

2012 年 07 月 06 日

計算機実習 IV (2012 年度)

第 12 回: gnuplot その 2

(<http://takeno.iee.niit.ac.jp/%7Eshige/math/lecture/comp4/comp4.html>)

目次

1	複数のグラフの重ね描き	1
2	gnuplot スクリプト	2
3	種々の設定 (set)	5
4	描画スタイル boxes	8
コラム:	gnuplot の高度な機能 その 1	10

1 複数のグラフの重ね描き

plot コマンドで複数のグラフを重ねて描画するには、範囲指定以外の <対象> とそれ以降のオプションを一つの単位として、それらを一つの plot コマンドの後ろにカンマ (,) 区切りで並べて書く。例えば、

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

は、線種 1 で $\sin x$ のグラフを、線種 2 で $\cos x$ のグラフをそれぞれ折れ線描画したものを重ねて表示する。

長い 1 行のコマンドを書く場合、対話ウィンドウでは行末に ¥ を書いて Enter を押すと、次の行は > から始まり、前の行が続いている状態となる。例えば、

```
gnuplot> plot [3:5] 2*x w l lw 2 not, ¥  
> "datafile" w lp lt 3
```

は、 $2x$ のグラフを線種 1、線幅 2、凡例なしで折れ線で描き、ファイル datafile のデータを折れ線と点 (記号) の線種 3 で表示する。

複数グラフの重ね描きに関しては、以下の点に注意せよ。

- 線種 (linetype) は、特に指定しなければ、グラフを指定した順番に自動的に 1 番の線種から順に使われる。
- 範囲指定部分はすべてのグラフに共通なので、個別に指定することはできない¹。
- 一つのデータファイルから 2 つ以上のグラフを生成して重ね描きする場合は、2 つ目以降のファイル名部分は空文字 ("") としてもよい。
- 関数グラフでは、同じ関数でも 2 つ目以降の関数は省略できない。

3 つ目のものは、例えば data の各行に 3 列のデータが含まれている場合、

```
plot "data" using 1:2 w l, "" using 1:3 w l
```

は、1 列目を x 座標とみなし、2 列目を y 座標としたグラフを線種 1 で、3 列目を y 座標としたグラフを線種 2 で重ね描きする。

課題 12-1. x^2 , $(x+1)^2$, $(x+2)^2$ の 3 つのグラフを、線幅がそれぞれ 2, 1, 3 で、色はそれぞれ赤、水色、青の折れ線で重ねて描く plot コマンドを示せ。

課題 12-2. $\sin x$ のグラフを線幅 1 の赤の折れ線で描き、さらに青の鉛直線の同じグラフと、点の大きさ 2 の黄緑色の三角形の点 (中が塗り潰された三角形) の同じグラフを重ねて描く plot コマンドを示せ。

課題 12-3. データファイル data の 1 列目が x 座標で 2 列目の絶対値が y 座標であるグラフを赤の折れ線で描き、同じ data の 1 列目が x 座標で 3 列目の 2 乗の値が y 座標であるグラフを青く塗り潰された三角形の点で重ねて描く plot コマンドを示せ。

2 gnuplot スクリプト

gnuplot の対話ウィンドウに与える命令は、ファイルに保存してそれを gnuplot に一度に処理させることができる。gnuplot 命令を書いたファイルを gnuplot スクリプトと呼ぶ。

gnuplot スクリプトを gnuplot に処理させる方法は何通りがある。例えばそのスクリプトファイル名を test.gp とする。

¹個別に別の軸を割り当てることはできるが、それは後で説明する。

1. gnuplot の対話ウィンドウで gnuplot スクリプトを読み込む命令として、load と call があり、それらの後ろにスクリプトファイル名を引用符で囲んで指定する²。

```
gnuplot> load "test.gp"
```

例えば test.gp に「plot sin(x)」と書いてあれば、それを対話ウィンドウに書いたのと同じことになる。

2. コマンドプロンプトで wgnuplot の起動時にスクリプトを指定して実行する (引用符は不要)。

```
Z:> wgnuplot -p test.gp
```

この 2. の方法での -p (または -persist) オプションは、wgnuplot が終了してもグラフウィンドウを残すためのオプションで、このオプションをつけず、スクリプト内に pause 命令もない場合は、グラフウィンドウは一瞬表示されるがすぐに閉じられてしまう。また、2. の場合はバッチ処理されるので、対話ウィンドウは表示されない (非対話型モード)。

- pause <秒数> {"<文字列>"}
 <秒数> が 0 以上 (実数も可) の場合は、指定秒の間、処理をそこで停止する。<秒数> が負の値の場合は、一時停止中を示すウィンドウを表示し、"<文字列>" が指定されていればそれをそのウィンドウに表示して Enter キーの入力かウィンドウボタンがクリックされるまで待つ。
- pause mouse {"<文字列>"}
 「pause -1 {"<文字列>"}」とほぼ同じであるが、一時停止ウィンドウは表示せず、グラフウィンドウ上でマウスクリックされるまで待つ。

例えば、test.gp が

```
pause -1 "まずはサインのグラフ"  
plot sin(x)  
pause -1 "次はコサインのグラフ"  
plot cos(x) w l lt 2  
pause -1 "終了"
```

であれば、 $\sin x$ と $\cos x$ のグラフを Enter キーで切り替えて表示する。この場合は pause があるので、2. の方法でも -p は不要。

gnuplot スクリプトの書き方について一般的な注意を上げる。


²load と call では、後者にはオプションをつけてそれをスクリプトファイル内で参照できるという違いがあるが、詳細は省略する。

- 空行は無視される。
- スペースやタブが複数並んでいても一つの空白とみなされる。
- ‘#’ から行末まではコメントとみなされる (AWK スクリプトと同じ)。
- 行末に ¥ を置けば、次の行まで 1 行の命令を続けられる (AWK スクリプトと同じ)。
- gnuplot はスクリプトの内容を 1 行ずつ実行するが、セミコロン (;) で区切ることで 1 行に複数の命令を書ける (AWK スクリプトと同じ)。逆に通常の行末にはセミコロンは不要。

例えば以下は、0.5 秒おきにグラフが切り替わる簡単なグラフアニメーションになる。

```
# 簡単なグラフアニメーション

plot sin(x) ; pause 0.5
plot sin(x+0.1) ; pause 0.5
plot sin(x+0.2) ; pause 0.5
plot ¥
    sin(x+0.3) # これはやや無理矢理
```

なお、拡張子が .gp か .plt のファイルであれば、実習室の otbedit は自動的に gnuplot モードになるが、otbedit の上の方にある  というアイコンで強制的に gnuplot モードにすることもできる。

また、2. の方法では複数のスクリプトを指定することもできる。

課題 12-4. plot 命令一つだけの 1 行のスクリプトを作成し、上の 1., 2. の方法を試してみよ。また、2. の -p をつけなかった場合どうなるかも確認せよ。

課題 12-5. 上の pause -1 を含むサンプルと、その pause -1 を pause mouse に変更したスクリプトの両者を実行し、それらの違いを確認せよ。

課題 12-6. pause を用いて $y = x^2$ のグラフが右上に少しずつ動いて消えていくようなグラフアニメーションを描画する gnuplot スクリプト kadai12-6.gp を作成せよ (範囲指定を使わないとうまく消えてくれない)。

課題 12-7. さらに、 $y = x^2$ のグラフが動くときに、過去のグラフも残しながら動く (段々描画するグラフが増える) ようなグラフアニメーションを描画する gnuplot スクリプト `kadai12-7.gp` を作成せよ。なおグラフの色は、動くグラフは青、過去のグラフは緑とせよ。

スクリプトがうまく動くかどうか、どこまで正しく動いているかを確認するには、`print` コマンドを使うとよい。`print` コマンドは式の値も表示できるので、簡単な関数電卓としても利用できる。

- `print "<文字列>"` : <文字列> を対話画面に表示する
- `print <式>` : <式> の値を対話画面に表示する

課題 12-8. `pause` と `print` で、 $x = 0^\circ$ から $x = 90^\circ$ までの 10° 毎の角度に対する $\sin x$ の値を、0.5 秒おきに表示する gnuplot スクリプト `kadai12-8.gp` を作成せよ。

課題 12-9. 別のスクリプトファイルを `load` するようなスクリプトを作成し、1. の形式でそれを `load` してどのように動作するかを `print` コマンドを用いて確認せよ。

3 種々の設定 (set)

set/unset/show

グラフに関する種々の設定は、`plot` コマンドでも多少は行えるが、多くは `set/unset` コマンドで行う。`set/unset` コマンドの形式は、基本的には以下の通り。

- `set <設定名> {<オプション>}` : <設定名> の設定を行う。
- `unset <設定名>` : <設定名> の設定をデフォルトに戻す。
- `show <設定名>` : <設定名> の現在の設定を表示する。

`set` で設定できる設定項目は非常に多いが、表 1 にその主なものをあげる。なお、`xlabel`, `xtics`, `xrange`, `xzeroaxis` などの x 軸用の設定項目は、 y 軸に対してもそれぞれに対応するもの (`ylabel`, `ytics`, `yrange`, `yzeroaxis`) がある³。同じく、`tmargin` (上) には同様の `bmarg` (下)、`lmarg` (左)、`rmarg` (右) がある。よく使うものをいくつか説明する (次回以降も適宜説明する)。

³後で説明するが、gnuplot には他にも軸があり、それらにも同様の設定項目がある。

設定名	意味	設定名	意味
title	グラフタイトル	boxwidth	箱 (棒) の幅
xlabel	x 軸のラベル	format	軸刻みの見出しの書式
xtics	x 軸の刻み	samples	関数のサンプリング数
xrange	x の範囲指定	arrow	任意箇所に線分、矢印を
xzeroaxis	x 軸表示	label	任意箇所に文字列を
key	凡例の設定	tmargin	上の余白の設定
grid	格子線	output	出力ファイル指定
logscale	対数軸	terminal	出力形式 (省略形 term)
pointsize	点のサイズ	datafile	データ用の設定
border	グラフの枠線	size	グラフの伸縮

表 1: set による主な設定項目名

set title/xlabel/ylabel

set title, set xlabel, set ylabel のオプションはほぼ同じで、

- set title "<文字列>" {font "<フォント名>,<フォントサイズ>"}

の形式で使用する。<文字列> には日本語も使え⁴、フォントもオプションで指定できる。2 重引用符 (" ") で文字列を囲む場合は、C 言語同様、その中に $\backslash n$ で改行文字を書くこともできる。例:

```
set xlabel "日付" font "MS 明朝,14"
set ylabel "人口 [人]"
set title "新潟県の人口の推移" font ",20"
plot "data" w l
```

この 3 行目のように、フォント名を省略してフォント指定した場合は、その設定項目のフォントは変更せず、サイズのみ変更する。よってこの場合は、

- x 軸のラベルのフォントは「MS 明朝」の 14 ポイント
- y 軸のラベルのフォントはデフォルトフォントのデフォルトサイズ
- グラフタイトルのフォントはデフォルトフォントの 20 ポイント

となる。

⁴正確に言えば、出力形式が日本語をサポートしていて、適切な日本語フォントが指定されていれば日本語が表示できる、ということ。wgnuplot の画面出力ではデフォルトで日本語が使えるようになっている (はず)。

なお、「MS 明朝」「MS ゴシック」は MS-Windows 付属の典型的な日本語フォントで、この名前の通り (あるいは英語名で「MS Mincho」「MS Gothic」) に指定しなければならない。日本語フォント名の場合は「M」「S」は全角、その次は半角スペース (英語名の方は全部半角)。

set コマンドの設定は、その後の plot コマンドに反映されるので、適用したい plot コマンドよりも前に行わなければならない。また、その設定は各項目に対する unset コマンド (または単一の reset コマンド) を実行しない限り有効で、その後の plot コマンドにずっと影響し続ける。

- reset : set コマンドで設定する (ほぼ) すべてのオプションの値をデフォルトの値に戻す。

また、直前の plot コマンドと同じものを再び実行したい場合、以下のコマンドも用意されている。

- replot : 直前に行った plot コマンドと同じものを実行する (オプションは不要)。

例えば、set コマンドの設定を変えたときにグラフがどう変化するかを見る場合は replot を使うと便利である。例:

```
gnuplot> plot sin(x) w l lt 3
gnuplot> set title "正弦"
gnuplot> replot
```

課題 12-10. set title, set xlabel, set ylabel に日本語文字列を指定して、それらが正しく表示されるかどうか確認せよ。

課題 12-11. set title のフォントを明朝体、ゴシック体などと切り替えて、フォントサイズも変更して表示させ、その場合グラフ領域の大きさがどう変わるか確認せよ。

課題 12-12. plot コマンドの title オプションと set title との違いを確認せよ。

課題 12-13. タイトルと x 軸のラベルを設定してグラフを表示し、その後 unset title して replot するとどう変わるか確認せよ。

課題 12-14. x 軸ラベルを「身長(cm)」、 y 軸ラベルを「体重(kg)」、グラフタイトルを「1年2組の身長と体重」とした、データファイル data12.dat の散布図 (凡例はなし、赤い塗りつぶしの丸) を描く plot コマンドを示せ。

xrange/yrange

set xrange, set yrange の設定は、基本的には plot コマンドの範囲指定と同じであるが、set xrange と plot コマンドの両方で範囲指定をした場合は、plot コマンドの方が優先する。なお、範囲指定は、大きい数値を先に書くと軸が反転したグラフになる。

set logscale を用いれば、特定の軸を対数軸にできる。

- set logscale <軸名> {<底>}
指定した軸を対数軸にする (デフォルトはいずれも線形軸)。<軸名> には、x や y (片対数軸)、および xy (両対数軸) などが指定できる (引用符は不要)。<底> は指定しなければ常用対数 (底 = 10) となる。

課題 12-15. set xrange と plot コマンドの範囲指定オプションの両方を指定して、どちらが優先されるか確認するような gnuplot スクリプト kadai12-15.gp を作成せよ。

課題 12-16. $y = 2^x$ と $y = 4x^3$ のグラフを、 x 軸のみが対数軸、 y 軸のみが対数軸、 x, y の両軸が対数軸、のそれぞれでどうなるかを確認するような gnuplot スクリプト kadai12-16.gp を作成せよ。

4 描画スタイル boxes

前回、gnuplot の基本的な描画スタイルを紹介したが、それ以外によく用いられる描画スタイルとして、ここでは boxes を紹介する。

boxes (省略名はなし) は、棒グラフ (箱グラフ) を描画し、関数グラフでもデータグラフでも使用できる。必要なデータ列は原則 2 列か 3 列で、それぞれ

- 2 列のデータ: $x \ y$
- 3 列のデータ: $x \ y \ width$

のようにみなされる。ここで、 x は棒の中心の x 座標、 y は棒の高さ、 $width$ は棒の横幅である。棒の幅のデフォルト値は set boxwidth でも設定できるが、デフォルトでは隣接する棒が丁度接する幅で描かれる。

with boxes には、棒の塗り潰し用の追加オプションがあるが、そのデフォルト値は set style fill でも設定できる。詳細は省略するが、典型的なものをいくつか紹介する。

- w boxes fs s 0.5 lt 3
棒を 0.5 の密度 (s = solid)、線種 3 (lt) で塗り潰す (fs = fillstyle)。密度は 0 から 1 までの値。
- w boxes fs pat 4 lt 5
棒を線種 5 で 4 番のパターン (pat = pattern) で埋める。各パターンがどのようなものであるかは、test コマンドで見ることができる。
- w boxes fs e lt 7
なにも塗られない (e = empty)、線種 7 の枠線だけの棒にする。

複数のデータを同時に棒グラフで表現する場合は、「横並び型」と「積み上げ型」の 2 通りあり、gnuplot ではいずれも histograms 描画スタイルで描画できるが、横並び型棒グラフの簡単なものなら with boxes でも、 x 座標のずらしと棒幅の調整で実現できる。例えば、1 列目のデータの間隔が 1.0 であるとき、

```
set boxwidth 0.2
plot "data" using 1:2 w boxes fs s 0.5 lt 1,¥
"" using ($1+0.2):3 w boxes fs s 0.5 lt 3
```

とすれば 2 列目のデータは $x = 1, 2, 3, \dots$ の場所に赤の棒グラフで、3 列目のデータは $x = 1.2, 2.2, 3.2, \dots$ の場所に青の棒グラフで表示される。なお、このようなグラフでは x 軸の刻みのラベルも変更したいことがあるだろうが、その方法については次回紹介する。

課題 12-17. 1 列目が 1,2,3,4,5 である 5 行 3 列のデータ data5.dat を手動で適当に作成して、1 列目を x 座標とし、2 列目のデータを赤の塗り潰しの棒グラフ、3 列目のデータを青の折れ線で描画する gnuplot スクリプト kadai12-17.gp を作成せよ。

課題 12-18. データ data5.dat のデータを、デフォルトの boxwidth のまま上のような x 座標のずらしをせずに 2 列目と 3 列目をそれぞれ with boxes の棒グラフで重ね描きするとどのように表示されるか確認せよ。

課題 12-19. データ data5.dat のデータの 2 列目、3 列目を、set boxwidth と using を使う方法で横並び型の 2 種類の棒グラフ (それぞれ赤の 3 番

のパターンと青の 4 番のパターン) で描画するような gnuplot スクリプト `kadai12-19.gp` を作成せよ。

コラム: gnuplot の高度な機能 その 1

現在の gnuplot はかなり高機能であり、本講義では紹介しきれないほど便利な機能がたくさんある。そのうちの幾つかを紹介しよう。

- 3 次元グラフ

本講義では 2 次元グラフ (1 変数関数 $y = f(x)$ のグラフ、または 2 次元データ) しか扱わないが、gnuplot には 2 変数関数 $z = f(x, y)$ のグラフ、または 3 次元データを描画するための `splot` という命令がある。

3 次元グラフは、もちろんある方向から見た形の 2 次元射影図として描くが、その視線方向は `set view` でコマンド駆動的に変更できるだけでなく、マウストラッグでグラフを直接回す、といったことも行えるようになっている。

一般には 2 変数関数の 3 次元グラフは曲面になるが、そのグラフの等高線描画 (`set contour`)、高さを色で表現するグラフ (`set pm3d`)、手前の曲面に隠れるべき奥の部分を消す隠線処理 (`set hidden3d`) などの機能もある。高さを色で表現して、それらを真下の xy 平面に射影した 2 次元表示 (色地図) も行える (`set view map`)。

- 回帰曲線 (非線形最小二乗法)

実験データなどを解析する際に、最小二乗法を用いて回帰直線を求めることがあるが、複雑な現象のデータでは直線的ではなく曲線的な傾向を示す場合もある。そのような場合に、理論などから予想される形の「曲線」に当てはめる「非線形最小二乗法」を用いることも多い。

gnuplot は、実験科学者が主な開発者であることもあり早くからその機能を持っていて、非線形最小二乗法の一つである Marquardt–Levenberg アルゴリズムを実装し、簡便に回帰曲線が求められるようになっている。

その計算で得た係数の値や計算の途中経過を見ることができ、結果のグラフも容易に描けるようになっていて、主にこの機能のために gnuplot を愛用するという研究者も少なくないようである。