

# gnuplot の使い方

gnuplot は数値データの並びをグラフとして可視化する (データの点をプロットする) ためのソフトであり、データを可視化するためのソフトの定番である。簡単な関数であれば直接入力してグラフをかかせることもできる。

## 1 起動

コマンドラインから gnuplot と入力すると起動する。  
終了するには exit と打ち込む。

## 2 簡単なグラフ

ここでは簡単な関数を指定して、それを与えられた範囲 (何も指定しなければ変数は  $-10$  から  $10$  まで動く) でプロットする。ここでの指定がすべての基本となる。あとは関数を指定していたところをデータのファイル名に置き換えることで、データをプロットさせることができる。

### 2.1 2次元グラフ

まず簡単なグラフを描画させる。次のように入力してみよ。

```
plot x
plot sin(x)
plot [-20:20] sin(x)
plot [-20:20][0.5:1] sin(x), x, cos(x)
```

使用できる演算子は  $+ - * / **$  の5つがある。最後の  $**$  は冪乗である。たとえば `plot 2**x` は関数  $y = 2^x$  のグラフをかく。使用できる関数には `sin(x)`、`cos(x)`、`exp(x)` (指数関数  $e^x$ )、`abs(x)` (絶対値)、`log(x)` (自然対数)、`log10(x)` (常用対数) 等のかなり基本的なものから、`gamma(x)` ( $\Gamma$  関数) などまで様々用意されている。なお `pi` は  $\pi$  を表わす定数として使用してよい。

### 2.2 3次元グラフ

次のように入力してみよ。

```
splot x**2 + y**2
splot [-1:1] sqrt( x**2 + y**2 )
splot [-5:5][10:15] sqrt( x**2 + y**2 )
splot [] [] [-5:5] sqrt( x**2 + y**2 )
splot [-pi:pi][-1:1][-5:5] sin(x*y) / cos(x**2)
```

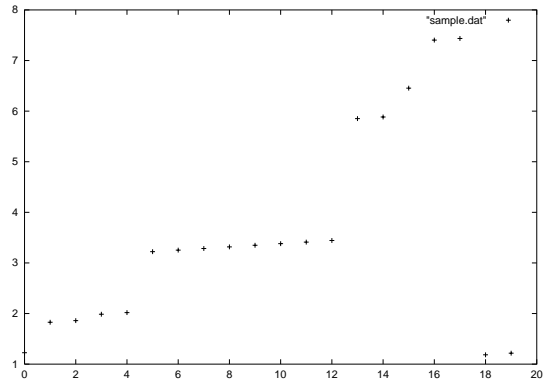
## 3 データのプロット

データファイルを指定して、そのデータをグラフ上にプロットすることができる。

### 3.1 2次元・単一データのプロット

たとえば次のようなデータファイル ( sample.dat ) ファイルがあるとすると。

```
# sample.dat
# 2d plotting sample
1.22592
1.82792
1.85960
1.98634
2.01802
3.22201
3.25370
3.28538
3.31707
3.34875
3.38043
3.41212
3.44380
5.85179
5.88347
6.45378
7.40431
7.43599
1.18449
1.21617
```



このようなデータファイルにおいては、# で始まる行はコメントで、データとしては扱われない。今のデータのように値のみ記されていた場合には、 $x$  の値は先頭行から順番に 1, 2, 3, ... と見なされる。

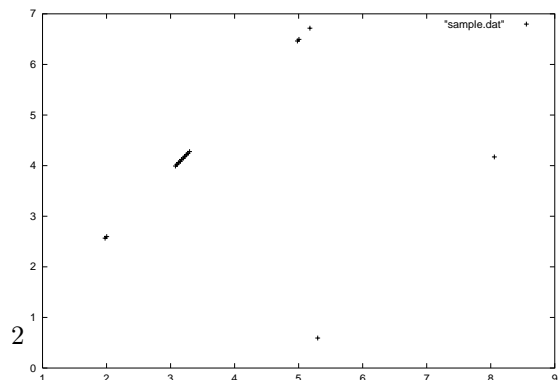
このデータファイル ( sample.dat ) をプロットするには次のように入力する。ただし gnuplot の「カレント・ディレクトリ」(作業フォルダ) と同じ場所にファイルがあるとすると。

```
plot "sample.dat"
```

### 3.2 座標付き 2次元グラフ

$x$  座標と  $y$  座標のペアを記したデータファイルを用意する。

```
1.9775390625 2.566405136567
2.0019531250 2.598089182820
3.0761718750 3.992185849578
3.1005859375 4.023869806568
3.1250000000 4.055553761317
3.1494140625 4.087237713960
3.1738281250 4.118921664640
3.1982421875 4.150605613509
```



3.2226562500 4.182289560721

このとき、各行がデータの座標  $(x, y)$  を意味している。データの区切りは空白（またはタブ文字）である。

ここでのサンプルは本当に点を打っただけであるが、あとで指定法を示すように、各点を直線でつなぐこともできる。このようなとき、データファイルに空行があるとそこでデータの区切りと見なされて、そこはグラフが繋がらなくなる。（色が変わったりはしない。一つのデータファイルと一つのグラフとは対応している。）

### 3.3 3次元データグラフ

3次元グラフでは  $x, y, z$  軸の3つが使用される。2次元のときと同様に  $z$  座標の値のみを並べたデータファイルと  $(x, y, z)$  座標の値を並べたデータファイルの2種類が用意できる。ところが  $z$  座標の値だけを並べたファイルは、 $x$  の値と  $y$  の値の2つを自動的に決めなければならないので2次元のときほど自明ではない。

また、3次元のときには「曲面」をかきたいのか、それともたとえば粒子が動いた軌跡のように「曲線」をかきたいのかによって描き方を根本的に変更しなければならない。曲線をかきたいときには2次元と同じように3次元のデータを  $x y z$  の順に1行ずつ記述する。

```
1.0 1.0 3.0
1.0 1.2 4.1
2.2 1.3 3.1
3.0 1.5 2.0
2.5 1.2 2.5
2.4 3.1 2.5
```

ただし、ここでは

```
plot "sample.dat" with lines
```

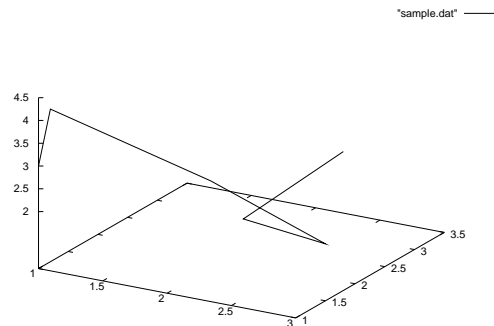
として点の間を直線で結ぶように指示した図を載せてある。詳しくは後述する。

一方曲面をかきたいときには、データを空行で区切られたブロックごとに並べてかいておく。そうすると各ブロックごとに生成された曲線間を適切に接続して曲面をかいてくれるようになる。

また  $x, y$  座標の値を省略して  $z$  値だけを記したデータファイルを準備する際には、データの中にある空行でデータの「ブロック」を分離する。 $x$  座標はブロックごとに振られ（最初のブロックが  $x = 1$  で次が  $x = 2$ ）、各ブロック内の各行ごとに  $y$  座標が振られる（各ブロックの1行目が  $y = 1$  で次の行が  $y = 2$ ）。

### 3.4 データ列の指定

データファイルは原則的には  $x \cdot y \cdot z$  の順に値が並んでいるが、事情によってデータ順が異なったり、別のデータが混在している場合もある。その場合には



- `plot "sample.dat" using 3:1`

のように `using` を使用して列番号を  $x, y, z$  (`splot` の場合) の順にコロンで区切って指定する (ファイルの先頭列が 1)。

また、次のように、たとえば 2 列目と 3 列目の積を使用するように指定することもできる。

- `plot "sample.dat" using 4:($2*$3)`

### 3.5 作業フォルダ

`gnuplot` の作業フォルダを表示するには、`pwd` と入力する。

```
pwd
```

これで現在の作業フォルダが画面に表示される。変更するには `cd` を用いる。

```
cd 'C:¥Documents and Settings¥F00¥My Documents¥data'
```

シングルクォートの代わりにダブルクォートを使うときにはディレクトリ区切り文字 `¥` は `/` を使わなければならない。いうまでもなく Unix では最初から `/` のみである。

```
cd "C:/Documents and Settings/F00/My Documents/data"
```

なお、シングルクォートを使用するときでもディレクトリ区切り文字は `/` を使ってかまわない。

## 4 グラフのスタイル

データをプロットするときには通常はデータファイルにある座標の位置に点が打たれるだけである。これを線で結んだり (スタイルの変更) または色などを変えたりすることができる。どのような種類が用意されているかは

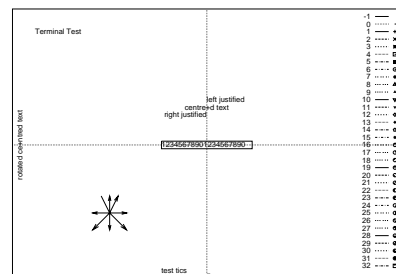
```
test
```

と打ち込めば表示される (画面や画像ファイルなど、出力する形式によっても異なるので注意)。よく使われるスタイルは次の通り:

<code>lines</code>	直線で結ぶ
<code>points</code>	データポイント (目立つ点)。既定。
<code>linepoints</code>	直線とデータポイントの組み合わせ
<code>dots</code>	小さな点

これらを変更するときには `plot` や `splot` の後ろに `with ...` と指定する。さらに `with ...` 線種 点種のように線の種類と点の種類を数字で (`test` の出力をみよ) 指定する。次の例では線の色を緑で、点の色を茶色で指定している (標準的な Windows 画面への出力の場合)。

```
plot "sample.dat" with lines 2 6
```



## 5 グラフの出力先

標準ではグラフはそのコンピュータでもっとも普通の出力画面に出力される。これではほかの場所で使いづらいので、最後に完成したグラフはファイルに保存しておく。(そうしなければ他の人に見せられない。)ところが画像の保存形式は、その使用目的に応じていろいろとある。ここでは代表的なものをいくつか紹介する。

### 5.1 ターミナル(端末)の設定

端末(画像ファイルの出力形式も含む)としてよく用いられるものには次のようなものがある。

windows	Windows の画面 (Windows の標準)
gif	GIF ファイル
png	PNG ファイル
postscript	PostScript 形式
latex	L <sup>A</sup> T <sub>E</sub> X の picture 環境形式
eepic	L <sup>A</sup> T <sub>E</sub> X の eepic 環境形式
tpic	L <sup>A</sup> T <sub>E</sub> X の tpic スペシャル命令
x11	X-Window の画面 (Unix の標準)
aqua	MacOSX の画面 (Macintosh の標準)

これらの詳細については下で述べる。それよりも重要なことはファイル形式を指定すると出力の形式はそのファイル形式になるが、出力そのものは画面に出てくるということである。ファイルには保存されるわけではない。ファイルに保存するには次の指定を組み合わせる。

### 5.2 出力ファイルの指定

出力をファイルに保存するには次のように打ち込む。

```
set output "foo.gif"
```

これで `foo.gif` という名前のファイルに保存される。もちろんこれは GIF 形式の場合であり、PNG 形式なら `foo.png`、EPS 形式なら `foo.eps` のように指定する。`foo` のところは `sample` だろうが `report` だろうが好きに指定してよい。

もう一つ重要なことは、保存し終わったら必ず出力ファイルを元に戻しておかないと、`gnuplot` は何も考えずに次々と出力をそのファイルに追加するので、ファイルがおかしくなることもある。元に戻すには単に `set output` とだけ指定しておく。

典型的には次のような操作をする。( `sample.dat` というファイルに用意されたデータを `report.png` という PNG ファイルに出力する場合 )

```
set terminal png
set output "report.png"
plot "sample.dat" with lines
set output
set terminal x11 (または aqua, windows など)
```

これで作業フォルダに report.png というファイルができた。クリックして表示できていれば成功である。最後の出力先を標準の画面に戻す操作では、MacOSX を使用している場合には「ターミナル」で起動した場合には aqua を、「x11」で起動した場合には x11 を指定する。

### 5.3 Postscript 形式

たとえば科学論文などで  $\text{T}_\text{E}\text{X}$  を使用して文書を作成し、それをその後 PDF ファイルに変換するようなことを考えている場合には、画像形式としては EPS ( Encapsulated PostScript ) 形式で用意することが好ましい ( PDF ファイルは EPS 形式と特に相性がよいからである<sup>1</sup> )。

```
set terminal postscript eps enhanced color solid "Times-Roman" 10
```

ここで enhanced 以降はオプションなので省略可能である。

enhanced	拡張 PostScript を使用 ( 標準 : noenhanced )
color	カラーで出力 ( 標準 : monochrome )
solid	ラインを実線にする ( 標準 : dashed : 破線 )
font-name	PostScript フォント名を指定 ( 標準 : Helvetica )
font-size	フォントサイズ ( 標準 : 10 )

フォント名は間違いなく使用できるものは Times-Roman · Helvetica · Courier だが、その他文書によってはいろいろ指定可能なこともある。ただしフォント指定は高度な知識を要するのでこれ以上は説明しない。

破線のスケールを指定する場合には dashlength 1.0 のように指定し、線幅のスケールを指定する場合には linewidth 1.0 のように指定する。EPS ファイルの大きさは、横 5 インチ、高さ 3.5 インチとなっているが、一般に拡大・縮小しても品質低下はないと思ってよい。

### 5.4 $\text{T}_\text{E}\text{X}$ 形式

標準的には  $\text{T}_\text{E}\text{X}$  の文書に図を挿入する場合には EPS 形式を推奨する。しかし場合によっては EPS 形式を使用できない場合がある ( システムが古いなど )。その場合には確実な方法は  $\text{T}_\text{E}\text{X}$  の命令を直接用いて図を出力することである。ただしこの方法には若干制限がある。

set terminal latex は  $\text{T}_\text{E}\text{X}$  のメモリ制限を超えやすい。set terminal eepic を使用するには  $\text{L}\text{T}_\text{E}\text{X}$  原稿の最初に  $\usepackage{eepic}$  が必要となる。また出力ドライバが Tpic を理解しなければならない。(理解できないドライバも存在する。dviout は大丈夫)。eepic パッケージを持っていないときには set terminal tpic を使用する。もちろんこの場合にも Tpic を理解する出力ドライバが必要である。

### 5.5 PNG · GIF 形式

PNG 形式や GIF 形式は図を WEB ページにおいたり、電子メール等で誰かに簡単に見せたりするのに最適である。印刷用にはあまり推奨できないとはいえ、多くのグラフィックソフトが対応している。ワードプロセッサで使用したい場合は、PNG 形式や GIF 形式なら直接扱うことができる

<sup>1</sup>より詳しくいえば、PDF ファイルはプリンタの制御命令コードの一体系である PostScript を基礎として定義された形式であり、EPS 形式は図版定義用に PostScript を改造して定義された形式である。

ものもあるし、比較的簡単に BMP 形式などに変換することもできるから、その点で大変手軽である。ただし GIF は一時期の著作権騒動の問題からバージョン等によっては対応していなかったりすることもあるので、ここではこれらの出力については細かなオプションは省略する。

```
set terminal gif
set terminal png
```

## 6 軸

2次元グラフを表示するときには上下に  $x$  軸、左右に  $y$  軸が付加される。3次元のときも  $x, y, z$  軸が表示される。これらの調整法をまとめる。

### 6.1 軸の範囲

軸の範囲はそのままグラフが表示される範囲でもある。手軽には `plot [x-min:x-max] [y-min:y-max] ...` のようにして `plot` や `splot` で指定する。そうでないときにはあらかじめ次のようにして軸の範囲を指定する。

- `set xrange [x-min:x-max]`
- `set yrange [y-min:y-max]`
- `set zrange [z-min:z-max]`
- `set x2range [x2-min:x2-max]`
- `set y2range [y2-min:y2-max]`

ただし  $x_2$  軸・ $y_2$  軸はそれぞれグラフの上側・右側に表示される軸であり、これらの範囲を指定しても表示されるグラフには影響を及ぼさない（目盛りの表示範囲が変わるだけ）。またそれぞれ `set xrange [0:1] reverse` のように `reverse` を指定すると軸の向きが逆になる（たとえば  $x$  軸では右の方が数値が小さくなる）。

これらを指定すると（`plot` の中でも）それ以降のグラフの表示まで影響を与える。標準的な「自動計算」の状態に戻すには `set autoscale` と入力する。`set autoscale x` のように指定すると  $x$  軸だけが元に戻る（ $x \cdot y \cdot z \cdot x_2 \cdot y \cdot xy$  が指定可能）。

注： 今後すべて軸に関する説明は  $x$  の部分を  $y \cdot z$  等に置き換えるだけで  $x$  軸以外にも適用されるので、 $x$  軸に関する説明のみをおこなう。

### 6.2 軸の目盛り

軸の目盛り位置を指定するには次のように入力する。

- `set xtics start, incr, end`

ここで `start` が目盛りの開始位置、`incr` が目盛りの間隔、`end` が終了位置を表わす。たとえば `set xtics -10, 1, 10` は  $-10$  から  $1$  刻みで  $10$  まで目盛りをつける。

ただし  $x$  軸・ $y$  軸の目盛り（の刻み記号）は自動的に  $x_2$  軸（右軸）や  $y_2$  軸（上軸）にも表示される。これを避けるには次のように入力する。

- `set xtics nomirror`

逆に  $x_2$  軸の刻み記号を  $x$  軸にも表示するには `set x2tics mirror` と指定する。

また、刻み位置を等間隔ではなく指定した位置のみに表示するには

- `set xtics (0, 1, 2, 4, 8)`

のように指定したい位置を括弧内にコンマで区切って並べる。標準では刻み位置にはその数値が表示されるが、それを文字列に変更するには今の指定に加えてそのラベルを指定する。

- `set xtics ("min" 0, "center" 4, "max" 8)`

これら目盛りの刻みは基本的には枠の外側に表示されるが、原点を通る水平位置 ( $y$  座標なら垂直位置) に表示するには次のようにする。

- `set xtics axis`

元の位置に戻すには `set xtics border` とする。

数値 (やラベル) は特に指定しなければ通常の向きで出力されるが、特に  $y$  軸ではそれを左に 90 度回転したいことがある。そうしたければ

- `set xtics rotate`

のように指定する。

今まで述べた `axis`・`mirror`・`rotate` の複数を 1 度の `set xtics` で指定したい場合には今記した順で記述すること。

軸は通常はグラフの内側に向かって表示されているが、これを外側にしたければ

- `set tics in`

とする。外側にしたければ `set tics out` とする (`xtics` ではない)。

また目盛りを記述したくないときには `set noxtics` (gnuplot 4.0 では `unset xtics`) と指定し、標準状態に戻したければ `set xtics` とだけ指定する。

また、標準目盛り以外に「小目盛り」を出力したければ

- `set mxtics number`

と指定する。 `number` は通常の間隔をいくつに刻むかを指定する。省略すれば 10 となる。 `set nomxtics` (gnuplot 4.0 では `unset mxtics`) と打ち込むと小目盛りは表示されなくなる

最後に、目盛りの刻み記号そのものの大きさを変えたければ

- `set ticscale size m-size`

で指定する。標準は `size` が 1.0 で `m-size` が 0.5 である。

### 6.3 軸のラベル

目盛りではなく軸そのものにラベルをつけるには

- `set xlabel "LABEL"`



と指定する。ラベルの出力位置は 2 次元グラフの場合には  $x$  軸の中央・ $y$  軸の左上である。3 次元グラフの場合には  $x \cdot y$  軸の場合には中央、 $z$  軸の場合には上部になる。

- `set xlabel "LABEL" r-shift, t-shift`

として `r-shift` やさらに `t-shift` で数値を指定するとラベルの出力位置がそれぞれ右及び上方向に指定した数値分だけずれて出力されるようになる。

## 6.4 座標軸

座標軸 ( $x$  軸 =  $y = 0$  の直線) を表示するには

- `set xzeroaxis`

とする。 `set zeroaxis` とすると  $x$  軸と  $y$  軸の両方が描かれる。ただしこれらは 2 次元グラフのみである。表示しなくさせるには `set noxzeroaxis` のようにする (gnuplot 4.0 では `unset xzeroaxis`)。

## 6.5 枠の非表示

グラフの枠そのものを表示したくなければ

- `set noborder`

とする (gnuplot 4.0 では `unset border`)。 `set border` とすると再び表示されるようになる。ただし、枠を非表示にしても目盛り刻みなどは表示されたままなのでそれらも表示したくなければ別個に指定する。

また 2 次元で右側と上側の枠を表示したくない (左側と下側のみにしたい) 場合は

- `set border 3`

とする。この数値の指定法は複雑なので述べない。

## 6.6 グリッドの表示

グラフ内に (方眼紙・グラフ用紙のように) グリッドを表示したい場合には

- `set grid`

とする。標準では小目盛りにはグリッドが付加されないが

- `set grid mxtics`

とすると  $x$  軸の小目盛りにもグリッド線が引かれるようになる。同様に `set grid xtics mxtics` のように個別に指定することもできる。

## 7 グラフのラベル

今更いうまでもなくグラフは `plot` や `splot` で出力する。その出力したグラフそのものに説明は、標準では指定した関数やデータファイルの名前が右上に表示される。これらを変更する方法を述べる。なお、説明はすべて `plot` に関して行うが `splot` でも同様である。

## 7.1 ラベルの指定

ラベル文字列は `plot` のオプション `title` を指定して行う。

- `plot expression title "LABEL"`

`expression` は関数やデータファイルの名前である。以下の説明でも同様である。`LABEL` が説明であり、グラフの枠の右上に表示される。

## 7.2 ラベルの出力位置

ラベルは標準ではグラフが表示される枠の右上部分に出力される。これを別の位置に変更するには

- `set key place`

と指定する。`place` は次のような位置指定の組み合わせを記述する。

<code>left</code>	グラフ内の左側
<code>right</code>	グラフ内の右側
<code>outside</code>	グラフ外の右側
<code>top</code>	グラフ内の上側
<code>bottom</code>	グラフ内の下側
<code>below</code>	グラフ外の下側
<code>(x,y)</code>	座標 $(x,y)$

これらは `set key top right` のように複数指定してよい。また、 $(x,y)$  の指定はグラフ内の  $(x,y)$  の座標位置に出力する。3次元グラフでは  $(x,y,z)$  のように指定する。

## 7.3 ラベルの出力順と線のサンプル

標準ではラベルは線のサンプルの右に出力される。

- `set key reverse`

とするとラベルの右に線のサンプルが出力されるようになる。

グラフ出力の線のサンプルは標準では4文字文の長さであるが、これをたとえば8文字分に変更するには

- `set key samplen 8`

のように指定する。

## 7.4 ラベルの出力体裁

標準ではラベルは縦方向に並べて出力されるが、その間には隙間が空いていない。これはラベル間の空白が1文字となっているためである(0文字なら重なってしまう)。これをたとえば2文字分に変更するには

- `set key spacing 2`

と指定する。

また、ラベルに四角い枠をつけるには

- `set key box linetype 1 linewidth 3`

のように指定する。`linetype` は線種 (標準は  $-1$ ) で `linewidth` は枠線の幅 (標準では  $1$ ) を指定する。線種の番号については `test` コマンドで確認することができる。

またラベルそのものを出力しなくするには

- `set nokey`

と指定する。(gnuplot 4.0 で `unset key`)

## 8 グラフの体裁

ここではグラフの全体的な出力に関わる指定を述べる。

### 8.1 グラフのタイトル

グラフそのもののタイトルを指定するには

- `set title "TITLE" r-shift, t-shift`

のようにして指定する。この場合 `TITLE` がグラフの上部の左右中央に出力される。なお、文字列中に `¥n` を指定するとそこで改行される。`r-shift` と `t-shift` はそれぞれ出力位置を右・上方向に何文字分ずらすかを指定する (省略可能)。

### 8.2 グラフの縦横比

グラフを正方形で出力するには

- `set size square`

と指定する。また  $x$  方向と  $y$  方向の比率を指定した長方形にするには

- `set size ratio 0.5`

のように指定する。ここで  $0.5$  は  $y$  軸の長さ /  $x$  軸の長さである。今の指定では縦方向が横方向の半分の長さになる。

### 8.3 サンプル数

関数などのグラフを描くときには、その値のサンプル点として  $100$  が選ばれますが、次のようにしてサンプルの個数を指定できる。

- `set samples 200`

## 8.4 対数グラフ

各座標軸を対数グラフにするには

- `set logscale x`

のように指定する。x は `xy` や `x2` のように指定してもよいし、省略してもよい(すべての軸が対数軸になる)。また `set logscale x 10` のようにして対数の底を指定することもできる。

## 8.5 媒介変数

通常では2次元グラフでは変数は  $x$  であり、3次元グラフでは変数は  $x, y$  が用いられている。グラフを媒介変数表示で指定するには

- `set parametric`

を指定する。こうして  $x, y$  (3次元ならさらに  $z$ ) 座標を表わす式をコンマで区切って並べる。ただし変数は2次元で  $t$  であり3次元で  $u, v$  である。たとえば円を表示するには

- `plot cos(t), sin(t)`

であるし、球を表示するには

- `splot cos(u)*cos(v), cos(u)*sin(v), sin(u)`

である。

## 8.6 等高線

等高線を表示するには

- `set contour base`

のようにする。このときには3次元グラフの  $x-y$  平面に等高線が描かれる `base` を `surface` にすると曲面上に描かれ、`both` とすると両方に描かれる。さらに

- `set nosurface`

とすると曲面自身が描かれなくなる。

## 8.7 曲面の色表示

gnuplot 4 では

- `set pm3d`

とすると曲面がその高さに応じて色づけして表示される。曲面上ではなく  $x-y$  平面に色づけを表示したい場合には

- `set pm3d at b`

と指定する。b を `t` とすると曲面の上部に表示されるようになるし `s` とすると曲面上に表示される(標準)。bs のような指定もできる。

## 8.8 隠線非表示

いうまでもなく、3次元グラフは2次元に投影されたものである。そのとき、みる角度によっては「本来」見えない部分が表示される。それを隠したければ

- `set hidden3d`

とする。

## 9 その他

ここではその他の便利な機能について述べる。

### 9.1 ヘルプ

ヘルプを表示するには `help` コマンドを使う。

- `help splot`

### 9.2 ファイル名の省略

特に長いファイル名を使用する場合など何度も入力するのはやっつけられない。その場合、単にファイル名を"`"`と空にしておくと、その直前のファイル名が自動的に用いられる。

### 9.3 スクリプト

いくつかの処理を行う場合など、いちいち `gnuplot` に指示をしていては間違いも誘発しやすくなるし、何よりも効率が悪い。そのようなときにはあらかじめよそでエディタ等を用いて（場合によっては専用のプログラムを作って）実行する手順を記したファイル（スクリプトファイル）を読み込ませればよい。そのようなファイルは

- `load "script-file"`

のようにして呼び出すことができる。さらに

- `load "sample.plt" "arg0" "arg1"`

のようにして呼び出しに引き数を指定することができる。このような引数はスクリプトファイルの中では `$0`, `$1`, ... のようにして 10 個まで使用することができる。

逆に `gnuplot` と対話的に行った操作を

- `save set "save-name"`

のようにして保存できる。今の指定は `set` コマンドで指定した内容を保存する。ほかに `var`（定義した変数）や `function`（定義した関数）を指定できるが、指定しなかった場合にはすべての設定と最後に実行した `plot` コマンドが保存される。

## 9.4 設定のリセット

いろいろな設定を試してみて、何が何だかわからなくなったとき、あるいはスクリプトなどの設定で強制的に既定の設定値に戻したいときには `reset` と入力する。

## 9.5 電卓

`print 2*3` のようにして計算値を表示させることができる。`print log10(3)` のような使い方もできる。

## 9.6 自宅などで使うには

<ftp://ftp.u-aizu.ac.jp/pub/gnu/gnuplot/gnuplot/> から Windows 用最新バージョンをダウンロードする。この文書が書かれているときには `gp400win32.zip` が最新版である。それらしいファイルをダウンロードしてくる。このファイルは Windows の「圧縮フォルダ」なので(古い Windows の場合には適切な展開ソフトを使用して中のファイルを取り出す)ハードディスクの好きな位置にすべてのファイルをコピーしておく。実行するプログラムの本体は `bin` フォルダの中にある `wgnuplot.exe` ファイルである。起動するために、`wgnuplot.exe` ファイルへのショートカットを作っておけば便利。

### 9.6.1 日本語化するには

<http://takeno.iee.niit.ac.jp/~foo/gp-jman/gp-jman.html> にはメニューやヘルプを日本語化するためのファイルが置かれている。その日本語化用ファイルは、そのページ内の「日本語化キット」または、この文書が書かれた時点での最新版のファイルは <http://takeno.iee.niit.ac.jp/~foo/gp-jman/data/wgp-jp/wgp400-20050625.zip> からダウンロードできる。この圧縮フォルダの中にある `wgnuplot.hlp` と `wgnuplot.mnu` の 2 つのファイルを先ほどの本体を展開した `bin` フォルダの中にある同じ名前のファイルに置き換える。

### 9.6.2 フォントがつぶれている場合

画面表示に用いるフォントは、ウィンドウの左上にあるシステムメニュー(を左クリックして) [ Options ] - [ Choose font ] を選択してフォントを変更する。フォント名は Terminal、サイズは 14 辺りが適当。その後やはりシステムメニューで [ Options ] - [ Update wgnuplot.ini ] を選択すると次回起動時以降もその設定が受け継がれる。

## 9.7 参考文献

次の本が大変よい。

- 「使いこなす gnuplot (改訂第 2 版)」(矢吹道郎・大竹敢 著、テクノプレス、2004)