

2012 年 07 月 13 日

計算機実習 IV (2012 年度)

第 13 回: gnuplot その 3

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

目次

1	種々の設定 (set) その 2	1
2	変数と関数定義	3
3	描画スタイル histograms	7
4	第 2 軸の利用	8
	コラム: gnuplot の高度な機能 その 2	9

1 種々の設定 (set) その 2

set grid/xzeroaxis/border

軸や枠線の見た目の変更は、set grid, set xzeroaxis, set border などで行う。主なオプション、使用法は以下の通り。

- set grid {<刻み指定>} {lt <n>} {lw <x>}
軸の刻みに合わせた格子線をつける (これを設定しないとつけない)。
<刻み指定> は格子線をどの軸のどの刻みにつけるかの指定。デフォルトは xtics (x 軸の主刻み) と ytics (y 軸の主刻み) の両方。
- set xzeroaxis {lt <n>} {lw <x>}
 x 軸の軸線 ($y = 0$) を描く (これを設定しないと描かない)。同様に set yzeroaxis は y 軸の軸線 ($x = 0$) を描く。さらに、set zeroaxis は、 x, y 両方の軸線を一度に設定する。
- set border <整数値> {lt <n>} {lw <x>}
グラフ領域の枠線の設定を行う。<整数値> には各枠線に対応する整数

値のビット和の値を指定するが、枠線の下、左、上、右がそれぞれ 0, 1, 2, 3 ビット (整数値としては 1, 2, 4, 8) に対応する。デフォルトは 15。

いずれも `lt`, `lw` で線種、線幅を変更できる。デフォルトの線幅 (`lw`) はいずれも 1.0 だが、デフォルトの線種 (`lt`) は、`grid` と `xzeroaxis` は 0 (点線の黒)、`border` は -1 (実線の黒) になっている。

例えば以下は、 x 軸の刻みに合う格子線を描き、 x, y 軸の軸線を枠線と同じ線種 (-1) で描き、左と下の枠線だけ ($2+1=3$) を描く。

```
set grid xtics
set zeroaxis lt -1
set border 3
plot sin(x)
```

課題 13-1. $y = \sin x$ のグラフを描き、次に縦横の格子線をつけたもの、横だけの緑の格子線をつけたもの、縦だけの赤の格子線をつけたものを `pause` で切り替えて表示するような `gnuplot` スクリプト `kadai13-1.gp` を作成せよ。

課題 13-2. グラフの枠線全部を消し、 x 軸と y 軸を太い緑の線で描き、 $y = \cos x$ のグラフを描く `gnuplot` スクリプト `kadai13-2.gp` を作成せよ。

set xtics/ytics

軸の刻み (目盛) と、各刻みにつけるラベルの設定は `set xtics`, `set ytics` で行う。`set xtics`, `set ytics` の設定は共通で、主に以下の 3 通りの書き方が用いられる。

- `set xtics <増分値>` : 刻みの間隔 (増分) のみを指定する
- `set xtics <開始値>, <増分値> {, <終了値>}`
刻みの開始位置、間隔、終了位置 (省略すれば端まで) を指定する
- `set xtics ("<文字列>" <位置>, "<文字列>" <位置>, ...)`
明示的な刻み、ラベル指定

`set xtics` には多くのオプションがあるが、よく使うものを以下にあげる。

- `mirror/nomirror` : 対辺に刻みをコピーする/しない (デフォルトは `mirror`)

- `font "<フォント名>,<サイズ>"` : 刻みラベル文字列のフォント指定

```
set border 3 # 下と左のみ
set xtics nomirror; set ytics nomirror # 刻みコピーなし
set xtics pi # 毎の刻み
plot cos(x)
```

x 軸の刻み、ラベルはデフォルトでは ON になっているが、`unset` で OFF にすることができる。

- `unset xtics` : x 軸の刻み、ラベルをすべて OFF にする

課題 13-3. $y = \sin x$ のグラフの横軸の刻み間隔を 1 に変えてみよ。また、デフォルトの場合と 1 間隔の場合で格子線のつきかたがどう変わるか確認せよ。

課題 13-4. `set xtics` を用いて x 軸の刻みラベルのフォントを 20 ポイントに変えてみよ。

課題 13-5. x の値 (1 列目) が 1,2,3 だけの 3 行のデータファイル `data1.dat` に対し、 x 軸の範囲を $0 \leq x \leq 4$ とし、 $x = 1$ には「太郎」、 $x = 2$ には「二郎」、 $x = 3$ には「花子」という軸ラベルをつけて、2 列目の値を高さとする棒グラフ (青の塗り潰し) を描く gnuplot スクリプト `kadai13-5.gp` を作成せよ。

2 変数と関数定義

gnuplot でも変数は利用できる。

- 変数に宣言は不要。
- 変数値は数値か文字列であり、数値は整数、実数、複素数が使用できる (複素数に関する説明は本講義では省略する)。
- 文字列は、二重引用符 (") で囲んでも一重引用符 (') で囲んでもよいが、どちらで囲むかで意味に違いがある場合もある。
- 文字列用の演算子、関数も用意されている

- 数値文字列から数値への変換は自動的に行なわれるが、数値から文字列への変換は自動的には行われない (AWK とは違う)

文字列用の主な演算子は以下の通り:

- `A . B` : 文字列 A と文字列 B の連結文字列を示す
- `A[n:m]` : 文字列 A の n 番目から m 番目までの部分文字列を示す (先頭文字は 1 番目)
- `A eq B / A ne B` : 文字列 A と文字列 B が等しければ真/等しくなければ真

文字列用の関数も多く用意されているが、`sprintf()` 以外は使うことはあまりないだろう。

- `sprintf(...)` : C 言語の `printf()` と同じ書式で、画面に表示する代わりに結果の文字列を返す

変数や式の結果は、前に紹介した `print` コマンドで知ることができる。

- `print <式> {, <式>, ...}` : 各 <式> の値を対話画面に表示する

例:

```
jfont1 = "MS Gothic" # 文字列変数
x1 = 1950; x2 = 2010 # 数値変数
s = sprintf('新潟県の人口 (%d年~%d年)', x1, x2)
set xrange [x1-5 : x2+5]
set title s font jfont1 . ",20"
print s[0:6]
.....
```

gnuplot では全角文字 1 文字を 2 の長さで勘定するので、上の例の `print` 文では「新潟県」が表示される¹。

なお、`sprintf()` では二重引用符 (" ") でなく一重引用符 (' ') で囲んでいるが、これはこの文字列の日本語文字コード (シフト JIS) の問題を回避するため、実際 " " で囲むとうまくいかない²。

課題 13-6. 「`x="1.0"`」「`y=2.0`」として、「`x+2.0`」「`y+2.0`」の結果、「`x . "05"`」「`y . "05"`」の結果を `print` コマンドで確認せよ。

¹講義で使用している `gawk` の方は 1 と勘定するので注意が必要。

²より詳しく言えば、この文字列中の全角文字「~」の 2 byte 目が `0x60` (=「'」) になっていて、「'」(逆クォート) は `gnuplot` の " " 中では特別な意味を持つからである。詳しくは `help quotes` を参照。

課題 13-7. 文字列変数 sei と mei に自分の姓と名 (日本語) を代入して、文字列を連結した結果と、名の 1 文字目だけを取り出した文字列を print コマンドで表示するような gnuplot スクリプト kadai13-7.gp を作成せよ。

gnuplot では自分で関数を定義することもできる。1 変数関数だけではなく、多変数の関数を作ることもできるし、変数以外のパラメータを持つ関数も許されている。関数定義時には、変数には任意の変数名が使えるが、plot するときは変数を x にする必要がある³。例:

```
f(t,a,b) = a*sin(b*t+c) # 定義時は t でよい
c = 1.5
plot f(x, 0.5, pi) # plot 時は x でないと不可
```

これは、結果として $y = 0.5 \sin(\pi x + 1.5)$ のグラフを描く。関数の中にあるパラメータ (今の場合は a, b, c) の値は、plot のときに決定していればよく、さらにこの c のように、すべてのパラメータを関数の引数とする必要もない。

```
f(t,a) = 2.0*sin(pi*t+a)
plot f(x, 0.0), f(x, 0.1), f(x, 0.3), f(x, 0.4)
```

この例のように、多変数関数はパラメータを変えながらグラフを重ね描きするのに便利である。

課題 13-8. 正規分布の密度関数 $f(x, p) = e^{-x^2/(2p)} / \sqrt{2p\pi}$ を定義し、 $p = 0.5, p = 1, p = 2$ に対する 3 本のグラフを重ね描きする gnuplot スクリプト kadai13-8.gp を作成せよ。

課題 13-9. $f(x) = \sin x, g(u) = 4u^3 - 3u, h(u) = 2u^2 - 1$ を定義し、 $y = g(f(x))$ と $y = h(f(x))$ のグラフを折れ線で重ね描きする gnuplot スクリプト kadai13-9.gp を作成せよ。

課題 13-10. $f(t) = \sin \pi t, g(t) = \cos \pi t$ と定義し、1 列目に -1.0 から 1.0 まで 0.1 刻みのデータが順に 21 行並ぶデータ data に対し、「plot "data" u (f(\$1)):(g(\$1)) w l」とするとどうなるか予想し、実際に実験してみよ。

三項演算子を用いれば、場合分け形の関数も定義できる。

³それを変更する set dummy なるものもあるが、説明は省略する。

- <条件> ? <式 1> : <式 2>
 <条件> が真のときは <式 1> を、偽のときは <式 2> を返す。

例えば、

$$f(x) = \begin{cases} 1 - x & (x \leq 1 \text{ のとき}) \\ 2x + 1 & (x > 1 \text{ のとき}) \end{cases}$$

という関数は、三項演算子を用いて

$$f(x) = (x \leq 1) ? (1 - x) : (2x + 1)$$

と書ける。特に、これはグラフの一部分を消したい場合によく使われ、

$$f(x) = x \geq 0 ? \sin(x) : 1/0$$

とすれば、これは $x \geq 0$ では $\sin x$ を意味するが、 $x < 0$ では無効値 $1/0$ なので、 $x < 0$ の部分はグラフ化されない。つまり、 $x \geq 0$ でのみ定義されているようなグラフになるわけである。

同様にデータファイルの `using` 指定内で 3 項演算子を使えば、データファイルのグラフの一部分をカットすることもできる:

```
# y 座標が 10 を越えている部分だけ表示を消す
plot "data" using 1:($2>10? 1/0: $2) w l
```

課題 13-11. 以下のような関数を定義してそのグラフを表示する gnuplot スクリプト `kadai13-11.gp` を作成せよ。

$$f(x) = \begin{cases} (x + 1)^2 & (x \leq -1 \text{ のとき}) \\ 0 & (-1 < x \leq 2 \text{ のとき}) \\ (x - 2)^2 & (x > 2 \text{ のとき}) \end{cases}$$

課題 13-12. $y = \sin x$ の値が 0.3 を越えている部分をカットするような関数 $f(x)$ を定義してそのグラフを表示する gnuplot スクリプト `kadai13-12.gp` を作成せよ。

課題 13-13. データファイル `data` の 1 列目を x 座標、2 列目を y 座標とするが、3 列目の値が 1 に等しい行のデータだけを点 (記号) で描画するような gnuplot スクリプト `kadai13-13.gp` を作成せよ。

3 描画スタイル histograms

簡単な棒グラフは `set boxwidth` と `with boxes` でも描けたが、複数の棒グラフを重ね描きするには `with histograms` (省略名 `w hist`) を用いるのがよい。ここではその使い方を簡単に紹介する。

`with histograms` を使う際には、次の点に注意する。

- 横軸値の指定はできず、データは 1 つのグラフに 1 列だけ指定する
- グラフは自動的に $x = 0, 1, 2, \dots$ の場所に描かれるので、`set xtics` の明示的なラベル指定で適切なラベルをつける必要がある
- `set yrange` の指定をしないとデータの最小値が `yrange` の最小値となり、一番低い棒グラフが表示されなくなる
- `set style fill` (棒の塗り方), `set style histogram` (並べ方) を先に設定してから `plot` する

`set style fill`, `set style histogram` に指定するオプションは以下の通り。

set style fill		
オプション名	省略名	意味
<code>empty</code>	<code>e</code>	塗らない
<code>solid <x></code>	<code>s</code>	密度 x で塗る ($0.0 \leq x \leq 1.0$)
<code>pattern</code>	<code>p</code>	パターン (模様) で埋める
set style histogram (省略名 set style hist)		
オプション名	省略名	意味
<code>clustered</code>	<code>clust</code>	横並び型
<code>rowstacked</code>	<code>rows</code>	積み上げ型 (1 つのグラフを横に)
<code>columnstacked</code>	<code>columns</code>	積み上げ型 (1 つのグラフを縦に)

例えば、データ `data` が

```
#番号 国語 算数
2100 50 70
2200 80 30
2300 60 80
```

であるとき、横並び型の棒グラフを描くには以下のようにする。

```
set yrange [0:100]
set xtics ( "2100" 0, "2200" 1, "2300" 2 )
set style fill s 0.5 bo lt -1 # 境界は黒
set style hist clust
plot "data" u 2 t "国語" w hist, "" u 3 t "算数" w hist
```

課題 13-14. 上の例のデータ `data` と同じものをエディタで作成し、`rows` の積み上げ型に変えて描画する `gnuplot` スクリプト `kadai13-14.gp` を作成せよ (適切に表示されるように `set yrange`, `set xtics`, `plot` の `title` オプションも必要ならば変更せよ)。

課題 13-15. 上の例を、`columns` の積み上げ型に変えて描画する `gnuplot` スクリプト `kadai13-15.gp` を作成せよ (適切に表示されるように `set yrange`, `set xtics`, `plot` の `title` オプションも必要ならば変更せよ)。

4 第 2 軸の利用

`gnuplot` の 2 次元グラフでは、横軸、縦軸にそれぞれ 2 軸ずつ用意されていて、通常の x 軸 (下軸)、 y 軸 (左軸) の他に x_2 軸 (上軸)、 y_2 軸 (右軸) があり、例えば単位の違う 2 つのグラフを重ね描きできる。これらの軸に関する設定は、通常の軸と同様に `set x2range`, `set x2tics`, `set x2label` などのコマンドで行う。

グラフをどの軸に合わせて描画させるかは、`plot` コマンドの `axes` オプションで指定する。これは関数描画でもデータ描画でも利用できる。例:

```
plot "data" u 1:2 w l , "" u 1:3 axes x1y2 w l
```

これは、各行 3 列のデータが並ぶデータファイル `data` に対して、1 列目を x 座標とし、2 列目は y 軸 (左軸) に沿って線種 1 の、3 列目は y_2 軸 (右軸) に沿って線種 2 の 2 つのグラフを描画することになる。

`axes` オプションで指定できる軸名は、`x1y1`, `x1y2`, `x2y1`, `x2y2` の 4 通りで、1 軸 (`x1y1`) は通常の軸を表す。

2 軸を使用する場合は、1 軸に対する目盛り (`tics`) の `nomirror` オプションを使用して鏡映の目盛りを消し、また、2 軸の目盛り (`tics`) を ON にする必要がある (デフォルトは 1 軸の刻みが `mirror` され、2 軸の刻みは OFF になっている)。

例:

```
set xlabel "[年]"
set ylabel "身長 [cm]"; set y2label "体重 [kg]"
set ytics nomirror ; set y2tics # y2tics を ON に
set xrange [10:20]
set yrange [100:200] ; set y2range [30:90]
plot "data" u 1:2 w lp, "" u 1:3 axes x1y2 w i lt 5
# 2 列目を折れ線 + 点で、3 列目を鉛直線で描画
```

課題 13-16. 上の例のグラフをまずはそのまま表示し、`pause` で y_1 軸と y_2 軸を入れ換え、身長は折れ線で、体重は棒グラフ (with boxes) で表示するものに切り替えるような gnuplot スクリプト `kadai13-16.gp` を作成せよ。

課題 13-17. 3 列のデータが並ぶ `data` で、1 列目が高度 [km] ($0 \sim 100$)、2 列目が温度 [K] ($180 \sim 300$)、3 列目が気圧 [hPa] ($3.0 \times 10^{-4} \sim 1.2 \times 10^3$) であるとき、横軸を高度とし、温度の値を右の縦軸に従って折れ線で、気圧の値を左の縦軸に従って鉛直線で表示する gnuplot スクリプト `kadai13-17.gp` を作成せよ。なお、 y 軸は気圧の軸を対数軸とし、軸の見出しも適当なものをつけること。

課題 13-18. 1 列目が時刻 [年] ($1920 \sim 2010$ 、5 年おき)、2 列目が日本の人口 [万人] ($4000 \sim 15000$) であるデータ `data` に対し、横軸は時刻、縦軸は人口で、下の軸は 1920 年から 1950 年まで、上の軸は 1980 年から 2010 年まで、とした 2 本の折れ線グラフを描く gnuplot スクリプト `kadai13-18.gp` を作成せよ。

コラム: gnuplot の高度な機能 その 2

前回に引き続き、gnuplot の高度な機能をいくつか紹介する。

- 構文

gnuplot 4.4 から、`set` や `plot` コマンドに `for` という繰り返し機能 (iteration) が実装され、複数の規則的な設定やグラフの重ね描きが容易に行えるようになり、さらに gnuplot 4.6 からは `if/else/while/do` といった基本的な構文、および中括弧 (`{ }`) による複数のコマンドのブロック化も可能になり、スクリプト言語らしくなっている。

- 日時変換

グラフのデータの横座標は、時刻や日付であることも多いが、gnuplot には早くからそのようなデータを処理する機能が備わっている。

日付や時刻は色々な書き方があるが、読み込む際にその日時データがどのような書式で書かれているかを指定 (`set timefmt`) することで様々な形式に対応でき、軸の刻みのラベルにそれらを出力するときも、データの形式とは別に書式を設定できるようになっている (`set xdata time, set format x`)。

- 軸のラベルや刻みラベルの表現

軸の単位が日付などの場合、軸の刻みのラベルが長くなることもあるが、その場合はラベル文字列を回転して重ならないようにしたり (`set xtics rotate`)、刻みから少しずらしたり (`set xtics offset`) することもできる。デフォルトでは刻みは枠の内側につくが、それを外向きにすることもできるし (`set xtics out`)、枠ではなく `zeroaxis` の軸線につけることもできる (`set xtics axis`)。デフォルトの刻みに対して、いくつかを手動で追加することも可能になっている (`set xtics add`)。

- 拡張文字列処理 (enhanced) モード

科学技術分野のグラフでは、タイトルなどにギリシャ文字や上付き文字 (累乗等)、下付き文字 (添字等)、記号などを用いることも多い。それらは、enhanced モードを使い、特別な書式を用いることで表示できる。