグラフを描く

左右に別々の y 座標を割り当てたグラフを描きます。y の代わりに y2 と指定します。例を示すので、それで理解できると思います。

```
gnuplot> set yrange [1:2]
gnuplot> set y2range [2:3]
gnuplot> set y2tics
gnuplot> set grid
gnuplot> plot "data.txt" using 1:2 axes x1y1,"data.txt" using 1:3 axes x1y2
```

ここで `set y2tics` を書かないと、 y_2 座標 (グラフの右側) の目盛には、数値が現れないようです。

同様に、2つの x 座標を (上下に) 持つこともできます。