

gnuplot によるグラフの作成

プログラミング演習 1#7

平成 26 年 5 月 30 日

すみません．今週やる予定だったネットワークの話は来週にして，今週は来週予定だった Gnuplot の話をします．

1 gnuplot

gnuplot は，関数式やデータからグラフを作成するためのツールです．gnuplot の起動は，コマンドラインから，

```
% gnuplot
```

とすだけです．起動するとプロンプトが

```
gnuplot>
```

となり，入力待ち状態になります．この状態で様々な指示を与えることでグラフを描くことができます．起動直後の状態では，作成したグラフは画面上に表れます．プロンプトを通じて gnuplot に与える指示を表 1 に示します．

今週のやってみよう (1)

1. gnuplot で $y = \sin x \cos x$ のグラフを画面に描く．
2. 別の端末エミュレータから `tdh8025` のホームディレクトリに移動し，`Lecture/H26-2014/PE1/data` の下にある `BP0SC.dat` というファイルの内容を表示・確認する．
3. 拡張子が `dat` のファイル全てを，自分のホームディレクトリの下に `Penshu1` ディレクトリにコピーし，そこへカレントディレクトリを移動する．
4. gnuplot を起動し `plot "BP0SC.dat"` として `BP0SC.dat` のグラフを描く．
5. `plot "BP0SC.dat" with lines` として，どのようなグラフが描かれるか確認する．
6. `plot "BP0SC.dat" title "line1" with linespoints` として，グラフを確認する．
7. `plot "BP0SC.dat" title "line1" with linespoints linetype 1 linewidth 1 pointtype 2 pointsize 2 linecolor 1` としてグラフがどのように変化するか確認する．
8. `plot "BP0SC.dat" title "line1" with linespoints linetype 0 linewidth 3 pointtype 7 pointsize 3 linecolor 2` としてグラフがどのように変化するか確認する．
9. 1 つのグラフに，`BP0SC.dat`，`BP1SC.dat` の 2 つのデータをグラフに表してみる．線や点のスタイルは好きなように決めてよい．
10. `set term postscript 18` および `set output "graph.ps"` を実行し，グラフのタイトルを自分の学籍番号として，`BP0SC.dat` および `BP1SC.dat` の 2 つのグラフを `plot` コマンドでポストスクリプトファイル上に描画する．
11. 描画したポストスクリプトファイルを確認する．削除せずにとっておくこと．

表 1: gnuplot へ与えるコマンド

set term foo	出力デバイスを foo に変更する。標準では x11 (画面)。set term postscript 18 とすると、文字サイズが 18 ポイントのポストスクリプト形式になる。
set output "bar"	出力ファイル名を "bar" とする指定する。
set title "foo"	グラフに "foo" というタイトルを付ける。
set xlabel "foo"	横軸に名前 foo をつける。
set ylabel "foo"	縦軸に名前 foo をつける。
set xtics A,B	横軸の目盛刻みを、値 A から B 刻みに指定する。例えば、set xtics 0,0.5 とすると、0 から 0.5 刻みとなる。
set ytics A,B	縦軸の目盛刻みを指定する。A,B の指示方法は xtics と同じ。
set logscale *	軸を対数目盛にする。*には x, y, xy のいずれかを指定する。元の線型目盛に戻すのは set nologscale x のようにする。
plot 関数	グラフを描く。正弦関数を描く場合は plot sin(x) とする。標準では直線で描かれる。関数の代わりにデータファイルを "と" で囲んで与えるとそのデータ点を折れ線で結んだグラフが作成される。カンマで区切って複数の関数名 (ファイル名) を与えれば、一つのグラフ中に複数の関数 (データ) をグラフ化することも可能。
help	マニュアルを見る。膨大なマニュアルが読めるが、英語なので注意。英語が苦手な人はググった方が早いかも。とはいえ、たとえば上記の set term postscript の後のオプションを表示するのに help set term postscript とすれば良いのは便利。複数ページにわたる項目では、Enter キーで読み進んでいく。項目の最後でどのサブピックを読むかきかれるので、サブピック名を指定すればそのサブピックに飛ぶし、単に Enter キーを押せば上の階層 (スーパーピック (?)) に移る。一番上の階層で Enter キーを押せば help から抜ける。つまり、help のどこにいても Enter キーを押し続けられれば抜かれる。

2 3次元空間中の曲面の描画

曲面の描画には、plot の代わりに splot を用います。たとえば $z = \sin x \sin y$ を描画するには

```
gnuplot> splot sin(x) * sin(y)
```

とします。プロット後にマウスでグリグリ回すといろんな方向から見る事ができます。が、やっぱりデフォルトの設定だと分かりにくいと思います。もう少しわかり易くするには、

- set isosample 50 として、サンプル間隔を狭くします。この例ではサンプル数を 50 にしていますが、デフォルトの設定だとサンプル数 10 になっています。
- set hidden3d を実行して、陰線処理を行う設定にします。

この2点を実行して再描画してみるといいでしょう。

3 媒介変数表示による描画

gnuplot では、

$$x = \cos t, y = \sin t \tag{1}$$

のような媒介変数表示によるグラフ (t を動かしていった時の (x,y)) の描画も可能です。媒介変数の機能を使うには、`set parametric` コマンドを使います。すると

```
gnuplot> set parametric
```

```
dummy variable is t for curves, u/v for surfaces
```

```
gnuplot>
```

のようになって媒介変数モードになります。メッセージの意味は、曲線 (2次元描画) のための媒介変数は t 、曲面 (3次元描画) のための媒介変数は u,v ということです。(ちなみに、元に戻るには `unset parametric` を実行します。) 実際の描画は、(1) 式の例だと

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

とすればよいです。複数プロットするときは媒介変数でない時と同様やっぱりカンマで区切っていくので、たとえば (1) 式と

$$x = \cos t, y = \sin 2t \quad (2)$$

を同時に表示するには、

```
gnuplot> plot cos(t), sin(t), cos(t), sin(2*t)
```

のように区切り目が分かりにくい書き方になります。もちろん、線のつなぎ方を変えるには

```
gnuplot> plot cos(t), sin(t) with points, cos(t), sin(2*t) with linespoints
```

のように指定できます。また、当然ですが、媒介変数表示中でも、

```
gnuplot> plot t, sin(t)
```

のように指定すれば、媒介変数表示を使わずに描画できるグラフを描画できます。

今週のやってみよう (2)

1. gnuplot で (2) 式のグラフと $y = \sin x \cos x$ のグラフを画面に同時に描く。
2. `set term postscript 18` および `set output "parametric.ps"` を実行し、グラフのタイトルを自分の学籍番号として、同じグラフをポストスクリプトファイル上に描画する。ファイル名が `graph.ps` のままだと上書きされてしまうので注意すること。
3. 描画したポストスクリプトファイルを確認する。削除せずにとっておくこと。

今週のやってみよう (3)

1. 球面の式は

$$x = r \cos \phi, y = r \sin \phi \cos \psi, z = r \sin \phi \sin \psi$$

で表されます。この式に基づいて球面を描いてみよう。また、式を色々変えてグラフがどう変化するか見てもみよう。

2. これについても、グラフのタイトルを自分の学籍番号として、`sphere.ps` という名前のポストスクリプトファイルで保存しておいて下さい。