

2012 年 06 月 29 日

計算機実習 IV (2012 年度)

第 11 回: gnuplot その 1

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

目次

1	gnuplot とは	1
2	plot コマンドのオプション	4
3	関数描画	6
4	データ描画	8
	コラム: gnuplot の特徴	10

1 gnuplot とは

gnuplot (グニュープロット、ニュープロット) は、コマンド駆動形式のグラフの描画、出力を行うフリーソフトであり、次のような特徴を持っている。

- Unix, MS-Windows, MacOS など、多くの環境で動作する。
- 対話的にコマンドを入力して実行することもできるが、コマンド列をスクリプトファイルに書いて、それを指定して実行することもできる。
- コマンドライン編集にヒストリ機能もついている。
- 非常に多くの出力形式 (画面、画像、プリンタ専用形式) に対応している
- 最新版 (4.6 以降) では、構文機能も実装されている。
- C や AWK プログラム内部から gnuplot を実行し、パイプや `system()` を使ってグラフを描かせることも可能。
- データだけでなく、ある程度の数式もグラフ化できる。
- MS-Excel のグラフに比べて品質が高い¹。

¹一般に科学論文では MS-Excel のグラフは使えないと言われていて、多くの科学者は論文用に MS-Excel 以外のグラフ描画ソフトを使っている。

現在の最新の正式リリース版は 4.6.0 であるが、本講義では、開発版 (4.7) を独自に make した MS-Windows 版の gnuplot を用いる。

- gnuplot の公式ホームページ: <http://www.gnuplot.info/>

「gnuplot の命令をスクリプトに書く」のは次回以降に行うこととし、今回は「対話型での利用方法」のみを紹介する。

対話形式の MS-Windows 版の gnuplot は、以下の 2 つがある。

- wgnuplot.exe : GUI 版の MS-Windows 用 gnuplot。対話ウィンドウとして GUI ウィンドウが立ち上がる。パイプ入力には対応していない。
- gnuplot.exe : CUI 版の MS-Windows 用 gnuplot。コマンドプロンプト上で対話環境が動作する。ヘルプやグラフは GUI 版と同様それぞれのウィンドウが立ち上がる。パイプ入力に対応している。

今回は「GUI 版の wgnuplot.exe」のみを利用する。コマンドプロンプトで、

```
Z:> wgnuplot
```

とすると、メニューのついた GUI 版 wgnuplot (対話ウィンドウ) が立ち上がり、バージョンなどのメッセージの最後に gnuplot プロンプトが

```
gnuplot>
```

と表示される。この gnuplot プロンプトの右に gnuplot の命令を入力していく形式で使用する。

2 次元グラフの描画命令は plot コマンドであり、その後ろに描画する関数 (数式)、あるいは描画するデータファイルを指定する。例えば、

```
gnuplot> plot 3*x*x-4*x+0.5
```

とすればグラフウィンドウが立ち上がり、その中で $y = 3x^2 - 4x + 1/2$ の 2 次関数のグラフが表示される。また、

```
1.0 -2.0  
2.0 7.0  
3.0 5.0  
6.0 0.0
```

というデータファイル data1.dat に対して

```
gnuplot> plot 'data1.dat'
```

とすれば、 $(x, y) = (1.0, -2.0), (2.0, 7.0), (3.0, 5.0), (6.0, 0.0)$ の 4 点がグラフウィンドウに描画される。グラフ描画は、以下のように行われる。

- 関数の描画は、実際には x の範囲を細かく分割して標本点を取り、それに対する関数の値を求め、その多数の点データのグラフを描いている。
- 関数のグラフはデフォルトでは折れ線で、データファイルのグラフはデフォルトでは点 (記号) で描くが、これらはオプションで変更可能。
- グラフの描画範囲は、指定しなければ自動的に設定される。

以下に、`wgnuplot` を使用する場合の注意をあげる。

- `wgnuplot` の終了は、対話ウィンドウの右上の終了アイコンをクリックするか、対話ウィンドウで `quit` か `exit` コマンドを入力する。
- 長いコマンド行を書く場合は、行末に `¥` をおいて改行すれば、次の行まで 1 行が続くと見なされる。
- 対話ウィンドウでは上下の矢印キーにより過去の入力行を呼び出せる²。
- 対話ウィンドウの「オプション」アイコンの「Choose Font」で対話ウィンドウの表示フォントを変更できる。変更後は、同じアイコンの「Update ... `wgnuplot.ini`」を実行しないと、次回の起動時に元の設定に戻ってしまう。また、グラフウィンドウのフォントの設定はここではなく、グラフウィンドウの上部の青いバー (タイトルバー) を右クリックして現れるメニューの「Choose Font」(と「Update ...」) で設定する。
- ヘルプは、対話ウィンドウの右上の「ヘルプ」アイコンをクリックするか、対話ウィンドウで `help` (あるいは `help [項目名]`) と入力すればヘルプウィンドウが表示される。

課題 11-1. `wgnuplot` を起動し、バージョン番号を確認せよ。

課題 11-2. `wgnuplot` の対話ウィンドウのフォントを見やすいものに変更し、`wgnuplot` を再起動してフォントの設定が正しく効いているか確認せよ。

課題 11-3. 対話ウィンドウで `plot sin(x)` と入力して三角関数のグラフを描画させ、対話ウィンドウやグラフウィンドウのアイコンの動作を確認せよ。

²コマンドプロンプト同様の入力を補佐する機能で、履歴機能 と呼ばれる。

課題 11-4. 上の data1.dat をエディタで作成し、そのデータを plot してみよ。

課題 11-5. 対話ウィンドウで help、または help quit (quit コマンドのヘルプの表示) と入力してヘルプを参照してみよ。

2 plot コマンドのオプション

2次元グラフを描画/更新するコマンドが plot コマンドである。plot コマンドにはたくさんのオプションがあるが、基本的には以下のように指定する³。

```
gnuplot> plot {<範囲指定>} <対象> {<オプション>}
```

<範囲指定> は、 x 軸、 y 軸の最小値、最大値の範囲を、コロン (:) 区切りで書いて [] で囲んで指定する。一つだけ範囲指定した場合は x 軸の範囲、2つ並べて指定した場合は x 軸と y 軸の範囲とみなされる。省略した場合、あるいは最小値、最大値を省略した (または * と書いた) 場合は自動的に範囲が選択される。例:

- [-1:5] : x は $-1 \leq x \leq 5$ の範囲、 y の範囲は自動設定
- [-1:][*:5] : x は $-1 \leq x$ で上の限界は自動設定、 y は $y \leq 5$ で下の限界は自動設定
- [] [3:5] : x の範囲は自動設定、 y は $3 \leq y \leq 5$ の範囲

課題 11-6. $y = 2 \sin(x + 3)$ のグラフを、 $-1 \leq x \leq 8$, $-3 \leq y \leq 3$ の範囲で描く plot コマンドを示せ。

<対象> は数式 (関数グラフの場合)、または "<データファイル名>" (データファイルグラフの場合) のいずれか。

<オプション> 部分に指定できる主なオプションは表 1 の通り。長いオプション名の代わりに省略名を使うこともできる。範囲指定オプション以外の表 1 のオプションには、with 以外は指定順に決まりはないので、<対象> の後ろ

³以降は、gnuplot のヘルプに合わせて、省略可能なオプションを { } で囲み ({} は実際には書かない)、実際に記述する内容を < > で囲んで示すことにする (< > も実際には書かない)。

オプション名	省略名	意味
title	t	凡例 (key) で使われる文字列を指定
notitle	not	凡例 (key) を描かない
with	w	描画スタイルの指定
using	u	データファイルの描画列を指定
every	ev	データ点の間引きを指定
axes	ax	描画軸の指定

表 1: plot コマンドの主なオプション一覧

に適当な順に並べて書けばよいが、with はさらに後ろにオプション (サブオプション) を複数つけられるので、原則最後に指定する。

with の後ろに指定する描画スタイルのうち、基本的なものは表 2 の通り⁴。関数グラフのデフォルト描画スタイルは折れ線 (with lines) で、データグ

スタイル名	省略名	意味
lines	l	折れ線
points	p	点 (記号)
linespoints	lp	折れ線と点 (記号)
dots	d	点 (小さい点)
impulses	i	細い鉛直線

表 2: 基本的な描画スタイル一覧

ラフのデフォルト描画スタイルは点 (記号) (with points) になっている。なお、通常の描画スタイルではデータ列は x と y の 2 列を必要とするが、描画スタイルに毎に必要なデータ列の個数には多少違いがある。それについては必要なら個別に説明するが、詳しくはマニュアルを参照せよ。

with によるスタイル指定の後ろには、さらに表 3 の追加オプション (サブオプション) をつけることができる⁵。pointtype と linecolor は、特に指定しなければ linetype の線種番号と同じ番号のものを使用する。with のサブオプションは、plot コマンドのオプションではなく with のサブオプションなので、with の後ろ書くのが正しい⁶。

線種、点種、線の色の一覧については、対話ウィンドウで test コマンドを実行すれば右端に上から順番に線種番号とそれに対応する線、点のサンプルが

⁴これら以外の他の描画スタイルについては後で紹介する。

⁵表 3 以外にもサブオプションはあるが、それは必要な場面で適宜紹介する。

⁶実際には with の前に書いても機能してしまう場合もある。

オプション名	省略名	意味
linetype	lt	グラフの線種を指定
linewidth	lw	グラフの線幅を指定
linecolor	lc	グラフの線色、点色を指定
pointtype	pt	グラフの点種を指定
pointsize	ps	グラフの点の大きさを指定

表 3: with の主なサブオプション一覧

表示される。

複数のグラフを重ねて描画する方法については、次回紹介する。

課題 11-7. test コマンドを実行し、線種番号、点種番号を確認せよ。

課題 11-8. 以下のオプションの意味を確認し、実際に実行し、基本的な描画スタイルを確認せよ。

- `plot sin(x) w p ps 2 pt 5 lt 1`
- `plot sin(x) t "正弦" w d lt 2`
- `plot sin(x) not w i lt 3`
- `plot "data1.dat" w l lt 4`
- `plot "data1.dat" t "データ" w lp ps 2 pt 6 lt 5`
- `plot "data1.dat" not w i lt 6`

課題 11-9. data1.dat のデータを、 $-1 \leq x \leq 7$ 、 $-5 \leq y \leq 10$ の範囲で、凡例なしの青色の折れ線と点のグラフ (線幅 1.5、点の大きさ 2、点種は中を塗らない三角形) で描く plot コマンドを示せ。

3 関数描画

gnuplot では、数式はほぼ C 言語と同じ形式で書けるし、数学関数も表 4 のように用意されているが⁷、数式の書き方については以下のことに注意が必要。

⁷この他にも、逆双曲線関数、ガンマ対数関数、ベッセル関数、ノイマン関数、完全楕円積分、Voigt 関数等の特殊関数や疑似乱数関数などの数学関数が用意されている。

- 累乗 x^y は `x**y` と書く⁸。また、階乗 $n!$ は `n!` と書ける。
- 円周率は `pi` として定義されている。
- 三角関数の角度の単位はデフォルトではラジアン⁹。
- `1/0` は (`0/0`, `2/0` など) エラーにはならず、無効値 (NaN) の代わりとして利用できる。
- `1/2` は整数の割り算で 0 になるので、`0.5` にしたければ `1.0/2.0` のように書かなければいけない¹⁰。

数学関数	意味	数学関数	意味
<code>abs(x)</code>	x の絶対値	<code>ceil(x)</code>	x 以上の最小の整数
<code>sgn(x)</code>	x の符号関数	<code>floor(x)</code>	x 以下の最大の整数
<code>sqrt(x)</code>	\sqrt{x}	<code>log(x)</code>	$\log_e x$ (自然対数)
<code>exp(x)</code>	e^x	<code>log10(x)</code>	$\log_{10} x$ (常用対数)
<code>cos(x)</code>	$\cos x$	<code>acos(x)</code>	$\arccos x$ (逆関数)
<code>sin(x)</code>	$\sin x$	<code>asin(x)</code>	$\arcsin x$ (逆関数)
<code>tan(x)</code>	$\tan x$	<code>atan(x)</code>	$\arctan x$ (逆関数)
<code>cosh(x)</code>	$\cosh x$ (双曲線関数)	<code>atan2(y, x)</code>	(x, y) に対する偏角
<code>sinh(x)</code>	$\sinh x$ (双曲線関数)	<code>tanh(x)</code>	$\tanh x$ (双曲線関数)
<code>erf(x)</code>	x の誤差関数	<code>inverf(x)</code>	<code>erf(x)</code> の逆関数
<code>norm(x)</code>	標準正規分布分布関数	<code>invnorm(x)</code>	<code>norm(x)</code> の逆関数
<code>gamma(x)</code>	ガンマ関数 $\Gamma(x)$	<code>lgamma(x)</code>	ガンマ対数 $\log \Gamma(x)$

表 4: 主な数学関数の一覧

課題 11-10. $y = \sin\left(x + \frac{\pi}{2}\right)$ のグラフ、および $y = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!}$ のグラフを描く `plot` コマンドを示せ。

課題 11-11. $y = \sqrt[3]{x}$, $y = \sin^2 x^3$, $y = \frac{\log\left|4x + \frac{2}{3}\right|}{\sqrt{2 + x^2 e^{-x^2}}}$ のグラフを描く `plot` コマンドを示せ。

課題 11-12. 表 4 の関数のグラフを一通り `plot` してみよ。

⁸AWK の `x^y` とは違うことに注意。この `x**y` の記法は FORTRAN に由来する。

⁹`set angles` を使えば度に変更もできるが詳細は省略する。

¹⁰この点も AWK とは異なる。AWK はすべて実数なので `1/2` で `0.5` になる。

4 データ描画

データファイルを描画する場合、データ処理をしながら描画するために、plot コマンドには using, every, index などのデータファイル描画専用のオプションが用意されている。ここでは、using について説明する。

デフォルトでは、データファイルの描画では、各行の 1 列目の値を x 座標、2 列目の値を y 座標としてグラフを描画するが、using オプションを使えば、任意の列のデータを x や y 座標にできるし、簡単なデータ処理も行える。

- using <指定 1>:<指定 2>
 <指定 1> がカッコなしの整数値の場合はその番号の列の値を x 座標にするが、丸カッコ () で囲まれた式の場合はその式の値を x 座標とする。同様に <指定 2> のものを y 座標とする。

丸カッコで囲んだ式の中では、AWK 同様、入力データの n 番目の列の値を $\$n$ (または column(n)) として参照できる。例:

- plot "data" using 2:1
 data の 2 列目を x 座標、1 列目を y 座標として描画。
- plot "data" using 1:(\$3-\$2)
 1 列目を x 座標、(3 列目の値) - (2 列目の値) を y 座標とする。

列番号でなくて式を書く場合は、必ず丸カッコが必要。逆に、列番号として指定する場合は丸カッコをつけてはいけない。例えば「using 1:(3)」の場合は y 座標は 3 列目ではなく、3 という定数値になる。

課題 11-13. 1 列目が 0 から 10 までの 1 ずつ増える整数値で、2 列目は 1 列目の 2 乗の値を持つデータ data2.dat を gawk で作成し、それを with lines と with points で描画させよ。

課題 11-14. data2.dat のデータの 2 列目を x 座標、1 列目を y 座標とした場合のグラフを描け。

課題 11-15. 1 列目が身長、2 列目が体重であるような 20 行のデータ data3.dat をエディタなどで作成し、そのデータの散布図 (with points のグラフ) を描け。

課題 11-16. data3.dat のデータを使って、 x 座標が身長、 y 座標は BMI である散布図を描く plot コマンドを示せ。なお BMI は、 $\text{BMI} = (\text{体重}) / (\text{身長})^2$ で定義される。

課題 11-17. data3.dat のデータを、体重が x 座標で、(身長 - 100) が y 座標の散布図 (点は青に塗りつぶされた逆三角形) で表示する plot コマンドを示せ。

plot では通常 2 列のデータが必要であるが、描画スタイルやそのオプションによって必要なデータの列には違いがある。その場合は、デフォルトでは 1 列目から順にデータが割り当てられるが、それを変更する場合は using でコロン (:) 区切りで必要な数だけ指定する。

特に、データが 1 列しかない場合、あるいは using で 1 列しか指定しなかった場合は、その値を y 座標とみなし、行番号を x 座標として使用することになっている (先頭行を 0 行目とみなす)。この行番号は、列番号 0 としても参照できるので、「using 2」「using 0:2」「using (\$0):2」はいずれもほぼ同じものを意味する。

逆に 1 列しかデータがなくても、「using 1:(sin(\$1))」のようにすれば、1 列目を x 座標とみなし、その $\sin x$ の値を y 座標としたグラフも描けることになる。これは、データと理想曲線を比較するときに利用できる。

課題 11-18. data2.dat のデータを用いて、 x 座標が 1 列目、 y 座標が $x^3 - 3x$ となるグラフを描け。

課題 11-19. gawk を用いて 1 列目は -2.0 から 2.0 までの 0.1 刻みの値、2 列目が x^2 の値となるデータ data4.dat のデータを作成し、2 列目が x 座標、1 列目が y 座標のグラフを描け。それと同じ形のグラフを gnuplot の関数描画でも描けるかどうか考えてみよ。

課題 11-20. (x, y) を原点を中心に反時計回りに θ 回転した点 (p, q) は、

$$p = x \cos \theta - y \sin \theta, \quad q = x \sin \theta + y \cos \theta$$

で表される。これを利用して、data4.dat のデータを原点の回りに 30° 回転した散布図 (with points のグラフ) を描画せよ。

コラム: gnuplot の特徴

gnuplot は 1989 年頃に T.Williams と C. Kelley (多分当時は学生) が作成を始めたグラフ描画用のフリーソフトで、最初からパソコン、ワークステーション、大型コンピュータなど、多くの環境で動作することを意図して作られた。

その目的のために、gnuplot は「コマンド駆動形式」であること、「多くの出力形式に対応」していること、「パイプ入力を受けつける」ことなどを大きな特徴として備えている。

「コマンド駆動形式」とは、通常の GUI 型グラフソフトがアイコンをクリックして設定などを行うのに対し、gnuplot は設定の変更などをすべてコマンド文字列で命令する方式であることを意味する。これは面倒に思えるかもしれないが、逆にそれによって色々な OS でも変わらずに動作し、それらの命令群をスクリプトファイルとして保存して再利用したり、バッチ処理したりすることが可能となっている。

「多くの出力形式に対応」とは、グラフを画面に出すだけでなく、画像ファイルとして保存したり、各種アプリケーションやプリンタの専用形式のファイルとして保存したりできることを意味する。古い出力形式は削除されることもあり、対応形式数は純粋に増え続けているわけではないが、現在でも 100 程度の出力形式に対応している。これは、多くのボランティアが開発に参加するオープンソースフリーソフトならではの性質であろう。

「パイプ入力」は、外部から gnuplot 命令を実行中の gnuplot プロセスの標準入りに流せる機能で、C 言語や AWK などのプログラム内部から gnuplot を実行/操作できるようになる。gnuplot のインラインデータ形式を使えば、一時的にデータファイルにデータを保存しなくても、データをパイプ経由で直接 gnuplot プロセスへ流せるので、自前のプログラムにグラフ機能を実装したり、シミュレーションからグラフの作成までを単独のプログラム内で作成したりすることが容易にできるようになる。

これらはいずれも、実験とそのデータの可視化を繰り返すような科学者にはありがたい機能で、大学等の研究機関の化学者、物理学者、工学者などが開発に関わってきたのものそのためだろう。