gnuplotの基本操作

電気情報工学科*　コンピュータシミュレーション
第8回

概要

この講義では，グラフ作成ソフトである gnuplot の基本操作を紹介する．

1　グラフ作成

1.1　gnuplotとは

gnuplot は簡単に 2D，3D のグラフが作成できるフリーのソフトウェアである．単純なグラフから，学術論文用の高品質なグラフまで作成可能で，広く普及している．gnuplot は,「ニュープロット (あるいは，ヌープロット)」と呼ばれる．間違いではあるが,「グニュプロット」と呼ばれることも多い．

gnuplot は UNIX に限らず Windows や Machintosh でも動作し，Excel とくらべものにならないくらい美しいグラフの作成が可能である．しかも，フリーである．卒業研究のグラフ作成に使うのをおすすめする．

さらに，gnuplot は C 言語などプログラミング言語と組み合わせて使うと，様々な処理が自動的にできる．

1.2　起動と終了

簡単な操作方法を述べるが，本当は，先ほど示した web ページを見て各自学習する方が良い．まずは，gnuplot を立ち上げてみよう．以下のコマンドを端末に入力する．

```
$ gnuplot
```

すると，gnuplot が立ち上がり，コマンド入力画面になる．終了したい場合は，

```
gnuplot> exit
```

とする．

*秋田工業高等専門学校

1.3　2 次元グラフ

グラフを描く前に，サンプル数を設定する．計算点の数が少ないと，グラフがでこぼこする．

```
gnuplot> set samples 1024
```

これで，計算する点の数が 1024 になった．これでもでこぼこする部分があれば，サンプル数を増やせ．

まずは，三角関数のグラフを書いてみよう．以下のコマンドを入力する．

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

sin 関数のグラフが描けただろう．描画範囲を変えたい場合は，

```
gnuplot> plot [0:6.28] [-1.5:1.5] sin(x)
```

とする．同時に複数のグラフを各場合は，次のように関数を並べればよい．

```
gnuplot> plot [-6.28:6.28] [-1.5:1.5] sin(x),cos(x),tan(x)
```

次のようにすると様々なグラフが描ける．

`gnuplot> plot x**3+x+1`	$x^3 + x + 1$
`gnuplot> plot x**0.5`	$x^{0.5}$
`gnuplot> plot log(x)`	$\log_e(x)$
`gnuplot> plot log10(x)`	$\log_{10}(x)$
`gnuplot> plot real(exp({0,1}*x))`	$\Re(e^{ix})$
`gnuplot> plot sqrt(x)`	$\sqrt{x}$

さらに，定義した関数のグラフを書くこともできる．

```
gnuplot> f(x)=sin(x)
gnuplot> g(x)=cos(x)
gnuplot> plot f(x)+g(x),
f(x)*g(x)
```

[練習 1] 以下の関数のグラフを作成してみなさい．

$\sin(x)\cos(x)$	$\sin^2(x)$	$\sin(x) + \cos(x)$
xe^{-x}	$x - \frac{x^3}{6} + \frac{x^5}{120} - \frac{x^7}{5040}$	$\cos(x)$, $\sin(x)$

gnuplot には付録の表**??**のような関数が用意されている．普通に使う関数はほとんど用意されているし，複素数もサポートされている．複素数は，実数部，虚数部のように記述する．すなわち，

`gnuplot> {1,0}`	実数の 1 を表す．
`gnuplot> {0,1}`	虚数単位 i を表す．
`gnuplot> {5.3,6.8}`	$5.3 + 6.8i$ を表す．

である．

1.4 3次元グラフ

3次元グラフも簡単にかける．3次元グラフの場合，右マウスでドラッグすると視点を変えることができるのでおもしろい．

```
gnuplot> splot          x^2 + y^2
x**2+y**2
gnuplot> splot          x sin(x + y)
x*sin(x+y)
```

3次元グラフで隠線処理が必要であれば，`set hidden3d` とする．また，表示するデータ点は，`set isosample` で設定する．たとえば，以下のようにすれば，隠線処理し，x方向とy方向とも40点のデータを出力する．

```
gnuplot> set hidden3d
gnuplot> set isosample 40,40
gnuplot> splot 1/(x*x+y*y+5)*cos(0.1*(x*x+y*y))
```

1.5 媒介変数表示

媒介変数を使ったグラフの作成もできる．

```
gnuplot> set parametric
gnuplot> plot sin(5*t), cos(2*t+pi/2)
```

媒介変数表示を止めるためには，

```
gnuplot> set noeparametric
```

とする．

1.6 ファイルのデータ描画

ファイルに格納されたデータをグラフ表示することもできる．講義では，三角関数の値が格納されたファイル[1]をダウンロードして，それをグラフにする．表の各行には，θ と $\sin\theta$，$\cos\theta$，$\tan\theta$ の値が書き込まれている．このファイルのデータをグラフ化するときには，`plot` コマンドを使う．引き続いて，ダブルクォーテーションでファイル名を囲む．最後に，`using` を使ってx座標とy座標が書かれている列を示す．具体的には，

```
gnuplot> plot "trifunc.txt" using 1:2
```

である．各データ点を線で結びたければ，

```
gnuplot> plot "trifunc.txt" using 1:2 with line
```

[1] http://www.akita-nct.jp/ yamamoto/lecture/2006/5E/gnuplot/trifunc.txt

とする．複数のデータを一度に描くためには，

```
gnuplot> plot "trifunc.txt" using 1:2 with line,
                          "trifunc.txt" using 1:3 with line,
                          "trifunc.txt" using 1:4 with line
```

とする．ただし，改行しないで (Enter キーを押さない) 記述する必要がある．これでは，$\tan(x)$ の値が大きすぎるので，プロットするレンジを変える．以下のように `set xrange[ymin:ymax]` でレンジを変えて，`replot` コマンドをつかう．

```
gnuplot> set yrange[-1.5:1.5]
gnuplot> replot
```

1.7 グラフのファイル出力

1.7.1 出力先の変更

gnuplot はディスプレイのみならず，グラフの図形ファイルを作成することができる．デフォルトの出力先はディスプレイとなっているので，これまでのグラフは画面に描画された．ディスプレイに表示されるだけでは不便なので，図形ファイルを作成する方法を示す．

諸君が使う代表的な出力先やファイルフォーマットを表 1 に示す．ディスプレイからファイルに出力先を変更するには，次のようにする．

```
gnuplot> set terminal emf
gnuplot> set output "hogehoge.emf"
```

これは，出直先を emf フォーマットのファイルに変更して，ファイル名を hogehoge.emf としている．

表 1: 代表的な出力デバイスとフォーマットを表す `set terminal` オプション

出力先	説明
`x11`	UNIX のディスプレイ
`windows`	Windows のディスプレイ
`emf`	Windows でよく使われるデータのフォーマット
`postscript`	UNIX で使われるデータフォーマット
`gif`	web などでおなじみ
`png`	これも web などでおなじみ
`epslatex`	LaTeX を使う場合便利

1.7.2 MS word に貼り付ける

gnuplot で作成したグラフを MS Word に貼り付ける時は，Windows 標準の emf フォーマットが便利である．emf フォーマットのグラフは次のようにして作成する．

```
gnuplot> set terminal emf
gnuplot> set output "hoge.emf"
gnuplot> plot sin(x)
```

最初の行で，出力先を emf ファイルに指定している．次の行で，出力ファイル名を hoge.emf としている．拡張子は重要で，emf としなくてはならない．プログラムは拡張子でファイルの種類を判断しているからである．最後の行で，三角関数のグラフを作成している．これで，正弦関数が書かれたファイル hoge.emf ができあがる．

これを MS Word に貼り付けるのは簡単である．

- Word を立ち上げて，メニューの挿入 → 図 → ファイルからを選択する．
- ファイルを選択する．

もし，emf フォーマットで張り付けができなければ，他のフォーマットを試してみる．

1.7.3 Starsuite に貼り付ける

学校の Linux では，office 環境として Sun microsystems の Starsuite が使えるようになっている．Starsuite へのグラフの張り付けもほとんど，MS Word と同じである．ここではあえて説明しないので，各自，実施して見よ．

1.7.4 LaTeX に貼り付ける

LaTeX というすばらしい文書作成ソフトウェアがある．使い方はちょっと難しいが，長い文書を作成する時は楽だし，なによりも出来上がりがすばらしい．本年度，みなさんに配布した講義ノートは全て LaTeX を使っている．以下のようにしてグラフのファイルを作成する．

```
gnuplot> set terminal epslatex
gnuplot> set output "hoge.eps"
gnuplot> plot sin(x)
```

そうすると，hoge.eps と hoge.tex というファイルができあがるので，それを LaTeX のソースに以下のように組み込めばよい．

```
\documentclass[10pt,a4paper]{jarticle}
\usepackage{graphicx}
\begin{document}

\begin{figure}[hbtp]
  \input{hoge}
```

```
  \caption{三角関数のグラフ}
\end{figure}

\end{document}
```

2 gnuplot のコマンド

2.1 ヘルプ

gnuplot にはかなり詳しいヘルプがある．英文であることと，コマンド入力のため初心者には使いにくいと思うが，使い込むとかなり便利である．ヘルプモードに入るためには，

```
gnuplot> help
```

とする．また，コマンドについて調べたければ，

```
gnuplot> help plot
```

のように，help の後にコマンド名を書く．

web ページの方が圧倒的に分かり易いが，ネットに接続されていない環境の場合やコマンドの使い方を忘れた場合には，かなり重宝する．

3 C 言語から gnuplot を操作する

3.1 パイプとは

gnuplot を C 言語のプログラムから制御するには，パイプを使うのが最も簡単である．ここでは，C 言語のプログラムによりパイプの生成し gnuplot を起動を行い，パイプを通してコマンドを送る．C 言語のプログラム内にコマンドを記述することにより，プログラマーの意図したとおりに gnuplot を操作することができる．

接続方法を説明する前に，UNIX のパイプという機能を簡単に説明しておく．UNIX のコマンドの大部分は，標準入力 (キーボード) からデータを受け取り，標準出力 (ディスプレイ) に処理した結果を出力するようになっている．例えば，

```
ls -l
```

などである．このように，コマンドをフィルター呼ぶ．

複数のフィルターコマンドを接続して，かなり複雑な処理ができる．最初のコマンドに処理すべきデータを与え，その標準出力を次のコマンドに渡すのである．例えば，

```
ls -l | sort -n -k +5
```

のようにする．最初のコマンドで，カレントディレクトリのファイルとディレクトリーの情報を調べ，次のコマンドでファイル容量の順に並べている．「ls -l」の出力が「sort -n -k +5」の入力になってる[2]．

このようにコマンドを連結する機能をパイプラインという．そして，連結する | をパイプという．あたかも，パイプにデータが流れているかのようである．もちろん，2 個以上のコマンドの連結が可能である．このようにパイプを使ってコマンドをつなぐことにより，UNIX ではかなり複雑な動作も簡単に記述できる．

3.2 パイプによる gnuplot へのコマンド送信

パイプを使うことにより，C 言語のプログラムを通して gnuplot を制御することができる．C 言語のプログラムから，gnuplot にパイプを通してコマンドを流し，プログラマーの思い通りに動作させるのである．これを実現するためには，(1) パイプを開く (2) パイプを通してコマンドを送る (3) パイプを閉じる—という一連の操作が必要である．

パイプを開くためには，ファイルポインターをつかう．そのためファイルポインターを格納する変数を用意しなくてはならない．パイプの先もファイルとして扱われるのである．

```
FILE *hoge;
```

次に gnuplot を立ち上げて，そこにパイプを接続する必要がある．次のようにする．

```
hoge = popen("gnuplot -persist","w");
```

popen() 関数がパイプを開く命令である．これで，gnuplot が立ち上がり，パイプを通して，コマンドを送ることができる．オプションの persist で，gnuplot が終了してもグラフが残るようにしている．そうしないと，コンピュータの動作は高速なので，gnuplot は一瞬にして終了し，グラフが消えてしまい，ほとんど動作内容が分からなくなる．popen() 関数の戻り値はパイプの情報を示すファイルポインタである．このファイルポインタを指定して，コマンドを送ることになる．以前，学習したファイル操作とほとんど同じである．

パイプを通して，gnuplot にコマンドを送るのは fprintf() 関数を使う．

```
fprintf(hoge, "plot sin(x)\n");
```

この fprintf を使って，gnuplot にいくらでもコマンドを送ることができる．あたかも，C 言語の向こう側で gnuplot が立ち上がって，それから命令を送っているかのように動作する．このようなことができるのが，コマンドを打ち込む Character-based User Interface(CUI) の良いところである．

すべての動作が終了したならば，パイプを閉じなくてはならない．これも，ファイルの操作と全く同じである．

```
pclose(hoge);
```

[2]各コマンドの詳細は「man ls」とか「man sort」とかタイプして，マニュアルを見よ．マニュアルは，f:次のページ，b:で前のページ，q:で終了の操作ができる．

3.3 プログラム例

3.3.1 グラフ作成

C 言語から gnuplot へパイプを使ってコマンドを送る方法を示す．リスト 1 はその例で，三角関数のグラフを描いている．パイプを開き fprintf() 関数を使って，直接 gnuplot にコマンドを送っているだけである．

リスト 1: パイプを使い C 言語から gnuplot にコマンドを送っている．

```
1  #include <stdio.h>
2
3  int main(void){
4    FILE *gp;
5
6    gp = popen("gnuplot -persist","w");
7    fprintf(gp, "plot sin(x)\n");
8
9    pclose(gp);
10
11   return 0;
12 }
```

3.3.2 配列のデータをグラフ化

数値計算の場合，配列にデータが格納されることが多い．配列に格納されたデータを gnuplot に送には，次のようにする．複数のデータがある場合は，付録に載せている．

1. まずは，配列のデータを送る準備が必要である．21 行目のように，

   `plot '-' [オプション]`

   とする．

2. データの送信は，24 行目にように

   `fprintf(パイプを表すファイルポインタ, データの並び)`

   とする．

3. e を送れば，データの終了となる．27 行目にように

   `fprintf(パイプを表すファイルポインタ,"e\n")`

   とする．

リスト 2: 配列に格納されたデータのグラフ化.

```
1  #include <stdio.h>
2  #include <math.h>
3  #define NX 720
4
5  int main(void){
6    FILE *gp;
7    int i;
8    double dx, x[NX+1], y[NX+1];
9
10   /* ----  ^^c7^^璽榛 成a1 ---- */
11   dx=4*M_PI/NX;
12   for(i=0; i<=NX; i++){
13     x[i]=-2*M_PI+i*dx;
14     y[i]=sin(x[i]);
15   }
16
17   /* ---- グア ^^d5^^ 成ba ---- */
18   gp = popen("gnuplot -persist","w");
19   fprintf(gp, "set xrange [-6.5:6.5]\n");
20   fprintf(gp, "set yrange [-1.5:1.5]\n");
21   fprintf(gp, "plot '-' with lines linetype 1 title \"sin\"\n");
22
23   for(i=0; i<=NX; i++){
24     fprintf(gp,"%f\t%f\n", x[i], y[i]);     //  ^^c7^^璽燭ν a1 ^^f1 ^^a4 ^^ad ^^
b9
25
26   }
27   fprintf(gp,"e\n");
28
29   pclose(gp);
30
31   return 0;
32 }
```

3.3.3 データファイルのグラフ作成

複雑な数値計算を行う場合，データをファイルに保存してから，それをグラフ化することもある．リスト 3 ではデータを一度ファイルに保存してから，それを呼び出してグラフ化している．ファイルに格納されたデータをグラフにする方法は簡単で，27 行目のように

```
plot "ファイル名" [オプション]
```

と直接，ファイル名を指定指定したコマンドを送る．

リスト 3: ファイルに格納したデータをグラフにする方法.

```
1  #include <stdio.h>
2  #include <math.h>
3  #define NX 720
4
```

```
5  int main(void){
6    FILE *data, *gp;
7    char *data_file;
8    int i;
9    double dx, x, y;
10
11   /*------  ^^c7^^璽織a1^^d5^^.ぅ    成a5 ---------- */
12   data_file="out.dat";
13   data = fopen(data_file,"w");
14
15   dx=4*M_PI/NX;
16   for(i=0; i<=NX; i++){
17     x=-2*M_PI+i*dx;
18     y=sin(x);
19     fprintf(data,"%f\t%f\n", x, y);
20   }
21   fclose(data);
22
23   /*------ グア ^^d5^^κ 成a4 ---------- */
24   gp = popen("gnuplot -persist","w");
25   fprintf(gp, "set xrange [-6.5:6.5]\n");
26   fprintf(gp, "set yrange [-1.5:1.5]\n");
27   fprintf(gp, "plot \"%s\" with lines linetype 1 title \"sin\"\n",data_file);
28   pclose(gp);
29
30   return 0;
31 }
```

4 グラフの装飾

もう少し複雑な，set コマンドを駆使した例をリスト 4 に示す．

リスト 4: パイプを使った gnuplot の制御．データファイルを作成して，それをプロットしている．軸なども細かく制御している．

```
1  #include <stdio.h>
2  #include <math.h>
3  void mk_triangle_data(char *a, double x1, double x2, int n);
4  void mk_graph(char *f, char *xlb, double x1, double x2,
5                char *ylb, double y1, double y2);
6
7  /*============================================================*/
8  /*    main function                                           */
9  /*============================================================*/
10 int main(void){
11
12   double pi = 4*atan(1);
13
14   mk_triangle_data("out.txt", -2*pi, 2*pi, 1000);
15   mk_graph("out.txt", "x", -2*pi, 2*pi, "y", -3, 3);
16
17   return 0;
```

```
18 }
19
20 /*============================================================*/
21 /*    make a data file                                        */
22 /*============================================================*/
23 void mk_triangle_data(char *a, double x1, double x2, int n){
24   double x, dx;
25   double y1, y2, y3;
26   int i;
27   FILE *out;
28
29   dx = (x2-x1)/n;
30
31   out = fopen(a, "w");
32
33   for(i=0; i<=n; i++){
34     x = x1+dx*i;
35     y1 = sin(x);
36     y2 = cos(x);
37     y3 = tan(x);
38
39     fprintf(out, "%e\t%e\t%e\t%e\n", x, y1, y2, y3);
40   }
41
42   fclose(out);
43 }
44
45 /*============================================================*/
46 /*    make a graph                                            */
47 /*============================================================*/
48 void mk_graph(char *f, char *xlb, double x1, double x2,
49               char *ylb, double y1, double y2)
50 {
51
52   FILE *gp;
53
54   gp = popen("gnuplot -persist","w");
55
56   fprintf(gp, "reset\n");
57
58   /* -------  set x grid ---------*/
59
60   fprintf(gp, "set grid\n");
61
62   /* -------  set x axis ---------*/
63
64   fprintf(gp, "set xtics 1\n");
65   fprintf(gp, "set mxtics 10\n");
66   fprintf(gp, "set xlabel \"%s\"\n", xlb);
67   fprintf(gp, "set nologscale x\n");
68   fprintf(gp, "set xrange[%e:%e]\n", x1, x2);
69
70   /* -------  set y axis ---------*/
71
```

```
72    fprintf(gp, "set ytics 1\n");
73    fprintf(gp, "set mytics 10\n");
74    fprintf(gp, "set ylabel \"%s\"\n", ylb);
75    fprintf(gp, "set nologscale y\n");
76    fprintf(gp, "set yrange[%e:%e]\n", y1, y2);
77
78          /* -------  plat graphs ---------*/
79
80    fprintf(gp, "set terminal x11\n");
81
82    fprintf(gp, "plot \"%s\" using 1:2 with line,\
83                      \"%s\" using 1:3 with line,\
84                      \"%s\" using 1:4 with line\n", f, f, f);
85
86    fprintf(gp, "set terminal png\n");
87    //fprintf(gp, "set output \"tri.png\"\n");
88
89    fprintf(gp, "replot\n");
90
91    pclose(gp);
92  }
```

5 練習問題

好きな関数あるいは計算結果を C 言語からパイプを使って gnuplot でグラフを作成してみよう.