GNUPLOTの利用法 : 2次元グラフを中心に

森保, 洋
九州大学大学院経済学研究科

https://doi.org/10.15017/1470258

出版情報：九州大学大型計算機センター広報. 27 (5), pp.538-555, 1994-12. 九州大学大型計算機センター
バージョン：
権利関係：

GNUPLOTの利用法

—2次元グラフを中心に—

森保 洋*

1　はじめに

GNUPLOTは2,3次元の関数・データを対話的にグラフ化するプログラムです．動作環境もUnix（もちろんUXPも），VMS，Atari，MS-DOS，MS-Windows，OS/2と豊富なため，大学ではUXP，自宅ではMS-WindowsでGNUPLOTを使い仕事を行うといったことも可能です．出力デバイスもディスプレイはもちろん，PostScriptがサポートされているので，美しい出力結果が得られます．出力デバイスにPostScriptまたはtpic specialを選択すれば，LaTeXで作成した文書の中にGNUPLOTで作ったグラフを取り込むことも可能です．

以下ではUXPを利用して2次元グラフをいかに美しく表示させるかについて紹介します．

1.1　最初に覚えること

X-windowを利用する場合は，GNUPLOT起動前にローカルのホスト名を指定しておかなければなりません．例えばローカルホスト名が`local-h`なら，`kyu-cc`側で

```
kyu-cc> setenv DISPLAY local-h:0.0
```

とし，`local-h`側で

```
local-h% xhost +kyu-cc
```

とします．

コマンドラインから`gnuplot`と入力することで，GNUPLOTが起動し，オープニングメッセージと共に`gnuplot>`とプロンプトが表示され，コマンド入力待ちになります．

ここで`help`コマンドを実行してみましょう．GNUPLOTの概略と共に，オンラインマニュアルが用意されているコマンドの一覧が表示されます．ここで表示したいコマンド名を入力すれば，そのコマンドの詳しい説明を見ることができます．コマンドによってはオプション名などについてさらに入力を促されることもあります．このオンラインマニュアルは`kyu-cc`に`/usr/local/doc/gnuplot.dvi`として保管されているGNUPLOTのリファレンスマニュアルを対話形式で読めるようにしたものですので，GNUPLOTの機能をほぼ完全に網羅しています．まめにオンラインヘルプを活用する癖をつけたほうがよいでしょう．

基本的にGNUPLOTでのコマンド入力はすべて小文字です．コマンド入力に際してはEmacsライクなラインエディットとヒストリ機能が使えます．

`quit`コマンドでGNUPLOTは終了します．今までの流れを図1に示します．

2　2次元グラフの描画コマンド–plot–

GNUPLOTでは2次元グラフを`plot`コマンドで描画します．このコマンドにより関数，データのグラフ化が可能です．

*九州大学大学院経済学研究科

```
kyu-cc> gnuplot

        G N U P L O T
        unix version 3.5
...
Terminal type set to 'x11'
gnuplot> help
 GNUPLOT is a command-driven interactive function plotting program. It
 is case sensitive (commands and function names written in lowercase
 are not the same as those written in CAPS). All command names may be
...
Help topics available:
      autoscale       bugs            cd              clear
      comments        environment     exit            expressions
      help            line-editing    load            pause
      plot            print           pwd             quit
      replot          save            set             shell
      show            splot           startup         substitution
      userdefined

Help topic: cd
 The `cd` command changes the working directory.

 Syntax:
         cd "<directory-name>"

 The directory name must be enclosed in quotes.

 Examples:
         cd 'subdir'
         cd ".."

Help topic:
gnuplot> quit
kyu-cc>
```

図 1: GNUPLOT 起動から終了まで

2.1 関数のプロット

関数をグラフ化するには plot コマンドに続いて，関数を指定します．具体的には

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

とすることで，図2が得られます．2種類以上の関数を同時に描画することも可能です．この場合，関数を “,” で区切ります．

```
gnuplot> plot sin(x) , 1.2*cos(2*x)
```

この結果が図3です．

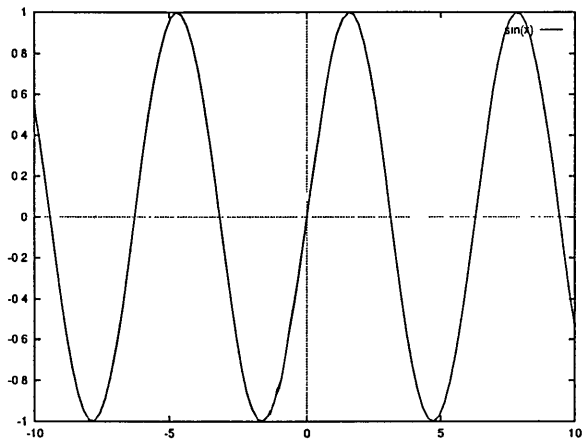

図 2: $y = \sin(x)$

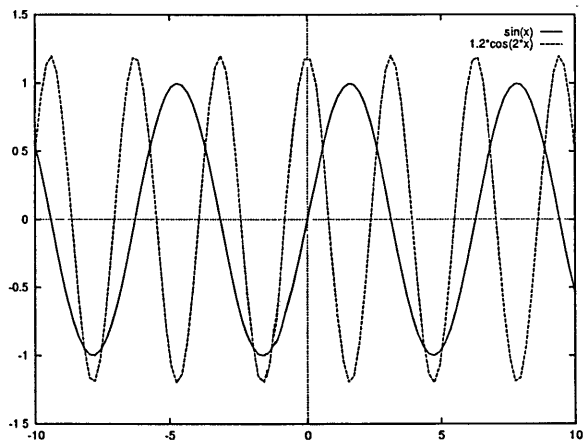

図 3: $y = \sin(x)$, $y = 1.2\cos(2x)$

GNUPLOTでは三角，指数関数をはじめとする数学関数が多数用意されています．詳しくはリファレンスマニュアルを参照してください．また，デフォルトでは関数の独立変数を x で表さなくてはなりません．演算子についてはC言語のものとほとんど同じですが，巾乗を表す ** や階乗を表す! などが追加されています．

あらかじめ用意されていない関数や定数は自分で定義することができます．

```
gnuplot> e = 2.71828
gnuplot> f(x) = e**(-x/5)*sin(x)
gnuplot> plot f(x)
```

2.2 様々なオプション

上のグラフではただ単に関数の形を指定しただけでした．グラフのタイトルや，値域，定義域などの設定は行なっていません．次にグラフをどうやって自分好みのものに仕上げるかをみてみましょう．

2.2.1 プロット範囲の設定

GNUPLOTではグラフに対する設定を種々の変数に記憶しています．その変数への代入は set コマンドで行ないます．例えば，X軸の範囲を $[-\pi, \pi]$ にして $\sin(x)$ を描かせるには

```
gnuplot> set xrange [-pi:pi]
gnuplot> plot sin(x)
```

とします．ここで， pi はあらかじめGNUPLOTで用意されている円周率を表す定数です．同様にY軸に関しても

```
gnuplot> set yrange [-1.2:1.2]
gnuplot> replot
```

とすることで，$[-1.2, 1.2]$ の範囲でプロットされることになります．replot コマンドは，現在設定されている変数内容にしたがって，最後に実行された plot コマンドを再実行するもので，グラフの設定の微調整を行なう時には非常に便利です．また，

```
gnuplot> set xrange [:10]
gnuplot> set yrange [-2:]
```

のように上限だけ，下限だけの指定も可能です．この場合，省略された部分の範囲は前回の設定が有効になります．

2.2.2 タイトル，X軸，Y軸のラベル

次にグラフにタイトル，X軸とY軸のラベルをつけてみます．

```
gnuplot> set title "sine curve"
gnuplot> set xlabel "X-axis"
gnuplot> set ylabel "Y-axis"
gnuplot> replot
```

上のように title,xlabel,ylabl のあとに指定したい文字列を二重引用符で括ります．残念ながら日本語の文字列は使えないようです．

2.2.3 グリッド

グラフに格子上の破線を入れるには

```
gnuplot> set grid
gnuplot> replot
```

とします．この設定を無効にするには

```
gnuplot> set nogrid
gnuplot> replot
```

です．

2.2.4 凡例

デフォルトではグラフの凡例がグラフの右上に表示されますが，この場所を変更したり，表示させないようにすることもできます．凡例をグラフの $(-2.5, 1)$ の位置に表示させたい場合，

```
gnuplot> set key -2.5,1
gnuplot> replot
```

のように，`set key` のあとに凡例を表示する X,Y 座標を入力します．凡例の文字列は `plot` コマンドから自動的に設定されるので，気に入らなければ，`plot` コマンド実行の際に `title` オプションをつけます．例えば，

```
gnuplot> plot sin(x) title "sine courve"
```

です．凡例を表示させたくない時には，

```
gnuplot> set nokey
gnuplot> replot
```

とします．

2.2.5 矢印と文字列

グラフ上に矢印と文字列を入れてみます．矢印を入れるにはオプション `arrow` を使います．

```
set arrow <tag> from <sx,sy> to <ex,ey>
```

<tag> は何本目の矢印かを区別するための番号で，座標 (sx,sy) から (ex,ey) への矢印を引きます．具体的には

```
gnuplot> set arrow 1 from 1,1.1 to pi/2,1
gnuplot> replot
```

のように使います．矢印の矢じりの部分を表示させたくない時は

```
gnuplot> set arrow 1 from 1,1.1 to pi/2,1 nohead
gnuplot> replot
```

のように，(ex,ey) を指定した後に `nohead` を追加します．
番号 1 の矢印を非表示にしたい場合には

```
gnuplot> set noarrow 1
gnuplot> replot
```

です．すべての矢印を非表示にするには noarrow のあとの番号を省略します．
グラフ上に文字列を表示させる例を以下に示します．

```
gnuplot> set label 1 "extremal value" at 1,1.1 right
gnuplot> replot
```

set label の後の番号が，何番目の文字列かを表す番号で，そのあとに実際に表示させる文字列を二重引用符で括って指定します．at の後には表示させる座標を入力します．最後の right は表示させる文字列を座標に右よせで表示させることを意味しています．right のほかに，center , left でそれぞれ中央よせ，左よせで表示させることができます．arrow 同様

```
gnuplot> set nolabel 1
gnuplot> replot
```

で番号1の文字列を非表示に，また nolabel のあとの番号を省略することですべての文字列を非表示にできます．

2.2.6 サンプリング間隔

図2で三角関数がなめらかに描かれていないと思った人は

```
gnuplot> set samples 200
gnuplot> replot
```

を実行してみてください．samples オプションで関数の値をサンプリングする点の数を変更することができます．デフォルトは100なので，100以上の値を設定すると，よりなめらかなグラフが描けます．

2.2.7 オプションの現在値表示

set コマンドで設定した内容は show コマンドですべて調べることができます．show のあとに all と指定すると，GNUPLOT のバージョン，設定できるオプションの現在値，ユーザー定義の関数，定数など，すべての情報が表示されます．all の代わりにオプション名を指定することでそのオプションの現在値だけが表示されます．ほかにもversion , function , variables を指定すると，それぞれGNUPLOT のバージョン，ユーザー定義の関数，ユーザー定義の定数を表示します．図4に show all の実行結果の一部を示します．

2.2.8 応用例

最後に以上のことを使った簡単な例と，その設定を図6と図5に示します．図5の最後の行に t "d.f.=6" という記述がありますが，これは title "d.f.=6" の省略形です．このようにGNUPLOT では他のコマンドと混乱しない限りにおいてコマンド名を短く入力することができます．また，コマンドが1行に入らない場合，継続記号として \ を使うことができます．

2.3 媒介変数表示の関数プロット

```
gnuplot> set parametric
```

とすることで，媒介変数を使ったグラフを表示するモードへ移行します．デフォルトでは媒介変数を t で表します．たとえば，$x = \cos 3t$, $y = \sin 5t$ を表示させるには次のようにします．

```
gnuplot> show all

        G N U P L O T
        unix version 3.5
        patchlevel 3.50.1.17, 27 Aug 93
        last modified Fri Aug 27 05:21:33 GMT 1993

        Copyright(C) 1986 - 1993   Thomas Williams, Colin Kelley

        Send comments and requests for help to info-gnuplot@dartmouth.edu
        Send bugs, suggestions and mods to bug-gnuplot@dartmouth.edu
        autoscaling is  x: OFF, y: OFF, z: ON
...
        data are plotted with points
        functions are plotted with lines
        grid is ON
        label 1 "modes" at 8.1,0.19,0 left
        arrow 1 from 8,0.19,0 to 2,0.18394,0
        arrow 2 from 8,0.19,0 to 4,0.135335,0
        arrow 3 from 8,0.19,0 to 6,0.112021,0
        key is ON
...
        terminal type is x11
        tics are IN,    ticslevel is 0.5
        x-axis tic labelling is computed automatically
        y-axis tic labelling is computed automatically
        z-axis tic labelling is computed automatically
        time is OFF, offset at 0, 0
        xrange is [0 : 15]
        yrange is [0 : 0.2]
        zrange is [-10 : 10]
        title is "chi-square distributions", offset at 0, 0
        xlabel is "x", offset at 0, 0
        ylabel is "f(x,df)", offset at 0, 0
        zlabel is "", offset at 0, 0
        zero is 1e-08
        last plot command was: plot chisq(x,4) title "d.f.=4",
        chisq(x,6) t "d.f.=6",chisq(x,8) t "d.f.=8"

        Variables:
        pi = 3.14159
        at is undefined

        User-Defined Functions:
        chisq(x,n) = 1/(2**(n/2)*gamma(n/2)) * x**(n/2-1) * exp(-x/2)

gnuplot>
```

図 4: show all の実行結果例

```
gnuplot> chisq(x,n) = 1/(2**(n/2)*gamma(n/2)) * x**(n/2-1) * exp(-x/2)
gnuplot> set xrange [0:15]
gnuplot> set yrange [0:0.2]
gnuplot> set title "chi-square distributions"
gnuplot> set xlabel "x"
gnuplot> set ylabel "f(x,df)"
gnuplot> set grid
gnuplot> set arrow 1 from 8,0.19 to 2,chisq(2,4)
gnuplot> set arrow 2 from 8,0.19 to 4,chisq(4,6)
gnuplot> set arrow 3 from 8,0.19 to 6,chisq(6,8)
gnuplot> set label 1 "modes" at 8.1,0.19 left
gnuplot> plot chisq(x,4) title "d.f.=4",chisq(x,6) t "d.f.=6",chisq(x,8) \
> t "d.f.=8"
```

図 5: 図 6 を作成するために入力したコマンド

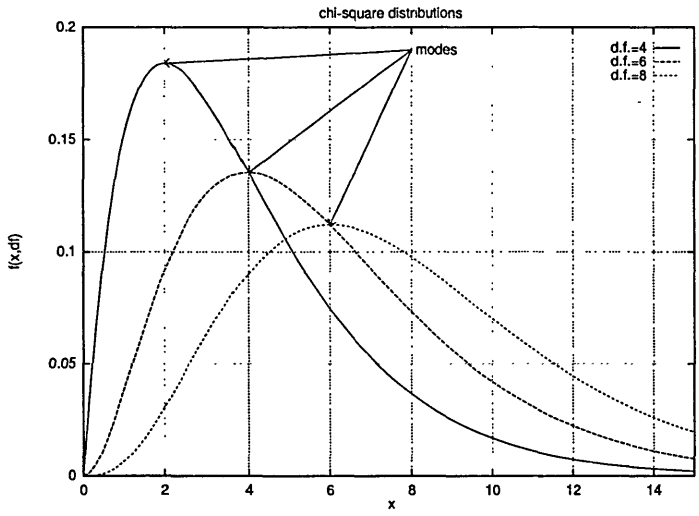

図 6: オプション設定の応用例

```
gnuplot> set trange [0:2*pi]
gnuplot> plot cos(3*t),sin(5*t)
```

媒介変数 t の範囲を $[0, 2\pi]$ に設定したあと，plot コマンドで，X 座標，Y 座標に対応する関数を“,”で区切って入力します．上の例の表示結果が図 7です．

媒介変数表示関数のプロットから通常の関数プロットモードに戻るには，

```
gnuplot> set noparametric
```

とします．

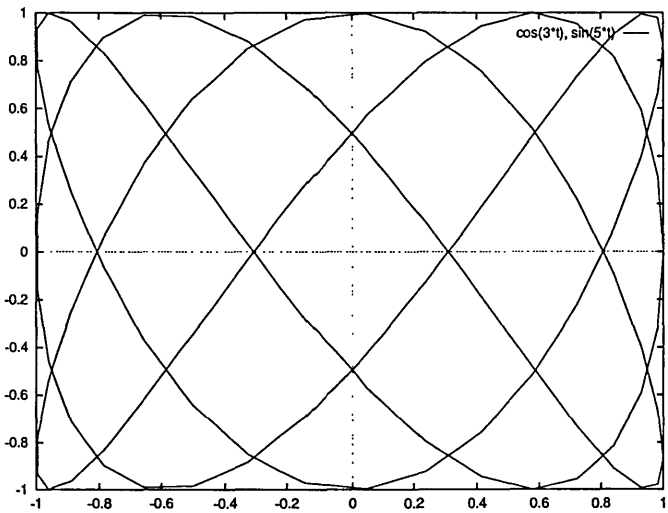

図 7: 媒介変数表示関数のプロット

2.4　データのグラフ化

実験データなどをグラフ化するには，あらかじめそのデータを別ファイルとして保存しておかなければなりません．一番簡単な例を示します．次のようなデータが pc1.dat という名前で保管されていたとします．

pc1.dat

```
#パソコン生産高 (億円)
2680.02
3357.40
3640.37
3829.49
4882.61
7989.34
9068.81
```

このとき

```
gnuplot> plot "pc1.dat"
```

とすると，図8のグラフが得られます．pc1.datのようにファイルにデータが縦1列に並んでいる場合，GNUPLOTはX座標に$0,1,\cdots$を，Y座標にファイルのデータを使い散布図を描きます．# 以下はコメントとして無視されますので，データ系列の説明などをつけておくとよいでしょう．

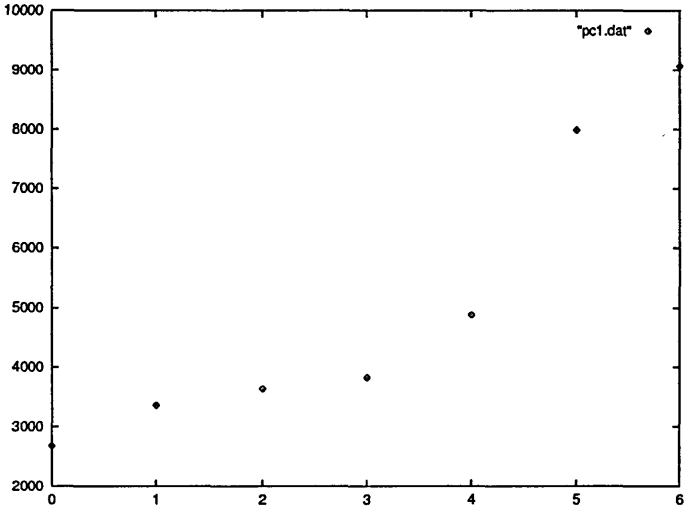

図 8: 一番簡単なデータプロット

このままでは，X座標を指定することができません．X,Y座標とも指定したい場合は次のようなファイル pc2.dat を用意します．

pc2.dat

```
#西暦    パソコン生産高
1984     2680.02
1985     3357.40
1986     3640.37
1987     3829.49
1988     4882.61
1989     7989.34
1990     9068.81
```

上のように，各列はタブまたはスペースで区切ります．このファイルを使って

```
gnuplot> plot "pc2.dat"
```

を実行すると，第1列をX座標に，第2列をY座標にとる散布図を表示します．また，using オプションを使えば，任意の列をX,Y座標に割り当てることができます．

```
plot "datafile" using <xcol>:<ycol>
```

例えば，第2,1列をX,Y座標としてグラフ化するには

```
gnuplot> plot "pc2.dat" using 2:1
```

を実行します．また，ファイル pc3.dat が

pc3.dat

```
#西暦   パソコン        汎用コンピュータ
1984    2680.02         6627.89
1985    3357.40         8337.46
1986    3640.37         9853.69
1987    3829.49         11655.53
1988    4882.61         13536.08
1989    7989.34         13762.16
1990    9068.81         12912.30
```

のときに，

```
gnuplot> plot "pc3.dat" using 1:3
```

とすればX,Y 座標に第1,3列を割り当てたグラフが表示されます．この場合にusing 以下を省略するとX,Y 座標に第1,2列が自動的に割り当てられます．

関数のプロット同様，“,” で区切ることによって，データに関しても複数の系列を同時に表示することができます．

```
gnuplot> plot "pc3.dat" using 1:2, "pc3.dat" using 1:3
```

この場合，同じファイル pc3.dat の第1列をX 座標に，第2,3列をY 座標にするグラフが表示されます．

もちろん，データと関数の同時表示も可能です．

```
gnuplot> plot "pc3.dat" , exp(7.83+0.22*(x-1984))
```

タイトル，X,Y 軸のラベルや範囲，凡例や矢印などは関数のプロットのところで説明したものがそのまま使えますので，そちらを参照してください．以上のことを使った具体的な例と出力結果を図9と図10に示します．

2.5　種々のグラフ表示

今までのグラフでは関数が実線，データが点で表示されていました．GNUPLOT では plot コマンドの with オプションを使ってラインスタイルを変更することができます．指定できるスタイルは lines , points , linespoints , impulses , dots , steps , boxes , errorbars , boxerrbars の8種類です．上で見てきたように関数では lines データではpoints がデフォルトとして設定されてい

```
gnuplot> set title "Domestic output of personal computers"
gnuplot> set xlabel "year"
gnuplot> set ylabel "100 million yen"
gnuplot> set grid
gnuplot> plot "pc3.dat" title "observations" , \
> exp(7.83+0.22*(x-1984)) t "predictions"
```

図 9: 図 10 を作成するために入力したコマンド

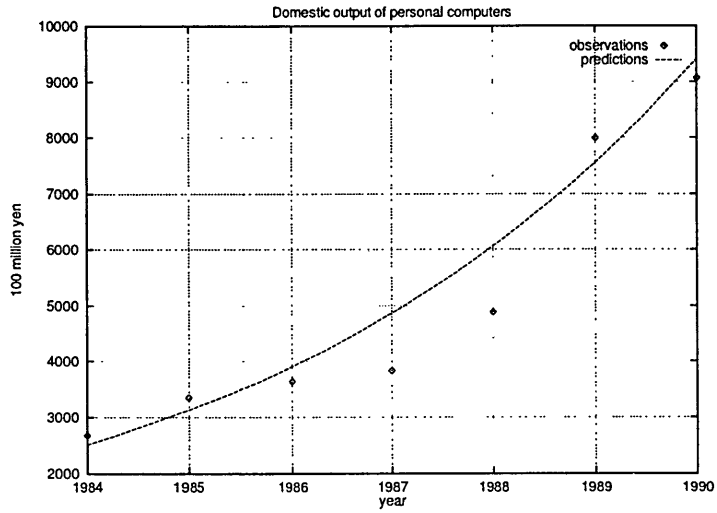

図 10: データプロットの具体例

ます．それぞれのスタイルの説明は具体的な例を見た方が早いでしょう．lines , points は上に例がありますので，まず，linespoints , impulses , dots , steps , boxes の例を図 11,12,13,14,15 に示します．図 13では非常に小さな点でプロットされています．大量のデータをプロットする時に好都合です．

残る errorbars , boxerrbars は観測値の他に誤差をグラフに表現したい時に用います．用意すべきデータファイルもX,Y 座標だけでなく誤差データも含んでいなければなりません．例えば次のようなファイルpc4.dat を用意します．

pc4.dat

#西暦	パソコン生産高	下限	上限
1984	2680.02	41.09141	3729.485
1985	3357.40	1216.92928	4673.111
1986	3640.37	2350.23110	5659.273
1987	3829.49	3435.22445	6693.744
1988	4882.61	4469.69510	7778.737
1989	7989.34	5455.85728	8912.039
1990	9068.81	6399.48341	10087.877

この時，

```
gnuplot> plot "pc4.dat" with errorbars
```

を実行したものが図16です．図を見てわかるように第1,2 列のデータをX,Y 座標として点でプロットし，第3 列と第4 列を線で結んで表示します．今まで同様，using を用いて任意の列を点の座標や誤差に割り当てることができます．using は with の前に指定しなければならないことに注意してください．

```
gnuplot> plot "pc4.dat" using 4:3:2:1 with errorbars
```

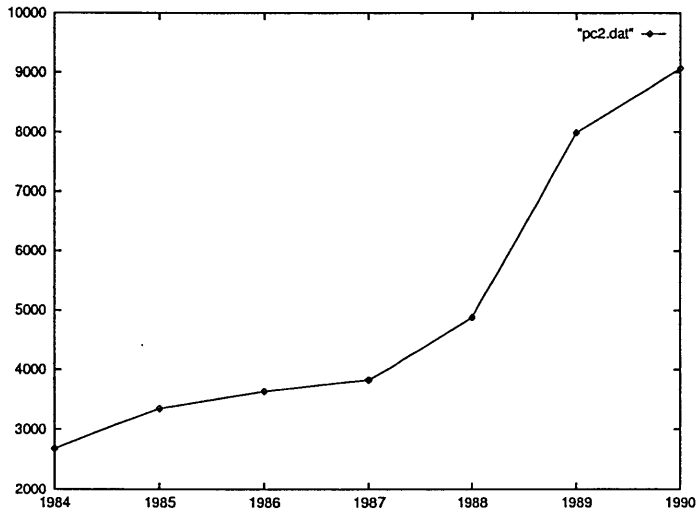

図 11: plot "pc2.dat" with linespoints

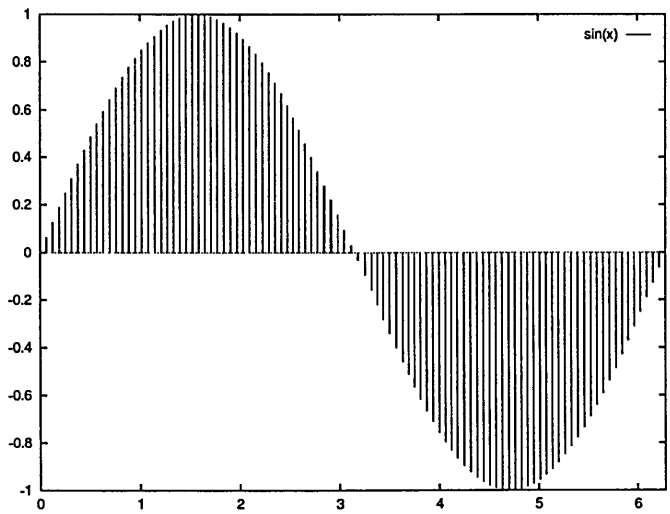

図 12: plot [0:2*pi] sin(x) with impulses

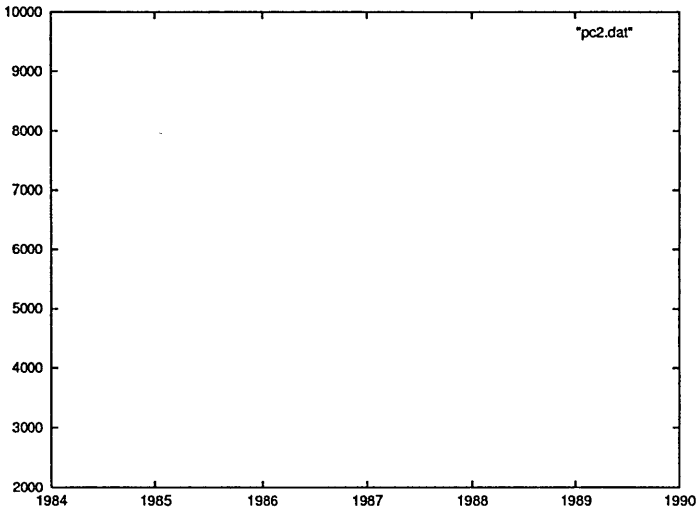

図 13: plot "pc2.dat" with dots

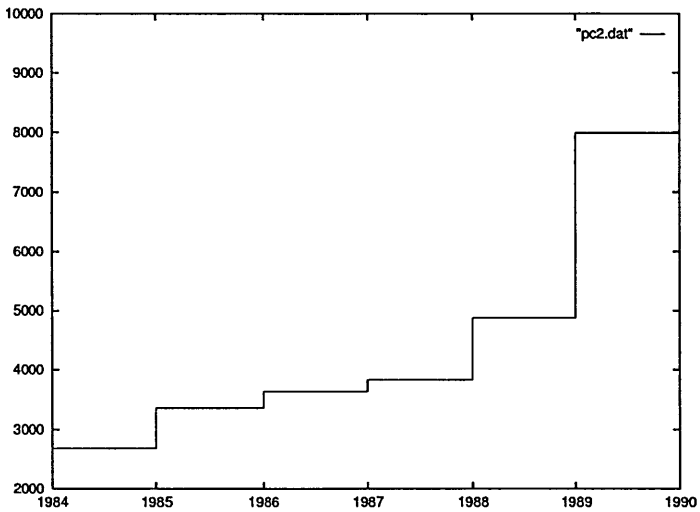

図 14: plot "pc2.dat" with steps

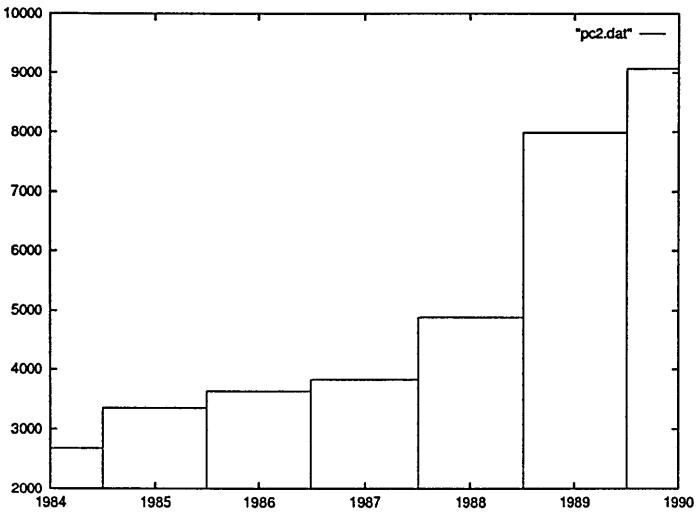

図 15: plot "pc2.dat" with boxes

また，データが 3 列しかないファイルでも errorbars , boxerrorbars を利用することができます．この場合，第 1,2 列が X,Y 座標となり，誤差の下限 = 第 2 列のデータ−第 3 列のデータ，誤差の上限 = 第 2 列のデータ + 第 3 列のデータ として処理されます．

3　プリントアウト

画面を見ながら対話的に作成したグラフをプリンタへ出力するには，グラフの出力先を画面からプリンタへ切替えなければなりません．その手順を以下に示します．

```
...
gnuplot> set term postscript landscape
Terminal type set to 'postscript'
Options are 'landscape monochrome dashed "Helvetica" 14'
gnuplot> set output "|lp -dps -Tps"
gnuplot> replot
gnuplot> set output
```

この作業によって，グラフがオープン端末室の PostScript プリンタに出力されます．再びグラフを画面に表示させるようにするには使っているターミナルタイプを改めて設定してやります．X-window を使用している場合，

```
gnuplot> set term x11
```

とします．

上の例で landscape の代わりに portrait を指定すると印刷方向を 90 度回転させることができます．この場合，A4 用紙にグラフが収まらなくなるので，

```
gnuplot> set size 0.5,0.5
```

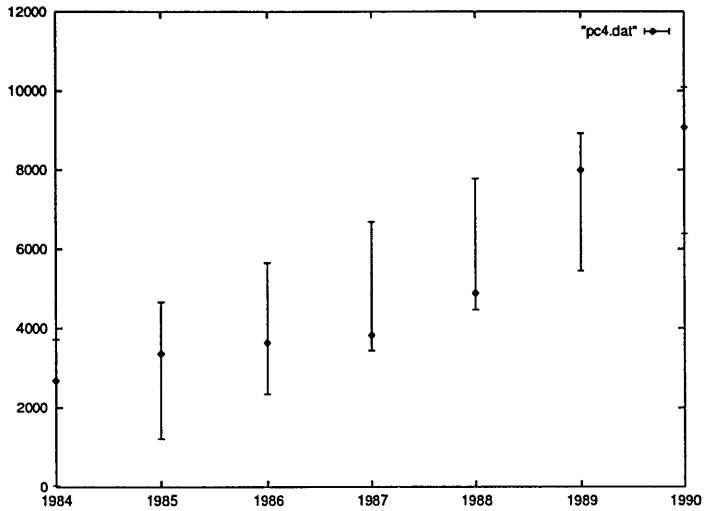

図 16: plot "pc4.dat" with errorbars

などとして，出力グラフのサイズを縮小してから印刷してください．size のあとの数値はそれぞれX方向，Y方向へのグラフの拡大率を示しています．

"|lp -dps -Tps" の代わりに二重引用符で括ったファイル名を指定すると，グラフがプリンタへ出力される代わりに，PostScript コードがそのファイルに書き込まれます．この方法を使えば，いったんファイルに PostScript コードを保存し，ローカルへファイル転送し，手近の PostScript プリンタで出力することもできます．

4 グラフを LaTeX へ取り込む

GNUPLOT の出力を LaTeX へ取り込むには様々な方法がありますが，ここでは二つの方法について説明します．一つは PostScript プリンタとスタイルファイル epsf.sty を組み合わせて用いる方法，もう一つはパソコンで広く用いられているプリンタドライバ dviprt を使う方法です．

4.1 PostScript プリンタを使う

まず，GNUPLOT で LaTeX に取り込みたいグラフを eps 形式で出力させます．希望のグラフを X-Window などに表示させた後，

```
...
gnuplot> set term postscript eps
Terminal type set to 'postscript'
Options are 'eps monochrome dashed "Helvetica" 14'
gnuplot> set output "sample.epsf"
gnuplot> replot
gnuplot> set output
```

とすることで，sample.epsf というeps形式のファイルが生成されます．出力デバイスを画面からPostScriptに変更してreplotコマンドでグラフを再描画させています．

次にLaTeX側で，sample.epsfを取り込むには，先頭行に

```
\documentstyle[epsf]{j-article}
```

のように，epsfオプションを追加します．そして，実際にグラフを表示させたいところで，

```
\begin{figure}
\epsfile{file=sample.epsf}
\caption{epsf 取り込み例 }
\label{fig:epsf}
\end{figure}
```

とします．グラフのサイズを変更したい場合には

```
\epsfile{file=epsfile.epsf}
```

の部分を

```
\epsfile{file=epsfile.epsf,height=8cm,width=15cm}
```

などとすることによって縦横のサイズを指定することができます．また，height,widthの片方だけを指定すると，もう片方はもともとのepsファイルに保存されている縦横比を変更しないように自動的に調整されます．

LaTeXのソースリストをfilename.texとすると，

```
kyu-cc> jlatex filename.tex
```

としてコンパイルした後に，

```
kyu-cc> dvi2ps filename.dvi | lp -dps -Tps
```

でセンターのPostScriptプリンタにグラフ入りのLaTeX出力が得られます．

4.2　パソコンとdviprtを使う

ネットワークを経由してkyu-ccを使っているが，プリントアウトをセンターまで取りに行くのは面倒だという人には，パソコン上で動作するdviprtとの組合せを使うとよいでしょう．MS-DOS上でLaTeXを使っている人のほとんどがプリンタドライバとしてdviprtを使っているのではないでしょうか．最新バージョンのdviprtはtpic specialに対応していますので，PostScriptにはかないませんが，GNUPLOTのグラフが美しく取り込めます．比較的古いバージョンのdviprtをお使いの人はこの際に新しいバージョン[†]にインストールし直すことをおすすめします．

GNUPLOT側でグラフのファイルに出力する方法はPostScriptプリンタに出力する場合とほぼ同様で，画面上にファイル出力させたいグラフを描画した後，

[†]94年10月1日現在の最新バージョンは2.42です．

```
...
gnuplot> set term tpic
Terminal type set to 'tpic'
Options are '40 6 0.100000'
gnuplot> set output "tpicfile.tex"
gnuplot> replot
gnuplot> set output
```

とすることによって，tpicfile.texがファイルに出力されます．このファイルをローカルのパソコンにファイル転送し，これを挿入したいLaTeXのソースのところで，

```
\begin{figure}
\input{tpicfile.tex}
\end{figure}
```

としてパソコン側でコンパイルすれば所望のグラフが取り込めます．パソコン側でのプレビュー，印刷はグラフを取り込まない時の方法と全く同じです．

5 情報源と入手先

GNUPLOTはThomas Williams, Colin Kelleyらによって開発され，その他数多くの人々によってサポートされているフリーソフトウェアです．GNUプロダクトで有名なFree Software Foundationが配布していることもあり，世界中に利用者がいます．USENETのcomp.graphics.gnuplotでは初歩的な質問から高度な利用法，バグ情報などが提供されています．

GNUPLOT FAQというGNUPLOTに関するQ&Aをまとめた文書がrtfm.mit.eduに/pub/usenet/news.answers/graphics/gnuplot-faqというファイル名で保管されています．これをanonymous FTPを使って入手することができます．また，これは定期的にUSENETのcomp.answersにポストされています．

最後にGNUPLOT自身の入手先を御紹介しておきます．上で述べたように，GNUPLOTは世界的に利用されているソフトウェアですので，日本中のさまざまなanonymous FTPサイトに置かれています．九州大学から一番近いFTPサイトはftp.kyushu-u.ac.jpでしょう．/pub/GNU/prepというディレクトリの中にgnuplot-3.5.tar.gzというファイル名で収められています．MS-Windows用を入手したい場合にはftp.kuis.kyoto-u.ac.jpに/WINDOWS/cica/util/gpt35win.zipとして保管されていますので，anonymous FTPして下さい．

参考文献

[1] 佐藤 周行, “UXP/M for the Working Scientists”, 1994, UXPの/usr/local/doc/uxpguide.dviから入手可能.

[2] Thomas Williams et al.“GNUPLOT reference manual”, 1994, UXPの/usr/local/doc/gnuplot.dviから入手可能.