

グラフ描画ソフト gnuplot を使う

目次

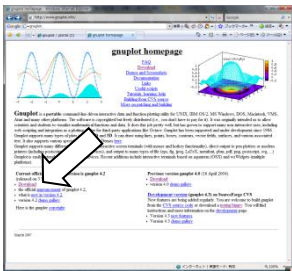
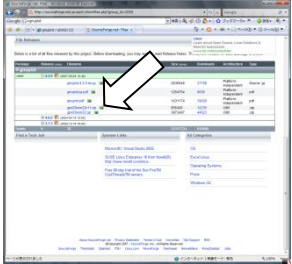

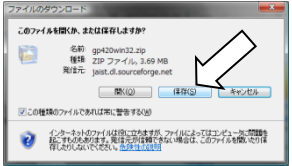
1. What is gnuplot ?	1
2. Download and Install gnuplot	1
3. How to use gnuplot	4
plot コマンドで 2 次元グラフが簡単に描けます.	4
データファイル text.txt を使って折れ線グラフを描くこともできます.	4
splot コマンドで 3 次元のグラフも簡単に描くことができます.	5
8 つの頂点データを使って立方体を描く	6
4. C 言語から gnuplot を操作することもできます.	7
6. gnuplot Tips	10
(1)パッケージに添付されてくる demo ファイルを見るといろいろなグラフの描き方が分かる.	10
(2)入力ファイルの中で if 文が使える. useps = 0 if (useps > 0) else	10
(3) 陰関数を直接描画できませんが, 工夫すれば可能.	10
(4)set multiplot を使えば, 一つのページに複数のグラフが描けます.	10
(5)Window 出力画面でのショートカットキー	10
(6)Terminal の設定と出力グラフの大きさ	10
(7) using オプション	11
(8)文字列操作 (help の説明文を日本語で解説)	11
(9)文字列変数を使ってフォント指定すると便利	12
(10)三項演算子との組み合わせ	12
(11)データファイルと every オプション	12
(12)画像 RAW ファイルを表示する	12
(13)棒グラフ histogram	13
(14)ベクトル描画 with vectors	13
(15) 凡例のタイトル指定 plot コマンドと title オプション	14
(16)データファイルを使って等高線グラフ	14
(17) 軸の設定	15
(18)3 項演算子ならびに Clip を使って特定の範囲のみ描画する方法	16
(19) with labels オプション, 3 列データファイルを使って, 指定した座標にラベルをプロットする.	16
(20) アニメーション制作	16
(21) Test パターンとカラー指定	18
7. 参考になる Web サイトと図書	19
gnuplot homepage	19
日本語解説サイト	19
参考図書	19

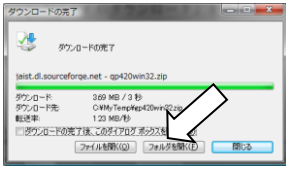

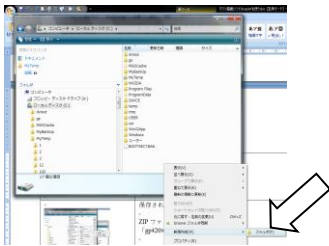
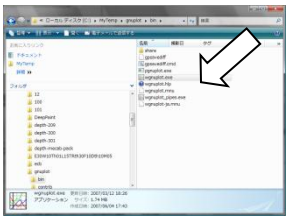
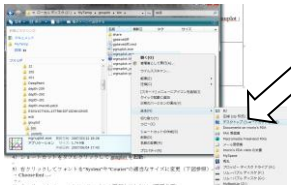

1. What is gnuplot ?

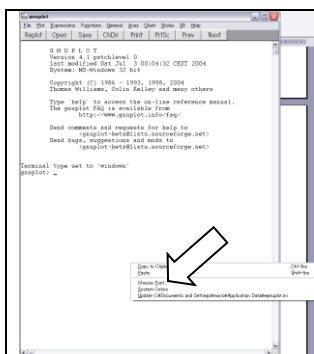
科学論文に掲載可能な高品質なグラフを作成できる無償ソフトです。読み方は「ニュープロット」で、gnuplot をすべて小文字で表記するそうです。1986年から開発が続いています。グラフ作成には描画コマンドを gnuplot プロンプトに続けて入力します。Excel を使ったグラフ作成に不満を持っている方には是非おすすめしたいソフトの一つです。

2. Download and Install gnuplot

インストール方法は簡単で、圧縮されたファイルをダウンロードし、特定のプログラムフォルダに解凍するだけです。以下に手順を示します。

	<p>Google で「gnuplot」とタイプすると良い。</p> <p>http://www.gnuplot.info/</p> <p>画面下の方の 「Download」ボタンをクリックする。</p>
	<p>gp420win32.zip をクリックする。</p> <p>(補足) Win32 と書かれているのは Windows 用のソフトという意味。さらに X11 が付いているものがあるが、Unix 系のソフトを Windows 上で動かしている方は、こちらのものでも良い。よく意味が分からない人は、x11 は選択しない。</p>
	<p>画面上部に 「セキュリティのため... ファイルのダウンロードがブロックされました。」と出た部分をクリックする。</p> <p>現れたメニューから 「ファイルダウンロード」をクリックする。</p>
	<p>「保存」を選択し、適切なフォルダを選択して</p>

	<p>真ん中のボタン 「フォルダを開く」を選択する。</p>
	<p>保存されたフォルダに移動した後、 ZIP ファイル形式で圧縮された 「gp420win32.zip」ファイルを解凍する。 解凍をすると 「gnuplot」フォルダが出る。</p>
	<p>「マイコンピュータ」のcドライブに 「Software」フォルダを新規に作成する。 作成した「Software」フォルダに、先ほど解凍した「gnuplot」 フォルダをコピーする。</p>
	<p>「gnuplot」フォルダ配下の「bin」フォルダにある 「wgnuplot.exe」ファイルの上で右クリック</p>
	<p>右クリックして現れるメニュー「送る」の 「デスクトップにショートカットを作成」をクリックする。</p>
	<p>デスクトップ領域に、wgnuplot のショートカットが作成さ れていることを確認する。</p>
	<p>ショートカットをダブルクリックして起動する。 gnuplot をはじめて使用する人は、文字がつぶれて読めな</p>

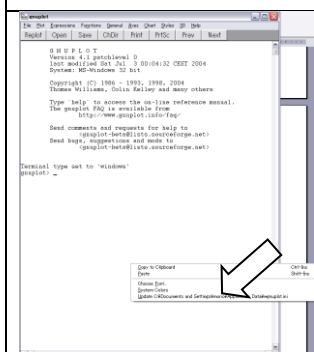


い状態になります。

白い背景を右クリックして表示されるメニューから

「Choose Font...」を選択してください。

「Courier」の「12pt」を選択して「OK」ボタンをクリックしてください。好きなフォントを選んでも良いです。



再度、背景を右クリックして表示されるメニューから、

「Update wgnuplot.ini」を忘れずに選択してください。

この操作は、フォントの設定をコンピュータに記憶させるためのものです。

以上で、インストール作業は終了です。

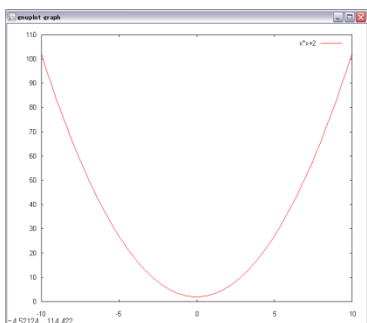
3. How to use gnuplot

plot コマンドで 2 次元グラフが簡単に描けます.

数式で表された $f(x) = x^2 + 2$ のグラフを描いてみましょう.

gnuplot > に続いて ”plot x*x+2” とタイプして Enter キーを押す.

gnuplot > plot x*x+2

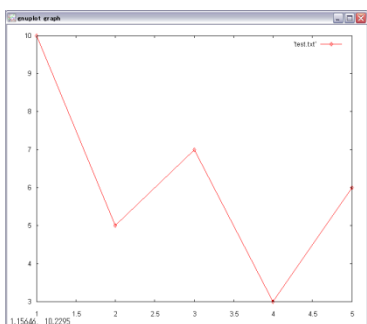


データファイル test.txt を使って折れ線グラフを描くこともできます.

gnuplot > plot 'test.txt' with linespoints

test.txt

1	10
2	5
3	7
4	3
5	6



(補足)

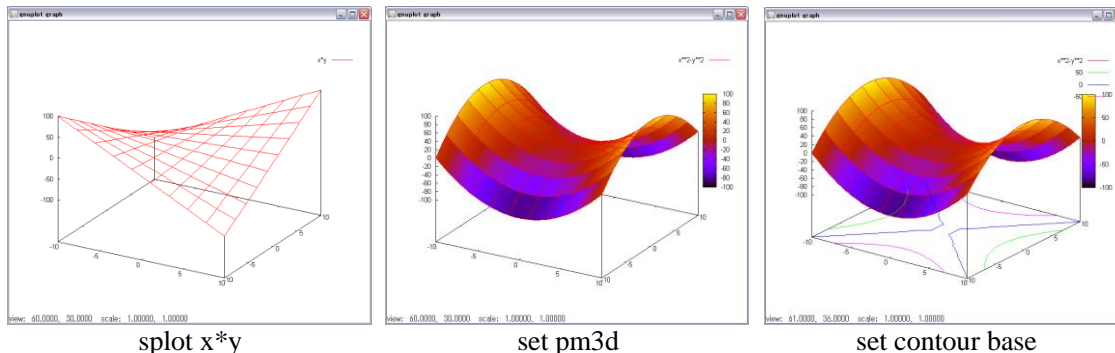
ファイルパスには絶対パスあるいは相対パスが指定できます. 作業ディレクトリを事前に変更しておくとう便利です.

(例) cd 'd:¥data' と入力すると作業ディレクトリが d:¥data に変更できます.

splot コマンドで 3 次元のグラフも簡単に描くことができます。

$f(x, y) = xy$ のグラフを描いてみましょう。その他、局面色付けや等高線表示も簡単にできます。

gnuplot> splot x*y



splot x*y

set pm3d

set contour base

(補足)

2次元同様、データファイルを使ってグラフを描くことができます。xyz 3列のデータファイルを用意してください。データブロックは空行で区切られています。空行で区切っていない場合は `set dgrid3d` コマンドでグリッドデータを作ります。

datafile.txt を使って 3D プロットする

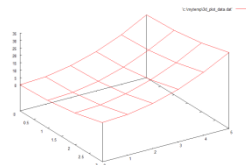
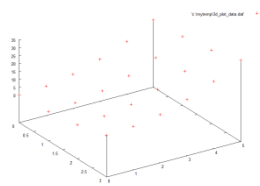
gnuplot> splot 'datafile.txt'

さらに with lines オプションで、グリッドを引く

gnuplot> splot 'datafile.txt' with lines

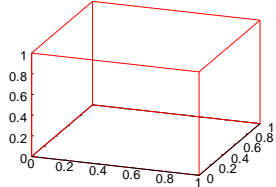
datafile.txt の内容

#x	y	z
0	0	0
0	1	1
0	2	4
0	3	9
0	4	16
0	5	25
1	0	1
1	1	2
1	2	5
1	3	10
1	4	17
1	5	26
2	0	4
2	1	5
2	2	8
2	3	13
2	4	20
2	5	29
3	0	9
3	1	10
3	2	13
3	3	18
3	4	25
3	5	34



8つの頂点データを使って立方体を描く

1辺の長さが1の立方体の上面と底面それぞれについて、4つの頂点を表すデータをグラフ
描画用ファイルに保存する.

cube.dat	gnuplot スクリプト	描画結果
<pre>#cube data #top face 0 0 1 1 0 1 1 1 1 0 1 1 0 0 1 (ここに空行を1行だけ入れる.) #bottom face 0 0 0 1 0 0 1 1 0 0 1 0 0 0 0</pre>	<pre>set xyplane 0 splot 'cube.dat' w l</pre>	

4. C 言語から gnuplot を操作することもできます.

必要なものはパイプ作成用ファイルポインタと, 4 種類の関数 `_popen()`, `fprintf()`, `fflush()`, `_pclose()` です.

```
#include "stdio.h"

void main(void)
{
    FILE *gid; // gnuplot 操作用ファイルポインタの宣言

    //_popen 関数で gnuplot へのパイプを作成する. 使い方は fopen 関数と同じ
    // (注意) _popen()で呼び出すファイルは, wgnuplot.exe ではなく,
    //pgnuplot.exe である. pgnuplot.exe へのファイルパスを指定する.

    //gnuplot をインストールしたフォルダが c:\gnuplot である場合の_popen()
    gid = _popen("C:/gnuplot/bin/pgnuplot","w");

    //パイプを作成して得たポインタ gid を通して, gnuplot コマンドを流し込む
    fprintf(gid, "plot x%n");
    fflush(gid); //バッファからコマンドを強制的に吐き出させる

    fprintf(gid, "pause -1"); //キーが押されるまで待機

    //使い終わったら閉じておく.
    _pclose(gid);

    return 0;
}
```


配列に格納されたデータを使ってグラフを作成するプログラム
plot '-' ではじめ, plot 'e\n' で終わる.

```
void gnuplot(void)
{
    int i;
    double x[10],y[10];

    for( i = 0; i < 10; i++){

        x[i] = i;
        y[i] = (i-1)*(i+2);

    }

    FILE *gid;

    gid = _popen("C:/Win32App/gnuplot/bin/pgnuplot","w");

    fprintf(gid, "plot '-' w l\n"); // with line

    for( i = 0; i < 10; i++){

        fprintf(gid,"%lf\t%lf\n",x[i],y[i]);

    }

    fprintf(gid,"e\n");

    fflush(gid);

    fprintf(gid, "pause -1");

    fflush(gid);

    _pclose(gid);

}
```

5. 入力ファイル (*.plt) を使った便利な定型処理機能

テキストエディタを使って次のような入力ファイルを作成すると散布図が簡単に描けます。多量のデータファイル进行处理する場合に非常に便利です。

```
#ファイル名 : sample.plt      #をつけるとコメント行になります。

#本ファイルを gnuplot ファイルメニューの「Open」から読み込めば、
#これ以降指定した手順でグラフを描画する。

#散布図を描く。 plot 'data-file' with points を使用する。
#test.dat ファイルの 1 列目と 2 列目を使って指定したマーカーをもつ散布図を描く。
#ただし、凡例は表示せず、x 軸のデータ範囲、刻み幅は指定する。
#x 軸タイトル、y 軸タイトルも指定する。
#縦軸と横軸のスケールを調整して正方形にする。

#ウィンドウ画面に出力
set terminal windows
set output

#拡張メタファイルに出力する場合は次のコマンドを使用する。
#set terminal emf monochrome 'arial' 20
#set output 'fig.emf'

#postscript file に出力する場合は次のコマンドを使用する。
#set terminal postscript eps enhanced monochrome 'arial' 20
#set output 'fig.eps'

#凡例を表示しない
unset key

#始点と終点の値を指定して x 軸の範囲を設定する。 y 軸も同様。
set xrange [0:10]
set yrange [0:20]

#軸ラベルを指定
#軸ラベルに上付文字や下付き文字を指定する場合には上付 '^', 下付 '_'を使用する。
#ただし、書式付き文字は Postscript ファイルへの出力時のみ有効である。
#Windows 画面への出力には対応していない。

set xlabel 'Weight [kg]'
set ylabel 'The length of spring [mm]'

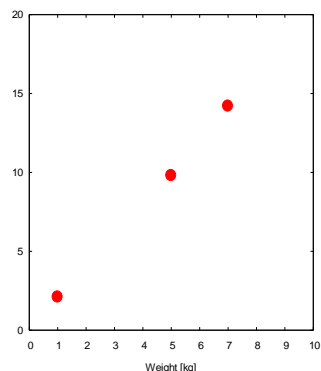
#軸の刻みを指定 下限値, 増分, 上限値
set xtics 0,1,10

#縦横サイズを調整して正方形にする
set size square

#データファイルの 1 列目と 2 列目を使って散布図を描画
set pointsize 2 #multiplier
plot 'sample.dat' using 1:2 with points pointtype 8

#gnuplot > call 'test.plt' test
#として実行すれば、引数$0 に test が指定できる。どんなファイル名にも対応可能。
#plot '$0.dat' using 1:2 with points pointtype 8

#出力先を Window に戻す
#set terminal windows
#set output
```



6. gnuplot Tips

(1)パッケージに添付されてくる demo ファイルを見るといろいろなグラフの描き方が分かる.

(2)入力ファイルの中で if 文が使える. useps = 0 if (useps > 0) else

(3) 陰関数を直接描画できませんが, 工夫すれば可能.

次の行を適当な定義に置き換えてください。

```
gnuplot> f(x,y) = x や y の入った楕円の式などをここに書く
gnuplot> set contour base #等高線を設定する
gnuplot> set cntrparam levels discrete 0.0 #等高線 Z=0 のグラフを描く
gnuplot> set nosurface
gnuplot> set terminal table #グラフを描画させるためのデータ系列を作成させる設定をする
gnuplot> set out 'ellipse.dat' #楕円のデータを保存するファイル名を指定
gnuplot> splot f(x,y) #データ系列を発生させる
gnuplot> set out #出力先を起動時に戻す
gnuplot> set terminal pop #画面出力に戻す
gnuplot> plot 'ellipse.dat' with lines #等高線データの入ったデータファイルを 2 次元 plot する.
```

(4)set multiplot を使えば, 一つのページに複数のグラフが描けます.

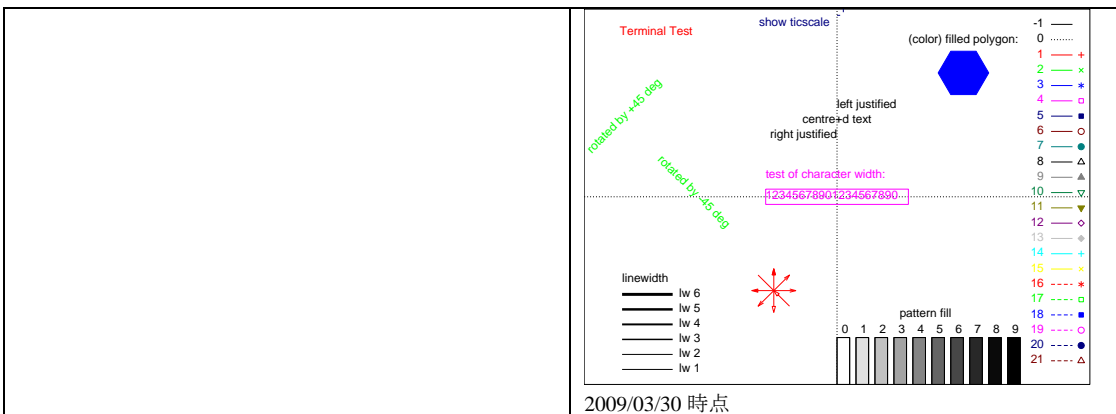
(5)Window 出力画面でのショートカットキー

Space	コマンド入力画面を表示
h	ショートカットキー一覧表示
m	マウス機能の ON/OFF
r	数値確認用のルーラを表示できる
u	ズームを解除する
Escape	`builtin-cancel-zoom` cancel zoom region

ここまでの Tips では, 新潟工科大学の竹野茂治先生が作成された <http://takeno.iee.niit.ac.jp/~shige/unix/gnuplot/gnuplot.html> に掲載されている内容を参考に, 必要と思われる項目について説明を行った.

(6)Terminal の設定と出力グラフの大きさ

EPS 形式 set terminal postscript eps enhanced monochrome 'Arial' 32	通常の Postscript 指定と異なり, EPS 指定では, 描画領域大きさ: 幅 10 インチ × 高さ 7 インチの半分の大きさ 5 インチ×3.5 インチ フォント大きさ: 半分の大きさ この例では, Arial フォントで 32pt の半分の 16pt の大きさになる.
PNG set term png enhanced size 640, 480	
EMF set terminal emf font "Arial, 18" set output "out_graph.emf"	



(7) using オプション

<p>plot 'data' using 1:(\$2/\$3) title column(N)</p>	<p>\$2 は、2 列目のデータの値 \$3 は、3 列目のデータの値 using 1:(\$2/\$3)は、1 列目のデータと 2 列目を 3 列目のデータで割った値を使う指定になる。</p> <p>(注意) call コマンドで呼び出すスクリプトファイルで\$2/\$3 を使用する場合は、call の引数でないことを表す必要があり、\$\$2/\$\$3 と\$を 2 重に入れること。</p> <p>title column(N)は、ファイルの 1 行目をタイトル文字列が格納されている行として取り扱うことを指定する。N は、N 行分だけ使うこと。col だけ表記する場合はあるが、これは column(0)のことであり、データが格納されたすべての列を使うということ。</p>
--	--

(8)文字列操作 (help の説明文を日本語で解説)

<p>four = "4"</p> <p>graph4 = "Title for plot #4"</p> <p>graph(n) = sprintf("Title for plot #%d",n)</p>	<p>文字"4"を変数 four に代入している。</p> <p>文字列"Title for plot #4"を変数 graph4 に代入している。</p> <p>引数 n を使って文字列を作成する関数 graph(n)を定義している。f(x)と同じ定義方法である。sprintf 関数で文字列を作成する。</p>
<p>plot 'data.4' title "Title for plot #4"</p> <p>plot 'data.4' title graph4</p> <p>plot 'data.4' title "Title for plot #".four</p> <p>plot 'data.4' title graph(4)</p>	<p>出力結果は全部同じ。</p>
<p>gnuplot スクリプトでの文字列操作事例</p> <p>f1='test' f2='.dat' No=1 f3=sprintf("%d", No)</p> <p>fname=f1.f3.f2</p> <p>plot fname title fname</p>	<p>変数 fname の内容は、変数 f1 と変数 f2, 変数 f3 を結合したものの「test1.dat」となる。</p> <p>test1.dat をプロットして、凡例タイトルを「test1.dat」に設定するスクリプト。</p>
<p>sprintf に指定する Format は</p> <p>%03d %3.1f %e</p>	<p>%lf は使用不可。</p>

(9)文字列変数を使ってフォント指定すると便利

label_font="Arial,30"	フォントを指定するための文字列変数を使うと便利.
set xlabel "x coordinates in the image plane" font label_font	例では, 文字列変数 label_font によって Arial フォントの 30pt を指定している.

(10)三項演算子との組み合わせ

A? B : C	条件 A が成立すれば B,成立しない場合 C を採用する.
plot 'datafile' using 1:(\$3= 0 ? \$2 : 1/0)	3列のデータをもつ datafile の第3列目の値が 0 であれば, 第1列目と第2列目でグラフを描き, 0 でなければ未定義の意味を表す 1/0 としてグラフを描画しない.
1/0	未定義データとして無視される.

(11)データファイルと every オプション

plot 'a.dat' every 2	プロットするデータを2つスキップする.
every ::0::0: 1行目だけプロット	every には6個のパラメータがあり, コロンで区切って指定する. every データ増分:ブロック増分:データ始点 : ブロック始点:データ終点:ブロック終点 every (増分) (始点) (終点) 大きく分けて3つの設定を, データ, ブロックの順に行う.
every :::0::0 0番目のブロックのみプロット	ブロックとは, データファイルを空行で区切ったデータの固まりである. (注意) ・データならびにブロックの番号は0から始まる. (重要) 空行が2行以上連続してしまう場合には, every のブロック認識ができないので, プログラム等でファイルを書き出す際に, 工夫する必要がある.

(12)画像 RAW ファイルを表示する

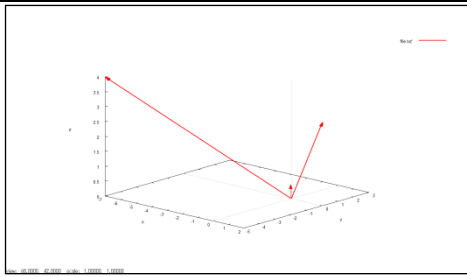
plot '画像ファイル' binary array=640x480 format='%uchar' with rgbimage	C 言語のソースファイル上では, ファイル名の指定に" (ダブルクォーテーション) を使用せず (シングルクォーテーション) を使用し, %の表記には%%とすることに注意する. example: fprintf(gid, "plot 'image.rgb' array=640x480 format='%uchar' with rgbimage¥n");
---	--

(13)棒グラフ histogram

<pre> set style data histogram set style histogram rowstacked set style fill solid border -1 set boxwidth 0.75 set key invert reverse Left outside set key autotitle columnheader set xtics ("1891-1900" 0.00000, "1901-1910" 1.00000, "1911-1920" 2.00000, "1921-1930" 3.00000, "1931-1940" 4.00000, "1941-1950" 5.00000, "1951-1960" 6.00000, "1961-1970" 7.00000) plot 'immigration.dat' using 2:xtic(1), '' using 3, '' using 4, ... </pre>	<pre> #45 度時計回りに軸目盛り文字が回転させる. set xtics nomirror rotate by -45 #ヒストグラム set style data histogram #ヒストグラムのデータは行毎に積み上げる. #列毎に積み上げる場合 columnstacked set style histogram rowstacked set style fill solid border -1 set boxwidth 0.75 set key invert reverse Left outside #1 行目を凡例タイトルとする. set key autotitle columnheader set xtics ("1891-1900" 0.00000, "1901-1910" 1.00000, "1911-1920" 2.00000, "1921-1930" 3.00000, "1931-1940" 4.00000, "1941-1950" 5.00000, "1951-1960" 6.00000, "1961-1970" 7.00000) 目盛りラベルを数字ではなく, 1891-1900 などの文字列 としたい場合は, "表示させたい文字列" 表示させたい座標の位置, の書式でラベルを一つ一つ指定する. このヒストグラムでは, データが 0,1,2,3,4,5,6,7 の 8 つ のセグメントに分けられ, 各セグメントにラベルが設定 される. xtic(1)は, xticlabels(1)の略であり, x 軸の目盛りラベル が格納されている列番号を指定する. :xtic(1):ytic(1):ztic(1)などと, 3 つの軸の目盛りタイトル を指定することも可能. 'data file' using 2:xtic(1), '' using 3, '' using 4,... 2 列目のデータと 1 列目の文字列を使って x 軸ラベルを 設定し, 後は順に 3 列目, 4 列目のデータを積み上げる. #using 3 ti col として 3 列目のデータでグラフを作成し, データの先頭行をタイトルに設定する. plot 'immigration.dat' using 2:xtic(1), "" using 3, "" using 4, ... </pre>
---	---

(14)ベクトル描画 with vectors

<pre> #3 つの軸タイトルを設定 set xlabel 'x' set ylabel 'y' set zlabel 'z' #ゼロ軸を描画 (線種を指定しないので点線で描画され る.) set xzeroaxis set yzeroaxis set zzeroaxis #xy 平面を底上げ描画しない set xyplane 0 </pre>	<p>3 次元のベクトルを描画するスクリプトを例に説明する.</p>
--	------------------------------------

<pre>#3つの軸スケールを同じにする. set size ratio -1 #原点から伸びる3つのベクトルを描画する. splot 'vector_data.txt' with vectors filled head lw 2 #####DATA FILE (vector_data.txt)##### 0 0 2 -2 1 0 0 0 -1 3 2 0 0 0 -7 -5 4</pre>	 <p>with vectors まで指定した場合は、→の先端が塗りつぶされない。</p> <p>3次元ベクトルの場合、データは6列で構成される。最初の3列にはベクトルの始点、残りの3列にはベクトルの終点（矢印の先端）を座標で入力する。</p>
---	--

(15) 凡例のタイトル指定 plotコマンドと title オプション

<pre>plot 'data-file' title "your title here" [example] plot 'datafile.dat' using 3:4 title "Logged in" with impulses, 'datafile.dat' using 3:5 t "Load average" with points, 'datafile.dat' using 3:6 t "%CPU used" with lines</pre>	<p>この例では、一つのデータファイル datafile.dat の3列目と4列目、3列目と5列目、3列目と6列目で3種類のグラフを同時に描画している。Title オプションを使って、それぞれのグラフ凡例を指定している。</p>
<pre>set key on bottom right Left reverse box plot 'datafile.dat' title "凡例のタイトルをここに"</pre>	<p>on: 凡例表示 bottom right: 凡例を右下に Left: 文字を左寄せに reverse: マーカとタイトルを左右反転 box: 凡例を四角で囲む。</p> <p>title: 凡例のタイトルを入れる</p>
<pre>plot 'datafile.dat' using 3:4 notitle</pre>	<p>凡例を消すには notitle オプションを使用する。</p>
<pre>すべての凡例を消す unset key</pre>	

(16) データファイルを使って等高線グラフ

<pre>#等高線を底面に描画 set contour base #真上から見る. set view 0, 0</pre>	
--	--

<pre>#等高線を 10 段階に分けて表示する. set cntrparam levels 10 #データ間を線で結ぶ set style data lines #縦横比を 1 set size square #xy それぞれ 51 個のデータでメッシュプロットする。 set dgrid3d 51,51, 1 #1 列目, 2 列目, 3 列目でグラフ描画 splot '2FootAngleX10x10.dat' using 1:2:3</pre>	
---	--

(17) 軸の設定

<p>スケールを無視して, 正方形の形に描画したい場合</p> <pre>set size square</pre> <p>軸目盛りのスケールを合わせたい場合</p> <pre>set size ratio -1</pre>	<p>縦横サイズ 1:1 プロット</p> <p>縦横の目盛りスケールが合う.</p>
<p>軸目盛り間隔とグリッド表示</p> <pre>set xrange [0:50] set yrange [50:0] set xtics 0,5,50 set mxtics 5 set mytics 5 set grid xtics mxtics ytics mytics</pre> <p>線種を同時に指定する場合は,</p> <pre>set grid ytics mytics lt 1 lw 2, lt 2 lw 1</pre>	<p>0 から 50 までを 5 刻みの主メモリ線を入れる.</p> <p>主メモリをさらに 5 等分する.</p>
<p>x 軸や y 軸に第 2 軸を追加したい場合</p> <pre>set xrange [0:100] set x2range [0:1] set yrange [20:100] set y2range [-1:1] plot 'datafile.txt' using 1:2 axis x1y1, '' using 3:4 axis x2y2</pre>	<p>plot 時に, axis 指定をする.</p> <p>何も指定しない場合は, axis x1y1 指定と同じ.</p>
<p>3D プロット時の Z 原点位置調整</p> <pre>set ticslevel 0.5 set xyplane 0.5 set xyplane at 0</pre>	<p>3D プロットの xy 底面を Z 範囲のどこに位置させるかを調節する. Default 値は 0.5 である.</p> <p>設定したい位置 pos があらかじめある場合, 設定値は $(pos - zmin)/(zmin - zmax)$ から計算される.</p> <p>Set ticslevel では, Z の範囲によって xy 底面の位置が相</p>

<pre>set xyplane 0 splot x*y</pre> <p>別の方法としひて set ticslevel 0 もあるらしい.</p>	<p>対的に変化するので、新たに用意されたコマンド <code>set xyplane at [Z-value]</code> を使用して、at 以下に具体的な Z 値を指定する.</p> <p>at を付けずに、ゼロのみ付けると xyz 軸の原点が変わったグラフになる.</p>
<pre>set isosamples 100,100</pre>	<p>splot x*y など、メッシュの細かさを調整する設定値. この例では、xy それぞれ 100 等分割される. Default では 10 に設定されている.</p>
<pre>set {x x2 y y2 z}zeroaxis linetype 1</pre>	<p>xzeroaxis: y=0 の直線を引く. yzeroaxis: x=0 の直線を引く.</p>

(18)3 項演算子ならびに Clip を使って特定の範囲のみ描画する方法

<pre>#set clip #default: noclip points, clip one, and noclip two set xrange[0:2*pi] set yrange[-1:1] unset clip f(x) = sin(x) g(x) = x<1 ? f(x) : x > 5 ? f(x) : -1000 plot f(x), g(x)</pre>	<p>3 項演算子 A?B:C A ならば B, 成り立たないならば C となることを利用して、関数 f(x) を場合分けした関数 g(x) を作り、描画する.</p>
---	--

(19) with labels オプション, 3 列データファイルを使って, 指定した座標にラベルをプロットする.

<pre>plot "block_label.dat" with labels font 'Arial,36</pre>	<p>block_label.dat には, x 座標, y 座標, テキストデータの 3 列のデータを格納しておく.</p>
--	---

(20) アニメーション制作

<p>アニメーション作成用親ファイル</p>	<pre>set term png enhanced size 320, 480 MAX=1000 f=0 set xrange [0:11000] set yrange [0:9000] call "animate_loop.plt" "\$0" set term win set output reset</pre>
------------------------	---

繰り返し処理内容記述ファイル

```
#set label 1 sprintf("[ %3d ]", f) at first 52,0  
set output sprintf("%03d.png", f)  
plot "$0" every :::f::f title sprintf("[ %3d ]", f)  
  
#pause -1  
  
f=f+1  
  
if( f <= MAX ) reread
```

(21) Test パターンとカラー指定

<p>EMF ファイル (2008/02/13 時点)</p> <p>Terminal Test</p> <p>show ticscale</p> <p>(color) filled polygon:</p> <p>left justified centre+d text right justified:</p> <p>test of character width: 12345678901234567890...</p> <p>this terminal supports text rotation only for truetype fonts</p> <p>rotated centred text rotated by +45 deg rotated by -45 deg</p> <p>linewidth</p> <ul style="list-style-type: none"> lw 6 lw 5 lw 4 lw 3 lw 2 lw 1 <p>pattern fill</p> <ul style="list-style-type: none"> 0 1 2 3 4 5 6 7 8 9 <p>Legend:</p> <ul style="list-style-type: none"> -1 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 	
<p>カラー指定</p> <p>rgbcolor "colname"</p> <p>rgbcolor "#RRGGBB"</p>	<p>#RRGGBB は 16 進数で</p>

7. 参考になる Web サイトと図書

gnuplot homepage

次のサイトには, gnuplot を使って描くことができる様々なグラフ形式とコマンドが掲載されています.
<http://gnuplot.sourceforge.net/demo/>

日本語解説サイト

Kawano Toshihiko (Los Alamos National Laboratory, Nuclear Physics Group)
<http://t16web.lanl.gov/Kawano/gnuplot/>

東京電機大学の松田先生のサイト
<http://ayapin.film.s.dendai.ac.jp/~matuda/Gnuplot/gnuplot.html>

パッケージに添付されてくる Help (英文) も非常に参考になります.
新潟工科大学の竹野茂治先生は Help を日本語に翻訳した文書を Web 上に公開されています.
<http://takeno.iee.niit.ac.jp/~shige/unix/gnuplot/gnuplot.html>

参考図書

矢吹道郎, 大竹敢: 使いこなす gnuplot (改訂第 2 版), テクノプレス
ISBN 4-924998-69-9
¥2,800

gnuplot の操作方法を非常にわかりやすく解説した本です. 是非お勧めします.

グラフ描画ソフト gnuplot を使う

Copy right 森尾吉成 2006

平成 18 年 3 月 3 日	第 1 版	第 1 刷発行
平成 19 年 6 月 4 日	第 2 版	第 1 刷発行
平成 19 年 12 月 6 日	第 3 版	第 1 刷発行

著者	森 尾 吉 成
発行者	MORIO FARM 社
	代表者 森尾 吉成
発行所	MORIO FARM 社
	郵便番号 514-8507
	三重県津市上浜町 1577
	三重大学生物資源学部
	電話 059-231-9602
	morio@bio.mie-u.ac.jp

Printed in Japan