

さくさくGnuplot : 図を貼り付けよう

牧嶋, 直子
九州大学大型計算機センター研究開発部

<https://doi.org/10.15017/1470355>

出版情報 : 九州大学大型計算機センター広報. 32 (3), pp.157-172, 1999-09. 九州大学大型計算機センター
バージョン :
権利関係 :

さくさく Gnuplot

～図を貼り付けよう～

牧嶋 直子†

Gnuplot*¹は計算結果などを図示する場合に便利な、コマンド入力方式の対話的な関数描画プログラム/グラフ作成ツールです。Gnuplotで作成した図はEPSやGIFなど様々なファイル形式で出力でき、以下の様に \LaTeX の原稿やHTML文書に簡単に取り込むことができます。また、OHPシートにカラー印刷することもできます。使用方法の解説の前に、まずは利用例をご覧ください。

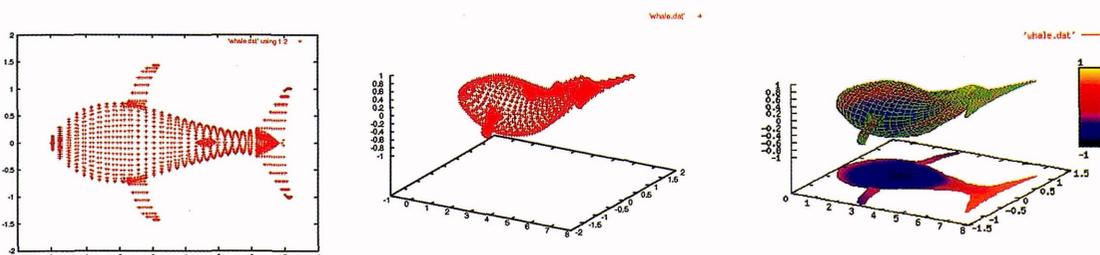


図 1: EPS, GIF 形式での出力

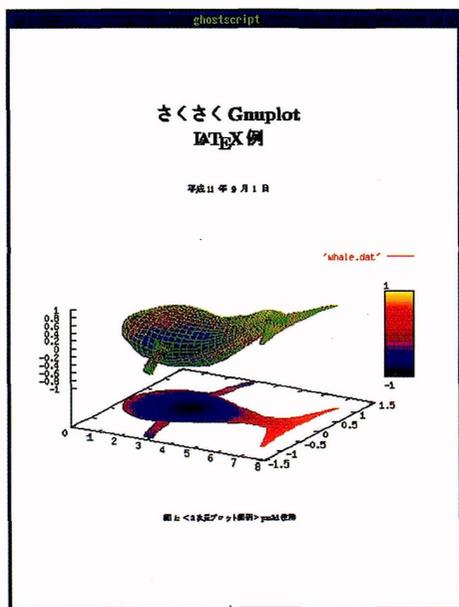


図 2: EPS 形式で \LaTeX 文書に貼り付け

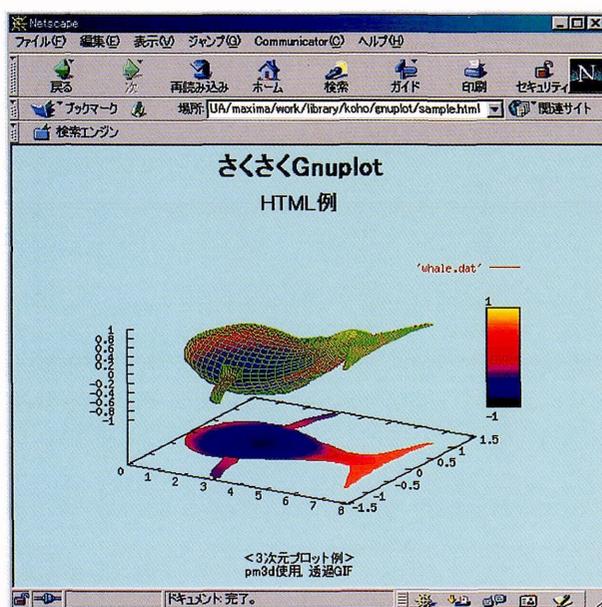


図 3: GIF 形式で HTML 文書に貼り付け

†大型計算機センター研究開発部 E-mail: maxima@cc.kyushu-u.ac.jp

*¹Thimas Williams 氏, Colin Kelly 氏らによって開発されたソフトウェア

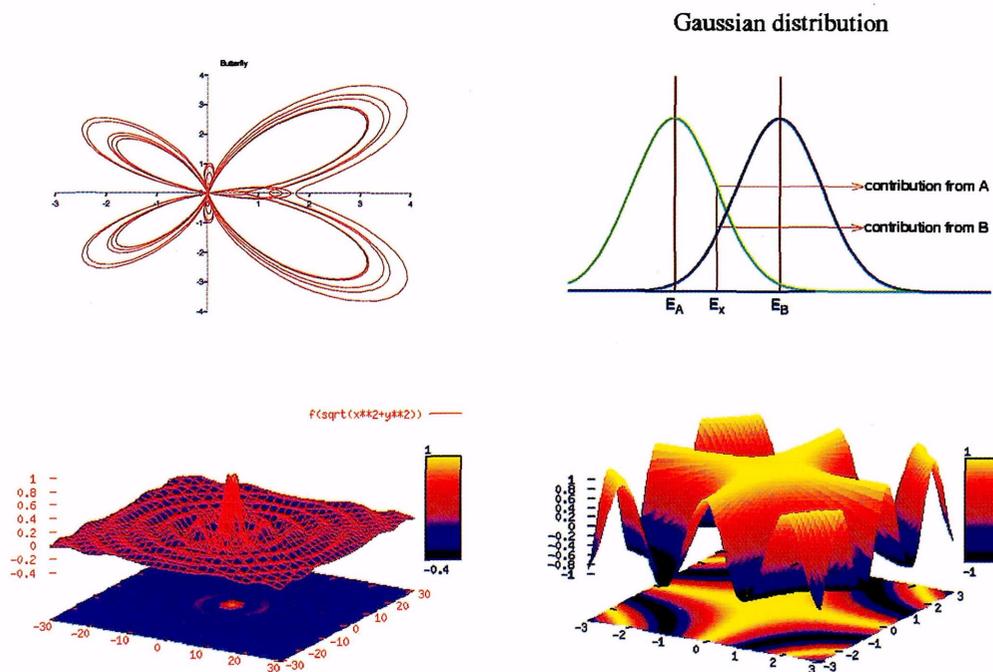


図 4: プロット図

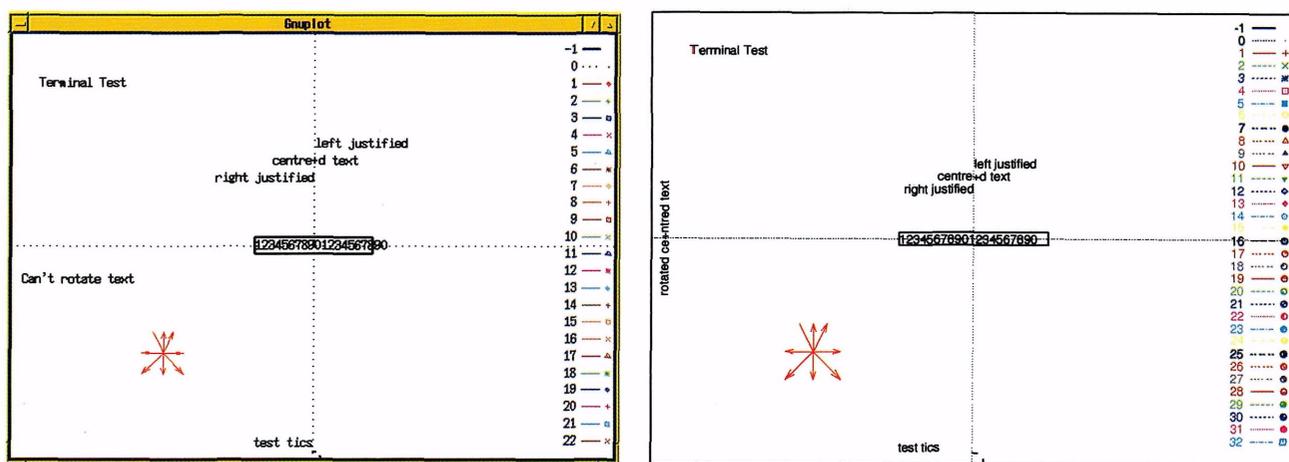


図 5: test コマンドの出力結果 (左: X11, 右: EPS)

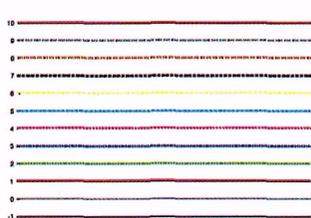


図 6: 使用可能な線種と色 (EPS)

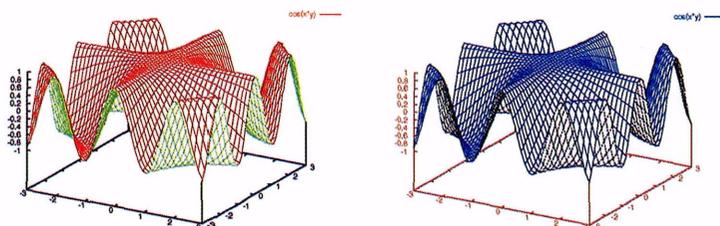


図 7: 色の変更

1 Gnuplot について

Gnuplot の特徴を簡単に述べます。

- 簡単な記述で 2 次元や 3 次元のグラフをプロット*²できます。
- 動作環境は Solaris, Linux, FreeBSD 等の UNIX OS や Windows など色々あり、かつ、フリーソフトウェアとして配布されています。そのため、九州大学大型計算機センターだけでなく、仕事場や自宅などでも使えます。
- PostScript, Tpic special など L^AT_EX で利用可能な各ファイル形式に出力できます。また、GIF 形式のファイルにも出力できますので、HTML 文書にも取り込めます。

Gnuplot の最新のバージョンは 3.7 です (1999 年 9 月現在)。最新版の情報は下記 URL から迎れます。

http://www.cs.dartmouth.edu/gnuplot_info.html

また、ソースコードと各種文書は下記 URL から入手できます。

<ftp://ftp.dartmouth.edu/pub/gnuplot>

2 Gnuplot の起動から終了まで

Gnuplot は対話的なツールですが、Gnuplot のコマンドを羅列したテキストファイルを読み込んで、非対話的に利用することも可能です。この記事では対話的な利用を中心に説明していきます。

2.1 前準備

Gnuplot は九州大学大型計算機センターでは、ライブラリサーバ wisdom*³ (IP アドレス: 133.5.9.9) で利用できます。X Window System を備えた計算機を使えば、遠隔地で画面を表示することができます。そのための環境変数の設定方法を以下に示します。画面に表示させない場合は、wisdom にログインするだけで、この設定は必要ありません。

まず、利用者側の計算機から xhost コマンドを実行します。ここで “userws %” は利用者計算機のプロンプトを表します。このコマンドで、ライブラリサーバ wisdom から利用者側の計算機への画面出力を許可します。

```
userws% xhost +133.5.9.9 ← (wisdom の IP アドレス)
133.5.9.9 being added to access control list
```

次に、telnet または rlogin など wisdom に入り、setenv で環境変数 DISPLAY の設定をします。この設定で、wisdom 側で実行した出力画面を利用者側の計算機に表示できます。

```
wisdom% setenv DISPLAY 133.5.7.68:0.0 ← (利用者計算機の IP アドレスの例)
```

これで起動の準備が整いました。

*²プロット “plot” には「点を結んで曲線を描く」という意味があります。

*³サーバ名は今後変更される可能性があります。

2.2 作業の流れ

Gnuplot の詳細な利用方法を述べる前に、実際の作業例を用いて一連の作業の流れを示します。まず最初に wisdom のコマンドラインに “gnuplot” と入力し Gnuplot を起動します。すると “gnuplot> ” というプロンプトが表示されます。ここに様々なコマンドを入力することで作図などを行います。主なコマンドはプロット方法を変更するための各種設定を行う set コマンドと、実際にプロットを行う plot, splot コマンドです。これらのコマンドの詳細は、第 5 節で説明します。exit あるいは quit と入力すると Gnuplot を終了できます。

```
wisdom% gnuplot          ←起動

      G N U P L O T
      Unix version 3.7
      patchlevel 0 (+1.1.9 1999/06/01)
      last modified Thu Jan 14 19:34:53 BST 1999

      Copyright(C) 1986 - 1993, 1998, 1999
      Thomas Williams, Colin Kelley and many others

      Type 'help' to access the on-line reference manual
      ...
      Send comments and requests for help to <info-gnuplot@dartmouth.edu>
      Send bugs, suggestions and mods to <bug-gnuplot@dartmouth.edu>

Terminal type set to 'x11'      ←起動時の terminal (出力ドライバ) は x11

gnuplot> plot ___              ←plot, splot コマンドで図をプロットする。
                               ___には関数やデータファイル名が入る。

gnuplot> set ___               ←set コマンドでプロット方法を変更する。
                               ___には様々なオプションが入る。

gnuplot> set terminal gif      ←例) terminal (出力ドライバ) を変更する。
Terminal type set to 'gif'
Options are 'small size 640,480 '

gnuplot> set output 'sample.gif' ←例) プロット図の出力先を指定する。

gnuplot> replot                ←plot, splot コマンドの再実行

gnuplot> save 'samle.plt'      ←プロット内容を保存する。

gnuplot> exit                  ←終了

wisdom% ■
```

replot コマンドは、最後に実行した plot あるいは splot コマンドを再実行します。途中で set コマンドで設定を変更した場合は、replot コマンドを実行すると、その変更内容が反映されます。save コマンドは、表示内容をファイル*4に保存します。このコマンドを用いてファイル (例えば sample.plt というファイル名) に保存しておけば、一旦 Gnuplot を終了した後も、同じ内容をプロットできます。これには、load コマンドで sample.plt を呼び出すか、Gnuplot 起動時のコマンドラインオプションに sample.plt を与えるかします。Gnuplot は対話的なプログラムですが、この様にプロット内容が書かれたファイルをコマンドラインオプションにして起動し、非対話的に使う方が便利な場合も多いです。この記事では対話的利用の例を用います。

```
gnuplot> load 'sample.plt'
```

```
wisdom% gnuplot sample.plt
```

*4ファイルはテキスト形式なので、テキストエディタで修正できます。

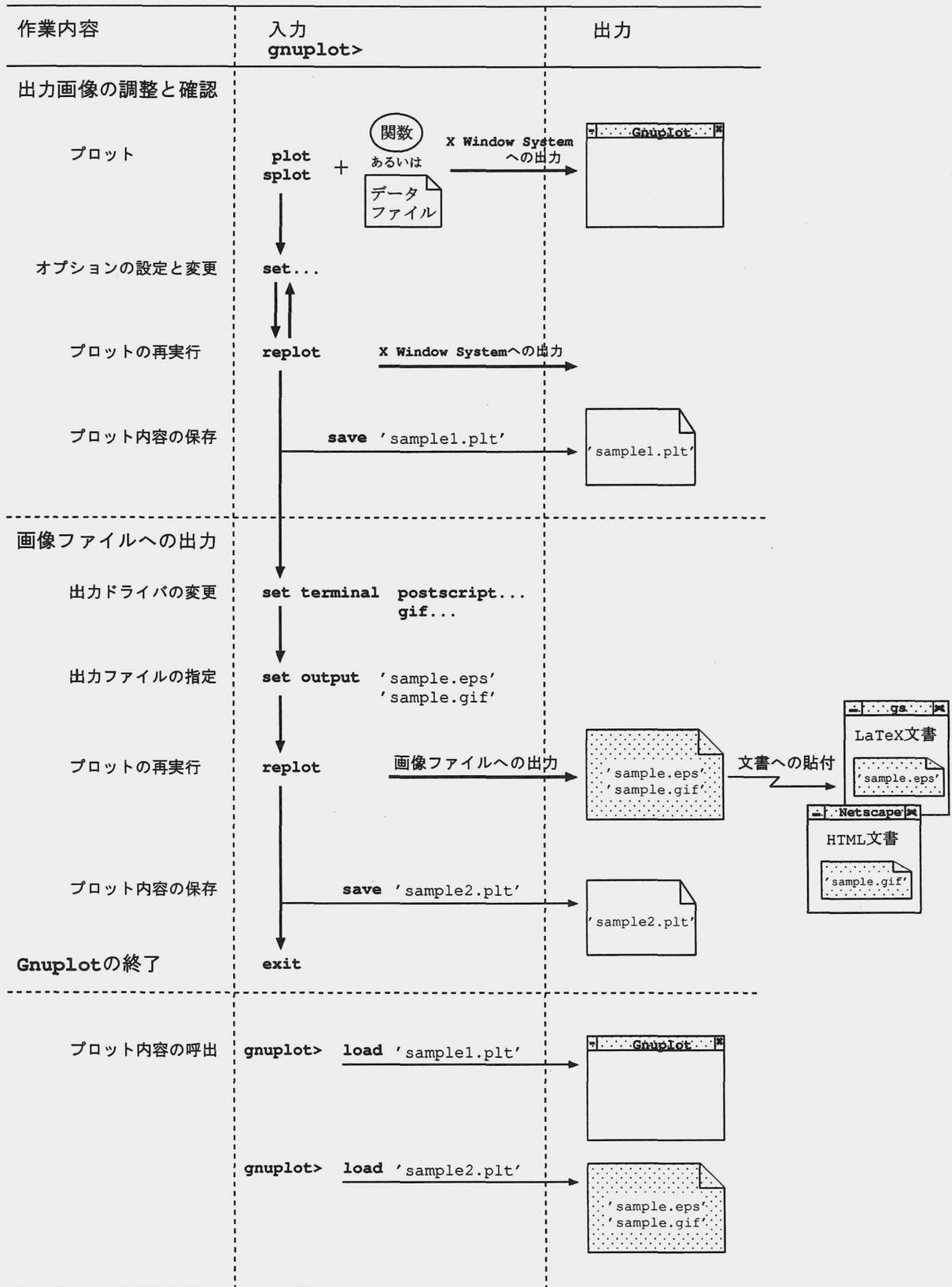


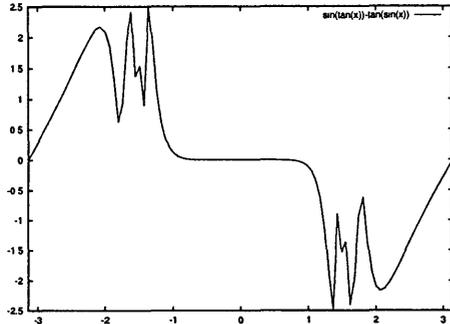
図 8: 作業の流れ

3 プロット図を描画してみる

Gnuplotでは、plotあるいはsplotコマンドによって関数や数値データをプロットできます。ここでは、それぞれ例をあげて説明します。

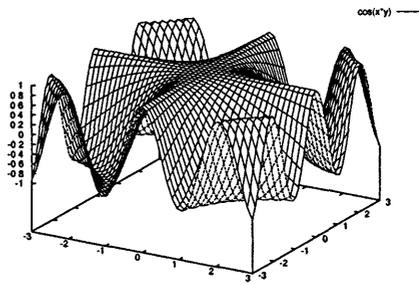
3.1 関数のプロット

まず関数を描画する方法について説明します。plotコマンドで2次元グラフを、splotコマンドで3次元グラフをプロットできます。



$y = \sin(\tan x) - \tan(\sin x)$ を $[-\pi, \pi]$ の範囲で2次元プロット (pi は π を意味します)

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



$z = \cos(xy)$ を $[-3, 3]$ の範囲で3次元プロット

```
gnuplot> set hidden3d          ←隠線処理の指定
gnuplot> set isosamples 45    ←解像度の指定
gnuplot> splot [-3:3] [-3:3] cos(x*y)
```

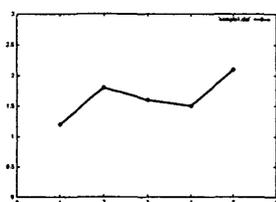
3.2 データファイルのプロット

理系の研究者の場合、Gnuplotを用いて関数を図示することより、実験で得られたデータや解析した結果をプロットすることの方が多くでしょう。そこで、ファイルに格納されたデータを使ってプロットする方法について説明します。

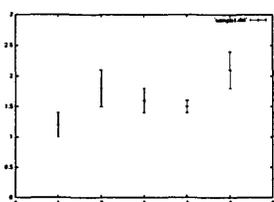
まず、以下の様な3列にわかれたデータがあるとします。このデータが入っているファイル名をsample1.datとします。“#”で始まる行はコメントとして扱われ無視されます。このデータをplotコマンドで2次元プロットしてみます。2種類以上のデータを同時に一つのグラフにプロットするときは、“,”(カンマ)で区切ります。入力行が長いときは“\”(バックslash)を行末に入れると、次の行も継続行とみなされます。また行が長いときは、withはwだけという様に短縮して入力することもできます。

sample1.dat		
#1 列目	2 列目	3 列目
1.0	1.2	0.2
2.0	1.8	0.3
3.0	1.6	0.2
4.0	1.5	0.1
5.0	2.1	0.3

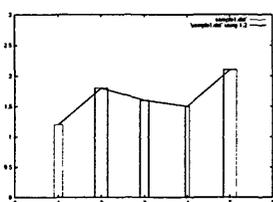
```
gnuplot> set xrange [0:6]
gnuplot> set yrange [0:3]
gnuplot> plot 'sample1.dat' with lines linewidth 5
gnuplot> plot 'sample1.dat' with yerrorbars
gnuplot> plot 'sample1.dat' w boxes, \
> 'sample1.dat' w lines
gnuplot> plot 'sample1.dat' with linespoints, \
> 'sample1.dat' using 1:3 w steps
```



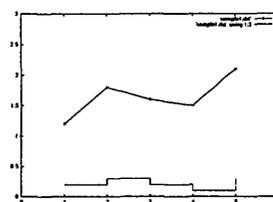
with lines



with errorbars



w boxes, w lines



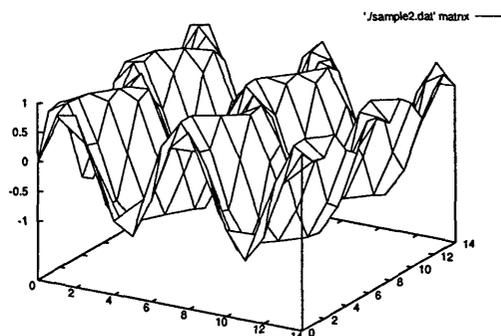
w linespoints, using 1:3 w steps

次に、以下の様な 15×15 の要素をもつ行列をプロットしてみます。行が x 座標、列が y 座標に対応しています。このデータが入っているファイル名を `sample2.dat` とします。このデータを `splot` コマンドで 3 次元プロットしてみます。

sample2.dat

```
0.00 0.84 0.91 0.14 -0.76 -0.96 -0.28 0.66 0.99 0.41 -0.54 -1.00 -0.54 0.42 0.99
0.75 0.96 0.30 -0.64 -0.99 -0.43 0.53 1.00 0.55 -0.40 -0.99 -0.66 0.27 0.96 0.76
0.79 0.94 0.23 -0.69 -0.98 -0.37 0.59 1.00 0.49 -0.47 -1.00 -0.61 0.34 0.97 0.72
0.14 0.91 0.84 0.00 -0.84 -0.91 -0.14 0.76 0.96 0.28 -0.66 -0.99 -0.41 0.54 1.00
-0.69 0.24 0.95 0.78 -0.10 -0.89 -0.86 -0.04 0.82 0.93 0.18 -0.73 -0.97 -0.32 0.63
-0.82 0.04 0.86 0.89 0.10 -0.78 -0.95 -0.24 0.69 0.98 0.37 -0.58 -1.00 -0.50 0.46
-0.28 0.66 0.99 0.41 -0.55 -1.00 -0.53 0.42 0.99 0.65 -0.29 -0.96 -0.75 0.15 0.91
0.61 1.00 0.47 -0.49 -1.00 -0.59 0.37 0.98 0.69 -0.23 -0.94 -0.79 0.09 0.89 0.87
0.84 0.91 0.15 -0.75 -0.96 -0.29 0.65 0.99 0.42 -0.54 -1.00 -0.55 0.41 0.99 0.66
0.40 0.99 0.67 -0.27 -0.96 -0.77 0.13 0.90 0.85 0.01 -0.83 -0.91 -0.15 0.75 0.96
-0.52 0.44 0.99 0.63 -0.31 -0.97 -0.74 0.17 0.92 0.82 -0.03 -0.86 -0.90 -0.11 0.78
-0.84 0.00 0.84 0.91 0.14 -0.76 -0.96 -0.28 0.66 0.99 0.41 -0.54 -1.00 -0.54 0.42
-0.51 0.45 0.99 0.63 -0.32 -0.97 -0.73 0.18 0.92 0.82 -0.04 -0.86 -0.89 -0.10 0.78
0.41 0.99 0.66 -0.27 -0.96 -0.76 0.14 0.91 0.84 0.00 -0.84 -0.91 -0.15 0.75 0.96
0.84 0.91 0.15 -0.75 -0.96 -0.29 0.65 0.99 0.42 -0.54 -1.00 -0.54 0.41 0.99 0.66
```

```
gnuplot> splot 'sample2.dat' matrix with lines
gnuplot> set hidden3d ←隠線処理
gnuplot> replot ←上記 splot コマンドの再実行
```



3.3 複数のグラフのプロット

前述の様に、“,” (カンマ) で区切ることにより 2 種類以上の関数やデータを一つのグラフに同時にプロットできます。この方法で、関数とデータを混在させてプロットすることもできます。これは、理論値と実測値を比較する場合によく利用されます。

4 文書に貼り付ける

Gnuplot で描画した図を文書に貼り付けるには、まず画像ファイルに出力する必要があります。Gnuplot は、PostScript, PPM, PNG, GIF など種々の画像形式をサポートしています。ここでは、 \LaTeX 文書に取り込むための PostScript 形式、HTML 文書に取り込むための GIF 形式のファイルに出力する方法について説明します。

4.1 L^AT_EX 文書に取り込む方法

Gnuplot の出力形式のうちで L^AT_EX に取り込める形式のものは、いくつかあります。ファイルサイズや出力品質を考えた場合、EPS(Encapsulated PostScript)形式で出力するのが最善でしょう。描画データを EPS ファイルに出力する場合は以下の様になります。'color' を付けることでカラー出力できます。省略するとモノクロになります。

```
gnuplot> set terminal postscript eps enhanced color
Terminal type set to 'postscript'
Options are 'eps enhanced color dashed defaultplex.'
gnuplot> set output 'sample2.eps'
gnuplot> set hidden3d
gnuplot> splot 'sample2.dat' matrix with lines
```

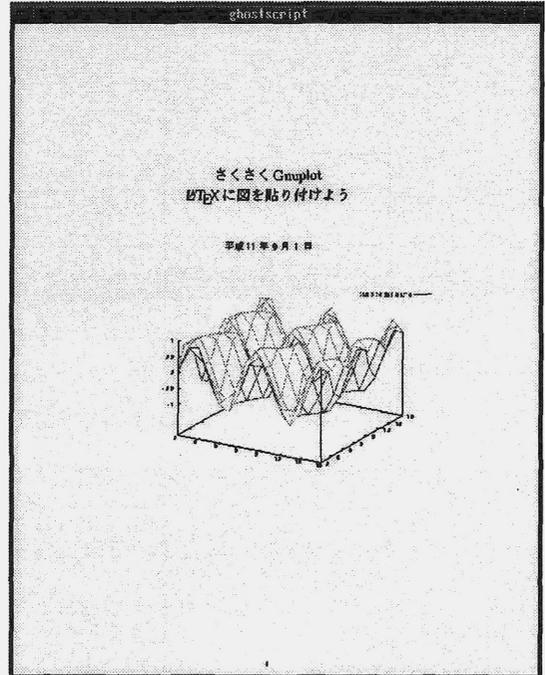
出力された図を L^AT_EX 文書に取り込んだものを gv で表示すると右図の様になります。用いた L^AT_EX のソースは以下のとおりです。L^AT_EX への画像の取り込み方や、platex コマンドでのコンパイル方法については、参考文献 [3]などを参照してください。

L^AT_EX のソース

```
\documentclass{jarticle}
\usepackage[dvips]{color,graphicx}
\begin{document}

\title{さくさく Gnuplot\\{\LaTeX}に図を貼り付けよう}
\maketitle
\includegraphics{./gazou/sample2.eps}

\end{document}
```



4.2 HTML 文書に取り込む方法

次に WWW ページの記述で用いられている HTML 文書で、Gnuplot の図を利用する方法について説明します。Gnuplot が直接生成できる画像形式は PPM と PNG です。Gnuplot のバージョン 3.7 からは直接 GIF 形式のファイルを出力できるようになりました。GIF 形式のファイルに出力するには次の様になります。

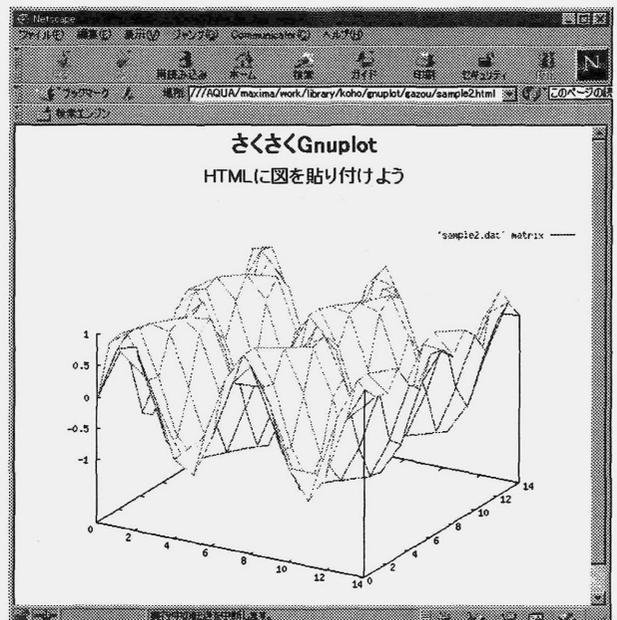
```
gnuplot> set terminal gif
Terminal type set to 'gif'
Options are 'small size 640,480 '
gnuplot> set output 'sample2.gif'
gnuplot> set hidden3d
gnuplot> splot 'sample2.dat' matrix with lines
```

出力された図表を HTML 文書に取り込み Netscape で表示してみると、右図の様になります。用いた HTML のソースは以下のとおりです。

HTML のソース

```
<html>
<body>
<center>
<h1>さくさく Gnuplot</h1>
<h2>HTML に図を貼り付けよう</h2>

</center>
</body>
</html>
```



但し、Gnuplot が生成する GIF 形式のファイルは圧縮されておらずファイルサイズが大きいので画像処理ツールで一度表示し、もう一度 GIF 形式で保存し直してください。また、最初から PNG 形式で図を作成すると、かなりファイルサイズは小さくなります（最近の WEB ブラウザは PNG 形式のファイルに対応しています）。

5 主なコマンドとオプション

Gnuplot には多くのコマンドがあります。その中で主要なコマンドについて説明します。ここに記述されていないコマンドについては、第 5.4 節の様に Gnuplot の help コマンドで参照してください。

コマンド	説明
plot オプション	関数やデータを 2 次元にプロットする。
splot オプション	関数やデータを 3 次元にプロットする。
replot	最後に実行した plot または splot コマンドを再実行する。あるプロットを異なる set オプションで見たり、同じプロットを別の terminal に出力するときに便利。
set オプション 設定値	様々なオプションを設定する。
reset	設定されたオプションを全て元の状態に戻す。
show オプション	set で設定されている（またはデフォルトで設定されている）設定値を表示する。show all で全てを表示する。
save オプション 'ファイル名'	ユーザ定義関数、定数、set で設定するオプション値のいずれかか、これら全てと、最後に実行した plot または splot コマンドを指定したファイルに保存する。オプションには function, variables, set のいずれかが入る。省略すると全てを保存する。
load 'ファイル名'	指定したファイルを読み込んで実行する。
clear	set output で指定されている出力先をクリアする。
help トピック名	オンラインマニュアルを参照する。
test	その出力ドライバ terminal で出力可能な線種や点種などのテスト表示を行う。
pause 時間 '文字列'	指定した時間、またはキーが入力されるまで待つ。-1 を指定するとリターンキーが押されるまで停止する。
cd 'ディレクトリ名'	指定したディレクトリにカレントディレクトリを変更する。
pwd	カレントディレクトリ名を画面に表示する。
shell	シェルを起動する。Gnuplot に戻るには exit を入力する。単一のシェルコマンドなら、コマンドラインの行頭に!の文字をつけることによっても実現できる。

5.1 plot コマンド, splot コマンド

plot コマンドで関数やデータを 2 次元にプロットし、splot コマンドで 3 次元にプロットします。

書式はそれぞれ以下のとおりです。{ } で囲みであるオプションは省略可能です。| は OR（あるいは）の意味です。

```
plot {範囲} [関数|'データファイル' {index リスト} {using リスト}] {title "文字列"} {with スタイル} { , [関数|'データファイル' ... ]
```

“範囲” は $[xmin : xmax] [ymin : ymax]$ の様に指定します。title で凡例につく文字列を設定します。2 種類以上の関数およびデータを一度に描画するときは、“,”(カンマ)で区切ります。

```

splot {範囲} [関数]'データファイル' {binary|matrix} {index リスト} {using リスト} {title '文字
列'} {with スタイル} { , [関数]'データファイル' ... }
    
```

“binary|matrix” は、splot 固有のもので、“データファイル” の形式を指定します。matrix はデータファイルが行列の形式であることを示します。それ以外のオプションは plot コマンドと同様です。

● with オプション

with オプションでグラフの表示スタイルを指定できます。書式は以下のとおりです。

```

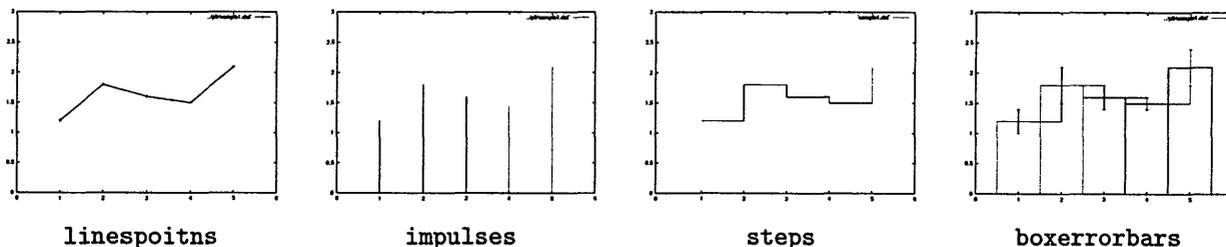
with 表示スタイル {線種} {点種}
    
```

“線種” と “点種” には番号 (数字) が入ります。どの番号がどの点種と線種になっているかは、出力ドライバ terminal の種類によって異なるので、第 5.3 節のように test コマンドで確認してください。

表示スタイル	説明
lines	折れ線。隣り合う点を直線で結ぶ。
points	各点に点を打つ。点種は出力ドライバ terminal によって異なる。
linespoints	lines + points
impulses	鉛直線。x 軸や x-y 平面からグラフの点に垂直に点を打つ。
dots	各点に小さめの点を打つ。
steps	点から点を階段状につなぐ。
errorbars	二次元グラフにおいて、各点に誤差指示線を引く。
boxes	二次元グラフで与えられた x 座標を中心に、棒グラフを描画する。
boxerrorbars	errorbars+boxes

with オプションを使用した例を以下に示します。

この例は第 3.2 節で例にあげたデータファイル “sample1.dat” を様々な表示スタイルで描画したものです。



● using オプション, index オプション

データファイルの中でプロットする列を選択するには using オプションを使います。また、一つのデータファイルの中に複数のデータセット (データの集まり) を入れておきたい場合があります。この場合、各データセットの間に 2 行以上の空白行を挿入することでデータを分割できます。複数のデータセットの中から任意のものを選択するには index オプションを使います。書式はそれぞれ以下のとおりです。

```

using y | x : y ← plot の場合
    
```

但し、表示スタイルによっては、必要な列数、書式が異なります。例えば、errorbars や boxerrorbars でプロットする場合は 3 列以上のデータを必要とします。

```

using z | x : y : z ← splot の場合
    
```

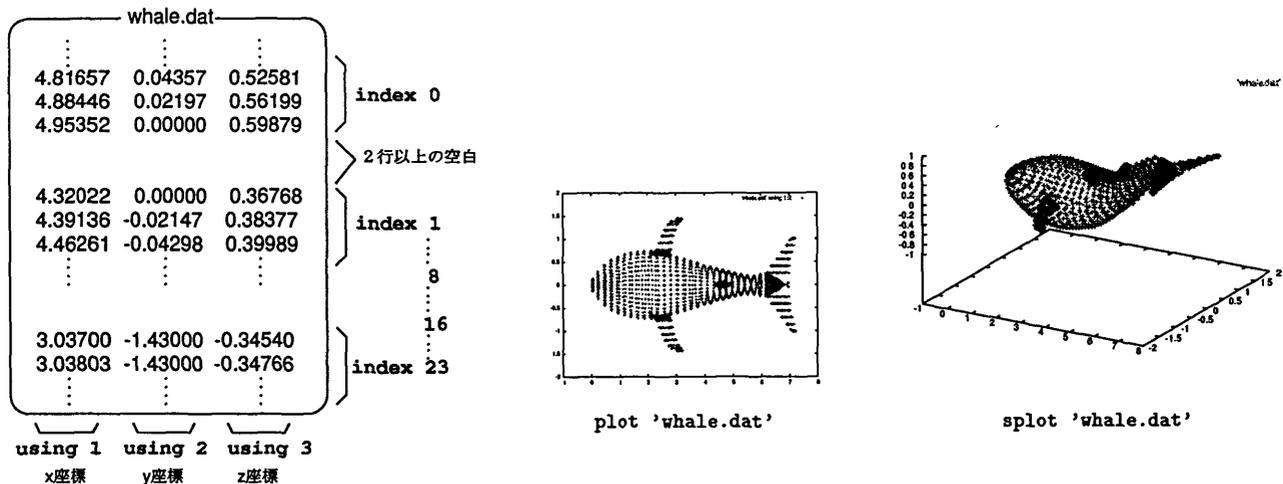
x, y, z には、それぞれ 1 以上の数値が入ります。

```
index m { : n { : p } }
```

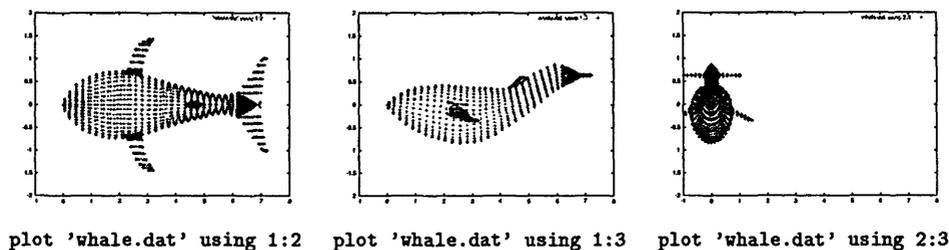
m , n , p には、それぞれ 0 以上の数値が入ります。“ $\{m:n\}$ ” と指定すると、 $m \sim n$ の範囲のデータセットを選択できます。“ $\{m:n\}:p$ ” と指定すると、 $m \sim n$ の範囲で p の数値だけ飛ばしながら、 $m, m+p, m+2p \dots$ というようにデータセットを選択できます。

using, index オプションを使用した例を以下に示します。

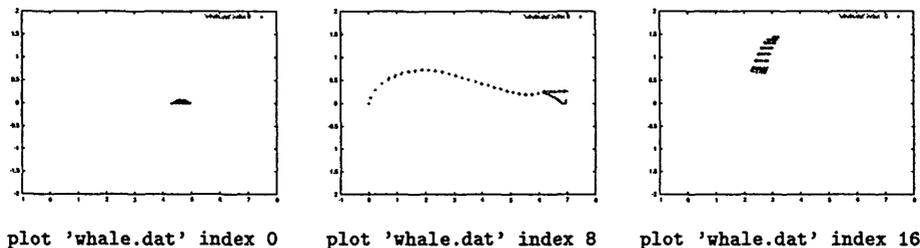
この例は Gnuplot のデモンストレーションとして提供されているデータファイルの中の “whale.dat” という鯨のプロットデータを使用しています。このデータは 24 個のデータセットから構成されており、1 列目に x 座標、2 列目に y 座標、3 列目に z 座標の値が格納されています。

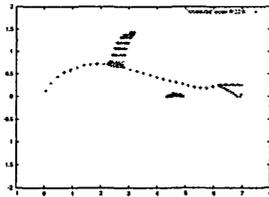


まず、using オプションを使用した例を示します。“using 1:2” で x - y 平面、“using 1:3” で x - z 平面、“using 1:2” で y - z 平面に投影した図をプロットできます。

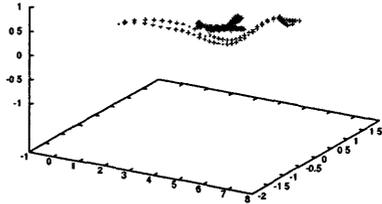


次に、index オプションを使用した例を示します。“index 0”, “index 8”, “index 16” で、鯨のかたちを構成するデータの一部ずつをプロットできます。“index 0:23:8” で、その 3 つを合わせたデータをプロットできます。





plot 'whale.dat' index 0:23:8



splot 'whale.dat' index 0:23:8

5.2 set コマンド

Gnuplot には、数多くのオプションが用意されています。オプションはプロットを行う上での各種設定のための環境変数です。Gnuplot を設定するには `set` コマンドを使用します。オプションの現在値を表示するには、`show` コマンドを使用します。`set` コマンドでの設定・変更は、その後に実行する `plot`, `splot` コマンド、または `replot` コマンドに反映されます。

set コマンドで設定可能な主なオプションについて説明します。

オプション	説明	no
<code>[x y z]range [min:max]</code>	軸の最大値・最小値を設定する。[:max], [min:] の様にどちらかの値だけでも指定可能。	
<code>title '文字列'</code>	図の表題を設定する。	○
<code>key location</code>	凡例の表示場所を変更する (デフォルトは右上 right top)。location には座標 x,y を直接指定するか, left, right, top, bottom, outside, below のいずれかが入る。	○
<code>[x y z]label '文字列'</code>	軸のラベルを設定する。	
<code>size xscale,yscale</code>	Gnuplot の出力を拡大・縮小する。xscale は x 軸方向, yscale は y 軸方向の倍率。	
<code>grid</code>	グラフに格子上の破線を表示する。	○
<code>arrow from x,y,z to x,y,z</code>	矢印を表示する。矢印の矢じり部分を非表示にしたい場合は nohead をつける。	○
<code>terminal</code>	グラフを出力する「端末」を設定する。	
<code>output 'ファイル名'</code>	グラフの出力先を指定したファイル名に設定する。	
<code>hidden3d</code>	グラフの表示が曲面による描画から面による描画にかわり、隠線処理が行われる。	○
<code>isosamples 数値</code>	3次元で関数表示するときの線の数の設定。すなわち、格子の密度を設定する。	
<code>view x, y, z</code>	3次元表示での視点を変更する。	
<code>contour location</code>	等高線を表示する。location には base, surface, both のいずれかが入る。base は xy 平面, surface はグラフの表面, both は両方を示す。デフォルト (省略値) は base。	○
<code>surface</code>	面の孤立線や網を表示する。nosurface は面を非表示にする。等高線を単独で表示したいときに便利。	○

no の項目が○になっているオプションは、接頭語として “no” を付けることができ、その設定を無効化できる。

hidden3d 以降は splot 用のオプションです。

● terminal オプション

terminal オプションで出力ドライバを変更できます。PostScript 形式、GIF 形式のファイルに出力するための書式をそれぞれ以下に示します。

○ PostScript 形式での出力

```
set terminal postscript {landscape|portrait|eps} {enhanced|noenhanced} {color|monochrome}
{solid|dashed} {フォント名} {フォントサイズ}
```

指定	説明
landscape	横長にする。
portrait	縦長にする。
eps	Encapsulated PostScript (EPS) 形式にする。
color	カラー出力する。
monochrome	モノクロ出力する。
solid	実線でプロットする。
dashed	破線でプロットする。
フォント名	フォントセットを指定する。
フォントサイズ	フォントサイズを指定する。

デフォルトの設定は “landscape noenhanced monochrome dashed defaultplex "Helvetica-Ryumin" 14”。

○ GIF 形式での出力

```
set terminal gif {transparent} {interlace} {small|medium|large} {size <x>,<y>} {<color0> ...}
```

指定	説明
transparent	透過 GIF にする。
interlace	インターレース GIF にする。
small	文字のフォントを変更する。
medium	small は 6×12 pixels, medium は 7×13 で Bold タイプ,
large	large は 8×16.
size <i>x,y</i>	<i>x, y</i> にはピクセル値を入れる。デフォルト値は 640×480.

透過とは画像の中で使われている色を透明にして、背景に溶け込ませてしまうように見せるものです。インターレースとは画像が表示される際に、全体を徐々に表示させていく方法で、大きな画像を使った場合に最後まで表示しなくても内容がわかるようにするものです。

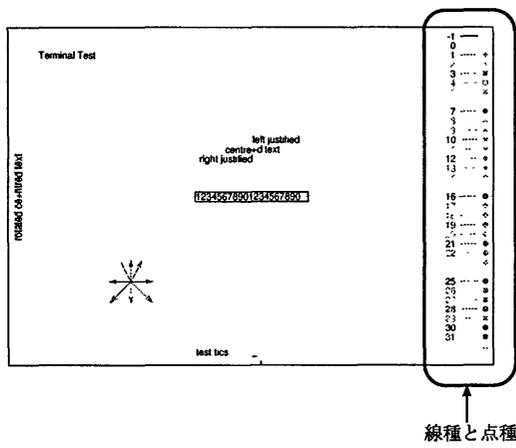
デフォルトの設定は “small size 640,480”。

5.3 test コマンド

Gnuplot では出力形式によって、描画の線種や点種が変わります。例えば、図 5 に示してある様に X Window System と PostScript 形式に出力したのでは線種や記号が異なっているのがわかります。その違いは線種や点種、色などのテスト表示を行う test コマンドで確認できます。

```
gnuplot> set terminal postscript eps enhanced
gnuplot> set output 'test.eps'
gnuplot> test
```

この'test.eps'を見ると、右図の様になります。
この図に表示されている番号は、その横にある線種と点種、色に対応しています。この番号を指定することで、出力するグラフの線種などを変更できます。



また、すでに PostScript 形式で出力されたグラフの色を変更することもできます。Gnuplot が PostScript のドライバを使って出力できる線の色と種類は図5と図6に示されているとおりです。これ以外の色を出力したい場合は、PostScript ファイルを直接 mule などのテキストエディタで書き直す方法があります。

出力された PostScript ファイルの中身を見ると、/LT0 ~ /LT8 の様に線種が定義されている部分があるはずですが、この部分が図6にある線種の1から9に対応しています。そして各行の“1 0 0”といった数値はカラーモデル RGB(Red, Green, Blue)を示しています。例えば、LT0のところではRだけが1なので赤になります。同様に LT1 が緑、LT2 が青です。つまり、この RGB 値を変更すれば、色を変更できます。

PostScript ファイルの中身

```

%!PS-Adobe-2.0 EPSF-2.0
%%Title: ./gazou/2_color.eps
%%Creator: gnuplot 3.7 patchlevel 0 (+1.1.9 1999/06/01)
%%CreationDate: Thu Aug 12 15:19:08 1999
%%DocumentFonts: (atend)
%%BoundingBox: 50 50 410 302
%%Orientation: Portrait
%%EndComments
/gnudict 256 dict def
gnudict begin
/Color true def
...
/LTb { BL [] 0 0 0 DL } def
/LTa { AL [1 dl 2 dl] 0 setdash 0 0 0 setrgbcolor } def
/LT0 { PL [] 1 0 0 DL } def
/LT1 { PL [4 dl 2 dl] 0 1 0 DL } def
/LT2 { PL [2 dl 3 dl] 0 0 1 DL } def
/LT3 { PL [1 dl 1.5 dl] 1 0 1 DL } def
/LT4 { PL [5 dl 2 dl 1 dl 2 dl] 0 1 1 DL } def
/LT5 { PL [4 dl 3 dl 1 dl 3 dl] 1 1 0 DL } def
/LT6 { PL [2 dl 2 dl 2 dl 4 dl] 0 0 0 DL } def
/LT7 { PL [2 dl 2 dl 2 dl 2 dl 2 dl 4 dl] 1 0.3 0 DL } def
/LT8 { PL [2 dl 2 dl 2 dl 2 dl 2 dl 2 dl 4 dl] 0.5 0.5 0.5 DL } def
...

```

この方法以外に、“set terminal tgif”とし、Tgifのファイルとして出力したものを色付けしたあと、EPSにする方法もあります。但し、複雑な3次元プロット図などで線や網の数が多い場合は細かい範囲指定ができず、色の変更は難しくなります。PostScript形式のファイルと同様に、Tgifのファイルを見ると、“red”や“black”など色を指定している箇所がありますので、これをUNIXのsedコマンドなどで一括置換しても、色の変更はできます。

5.4 help コマンド

Gnuplot にはヘルプ機能があり、help コマンドによってオンラインマニュアルを参照できます。help コマンドを実行すると、Gnuplot についての簡単な説明が表示されたあと、細目名 “Topic” の入力を要求されます。さらに、さらなる細目名 “Subtopic” がある場合は、その入力が要求されます。help に続いて、“Topic” と “Subtopic” を一度に入力しても表示できます。

```
gnuplot> help      ← help コマンドの実行
'gnuplot' is a command-driven interactive function and data 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 abbreviated
...
Help topics available:
  batch/interactive bugs      commands      comments
  coordinates      copyright      environment      expressions
  glossary      graphical      introduction      line-editing
...
Help topic: set      ←例) コマンド set の説明を表示
The 'set' command can be used to sets _lots_ of options. No screen is
drawn, however, until a 'plot', 'splot', or 'replot' command is given.
...
Subtopics available for set:
  angles      arrow      autoscale      bar
  bmargin      border      boxwidth      clabel
  clip      cnttparam      contour      data
...
Subtopic of set: angles      ←例) オプション angles の説明を表示
By default, 'gnuplot' assumes the independent variable in polar graphs is in
units of radians. If 'set angles degrees' is specified before 'set polar',
then the default range is [0:360] and the independent variable has units of
degrees. This is particularly useful for plots of data files. The angle
setting also applies to 3-d mapping as set via the 'set mapping' command.

Syntax:
  set angles {degrees | radians}
  show angles
...
gnuplot> help set angles      ←例) help コマンドの後にトピックを入力
By default, 'gnuplot' assumes the independent variable in polar graphs is in
units of radians. If 'set angles degrees' is specified before 'set polar',
then the default range is [0:360] and the independent variable has units of
...
```

また、ライブラリサーバ wisdom には Gnuplot のドキュメントを TeX の DVI ファイルと PS ファイルとして置いてあり、以下のコマンドで参照できます。

```
wisdom% xdvi /usr/local/gnuplot/share/gnuplot.dvi
wisdom% gv /usr/local/gnuplot/share/gnuplot.ps
```

6 pm3d

splot コマンドで 3 次元グラフをプロットする際に、pm3d という拡張機能を使えます。オプションとして、この pm3d を設定すると、3 次元グラフの表面に色づけでき、*Mathematica*^{*5}にもひけをとらない作図が可能となります。pm3d を組み込んだ Gnuplot は gnuplot コマンドとは別に、“gnuplot-pm3d” というコマンドを wisdom のコマンドラインに入力し起動できます。使い方は前述までの Gnuplot と同じです。これは、Gnuplot3.7 の β

*5 Stephen Wolfram 氏により開発された汎用数学処理システム。参考文献 [3] 参照。

版に組み込んでおり、日本語も使えない様なので、別コマンドとして提供しています。出力ドライバ `terminal` は、`gif` もしくは `postscript` を利用できます。但し、X Window では表示できません。使用方法に関しては“<http://www.sci.muni.cz/~mikulik/gnuplot.html>”を参考にしてください。

```
wisdom% gnuplot-pm3d
```

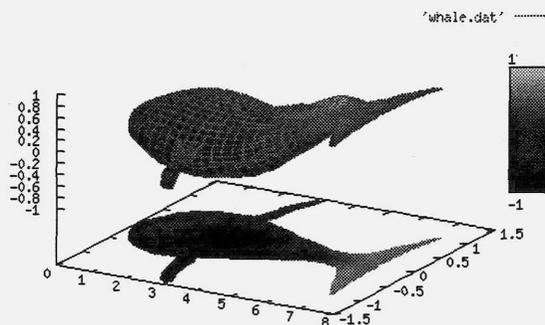
書式は以下のとおりです。

```
set pm3d at location
```

`location` には、`s`, `t`, `b` が入ります。“`sb`”の様に2つ以上の指定もできます。`s` は“`surface`”でグラフの表面、`t` は“`top`”で x - y 平面の上面、`b` は“`bottom`”で x - y 平面の下面を意味します。

`pm3d` オプションを使用した例を以下に示します。この例は第 5.1 節で例にあげたデータ“`whale.dat`”を使用しています。

```
gnuplot> set pm3d at bs
gnuplot> set hidden3d
gnuplot> set terminal gif transparent
gnuplot> set output "pm3d_whale.gif"
gnuplot> splot 'whale.dat' with lines
```



7 印刷

Gnuplot で作成した図そのもの、もしくはその図を L^AT_EX 文書に取り込んだ PostScript 形式のファイルは、九州大学大型計算機センター 2 階に設置されているネットワークプリンタ (モノクロ印刷)、A4 判フルカラーポストスクリプトプリンタ (カラー印刷) に出力できます。これらのプリンタへの出力方法は、センターのホームページ“<http://www.cc.kyushu-u.ac.jp/library/TeX/TeX.html>”や参考文献 [3]などを参考にしてください。

参考文献

- [1] 森保 洋, “GNU PLOT の利用法 -2 次元グラフを中心に-”, 九州大学大型計算機センター広報, Vol.28, No.2, pp.100-128(1995).
- [2] 渡部善隆, “《急ぐ人のための》Mathematica Graphics 入門”, 九州大学大型計算機センター広報, Vol.28, No.2, pp.100-128(1995).
- [3] 牧嶋直子, “わくわく pL^AT_EX 2_ε ~ graphics パッケージをあけてみよう~”, 九州大学大型計算機センター広報, Vol.31, No.1, pp.19-33(1998).
- [4] 乙部巖己+江口庄英 著, “pL^AT_EX 2_ε for WINDOWS Another Manual Vol.2 Extended Kit”, ソフトバンク (1997).